

Primeiros Passos Com O

Atualizado
para 9.7

DB2 Express-C

Ideal para desenvolvedores e administradores de sistemas



Raul F. Chong

com

Ian Hakes

Rav Ahuja

Prefácio por

Dr. Arvind Krishna

Um livro da comunidade para a comunidade

PRIMEIROS PASSOS COM O

DB2 Express-C

Um livro da comunidade para a comunidade

RAUL CHONG, IAN HAKES, RAV AHUJA

PREFÁCIO POR DR. ARVIND KRISHNA



TERCEIRA EDIÇÃO

Terceira edição (junho de 2009)

Terceira impressão (outubro de 2010)

Esta edição foi atualizada para o IBM® DB2® Express-C versão 9.7.2 para Linux®, UNIX® e Windows®.

© Copyright IBM Corporation 2007, 2010. Todos os direitos reservados.

Índice

Sobre este livro	11
Avisos e marcas comerciais	11
Por que eu devo ler este livro?	12
Como este livro está estruturado?	12
Um livro para a comunidade	13
Autores e colaboradores.....	14
Agradecimentos	14
Prefácio	15
PARTE I – VISÃO GERAL E CONFIGURAÇÃO.....	17
Capítulo 1 – O que é o DB2 Express-C?	19
1.1 Gratuito para desenvolvimento, implementação e distribuição...sem limites!	20
1.2 Download do DB2 Express-C	20
1.3 Assistência ao usuário e suporte técnico	21
1.4 Servidores DB2.....	21
1.5 Clientes e drivers do DB2	22
1.6 Liberdade de desenvolvimento de aplicativos.....	23
1.7 Versões do DB2 versus edições do DB2	24
1.8 Passando para outra edição do DB2.....	24
1.9 Manutenção e atualizações para o DB2 Express-C.....	25
1.10 Software gratuito e componentes do DB2 relacionados	25
1.10.1 IBM Data Studio.....	26
1.10.4 DB2 Text Search	26
1.10.5 WebSphere Application Server – Community Edition	27
1.11 Resumo.....	27
Capítulo 2 – Funções e produtos relacionados.....	29
2.1 Funções incluídas na assinatura do DB2 Express (FTL)	32
2.1.1 Fix packs.....	32
2.1.2 Alta disponibilidade e recuperação de desastres (HADR)	32
2.1.3 Replicação de dados	33
2.2 Funções não disponíveis no DB2 Express-C	34
2.2.1 Particionamento de banco de dados	34
2.2.2 Concentrador de conexão	35
2.2.3 Geodetic Extender	35
2.2.4 Acesso baseado em rótulo (LBAC)	35
2.2.5 Workload Manager (WLM).....	36
2.2.6 Deep compression.....	37
2.2.7 Compatibilidade com SQL.....	38
2.3 Produtos com taxa relacionados ao DB2	39
2.3.1 DB2 Connect	39
2.3.2 InfoSphere Federation Server	40
2.3.3 InfoSphere Replication Server.....	41
2.3.4 Optim Development Studio (ODS)	41
2.3.5 Optim Database Administrator (ODA)	42
2.4 Ofertas do DB2 na Amazon Elastic Compute Cloud	42

2.5 Resumo.....	42
Capítulo 3 – Instalação do DB2	43
3.1 Pré-requisitos de instalação	43
3.2 Autoridade de instalação do sistema operacional	43
3.3 Assistente de instalação	44
3.4 Validação da instalação	51
3.5 Instalação silenciosa.....	53
3.6 Resumo.....	54
3.7 Exercícios	54
Capítulo 4 – Ambiente do DB2.....	59
4.1 Configuração do DB2	67
4.1.1 Variáveis de ambiente	68
4.1.2 Arquivo de configuração de gerenciador de banco de dados (dbm cfg).....	69
4.1.3 Arquivo de configuração de banco de dados (db cfg).....	71
4.1.4 Registro de perfil do DB2.....	72
4.2 O DB2 Administration Server (obsoleto)	73
4.3 Resumo.....	74
4.4 Exercícios	74
Capítulo 5 – Ferramentas do DB2	79
5.1 IBM Data Studio.....	81
5.2 Control Center (obsoleto)	82
5.2.1 Como iniciar o Control Center	85
5.3 Command Editor (obsoleto).....	86
5.3.1 Inicialização do Command Editor	86
5.3.2 Como adicionar uma conexão com banco de dados	87
5.4 Assistente SQL Assist (obsoleto)	88
5.5 Botão Show SQL (obsoleto)	89
5.6 Task Center (obsoleto)	90
5.6.1 O banco de dados Tools Catalog (obsoleto).....	90
5.7 Journal (obsoleto)	92
5.7.1 Inicialização do Journal	93
5.8 Health Monitor (obsoleto)	93
5.8.1 Health Center (obsoleto).....	94
5.9 Gerenciador de memória autoajustável.....	96
5.10 Criação de scripts	96
5.10.1 Scripts SQL.....	96
5.10.2 Scripts de sistema operacional (shell).....	98
5.11 Considerações sobre o Windows Vista	98
5.12 Resumo.....	99
5.13 Exercícios	99
PART II – APRENDENDO O DB2: ADMINISTRAÇÃO DO BANCO DE DADOS.....	105
Capítulo 6 – Arquitetura do DB2.....	106
6.1 O modelo de processo do DB2.....	106
6.2 O modelo de memória do DB2	108
6.3 O modelo de armazenamento do DB2	109

6.3.1 Páginas e extensões	110
6.3.2 Bufferpools.....	110
6.3.3 Tablespaces	112
6.4 Resumo.....	117
6.5 Exercícios	117
Capítulo 7 – Conectividade com o cliente.....	121
7.1 Diretórios do DB2.....	121
7.1.1 Diretório de banco de dados do sistema	121
7.1.2 Diretório de banco de dados local	122
7.1.3 Diretório do nó	122
7.1.4 Diretório DCS.....	122
7.2 Configuration Assistant (obsoleto).....	122
7.2.1 Configuração necessária no servidor	123
7.2.2 Configuração necessária no cliente	126
7.2.3 Como criar perfis de cliente e servidor	130
7.3 Resumo.....	133
7.4 Exercícios	133
Capítulo 8 – Como trabalhar com objetos de banco de dados.....	137
8.1 Esquemas	137
8.2 Sinônimos (ou aliases) públicos	138
8.3 Tabelas	139
8.3.1 Tipos de dados	139
8.3.2 Colunas de identificação.....	144
8.3.3 Objetos de sequência	145
8.3.4 Tabelas de catálogo do sistema	145
8.3.5 Tabelas temporárias globais declaradas (DGTTs).....	146
8.3.6 Criar tabelas temporárias globais (CGTTs).....	148
8.4 Visualizações	149
8.5 Índices.....	149
8.5.1 Design Advisor.....	150
8.6 Integridade das referências	151
8.7 Evolução do esquema	152
8.8 Resumo.....	154
8.9 Exercícios	154
Capítulo 9 – Utilitários de movimentação de dados.....	157
9.1 Utilitário EXPORT	158
9.2 Utilitário IMPORT	159
9.3 Utilitário LOAD	160
9.4 O utilitário db2move.....	162
9.5 O utilitário db2look	162
9.6 Resumo.....	165
9.7 Exercícios	165
Capítulo 10 – Segurança de banco de dados	169
10.1 Autenticação	170
10.2 Autorização.....	171

10.2.1 Privilégios	171
10.2.2 Autoridades.....	172
10.2.3 Funções	177
10.3 Considerações sobre privilégio de grupo	178
10.4 O grupo PUBLIC.....	178
10.5 As instruções GRANT (conceder) e REVOKE (revogar)	179
10.6 Autorização e verificação de privilégio	179
10.7 Segurança estendida no Windows	181
10.8 Resumo.....	181
10.9 Exercícios	182
Capítulo 11 – Backup e recuperação	187
11.1 Log do banco de dados	187
11.2 Tipos de logs.....	188
11.3 Tipos de logs.....	189
11.3.1 Log circular	189
11.3.2 Log de arquivamento.....	190
11.4 Log do banco de dados no Control Center.....	191
11.5 Parâmetros de log	192
11.6 Backup do banco de dados	193
11.7 Recuperação do banco de dados.....	195
11.7.1 Tipos de recuperação.....	195
11.7.2 Restauração do banco de dados.....	195
11.8 Outras operações com BACKUP e RESTORE	196
11.9 Resumo.....	196
11.10 Exercícios	197
Capítulo 12 – Tarefas de manutenção	201
12.1 REORG, RUNSTATS, REBIND.....	201
12.1.1 O comando REORG	202
12.1.2 O comando RUNSTATS.....	202
12.1.3 BIND / REBIND.....	203
12.1.4 Tarefas de manutenção do Control Center (Centro de Controle)	204
12.2 Opções de manutenção.....	205
12.3 Resumo.....	207
12.4 Exercícios	207
Capítulo 13 – Concorrência e bloqueios	211
13.1 Transações	211
13.2 Concorrência.....	212
13.3 Problemas sem o controle de concorrência	213
13.3.1 Perda de atualização	213
13.3.2 Leitura sem alocação.....	214
13.3.3 Leitura não reproduzível.....	215
13.3.4 Leitura fantasma	216
13.4 Níveis de isolamento	216
13.4.1 Uncommitted read (Leitura sem alocação)	217
13.4.2 Cursor stability (Estabilidade do cursor).....	217

13.4.3 Estabilidade de leitura	219
13.4.4 Leitura reproduzível	219
13.4.5 Comparação de níveis de isolamento	220
13.4.6 Configuração do nível de isolamento	220
13.5 Ampliação de bloqueio	222
13.6 Monitoramento de bloqueio	223
13.7 Espera de bloqueio	224
13.8 Causas e detecção de impasse de bloqueios	224
13.9 Melhores práticas de concorrência e bloqueio	226
13.10 Resumo	227
13.11 Exercícios	227
PARTE III – APRENDENDO O DB2: DESENVOLVIMENTO DE APLICATIVOS	235
Capítulo 14 – Introdução ao desenvolvimento de aplicativos DB2	237
14.1 Desenvolvimento de aplicativos no DB2: O quadro geral	237
14.2 Desenvolvimento no lado do servidor	239
14.2.1 Procedimentos armazenados	239
14.2.2 Funções definidas pelo usuário	240
14.2.3 Disparadores	240
14.3 Desenvolvimento no lado do cliente	241
14.3.1 SQL incorporado	241
14.3.2 SQL estático x SQL dinâmico	242
14.3.3 CLI e ODBC	245
14.3.4 JDBC, SQLJ e pureQuery	247
14.3.5 OLE DB	249
14.3.6 ADO.NET	250
14.3.7 PHP	251
14.3.8 Ruby on Rails	252
14.3.9 Perl	252
14.3.10 Python	253
14.4 XML e DB2 pureXML	253
14.5 Serviços de web	254
14.6 APIs administrativas	255
14.7 Outros tipos de desenvolvimento	255
14.7.1 Como trabalhar com Microsoft Access e Microsoft Excel	256
14.8 Ferramentas de desenvolvimento	257
14.9 Amostras de programas	257
14.10 Resumo	258
Capítulo 15 – DB2 pureXML	259
15.1 Como usar XML com bancos de dados	260
15.2 Bancos de dados XML	260
15.2.1 Bancos de dados com capacidade para XML	260
15.2.2 Bancos de dados XML nativos	261
15.3 XML no DB2	262
15.3.1 As vantagens da tecnologia pureXML	263
15.3.2 Fundamentos de XPath	265

15.3.3 Fundamentos do XQuery	269
15.3.4 Inserção de documentos XML.....	270
15.3.5 Como consultar dados XML	273
15.3.6 Operações Join com SQL/XML	280
15.3.7 Joins com XQuery	281
15.3.8 Operações de atualização e exclusão.....	282
15.3.9 Indexação XML.....	284
15.4 Como trabalhar com esquemas XML	285
15.4.1 Registro dos esquemas XML	285
15.4.2 Validação de esquema XML.....	288
15.4.3 Outro suporte a XML	289
15.6 Resumo.....	290
15.7 Exercícios	290
Apêndice A – Solução de problemas.....	293
A.1 Obter mais informações sobre os códigos de erro	294
A.2 SQLCODE e SQLSTATE	294
A.3 Log de notificação de administração do DB2.....	295
A.4 db2diag.log	295
A.5 Rastreamentos CLI	296
A.6 Defeitos e correções do DB2	296
Apêndice B – Referências e recursos	297
B.1 Referências	297
B.2 Sites:.....	297
B.3 Livros	298
B.4 Emails para contato.....	299

Sobre este livro

Avisos e marcas comerciais

© Copyright IBM Corporation 2007, 2010

Todos os direitos reservados.

IBM Canada

8200 Warden Avenue

Markham, ON

L6G 1C7

Canadá

É proibida a cópia ou reprodução desta documentação ou de qualquer parte da mesma, usando qualquer forma ou método, ou ainda sua tradução para outro idioma, sem o consentimento prévio de todos os detentores de direitos autorais acima mencionados.

A IBM não oferece nenhuma representação ou garantia no que diz respeito ao conteúdo deste livro, e nega especificamente quaisquer garantias de comerciabilidade ou adequação para um propósito específico. A IBM não assume a responsabilidade por qualquer erro que possa aparecer neste documento. As informações contidas neste documento estão sujeitas a alterações sem aviso prévio. A IBM se reserva o direito de fazer qualquer alteração sem a obrigação de notificar qualquer pessoa sobre tais revisões ou alterações. A IBM não assume qualquer compromisso de manter atualizadas as informações aqui contidas.

As informações neste documento sobre produtos não pertencentes à IBM foram obtidas dos fabricantes desses produtos. A IBM não testou esses produtos e não pode confirmar a precisão dos dados sobre desempenho, compatibilidade ou qualquer outra afirmação relacionada a tais produtos não pertencentes à IBM. As dúvidas sobre os recursos dos produtos não pertencentes à IBM devem ser encaminhadas aos fornecedores desses produtos.

IBM, o logotipo IBM e ibm.com são marcas comerciais ou registradas da International Business Machines Corp., registradas em muitas jurisdições em todo o mundo. Outros nomes de produtos e serviços podem ser marcas comerciais da IBM ou de outras empresas. Uma lista atual das marcas comerciais da IBM está disponível na Web em "Copyright and trademark information" (Informações sobre direitos autorais e marcas comerciais) em www.ibm.com/legal/copytrade.shtml (em inglês).

Java e todas as marcas comerciais baseadas em Java e os logotipos são marcas comerciais da Sun Microsystems, Inc. nos Estados Unidos e/ou em outros países.

Microsoft e Windows são marcas registradas da Microsoft Corporation nos Estados Unidos, em outros países ou ambos.

Linux é marca comercial registrada de Linus Torvalds nos Estados Unidos, em outros países e ou em ambos.

UNIX é uma marca registrada de The Open Group nos Estados Unidos e/ou em outros países.

Outras empresas, nomes de serviços ou produtos podem ser marcas comerciais ou marcas de serviço de terceiros.

As referências feitas nesta publicação a produtos ou serviços IBM não pressupõem uma intenção da IBM de torná-los disponíveis em todos os países nos quais opera.

Por que eu devo ler este livro?

Este livro destina-se a qualquer pessoa que trabalhe ou pretenda trabalhar com bancos de dados, como os administradores de bancos de dados (DBAs), desenvolvedores de aplicativos, consultores, arquitetos de software, gerentes de produto, instrutores e estudantes.

Como este livro está estruturado?

A Parte I, Visão geral e configuração, explica o que é o DB2 Express-C edition, apresenta a família de produtos e funções do DB2, auxilia a instalação e criação de bancos de dados e explora as ferramentas disponíveis no DB2.

Parte II, Aprendendo o DB2: Administração de banco de dados, foi criada para familiarizar você com o ambiente DB2, arquitetura, conectividade remota, objetos de bancos de dados, movimentação de dados (importação/exportação/carregamento), segurança, backup e recuperação, concorrência e bloqueio e outras tarefas comuns de manutenção.

Parte III - Aprendendo o DB2: Desenvolvimento de aplicativos, apresenta o desenvolvimento de aplicativos DB2, incluindo o desenvolvimento pelo lado do servidor e pelo lado do cliente. Ela também discute SQL/XML, XQuery e pureXML®.

O Apêndice contém informações úteis sobre solução de problemas.

A maioria dos capítulos contém exercícios, e os arquivos de entrada de dados necessários para esses exercícios de laboratório foram fornecidos no arquivo compactado `expressc_book_exercises_9.7.zip` que acompanha este livro.

Os materiais neste livro também são usados nos cursos oferecidos como parte do **Programa DB2 on Campus** (DB2 no Campus), e correspondem o máximo possível às apresentações em vídeo de e-learning disponíveis em www.channelDB2.com/oncampus (em inglês). Você pode ler mais sobre o programa DB2 on Campus no site do DB2 Express-C: www.ibm.com/db2/express/students.html.

Observação:

Para obter mais informações sobre o programa DB2 on Campus, assista ao vídeo em: <http://www.channeldb2.com/video/video/show?id=807741:Video:3902>

Agora em sua terceira edição, fizemos algumas alterações e adições. Para aqueles que leram a segunda edição do livro, abrangendo o DB2 9.5, estamos facilitando a busca de alterações no livro que correspondem às novas funções ou atualizações na versão 9.7 do DB2. As alterações podem ser facilmente identificadas através deste ícone:



Um livro para a comunidade

Este livro foi criado pela equipe do DB2 Express-C. A versão online será disponibilizada para a comunidade do DB2 Express-C gratuitamente. No momento de sua redação, este livro já atingiu mais de 85.000 downloads e foi traduzido para 9 idiomas por voluntários em todo o mundo. Um verdadeiro esforço comunitário! Se você quiser fornecer feedback, contribuir com novo material, aperfeiçoar o material existente ou ajudar a traduzir este livro para outro idioma, envie um email com seus planos de contribuição para db2x@ca.ibm.com com o assunto "DB2 Express-C book changes" (Alterações no livro DB2 Express-C).

O sucesso deste livro tem sido a inspiração para desenvolver mais de 25 novos e-books gratuitos sobre produtos IBM, e também sobre tecnologias não-IBM. Os livros são parte da **Série de livros DB2 on Campus** (DB2 no Campus), que foi lançada em janeiro de 2010.

Para obter mais informações sobre este livro ou sobre a série DB2 on Campus, visite o site do IBM® DB2 Express-C em ibm.com/db2/express

Autores e colaboradores

As pessoas abaixo forneceram conteúdo e outras contribuições significativas para este livro.

Raul F. Chong – Autor principal

Raul é o gerente do programa DB2 on Campus no Laboratório da IBM em Toronto.

Ian Hakes – Co-autor e editor

Ian é um ex-facilitador da comunidade DB2 Express-C e agora trabalha como especialista em capacidade de uso no Laboratório da IBM em Toronto.

Rav S. Ahuja – Co-autor e publicação

Rav é gerente de produto sênior do DB2 no Laboratório da IBM em Toronto.

Agradecimentos

Agradecemos imensamente aos indivíduos a seguir, por sua assistência e pelos materiais de desenvolvimento mencionados neste livro:

- Ted Wasserman, Clara Liu e Paul Yip do Laboratório da IBM em Toronto, que desenvolveram materiais que serviram como a estrutura deste livro.
- Don Chamberlin e Cindy Saracco por seus arquivos sobre XQuery no IBM developerWorks e Matthias Nicola por suas apresentações sobre pureXML.
- Kevin Czap e Grant Hutchison por desenvolver materiais de resumo técnico sobre DB2.
- Katherine Boyachok e Natasha Tolub pelo design da capa deste livro.
- Susan Visser por revisar e fornecer assistência na publicação deste livro.

Prefácio

A inovação é a base do progresso tecnológico. Na IBM, inovação é uma parte integral da evolução de nosso servidor de dados. Pioneira das técnicas de gerenciamento de dados dos anos 1960 e 1970, a IBM continua a fornecer tecnologias inovadoras de gerenciamento de informação, o que se reflete nos milhares de patentes de gerenciamento de dados de autoria dos tecnólogos da IBM. Como resultado, algumas das maiores organizações do mundo dependem de produtos IBM, incluindo o DB2, para sustentar suas soluções de gerenciamento de dados mais exigentes e de missão crítica.

Entretanto, o DB2 não é mais apenas destinado a grandes empresas. Com o lançamento do DB2 Express-C, a premiada tecnologia DB2 agora está disponível para pequenas e médias empresas – e sem custo! Embora haja outros servidores de dados gratuitos ou de código aberto disponíveis, o DB2 Express-C oferece vantagens exclusivas sobre essas alternativas.

Há muitos avanços tecnológicos presentes no DB2 Express-C. Estas inovações fornecem novas capacidades, reduzem os encargos administrativos, aumentam o desempenho e reduzem o custo com infraestrutura.

A tecnologia híbrida do DB2 Express-C é capaz de gerenciar dados relacionais e XML em seus formatos nativos. Isto torna o DB2 ideal para estruturar a nova geração de aplicativos SOA e Web 2.0, na qual os dados XML fluem em abundância. Diferente de outros servidores de dados “gratuitos”, o DB2 Express-C não limita a quantidade de dados que podem ser armazenados em um banco de dados, ou o número de bancos de dados que podem ser criados em um sistema. E, é claro, se você precisar de suporte ou assistência da IBM, a ajuda está à distância de um clique.

Este livro serve como guia de primeiros passos e utilização do DB2 Express-C. Ele irá ajudá-lo a compreender os conceitos do DB2 e possibilitar que você desenvolva habilidades de administração e desenvolvimento de aplicativos do DB2. As habilidades e conhecimentos que você irá obter são relevantes para as outras edições avançadas de DB2 em Linux, UNIX e Windows.

Embora o DB2 Express-C não seja um produto de código aberto, a IBM acredita muito em fornecer suporte e promover as iniciativas comunitárias. Estou encantado por este livro ser desenvolvido pelos membros da comunidade de DB2 Express-C e estar disponível gratuitamente para qualquer pessoa da comunidade. Encorajo você a enriquecer e atualizar este livro com seus próprios conhecimentos e experiências, e também a auxiliar na tradução deste livro para outros idiomas, para que outras pessoas possam se beneficiar também.



Arvind Krishna
Gerente geral
Gerenciamento de informação, IBM Software Group

PARTE I – VISÃO GERAL E CONFIGURAÇÃO

1

Capítulo 1 – O que é o DB2 Express-C?

O software de servidor de dados DB2 Express-C (“DB2 Express-C”) é membro da família IBM DB2, um poderoso software de servidor de dados para gerenciar dados relacionais e XML. O [DB2 Express-C](#) é uma edição gratuita, ilimitada e fácil de usar do DB2. O ‘C’ em DB2 Express-C significa Comunidade. Uma comunidade de usuários do DB2 Express-C que se une para ajudar uns aos outros, seja on-line ou off-line. A comunidade do DB2 Express-C consiste em todo tipo de pessoas e empresas que projetam, desenvolvem, implementam ou utilizam soluções de bancos de dados. Os membros da comunidade incluem:

- Desenvolvedores de aplicativos que necessitam de um software de banco de dados com padrões abertos para construir aplicativos independentes, cliente-servidor, baseados na web e corporativos
- ISVs, fornecedores de hardware, fornecedores de pilha de infraestrutura e fornecedores de outros tipos de solução que desejam montar um pacote ou incorporar um servidor de dados completo como parte de suas soluções
- Consultores, administradores de bancos de dados e arquitetos de TI que necessitam de um servidor de dados robusto para treinamento, capacitação, avaliação e prototipagem
- Empresas recém-iniciadas, pequenas e médias que necessitam de um servidor de dados confiável para suas aplicações e operações
- Hobistas de bancos de dados e entusiastas de tecnologia de ponta que desejam um servidor de dados fácil de usar para construir aplicativos Web 2.0 e de próxima geração
- Estudantes, professores e outros usuários acadêmicos que desejam um servidor de dados altamente versátil para ensinar, material de curso, projetos e pesquisa

O DB2 Express-C compartilha a mesma funcionalidade principal e base de código das outras edições pagas do DB2 para Linux, UNIX e Windows. O DB2 Express-C pode ser executado em sistemas de 32 ou 64 bits com os sistemas operacionais Linux ou Windows. Ele também está disponível em Solaris (x64) e como versão beta em Mac OS X (x64). Pode ser executado em sistemas com qualquer quantidade de processadores e memória, e não tem qualquer requisito especial de armazenamento ou configuração de sistema. O DB2 Express-C também inclui pureXML sem qualquer custo. O pureXML é uma tecnologia exclusiva do DB2 que armazena e processa documentos XML na forma nativa.

1.1 Gratuito para desenvolvimento, implementação e distribuição...sem limites!

Esta frase resume os principais ideais por trás do DB2 Express-C:

- **Gratuito para desenvolvimento:** Se você for um desenvolvedor e necessita de um banco de dados para seu aplicativo, você pode usar o DB2 Express-C.
- **Gratuito para implementação:** Se estiver trabalhando em um ambiente de produção e precisar de um sistema de gerenciamento de dados para armazenar seus registros vitais, você pode usar o DB2 Express-C.
- **Gratuito para distribuição:** Se estiver desenvolvendo um aplicativo ou uma ferramenta que exige um servidor de dados incorporado, você pode incluir o DB2 Express-C. E mesmo quando o DB2 Express-C for incorporado ao seu aplicativo e distribuído todas as vezes que você o vender, ele ainda será gratuito. Você precisará se registrar na IBM para poder redistribuir o DB2 Express-C, mas o registro também é gratuito.
- **Sem limites:** Embora outras ofertas de bancos de dados concorrentes estabeleçam limites sobre o tamanho do banco de dados, o número de bancos de dados e o número de usuários, no DB2 Express-C NÃO há limites de tamanho de dados. Seu banco de dados pode continuar a crescer sem violar o contrato de licenciamento. Também não há limites de licenciamento impostos sobre o número de conexões ou usuários por servidor.

Observação:

Para saber mais sobre o DB2 Express-C e seu papel no universo da informação sob demanda e da Web 2.0, assista a esta apresentação em vídeo:

<http://www.channeldb2.com/video/video/show?id=807741:Video:3922>

1.2 Download do DB2 Express-C

Todas as imagens do DB2 Express-C podem ser baixadas em ibm.com/db2/express e usadas gratuitamente. Estão disponíveis as imagens abaixo:

- DB2 Express-C 9.7.2 para Windows
- DB2 Express-C 9.7.2 para Windows 64 bits
- DB2 Express-C 9.7.2 para Linux
- DB2 Express-C 9.7.2 para Linux 64 bits
- DB2 Express-C 9.7.2 para Linux no Power
- DB2 Express-C 9.7.2 para Solaris x86-64
- DB2 Express-C 9.5.2 versão beta para Mac OS X

Observação:

No Windows, o DB2 Express-C também é oferecido em uma versão mais leve que é 44% menor que a versão regular. Ele está disponível somente em inglês e não inclui as ferramentas de GUI e o recurso de pesquisa de texto.

1.3 Assistência ao usuário e suporte técnico

Se você tiver alguma dúvida técnica sobre o DB2 Express-C, poste suas perguntas no [fórum do DB2 Express-C](#). Este fórum aberto é monitorado por especialistas em DB2 da IBM, embora seja a comunidade quem fornece a maioria das respostas voluntariamente.

A IBM também oferece aos usuários a escolha de adquirir, por um custo baixo, uma assinatura anual de software servidor de dados do DB2 Express (“DB2 Express”) (também conhecida como uma Licença por período fixo ou FTL). Esta assinatura é acompanhada pela assistência da IBM para suporte técnico 24 horas por dia e atualizações de software. Além de suporte e manutenção de software, com a taxa de assinatura anual de baixo custo (aproximadamente US\$1.990 por servidor por ano nos EUA – podendo variar em outros países) você também pode usar os recursos adicionais: HADR (clustering para alta disponibilidade e recuperação de desastres), replicação de SQL (para a replicação de dados com outros servidores DB2) e compressão de backup (para a criação de cópias compactadas de backup do banco de dados). Você encontrará mais informações sobre a opção de assinatura em: ibm.com/db2/express/support.html

1.4 Servidores DB2

Todas as edições do servidor DB2 contêm os mesmos componentes principais; os pacotes são montados de tal maneira que os usuários podem escolher as funções de que necessitam pelo preço mais adequado. A *Figura 1.1* ilustra as diferentes edições do produto DB2.

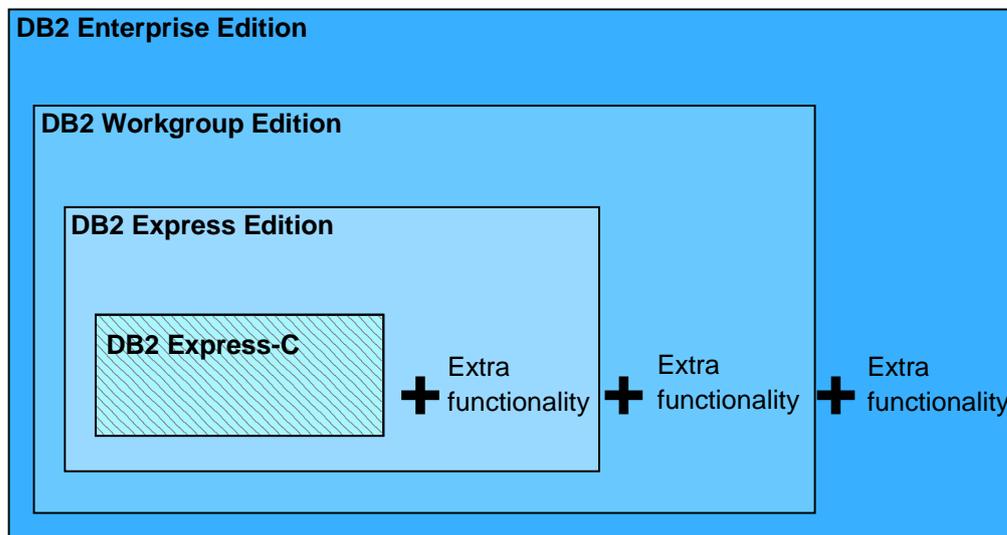


Figura 1.1 – Servidores DB2

Como mostrado na *Figura 1.1*, o DB2 Express-C é o mesmo que o DB2 Express sem alguns componentes. O DB2 Express-C é gratuito para a comunidade. A assistência técnica está disponível através de um fórum on-line gratuito, ou você pode receber suporte técnico oficial da IBM DB2 durante 24 horas por dia caso adquira a assinatura anual (Licença por período fixo do DB2 Express).

A *Figura 1.1* também explica porque a atualização do DB2 Express-C é tão fácil. Se você deseja atualizar para qualquer um dos servidores DB2 no futuro, todos os servidores DB2 têm os mesmos componentes principais. Isto também significa que qualquer aplicativo desenvolvido para uma edição funcionará, sem modificações, nas outras edições. E qualquer habilidade que você aprender em uma edição poderá ser aplicada às outras edições.

1.5 Clientes e drivers do DB2

Um cliente do DB2 inclui a funcionalidade necessária para se conectar a um servidor DB2; entretanto, nem sempre é necessário instalar um cliente do DB2. Por exemplo, um aplicativo JDBC Tipo 4 requer somente a instalação de um driver JDBC para se conectar a um servidor DB2. Os clientes e drivers de DB2 são fornecidos em diversas versões diferentes:

- IBM Data Server Client: mais completo, inclui ferramentas de GUI, drivers
- IBM Data Server Runtime Client: um cliente leve com funcionalidade básica e inclui drivers
- DB2 Runtime Client Merge Modules for Windows: usado principalmente para incorporar um cliente DB2 Runtime como parte da instalação de um aplicativo de Windows
- IBM Data Server Driver for JDBC and SQLJ: permite que aplicativos Java se conectem aos servidores DB2 sem a necessidade de instalar um cliente completo
- IBM Data Server Driver for ODBC and CLI: permite que os aplicativos ODBC e CLI se conectem a um servidor DB2 sem a grande ocupação devida à necessidade de instalar um cliente
- IBM Data Server Driver Package: Inclui um driver específico par Windows com suporte a ambientes .NET além de ODBC, CLI e código aberto. Este driver era conhecido anteriormente como o IBM Data Server Driver para ODBC, CLI e .NET.

new in
V9.7

A *Figura 1.2* mostra os diferentes clientes e drivers disponíveis para DB2.

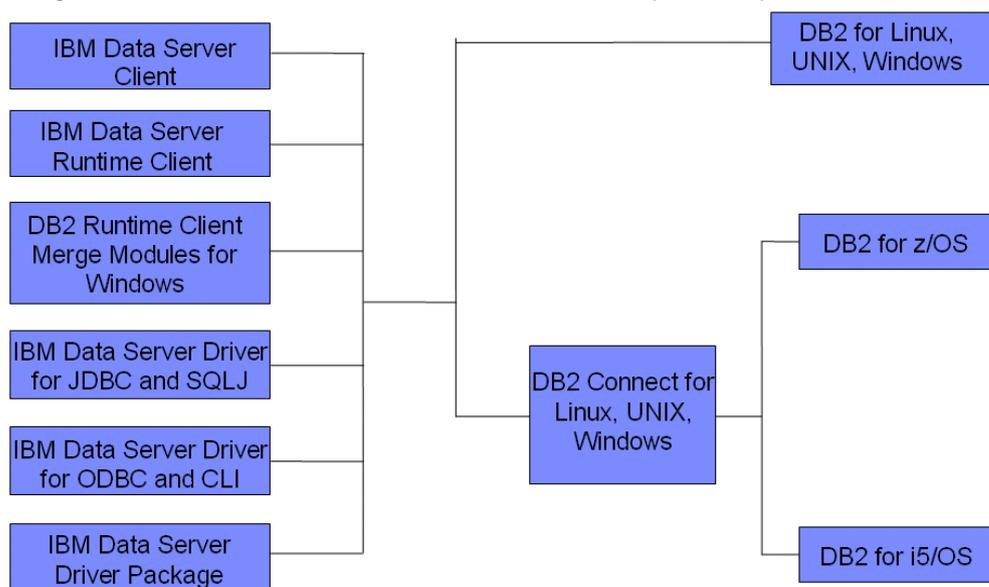


Figura 1.2 – Clientes e drivers do DB2

No lado esquerdo da *Figura 1.2* são mostrados todos os clientes e drivers do DB2. Embora todos os clientes do DB2 incluam os drivers necessários, a partir do software de servidor de dados DB2 (“DB2”) v.9 fornecemos os drivers individuais também. Todos os clientes e drivers do DB2 são gratuitos e estão disponíveis para download do site do DB2 Express-C. Os clientes e drivers podem ser usados para se conectar a um servidor DB2 em Linux, UNIX ou Windows. Para se conectar a um DB2 para z/OS® ou DB2 para servidor i5/OS®, será necessário passar por um servidor DB2 Connect™ (mostrado no meio da *Figura 1.2*). Discutiremos o software DB2 Connect (“DB2 Connect”) no *Capítulo 2*.

Observação:

Embora o foco deste livro seja sobre o servidor de dados DB2, os clientes do IBM Data Server também podem se conectar a outros servidores de dados da família IBM, como o Informix. Daí o nome genérico “IBM Data Server client” (cliente de servidor de dados da IBM) em vez do mais específico “DB2 client” (cliente DB2).

1.6 Liberdade de desenvolvimento de aplicativos

O DB2 oferece um ambiente de desenvolvimento de aplicativos que é baseado em padrões e transparente em toda a família DB2. A padronização SQL por toda a linha de produtos DB2 fornece um conjunto comum de interfaces de programação de aplicativos para acesso a banco de dados.

Além disso, cada produto DB2 fornece pré-compiladores SQL e interfaces de programação de aplicativos (APIs) que permitem aos desenvolvedores incorporar SQL estático e dinâmico em aplicativos portáteis. O DB2 possui até mesmo um provedor gerenciado .NET nativo e integração às ferramentas do Microsoft® Visual Studio.

As linguagens e padrões que você pode usar com o DB2 incluem:

- SQL, XQuery, XPath
- C/C++ (CLI, ODBC e SQL incorporado)
- Java (JDBC e SQLJ)
- COBOL
- PHP
- Perl
- Python
- Ruby on Rails
- linguagens .NET
- OLE-DB
- ADO
- MS Office: Excel, Access, Word
- Serviços web

1.7 Versões do DB2 versus edições do DB2

Se você for um novato em DB2, pode ficar um pouco confuso quanto à distinção entre uma versão do DB2 e uma edição do DB2.

De tempos em tempos, a IBM lança publicamente uma nova versão do DB2. Uma versão inclui novas funções e melhorias significativas no produto. Atualmente, a IBM suporta oficialmente o DB2 Versão 9. Uma versão também pode sofrer algumas alterações, que são atualizações que podem incluir alguma nova funcionalidade, mas normalmente não são significativas o suficiente para assegurar uma nova versão. Por exemplo, 9.5 e 9.7 são níveis de atualização do DB2 Versão 9. Nos últimos anos, a IBM vem lançando uma nova atualização da versão do DB2 a cada 1 ou 2 anos; entretanto, as novas versões normalmente são lançadas com período de intervalo de 3 anos ou mais. A versão mais recente é a V9.7, que se tornou disponível ao público em geral (GA) em junho de 2009. Cada atualização de versão pode ter diversos níveis de modificação, normalmente contendo correções ou correspondendo a níveis de fix pack e raramente fornecendo novas funcionalidades. No momento da redação deste livro, o nível de versão, atualização e modificação (V,R,M) mais atuais do DB2 Express-C é 9.7.0, o que corresponde a um nível de código de 9.7 com o Fix pack 0, o que significa que está no nível de GA.

Por outro lado, as edições são ofertas selecionadas ou agrupamentos em pacote dentro de cada versão. Como discutido anteriormente, uma edição é um pacote de funções diferentes por um dado preço e uma certa licença. O DB2 Versão 9.7 (também conhecido por DB2 9.7) possui diversas edições; por exemplo, DB2 Express-C 9.7, DB2 Express 9.7, DB2 Workgroup 9.7 e DB2 Enterprise 9.7 (veja a *Figura 1.1*).

1.8 Passando para outra edição do DB2

Com o crescimento de suas necessidades de banco de dados, você pode necessitar de atualização para uma edição do DB2 que suporte uma configuração de hardware maior. Se esta situação ocorrer, é fácil atualizar para outra edição do DB2:

- Se você for atualizar do DB2 Express-C para o DB2 Express (Licença por período fixo) no mesmo computador, é necessário apenas aplicar a licença com o comando `db2licm`.
- Se estiver atualizando para outra edição do DB2 no mesmo sistema de computador, desinstale o DB2 Express-C e instale em seguida a nova edição do DB2. Ao desinstalar o DB2 Express-C, seus bancos de dados não serão excluídos (mas sempre é recomendável fazer um backup).
- Se estiver atualizando o DB2 e a nova edição for instalada em um computador diferente e maior usando o mesmo sistema operacional, instale a nova edição do DB2 no computador maior, faça backup dos bancos de dados no computador menor, mova as imagens de backup para o computador maior, e restaure as imagens de backup nos bancos de dados no computador maior. Também é necessário salvar as configurações da instância (`dbm cfg`) do seu computador menor e aplicá-las ao computador maior. Os comandos de backup e restauração são discutidos em maiores detalhes no *Capítulo 11, Backup e recuperação*. O `dbm cfg` é discutido em maiores detalhes no *Capítulo 5, O ambiente do DB2*.
- Em qualquer um dos casos, seu aplicativo cliente não necessita de modificações.

1.9 Manutenção e atualizações para o DB2 Express-C

As imagens de instalação do DB2 Express-C são atualizadas periodicamente. Estas atualizações geralmente coincidem com a disponibilidade das novas atualizações ou versões, ou quando há um número significativo de correções acumuladas para o produto. No passado, as atualizações do DB2 Express-C normalmente eram disponibilizadas uma vez ao ano. Entretanto, observe que o DB2 Express-C, sendo uma oferta gratuita sem garantia, não é fornecido com nenhuma versão de manutenção oficial ou fix packs regularmente programados (que são liberados diversas vezes ao ano). Após a disponibilização de uma nova atualização ou versão do DB2 Express-C, as versões anteriores do DB2 Express-C não são mais mantidas.

Como discutido anteriormente, se você necessitar de acesso a correções de segurança e atualizações de software ou fix packs com correções regulares, a IBM oferece a licença com assinatura anual (FTL) do DB2 Express. Ao adquirir esta assinatura, sua instalação do DB2 Express-C pode ser atualizada com a chave de licença para FTL, que lhe dá o direito a suporte técnico ao DB2 e acesso a atualizações e fix packs durante o período de validade da licença de assinatura. A licença de assinatura também dá direito a atualizações de versão gratuitas, ou se você preferir a flexibilidade de permanecer em uma versão ou atualização de versão específica, basta aplicar os fix packs e correções de segurança durante o período em que tal versão for suportada e sua assinatura anual mantida.

1.10 Software gratuito e componentes do DB2 relacionados

Todo o software disponível para download na página de download do DB2 Express-C (www.ibm.com/db2/express/download.html) é gratuito. Além do software DB2 Express-C, há outros pacotes de software úteis para download e uso gratuito:

- Recursos adicionais do Visual Studio
- DB2 Spatial Extender

Há também os kits de ferramentas de iniciante adicionais baseados no DB2 Express-C, disponíveis para download o site do IBM Alphaworks (www.alphaworks.ibm.com/datamgmt) que podem ser úteis:

- Starter Toolkit for DB2 on Rails (www.alphaworks.ibm.com/tech/db2onrails/)
- Web 2.0 Starter Toolkit for DB2 (www.alphaworks.ibm.com/tech/web2db2)

Se estiver procurando um servidor de aplicativos leve gratuito, a IBM oferece:

- WebSphere® Application Server – Community Edition (WAS CE)

Observação:

Embora as capacidades espaciais com o DB2 já estejam disponíveis há quase 10 anos, poucos usuários as conhecem. Você pode tirar proveito do DB2 Spatial Extender gratuitamente em todas as edições do DB2, incluindo o DB2 Express-C. O Spatial Extender permite trabalhar com dados espaciais e geodéticos usando SQL. Por exemplo, esta capacidade pode ajudar a responder perguntas como “Qual é a loja de varejo mais próxima de cada cliente que reside em Toronto e gastou mais de US\$ 3000 no ano passado conosco?” Você pode até mesmo usar o DB2 Spatial Extender em aplicações de saúde. Por exemplo, ele poderia ajudar a responder à pergunta “Quais são os padrões de células malignas em uma ressonância magnética do cérebro?”

Para obter mais informações, consulte o tópico [Sobre o DB2 Spatial Extender](#) no Centro de Informações do IBM DB2 Database para Linux, UNIX e Windows Information.

1.10.1 IBM Data Studio

O IBM Data Studio é uma ferramenta baseada em Eclipse que permite a você gerenciar seus bancos de dados e o ajuda a desenvolver scripts XQuery, SQL, funções definidas pelo usuário e procedimentos armazenados. Há um depurador integrado incluído. Além disso, o IBM Data Studio permite trabalhar com diagramas de modelo de dados físico (PDM) para compreender os relacionamentos de entidade entre tabelas. Ele também pode ajudar a desenvolver e publicar dados como um serviço de web, usando o método de arrastar e soltar que não necessita de programação. O IBM Data Studio substitui ferramentas do DB2 como o Control Center e o Command Editor, que agora estão obsoletas (elas foram incluídas com o DB2, mas não estão mais em desenvolvimento). O IBM Data Studio é discutido em mais detalhes no *Capítulo 5, Ferramentas do DB2*.

1.10.4 DB2 Text Search

O DB2 Text Search é um componente integrado opcional do DB2. Ele tem como base a tecnologia OmniFind™ da IBM, e permite realizar poderosas pesquisas de texto completo rápidas e detalhadas em documentos de texto, incluindo qualquer documento XML armazenado nativamente no DB2. Este componente utiliza processamento linguístico para encontrar formatos diferentes do termo de pesquisa dentro do texto. Por exemplo, se você estiver procurando a palavra “estudar”, o DB2 Text Search também encontra outras formas da palavra, como “estuda” ou “estudou”.

Para instalar o componente DB2 Text Search, selecione uma instalação personalizada do DB2 Express-C e selecione a função DB2 Text Search dentro da categoria de suporte do servidor.

Observação:

Uma funcionalidade semelhante também está disponível em uma extensão do DB2 denominada Net Search Extender (NSE). A NSE está sendo substituída em favor do DB2 Text Search,

1.10.5 WebSphere Application Server – Community Edition

O IBM WebSphere Application Server - Community Edition (WASCE) é um servidor de aplicativos Java EE 5 leve disponível gratuitamente. Construído a partir da tecnologia Apache Geronimo, ele aproveita as mais recentes inovações da comunidade de código aberto para fornecer uma base integrada, acessível e flexível para o desenvolvimento e implementação de aplicativos Java. O suporte técnico opcional para WASCE está disponível através de uma assinatura anual.

1.11 Resumo

A edição DB2 Express-C oferece gratuitamente um produto que está entre os melhores do mercado. Ele oferece a liberdade de desenvolver, implementar e distribuir sem qualquer limitação de tamanho de banco de dados, enquanto ainda inclui a mesma funcionalidade principal e tecnologia pureXML das outras edições do DB2. O DB2 Express-C suporta uma ampla variedade de clientes, drivers e linguagens de desenvolvimento e fornece um caminho de atualização simples para as outras edições do DB2.

2

Capítulo 2 – Funções e produtos relacionados

Este capítulo descreve as funções do DB2 incluídas na compra de uma licença de assinatura anual (Licença por período fixo ou FTL) do DB2 Express. Ele também descreve as funções incluídas nas outras edições do DB2, em alguns casos por um valor adicional.

As diferenças entre o DB2 Express-C gratuito (sem garantia) e a opção de Assinatura anual para o DB2 Express estão realçadas abaixo, na *Tabela 2.1*.

Característica	Gratuito (sem garantia)	Assinatura paga* (FTL)
Capacidades principais do DB2	Sim	Sim
Ferramentas de administração gratuitas	Sim	Sim
Ferramentas de desenvolvimento gratuitas	Sim	Sim
Capacidades autônomas	Sim	Sim
Recurso pureXML	Sim	Sim
Assistência gratuita baseada em comunidade***	Sim	Sim
Suporte oficial da IBM 24 horas por dia	Não	Sim
Fixpacks	Não	Sim
Alta disponibilidade (HADR)	Não	Sim
Replicação de dados SQL	Não	Sim
Compactação de backup	Não	Sim
Utilização máxima de processador	2 núcleos	4 núcleos (máximo de 2 soquetes)
Utilização máxima de memória	2GB	4GB
Disponibilidade de atualização	Atualizações completas nas novas versões, geralmente uma vez por ano	Correções de segurança e fix packs diversas vezes ao ano
Acesso para instalar imagens de versões/atualizações anteriores	Não, estão disponíveis somente as imagens da atualização de versão e beta atuais	Sim, através do IBM Passport Advantage
Preço por servidor por ano**	0	US\$ 2.995

Tabela 2.1: Comparação entre o DB2 Express-C GRATUITO e a Assinatura paga (FTL)

30 Primeiros passos com o DB2 Express-C

* As funções incluídas na assinatura estão disponíveis somente durante a validade das assinaturas.

** O preço da assinatura é para os Estados Unidos e sujeito a alterações sem aviso prévio. O preço em outros países pode variar.

*** A assistência gratuita baseada em comunidade está disponível através do fórum on-line.

A *Tabela 2.2* lista as funções do produto e indica se estão incluídas nas diferentes edições do DB2 9.7. As funções adquiridas separadamente são listadas por nome na edição correspondente do DB2 e realçadas com fundo cinza claro.

Função	Assinatura do DB2 Express (FTL)	DB2 Express Edition	DB2 Workgroup Server Edition (WSE)	DB2 Enterprise Server Edition (ESE)
Replicação SQL homogênea	Sim	Sim	Sim	Sim
Federação homogênea	Sim	Sim	Sim	Sim
Net Search Extender, DB2 Text Search	Sim	Sim	Sim	Sim
Spatial Extender	Sim	Sim	Sim	Sim
Compactação de backup	Sim	Sim	Sim	Sim
Tecnologia pureXML	Sim	Sim	Sim	Sim
Alta disponibilidade recuperação de desastres	Sim	Recurso de alta disponibilidade	Sim	Sim
Tivoli® System Automation	Sim		Sim	Sim
Serviços avançados de cópia	Não		Sim	Sim
Reorganização on-line	Não		Sim	Sim
MQT	Não	Não	Não	Sim
MDC	Não	Não	Não	Sim
Paralelismo de consulta	Não	Não	Não	Sim
Concentrador de conexão	Não	Não	Não	Sim
Particionamento de tabela	Não	Não	Não	Sim
Governador	Não	Não	Não	Sim

Compressão: nível de linha, índice, XML, tabela temporária	Não	Não	Não	Recurso de otimização de armazenamento
Controle de acesso baseado em rótulo (LBAC)	Não	Não	Não	Recurso de controle de acesso avançado
Geodetic Extender	Não	Não	Não	Recurso de gerenciamento de dados geodéticos
Query Patroller	Não	Não	Não	Recurso de otimização de desempenho
Gerenciamento de carga de trabalho do DB2	Não	Não	Não	
Performance Expert	Não	Não	Não	
Replicação Q homogênea	Não	Não	Não	Recurso de replicação homogênea para ESE
Particionamento de banco de dados	Não	Não	Não	Não

Tabela 2.2: Edições do DB2 Versão 9.7: suporte a recurso e função

Os recursos disponíveis com as outras edições do DB2 são:

Recursos do DB2 Express Edition passíveis de cobrança

- Recurso de alta disponibilidade

Recursos incluídos gratuitamente no DB2 Workgroup Edition:

- Alta disponibilidade e recuperação de desastres (HADR), Tivoli System Automation, reorganização on-line, Serviços avançados de cópia
- Disponibilidade do DB2 em plataformas UNIX adicionais: AIX®, Solaris e HP-UX

Recursos incluídos gratuitamente no DB2 Enterprise Edition:

- Particionamento de tabela (faixa)
- Tabelas de consulta materializada (MQT)
- Clustering Multidimensional (MDC)

- Paralelismo de consulta
- Concentrador de conexão
- Governador

Recursos do DB2 Enterprise Edition passíveis de cobrança

- Recurso de otimização de armazenamento (inclui compactação)
- Controle avançado de acesso (refinado e segurança avançada)
- Otimização de desempenho (Gerenciamento de carga de trabalho, Performance Expert, Query Patroller)
- Gerenciamento de dados geodéticos (análise de localização geográfica)

Produtos relacionados ao DB2 com taxa:

- DB2 Connect
- InfoSphere Warehouse Editions
- InfoSphere Balanced Warehouse
- WebSphere Federation Server
- WebSphere Replication Server

2.1 Funções incluídas na assinatura do DB2 Express (FTL)

Esta seção descreve os Fix packs do DB2, HADR e replicação de SQL.

2.1.1 Fix packs

Um fix pack do DB2 é um conjunto de correções de código aplicado a um produto DB2 instalado para corrigir problemas diferentes relatados após o lançamento do produto. Com uma licença de assinatura instalada, os fix packs têm download e instalação gratuitos. Normalmente, eles são disponibilizados a cada quatro meses ou conforme autorizado.

Para fazer o download do fix pack mais recente, visite o site de suporte técnico do DB2 em http://www.ibm.com/software/data/db2/support/db2_9/

2.1.2 Alta disponibilidade e recuperação de desastres (HADR)

A alta disponibilidade e recuperação de desastres (HADR) é um recurso de confiabilidade de banco de dados que fornece uma solução de alta disponibilidade e recuperação de desastres para falhas completas ou parciais no local. Um ambiente HADR geralmente consiste em dois servidores de dados, o principal e o secundário (que podem estar em localizações geográficas separadas). O servidor principal é onde o banco de dados fonte é armazenado e acessado pelos aplicativos cliente. À medida que as transações são processadas no banco de dados principal, os registros de log do banco de dados são enviados automaticamente pela rede para o servidor secundário. O servidor secundário possui uma cópia clonada do banco de dados principal, normalmente criada através de backup do banco de dados principal e restauração no sistema secundário. Quando são

recebidos os logs do banco de dados principal, eles são repetidos e aplicados ao banco de dados secundário. Através de uma repetição contínua dos registros de log, o banco de dados secundário mantém uma réplica sincronizada do banco de dados principal, que pode assumir caso ocorra uma falha do banco de dados principal.

Uma solução HADR com suporte a DB2 completa oferece:

- Capacidade de failover ultra-rápida, com transparência completa para clientes e aplicativos cliente
- Atomicidade completa de transações para evitar a perda de dados
- A capacidade de atualizar sistemas ou aplicativos sem interrupção visível dos serviços
- Failover remoto do sistema, fornecendo recuperação completa de desastres locais que tenham atingido o centro de dados
- Gerenciamento simples com as ferramentas gráficas do DB2
- Tudo isso com um impacto desprezível sobre o desempenho geral do sistema

Observação:

Para assistir a uma demonstração do funcionamento do HADR, visite:

<http://www.ibm.com/software/data/db2/express/demo.html>



O DB2 9.7 terá a nova capacidade de permitir que os clientes leiam o servidor de reserva. Esta capacidade de "leitura de reserva" deverá estar disponível com o DB2 9.7 Fixpack 1.

2.1.3 Replicação de dados

Esta função permite a replicação dos dados entre o servidor fonte, onde são capturadas as alterações dos dados, e um servidor de destino, onde são aplicadas as alterações dos dados. A *Figura 2.1* fornece uma visão geral do funcionamento da replicação.

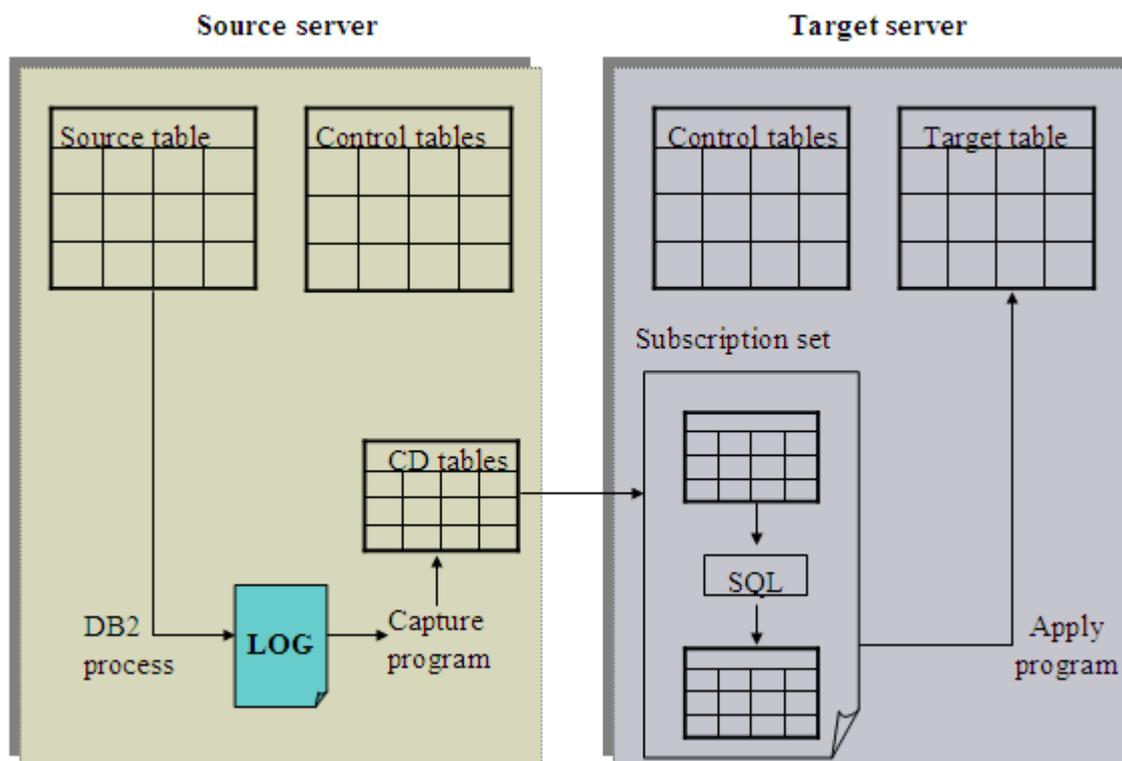


Figura 2.1 – Replicação de SQL

Na *Figura 2.1* há dois servidores, um servidor fonte e um servidor de destino. No servidor fonte, um programa Capture captura as alterações realizadas no banco de dados. No servidor de destino, um programa Apply aplica as alterações à réplica do banco de dados. A replicação é útil para diversas finalidades que exigem dados replicados, incluindo capacidade auxiliar, alimentação de depósitos de dados e mercados de dados, e auditoria do histórico de alterações. Usando a função de replicação de SQL, você pode replicar os dados entre o DB2 Express e outros servidores de dados IBM, incluindo os usados em outros sistemas Linux, UNIX, z/OS e i5/OS.

2.2 Funções não disponíveis no DB2 Express-C

Esta seção descreve algumas das funções disponíveis em outras edições do DB2, mas ausentes no DB2 Express-C ou na licença de assinatura anual do DB2 Express.

2.2.1 Particionamento de banco de dados

A função de particionamento do banco de dados (DPF) fornece processamento de consultas distribuído por um cluster de servidores de banco de dados. Ela está disponível somente com as InfoSphere Warehouse Editions e permite a distribuição dos dados entre diversas partições de banco de dados ou nós, que podem residir em diversos servidores diferentes. O DPF é baseado em uma arquitetura onde nada é compartilhado, na qual

cada partição de banco de dados possui um subconjunto dos dados gerais em seus próprios discos independentes.

À medida que é adicionado ao cluster de bancos de dados, cada computador traz poder de processamento de dados adicional com suas próprias CPUs, memória e discos, permitindo que tarefas grandes e consultas complexas sejam divididas em pedaços menores e distribuídas pelos diversos nós de bancos de dados para serem executadas em paralelo. Isto resulta em uma concorrência mais alta e tempos de resposta mais rápidos do que seria possível se o banco de dados residisse em um único servidor. O DPF é particularmente útil em grandes ambientes de depósito de dados e cargas de trabalho de inteligência de negócios envolvendo qualquer valor de uma centena de gigabytes até diversas centenas de terabytes de dados.

2.2.2 Concentrador de conexão

O concentrador de conexão é uma função que permite suportar uma grande quantidade de usuários conectados simultaneamente. Antes, cada conexão com banco de dados exigia um agente de banco de dados. O concentrador de conexão apresenta o conceito de um “agente lógico”, permitindo que um agente lide com diversas conexões. Os agentes são discutidos em mais detalhes no *Capítulo 6, Arquitetura do DB2*.

2.2.3 Geodetic Extender

O DB2 Geodetic Extender está disponível como versão paga do DB2 Enterprise Server Edition. Esta extensão torna muito mais fácil o desenvolvimento de aplicativos para inteligência de negócios e e-governo que exigem análise de localização geográfica. O DB2 Geodetic Extender pode construir um globo virtual em qualquer escala. A maioria das informações de localização é coletada com o uso de sistemas mundiais, como os satélites de posicionamento global (GPS), e pode ser representada em coordenadas latitude/longitude (geocode). Os dados de negócios, como endereços, podem ser convertidos em um geocódigo pelo DB2 Geodetic Extender, e os aplicativos corporativos funcionam melhor quando mantêm os dados neste formato não projetado, deixando as projeções de mapa (terra para mapa plano) onde pertencem: na camada de apresentação, para exibir e imprimir os mapas.

2.2.4 Acesso baseado em rótulo (LBAC)

O controle de acesso baseado em rótulo fornece segurança granular no nível de linha e coluna. Ele usa uma etiqueta associada a sessões de usuário e linhas ou colunas de dados, para dar acesso aos dados em sua mesa. A *Figura 2.2* ilustra como o LBAC funciona.

	No LBAC	SEC=254	SEC=100	SEC=50	ID	SALARY
<pre>SELECT * FROM EMP WHERE SALARY >= 50000</pre>					255	60000
					100	50000
					50	70000
					50	45000
					60	30000
					250	56000
					102	82000
					100	54000
					75	33000
					253	46000
					90	83000
					200	78000

Figura 2.2 – Um exemplo do funcionamento do LBAC

Na figura, a tabela *EMP* possui uma coluna, *SALARY*, e uma coluna interna *ID* contendo o rótulo para uma dada linha. As outras colunas da figura são usadas somente para fins de ilustração. Se a consulta mostrada na figura for executada, dependendo do rótulo do usuário, ele será capaz de ver linhas diferentes. A coluna com o título “No LBAC” (Sem LBAC) representa as linhas que seriam selecionadas caso o LBAC não fosse implementado. Como você pode ver, todas as linhas com salário superior ou igual a 50.000 são selecionadas.

Agora, digamos que o usuário emissor da consulta tenha rótulo de segurança 100. Você pode ver as linhas selecionadas, neste caso na coluna três, contando a partir da esquerda. Neste caso, o DB2 encontrará as linhas onde o salário é superior ou igual a 50.000, e irá verificar a segurança da linha. Por exemplo, a primeira linha tem salário 60.000 e ID de rótulo 255. Como este usuário possui ID de rótulo 100, menor do que 255, ele não poderá ver esta linha e, portanto, a saída da consulta não irá retorná-la.

A segurança LBAC deve ser implementada por um administrador de segurança que possua autoridade SECADM.

2.2.5 Workload Manager (WLM)

O Workload Manager gerencia cargas de trabalho por todo o banco de dados, com base nos usuários e prioridades de aplicativo combinados com a disponibilidade de funções e limites de carga de trabalho. Isto permite regular a carga de trabalho e as consultas ao banco de dados, de maneira que as consultas de prioridade máxima podem ser executadas imediatamente e evitar que as consultas ‘indesejadas’ monopolizem os recursos do seu sistema, assegurando que este seja executado de maneira eficiente.

O WLM foi aperfeiçoado ainda mais no DB2 9.7 e fornece capacidades mais poderosas que o Query Patroller e as ferramentas DB2 Governor disponíveis com as versões anteriores do DB2.

2.2.6 Deep compression

O DB2 suporta diversos tipos de compressão:

- NULL e compressão de valor padrão

Este tipo de compressão é aplicado às colunas cujos valores normalmente são NULL ou valores padrão de sistema como 0, onde não é consumido espaço de armazenamento em disco.

- Clustering multidimensional

As tabelas de clustering multidimensional (MDC) nas quais as páginas de dados físicos são agrupadas em diversas dimensões. Elas utilizam índices de bloco que, de certo modo, são uma maneira de compactar os índices porque apontam para um bloco de registros em vez de apontar para um único registro.

- Compressão de backup de banco de dados

Isto se aplica às imagens de backup. Os índices etablespace de LOB são compactados.

- Compressão de linha de dados

A compressão de linha funciona através da substituição de cadeias de texto repetidas dentro de uma linha de dados por um símbolo muito menor. O mapeamento deste símbolo menor e da cadeia de texto é mantido em um dicionário. A compressão de linha pode melhorar drasticamente o desempenho nas cargas de trabalho de E/S, uma vez que mais linhas podem ser transportadas do disco para a memória (e vice-versa) por serem menores. Você também pode se beneficiar da economia de armazenamento, que normalmente representa uma das maiores despesas do orçamento de TI das empresas. Para cargas de trabalho de CPU, pode haver alguma sobrecarga, pois as linhas compactadas precisam ser descompactadas antes do processamento. Observe, também, que os dados de log dos registros compactados também estão em formato compactado.

Em geral, ao acessar colunas de XML e LOB, o DB2 não utiliza o bufferpool (memória), mas executa E/S diretamente no disco. Isto ocorre porque o XML e os LOBs normalmente têm tamanho grande; assim, levá-los para a memória causaria a remoção de páginas necessárias da memória. Entretanto, com o DB2 9.5, é permitida a expansão de XML em linha para pequenos documentos XML (menores do que 32K). Isto significa que os documentos XML pequenos podem ser armazenados com as linhas da tabela base, e não em um objeto de armazenamento interno separado conhecido por XDA. As vantagens desta abordagem são duplas: Primeiro, os documentos XML agora podem ser acessados através do bufferpool e, em segundo lugar, os documentos XML também podem tirar proveito da compressão de linhas de dados.

new in
V9.7

O DB2 9.7 traz novos e mais aperfeiçoamentos na compressão:

- Os objetos internos XDA (onde o XML é armazenado) agora também podem ser compactados.
- Os índices e tabelas temporais (sistema e usuário) podem ser compactados.
- Os LOBs podem ser expandidos em linha de maneira semelhante à expansão de XML em linha.

new in
V9.7

2.2.7 Compatibilidade com SQL

Embora muitos fornecedores sigam os padrões de SQL 92 e SQL/PSM, nem todos os recursos dos padrões são suportados e, por outro lado, são suportados outros recursos não incluídos nos padrões. Com o recurso de compatibilidade de SQL do DB2 9.7, o DB2 agora pode suportar a maior parte da sintaxe de PL/SQL que é suportada por outros fornecedores de RDBMS, além do próprio SQL PL do DB2. A *Figura 2.3* resume como este suporte funciona.

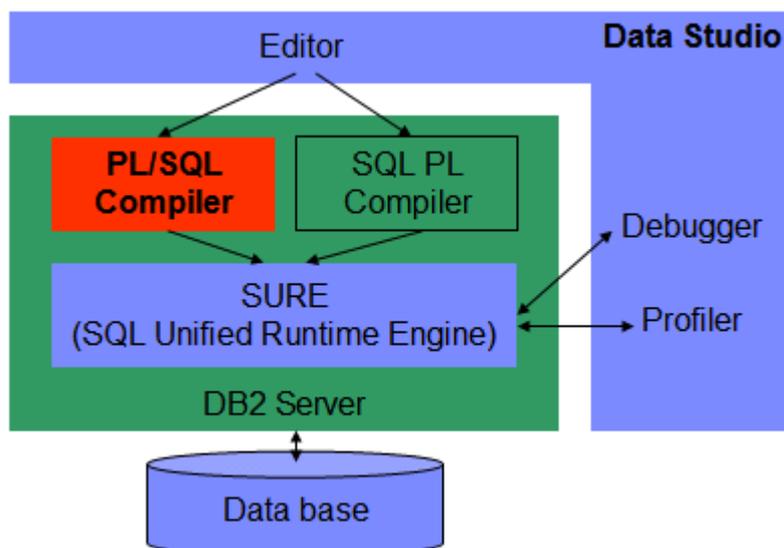
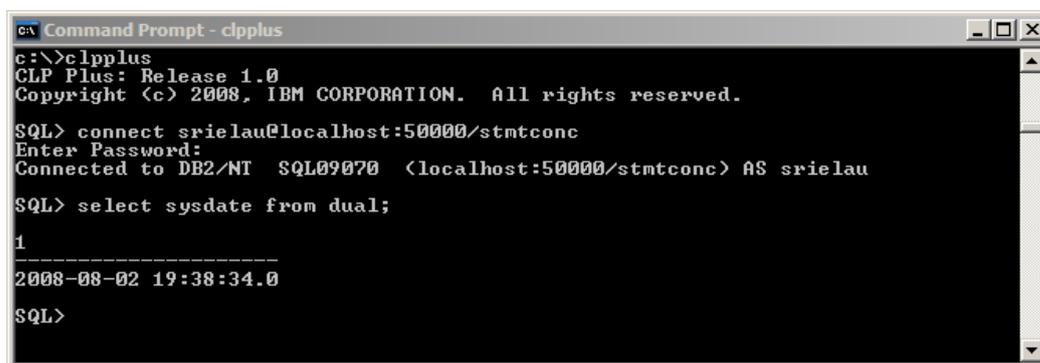


Figura 2.3 –Suporte a PL/SQL no DB2

Na figura, você poderá observar que um compilador PL/SQL foi desenvolvido e incorporado ao mecanismo do DB2.

O recurso de compatibilidade com SQL também inclui suporte a uma ferramenta denominada CLPPlus. O CLPPlus é uma ferramenta de linha de comando que permite executar SQL e outros comandos. Ele é similar ao Processador de linha de comando (CLP) existente do DB2. A *Figura 2.4* ilustra a ferramenta CLPPlus.



```
GA Command Prompt - clpplus
c:\>c:\clpplus
CLP Plus: Release 1.0
Copyright (c) 2008, IBM CORPORATION. All rights reserved.

SQL> connect srielau@localhost:50000/stmtconc
Enter Password:
Connected to DB2/NT  SQL09070  (localhost:50000/stmtconc) AS srielau

SQL> select sysdate from dual;

1
-----
2008-08-02 19:38:34.0

SQL>
```

Figura 2.4 –CLPPlus

O suporte à maioria dos tipos de dados do PL/SQL também foi incorporado, como BINARY_INTEGER, RAW e assim por diante. Outros tipos de dados da Oracle, como o VARCHAR2, são suportados sem necessitar do recurso de compatibilidade de SQL, mas você precisa ativá-los, usando a variável de registro DB2_COMPATIBILITY_VECTOR. Mais adiante no livro você poderá encontrar maiores explicações sobre os tipos de dados Oracle e esta variável de registro.

Os recursos de compatibilidade de SQL descritos acima estão disponíveis atualmente com as edições DB2 9.7 Workgroup e Enterprise. Espera-se que estejam disponíveis, em um futuro próximo, na edição do DB2 Express (incluindo a opção de assinatura ou FTL anual).

Embora o suporte a PL/SQL e os recursos CLPPlus não estejam disponíveis no Express-C 9.7, os outros recursos incluídos simplificam a ativação dos aplicativos Oracle no DB2. Estes incluem novos tipos de dados, novas funções de dimensionamento, suporte a módulos e semântica currently committed (CC, atualmente comprometido) para o nível de isolamento de cursor stability (CS, estabilidade do cursor). Estes recursos serão discutidos mais adiante no livro.

2.3 Produtos com taxa relacionados ao DB2

Esta seção fornece uma descrição rápida dos produtos e ofertas com taxa que podem ser usados com o DB2.

2.3.1 DB2 Connect

O DB2 Connect é um software pago que permite que um DB2 para cliente Linux, UNIX ou Windows se conecte a um DB2 para servidor z/OS ou i5/OS conforme mostrado na *Figura 2.5*. O DB2 Connect não é necessário quando a conexão ocorre na direção oposta; ao conectar-se do DB2 para z/OS ou DB2 para i5/OS de um DB2 para servidor Linux, UNIX ou Windows. O DB2 Connect é fornecido em duas edições principais, dependendo de suas necessidades de conexão: Uma edição DB2 Connect Personal Edition e outra DB2 Connect Enterprise Edition.

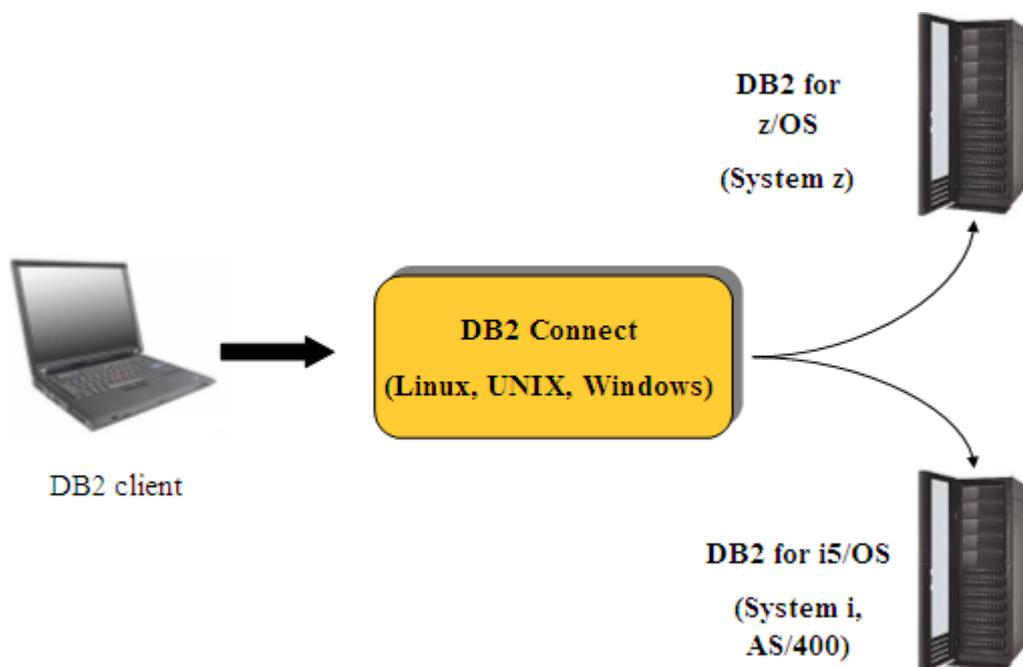


Figura 2.5 – DB2 Connect

2.3.2 InfoSphere Federation Server

Conhecido anteriormente por WebSphere Information Integrator (para suporte a federação), o WebSphere Federation Server permite a federação de bancos de dados, o que significa que você pode executar consultas a bancos de dados que podem trabalhar com objetos de diferentes sistemas de bancos de dados relacionais. Por exemplo, se você comprar o WebSphere Federation Server, poderá executar a consulta mostrada na *Listagem 2.1* abaixo.

```
SELECT *
FROM    Oracle.Table1  A
        DB2.Table2    B
        SQLServer.Table3 C
WHERE
        A.col1 < 100
        and B.col5 = 1000
        and C.col2 = 'Test'
```

Listagem 2.1 – Uma consulta federada

A *Figura 2.6* fornece uma ilustração do uso do WebSphere Federation Server.

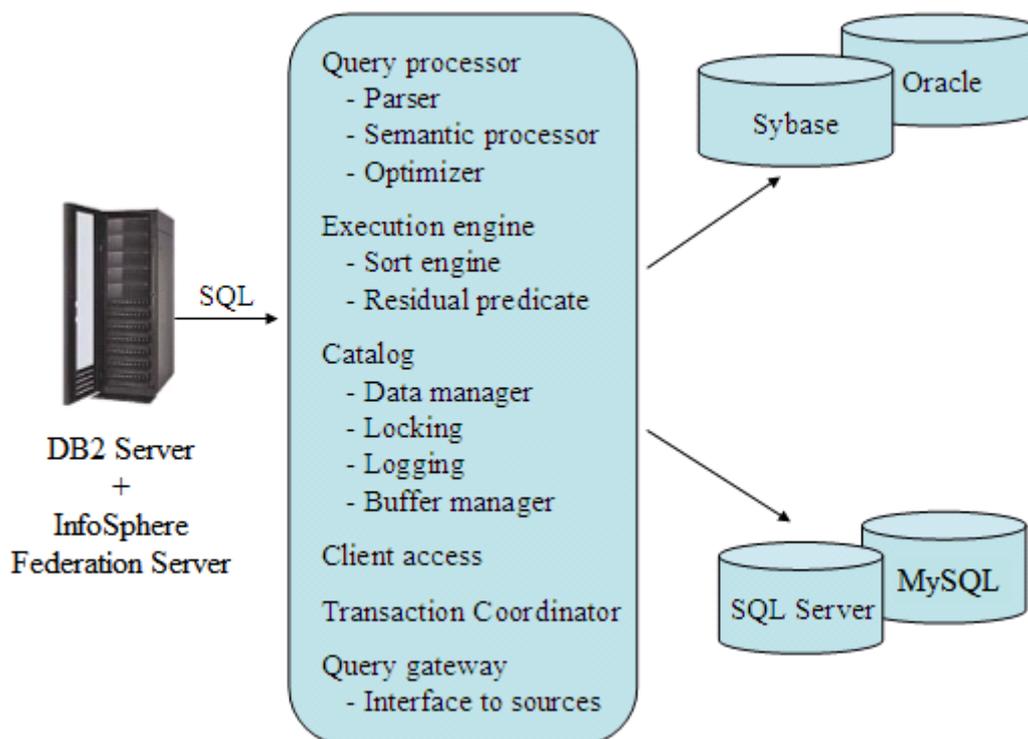


Figura 2.6 – InfoSphere Federation Server

Para sistemas de gerenciamento de bancos de dados relacionais que fazem parte da família IBM, o suporte a federação é integrado ao DB2 Express-C. Isto significa que o produto WebSphere Federation Server não é necessário quando você deseja, por exemplo, executar uma consulta entre dois bancos de dados DB2 diferentes, ou entre um banco de dados DB2 e um banco de dados Informix (o Informix faz parte da família IBM).

2.3.3 InfoSphere Replication Server

Conhecido anteriormente como WebSphere Information Integrator (para suporte a replicação), o InfoSphere Replication Server permite a replicação SQL de registros de bancos de dados quando há servidores de dados não IBM envolvidos. Ele também inclui um recurso conhecido por Q-Replication, para a replicação de dados usando filas de mensagens.

2.3.4 Optim Development Studio (ODS)

Conhecido anteriormente como Data Studio Developer, o ODS é uma ferramenta baseada em Eclipse que pode ser facilmente integrada ao Data Studio e compartilhar o mesmo Eclipse. O ODS pode ajudá-lo a criar bancos de dados de desenvolvimento usando a função de copiar e colar de bancos de dados Oracle ou DB2 existentes.

2.3.5 Optim Database Administrator (ODA)

Conhecido anteriormente como Data Studio Administrator, o ODA é uma ferramenta baseada em Eclipse que pode ser facilmente integrada ao Data Studio e compartilhar o mesmo Eclipse. O ODA fornece uma capacidade de gerenciamento de mudanças e de automatizar alterações de esquema com maior facilidade.

2.4 Ofertas do DB2 na Amazon Elastic Compute Cloud

É importante mencionar que a IBM formou uma parceria com a Amazon Web Services (AWS) para executar o DB2 na Elastic Compute Cloud (EC2) da Amazon. A AWS fornece um conjunto de serviços integrados que compõem uma plataforma de computação “em nuvem”, e está disponível em um modelo de pagamento de acordo com o uso. Ou seja, a AWS permite “alugar” capacidade de computação (servidores e armazenamento virtuais) e pagar apenas pela capacidade utilizada. Por exemplo, digamos que você provisione um servidor virtual EC2 para operações de banco de dados normais e, durante os momentos de pico ou para necessidades sazonais, provisione um servidor de banco de dados extra durante algumas horas. Neste exemplo, você pagaria à AWS pelo uso do servidor de banco de dados extra somente pelas horas em que este fosse executado.

A IBM oferece três opções diferentes de implementação do DB2 na plataforma de nuvem da Amazon:

- AMIs do DB2 Express-C para avaliação e desenvolvimento
- AMIs prontos para produção, com pagamento de acordo com o uso, com o DB2 Express e o DB2 Workgroup
- Capacidade de criar seus próprios AMIs usando as licenças DB2 que você possui

Para obter mais informações e saber como iniciar com o DB2 na Amazon EC2, visite: ibm.com/db2/cloud

2.5 Resumo

O DB2 Express-C fornece uma base gratuita e fácil de usar para o desenvolvimento de aplicativos de bancos de dados, sua implementação na produção, e até mesmo incorporação e distribuição dos aplicativos em soluções de terceiros. É ideal caso você fique à vontade com assistência baseada em comunidade e não tenha necessidade das correções mais recentes ou de recursos avançados. Entretanto, se você necessita de suporte técnico formal da IBM, atualizações de software (fixpacks) regulares ou utilização de recursos adicionais e suporte a clustering de alta disponibilidade, a IBM oferece uma licença de assinatura (FTL) do DB2 Express por uma taxa anual baixa. Se necessitar de recursos ainda mais avançados para cargas de trabalho de missão crítica e aplicações de banco de dados em larga escala, a IBM oferece edições do DB2 mais redimensionáveis e produtos relacionados. Isto permite que você comece pequeno com o DB2 Express-C e possa expandir para novos patamares de acordo com a demanda dos seus negócios.

3

Capítulo 3 – Instalação do DB2

A instalação do DB2 é razoavelmente simples e, em uma instalação típica, basta selecionar as opções padrão para ter o servidor DB2 instalado e funcionando em pouco tempo.

Primeiro, faça o download da imagem do DB2 Express-C apropriada à sua plataforma no site do DB2 Express-C (www.ibm.com/db2/express).

3.1 Pré-requisitos de instalação

O DB2 Express-C está disponível em Linux®, Sun Solaris (x64) e Microsoft Windows® 2003, XP, Vista e Windows 7. Ele também está disponível em uma versão beta para Mac OS X. As arquiteturas de processador disponíveis são 32 bits, 64 bits e PowerPC (Linux). Se você precisa executar o DB2 em outra plataforma (como o UNIX), você deve adquirir uma das diferentes edições de servidor de dados descritas anteriormente neste livro. Os requisitos de sistema operacional para todas as edições do DB2 também estão descritas neste documento: www.ibm.com/software/data/db2/udb/sysreqs.html

Em termos de recursos de hardware, o DB2 Express-C pode ser instalado em sistemas com diversos núcleos de CPU e memória; entretanto, ele utiliza somente até 2 núcleos e 2 GB de memória para a versão de licença gratuita sem garantia, e até 4 núcleos e 4 GB de memória para a versão paga assinada do DB2 Express. Os sistemas podem ser físicos ou virtuais, criados através de particionamento ou execução de software de máquina virtual. Se preferir, obviamente você pode executar em sistemas menores, por exemplo, um sistema com um único processador e 1 GB de memória.

Para obter as informações mais recentes sobre os requisitos de hardware do DB2 Express-C, leia a página do DB2 Express-C na web em www.ibm.com/software/data/db2/express/about.html

3.2 Autoridade de instalação do sistema operacional

Para instalar o DB2 Express-C no Linux ou Windows, é necessário um usuário de sistema operacional com autoridade suficiente.

Para o **Linux**, você precisa ter acesso root (o superusuário) para instalar o DB2 Express-C. Também é possível instalar o DB2 Express-C como usuário sem acesso root; entretanto, a funcionalidade do produto ficará limitada. Por exemplo, em uma instalação

sem acesso root, não é possível criar mais instâncias do que a única instância padrão criada no momento da instalação.

No **Windows**, a conta de usuário deve pertencer ao grupo dos Administradores na máquina onde a instalação será realizada. Como alternativa, no Windows 2008, Windows Vista ou posterior, um não administrador pode realizar uma instalação, mas as credenciais de administrador serão solicitadas pelo Assistente de configuração do DB2.

O ID do usuário de instalação deve pertencer ao grupo dos Administradores de domínio caso a instalação exija a criação ou confirmação de uma conta de domínio.

Você também pode usar a conta integrada do sistema local para executar a instalação, embora não seja recomendado. A conta de sistema local não exige senha, mas não pode acessar os recursos de rede.

A conta de usuário também deve ter a permissão "Acessar este computador através da rede".

Observação:

Assista a uma apresentação em vídeo sobre a instalação do DB2 Express-C neste link. Embora esta apresentação esteja demonstrando o DB2 9.5, a única diferença de uma instalação do DB2 9.7 está na cor dos painéis de instalação:

<http://www.channeldb2.com/video/video/show?id=807741:Video:4442>

3.3 Assistente de instalação

Embora existam diversos métodos de instalação do DB2 Express-C, o método mais fácil é usar o Assistente de instalação do DB2 baseado em GUI. Após fazer o download e descompactar a imagem do DB2 Express-C, você pode iniciar o assistente na forma a seguir:

- Windows: Execute o arquivo **setup.exe** localizado no diretório EXP/image/
- Linux: Execute o comando **db2setup** localizado no diretório exp/disk1/

A instalação do DB2 Express-C é simples, basta seguir as instruções do assistente de instalação do DB2. Na maioria dos casos, a configuração padrão é suficiente, bastando aceitar a licença, clicar no botão *Avançar* até o botão "*Finish*" (Concluir) estar ativo e, em seguida, clicar no botão "*Finish*". Após alguns minutos, sua instalação estará completa e o DB2 estará funcionando!

A *Figura 3.1* mostra o DB2 Setup Launchpad. Clique em *Install a Product* (Instalar um produto) e selecione *Install New* (Instalar novo) para instalar uma nova cópia do DB2 Express-C em seu sistema. Se você já instalou o DB2 Express-C ou outra edição do DB2 anteriormente, poderá haver um botão com o nome “*Work with Existing*” (Trabalhar com o existente). No DB2, você pode instalar o produto diversas vezes, e essas instalações podem ser de versões ou níveis de atualização diferentes.

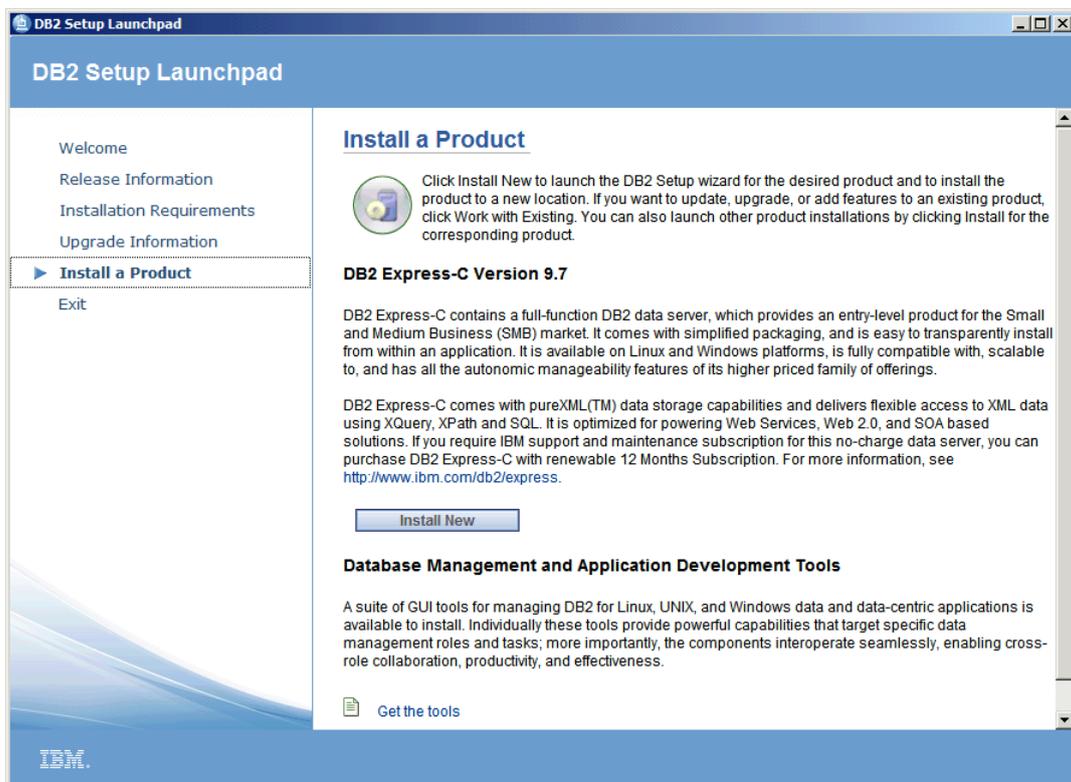


Figura 3.1 – O DB2 Setup Launchpad

Após aceitar a licença, normalmente é suficiente selecionar a instalação “*Typical*” (Típica) como mostrado na *Figura 3.2*. Se desejar incluir o componente DB2 Text Search, selecione “*Custom*” (Personalizada).

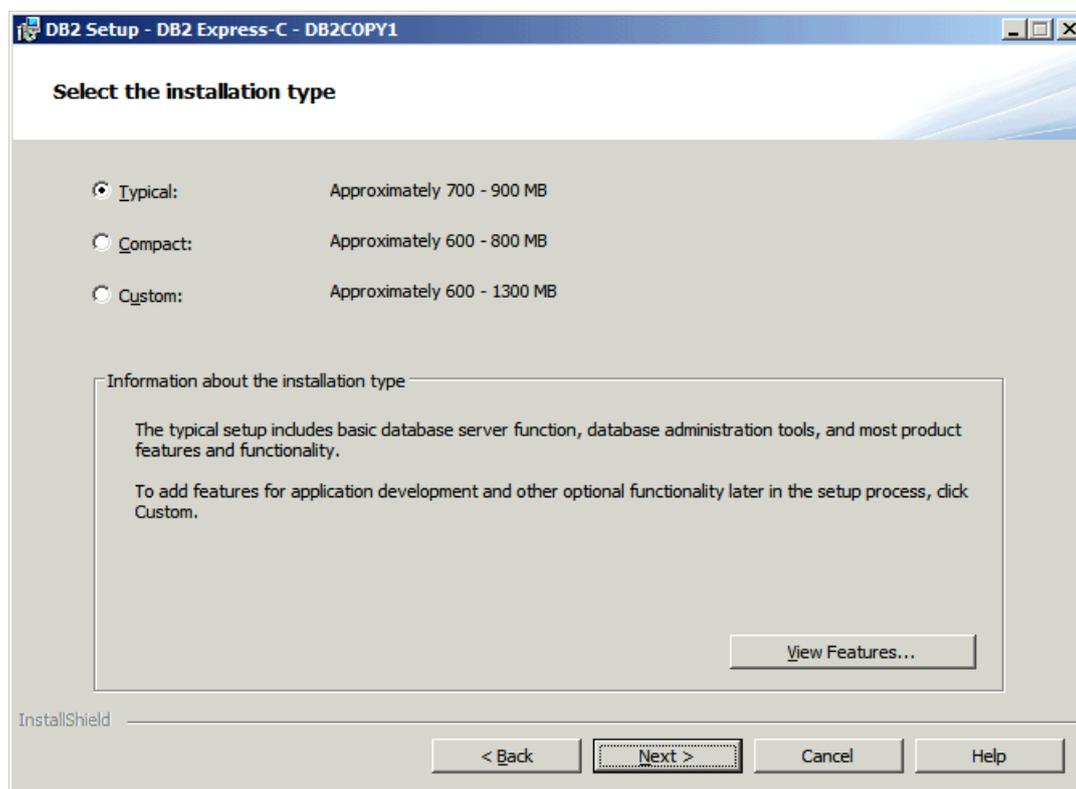


Figura 3.2 – Tipos de instalação

Na próxima etapa, como mostrado na *Figura 3.3*, você pode optar por instalar o produto, criar um arquivo de resposta ou ambos. Os arquivos de resposta são discutidos na seção 3.4, *Instalação silenciosa*. Escolha a opção padrão (“*Install IBM DB2 Express Edition on this computer and save my settings in a response file*”) (Instalar o IBM DB2 Express Edition neste computador e salvar minha configuração em um arquivo de resposta).

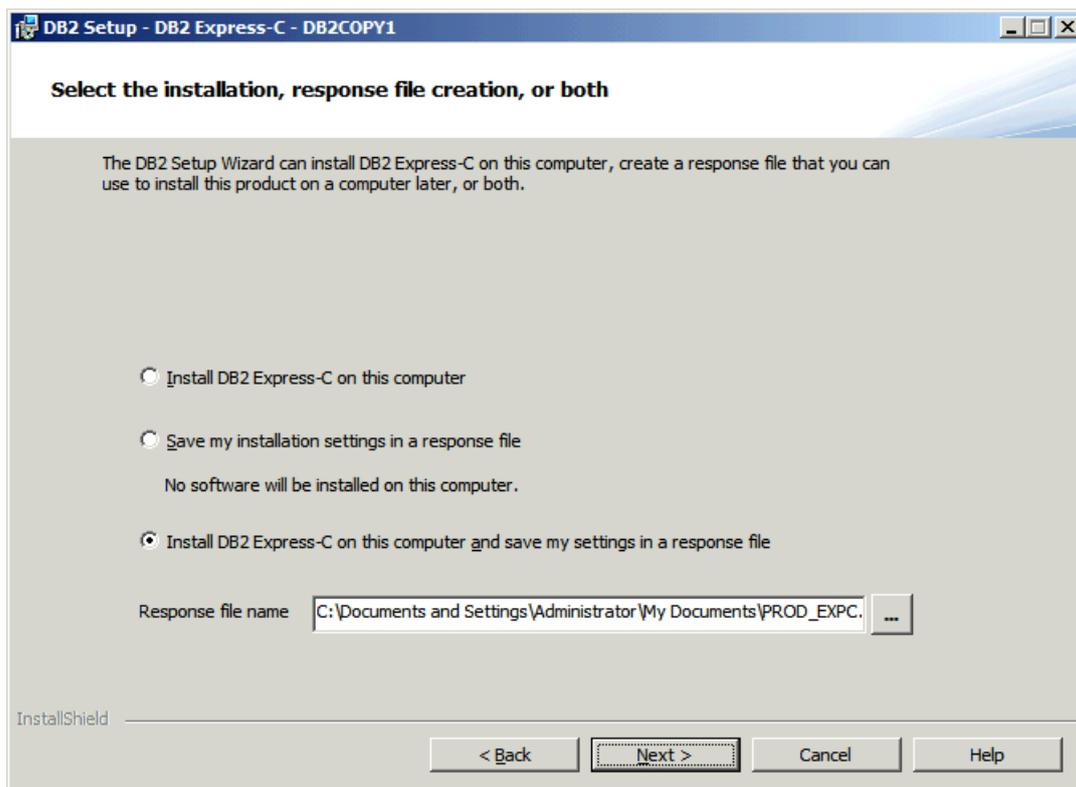


Figura 3.3 – Selecionando a instalação

Selecione os valores padrão das próximas telas. Ao chegar no painel mostrado na *Figura 3.4*, você precisará digitar um ID de usuário, que será usado para configurar e executar a instância e outros serviços.

Se usar um ID de usuário existente, este deve fazer parte do grupo Administrador local do Windows.

Se o ID de usuário não pertencer a um usuário existente, ele será criado como Administrador local. Você pode deixar o campo domínio em branco se o ID de usuário não pertencer a um domínio.

O nome do ID de usuário padrão do Windows é `db2admin`. No caso do Linux, o ID de usuário padrão criado é denominado `db2inst1`.

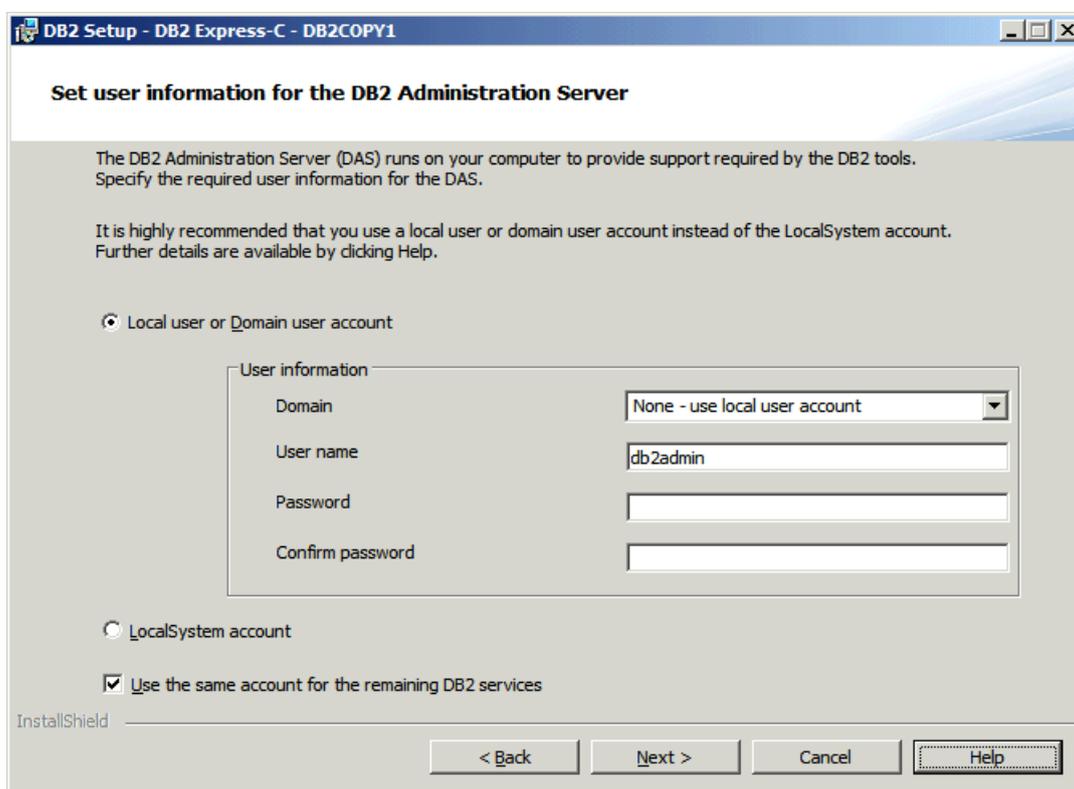


Figura 3.4 – Especificando as informações de usuário do DB2 Administration Server

Finalmente, na *Figura 3.5*, o assistente de instalação exibe um resumo do que será instalado e de todos os valores de configuração diferentes fornecidos nas etapas anteriores. Ao clicar no botão “*Finish*” (Concluir), a instalação é iniciada e os arquivos do programa são colocados em seu sistema.

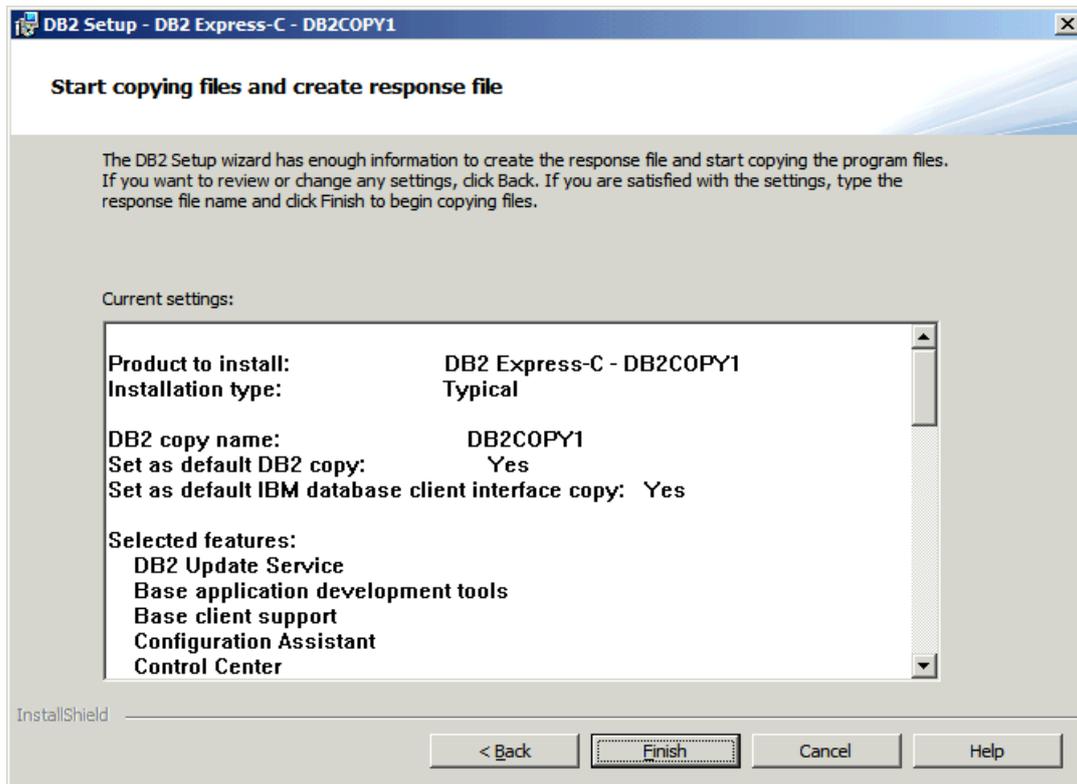


Figura 3.5 – Resumo do que será instalado

Ao terminar a instalação, aparece uma janela semelhante à da *Figura 3.6*, informando os resultados do processo do assistente de instalação além de qualquer outra etapa que seja necessária para concluir a instalação.

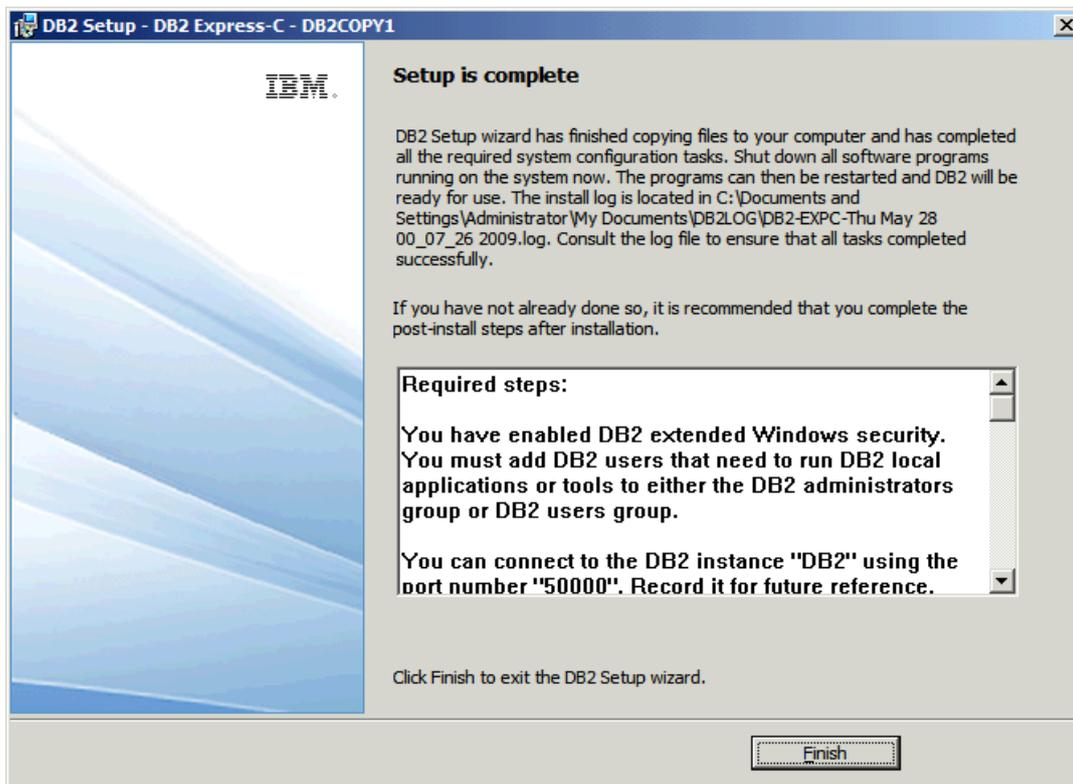


Figura 3.6 – A instalação está completa

Ao clicar em “*Finish*” (Concluir) na janela de resultados de instalação mostrada na *Figura 3.6*, o aplicativo *DB2 First Steps* é iniciado, como mostrado na *Figura 3.7*.

Este pequeno aplicativo descreve as diversas opções diferentes para você começar a usar o DB2, como criar um banco de dados de amostra padrão (apropriadamente denominado **SAMPLE**) ou criar um novo banco de dados. Se você não deseja explorar o DB2 através do *First Steps* no momento, feche a janela e carregue-o posteriormente.

Para iniciar o *DB2 First Steps* manualmente no Windows, selecione *Iniciar -> Programas -> IBM DB2 -> DB2COPY1 (Default) -> Set-up Tools -> First Steps* ou execute o comando `db2fs` na linha de comando.

No Linux, execute o comando `db2fs` em uma janela de terminal.

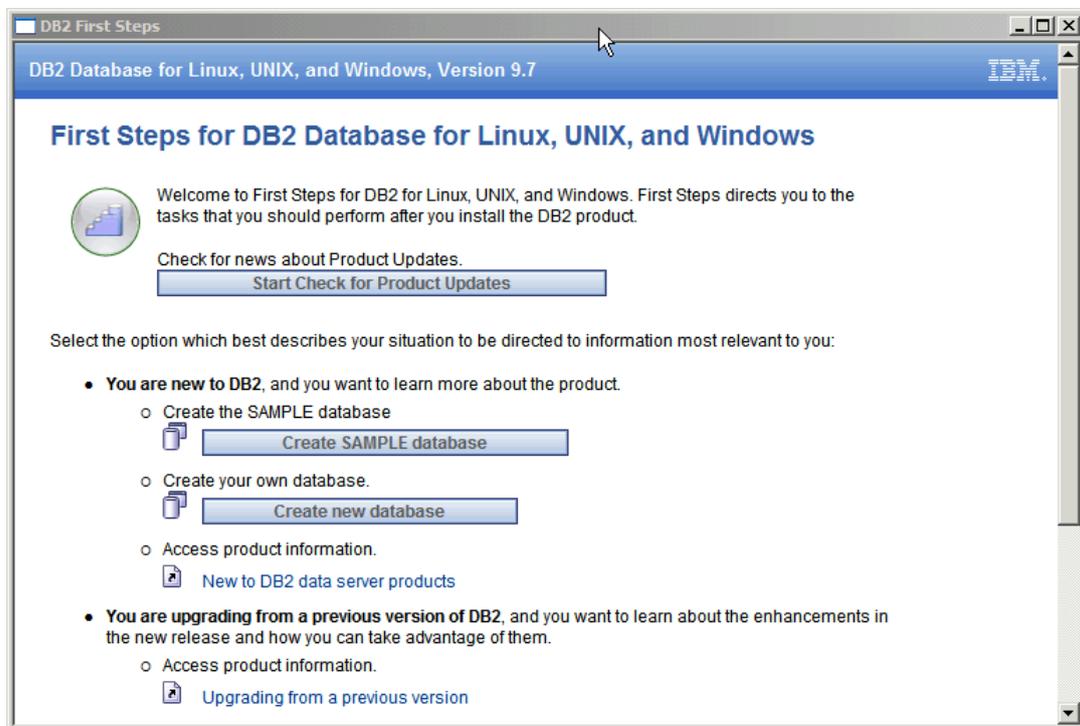


Figura 3.7 – First Steps

3.4 Validação da instalação

Após instalar o DB2, você pode executar três comandos na janela de comando do DB2 (no Windows) ou no terminal (no Linux) para verificar se sua instalação está adequada:

- `db2level`: Este comando exibe as informações sobre o produto DB2 instalado, nível de fix pack e outros detalhes.
- `db2licm -1`: Este comando lista todas as informações de licenciamento específicas aos produtos DB2 instalados

new in
V9.7

- **db2val**: Este é um novo comando disponível no DB2 9.7. Ele valida sua instalação através da verificação da funcionalidade principal de sua cópia do DB2. Isto assegura que suas instâncias estejam consistentes e que a criação de banco de dados e as conexões de banco de dados funcionem.

A Figura 3.8 abaixo mostra um exemplo da saída destes três comandos.

```

C:\Program Files\IBM\SQLLIB\BIN>db2level
DB21085I Instance "DB2" uses "32" bits and DB2 code release "SQL09070" with
level identifier "08010107".
Informational tokens are "DB2 v9.7.0.441", "s090521", "NT3297", and Fix Pack
"0".
Product is installed at "C:\PROGRA~1\IBM\SQLLIB" with DB2 Copy Name "DB2COPY1"

C:\Program Files\IBM\SQLLIB\BIN>db2licm -1
Product name:          "DB2 Express-C"
License type:         "Unwarranted"
Expiry date:          "Permanent"
Product identifier:   "db2expc"
Version information:  "9.7"
Max number of CPUs:  "2"
Max amount of memory <GB>: "2"

C:\Program Files\IBM\SQLLIB\BIN>db2val

DBI1379I The db2val command is running. This can take several minutes.
DBI1333I Installation file validation for the DB2 copy DB2COPY1
was successful.
DBI1339I The instance validation for the instance DB2 was
successful.
DBI1343I The db2val command completed successfully. For details, see
the log file C:\DOCUME~1\ADMINI~1\MYDOCU~1\DB2LOG\db2val-Fri May 29 04_1
40 2009.log.
C:\Program Files\IBM\SQLLIB\BIN>

```

Figura 3.8 – Os comandos **db2level**, **db2licm** e **db2val** para validar sua instalação

Na figura, a saída do comando **db2level** indica que o DB2 9.7 (DB2 v9.7.0.441) com Fix Pack 0 está sendo executado, o que significa que seu código do DB2 está no nível básico (GA) sem nenhuma correção aplicada. O comando **db2licm -1** indica que você instalou a edição DB2 Express-C, que possui uma licença permanente e sem garantia permitindo o uso de até dois núcleos e até 2GB de memória. A saída do comando **db2val** explica-se por si mesma.

Observação:

Se desejar validar a consistência de um banco de dados a qualquer momento, use o utilitário INSPECT.

3.5 Instalação silenciosa

Pode haver situações em que seja necessário instalar um cliente DB2 em diversos computadores, ou então incorporar um servidor de dados do DB2 como parte de seu aplicativo, e você deseja instalá-lo como parte do processo de instalação do seu aplicativo. Nessas situações, a maneira ideal de instalar o DB2 é a instalação silenciosa.

O DB2 ativa a instalação silenciosa através do uso dos arquivos de resposta, que armazenam as informações de instalação como opções de texto simples. A *Listagem 3.1* mostra um trecho de uma amostra de arquivo de resposta.

```
PROD=UDB_EXPRESS_EDITION
LIC_AGREEMENT=ACCEPT
FILE=C:\Program Files\IBM\SQLLIB\
INSTALL_TYPE=TYPICAL
LANG=EN
INSTANCE=DB2
DB2.NAME=DB2
DEFAULT_INSTANCE=DB2
DB2.SVCENAME=db2c_DB2
DB2.DB2COMM=TCPIP
...
```

Listagem 3.1 – Uma amostra de arquivo de resposta

Há diversas maneiras de gerar um arquivo de resposta:

- Instale o DB2 Express-C uma vez em um computador usando o assistente de instalação do DB2. Uma das primeiras opções do assistente (como mostrado na *Figura 3.3*) permite selecionar a opção de salvar suas respostas de instalação em um arquivo de respostas. Ao final da instalação, o assistente gera um arquivo de resposta em um diretório e com um nome de arquivo especificados. Este é um arquivo de texto que pode ser editado manualmente mais tarde.
- Edite a amostra de arquivo de resposta fornecido com a imagem do DB2 Express-C. Esta amostra de arquivo (indicado pela extensão de arquivo .rsp) está localizada no diretório `db2/platform/samples/`
- No Windows, você também pode usar o comando gerador de arquivo de resposta:

```
db2rspgn -d <output directory>
```

Finalmente, para instalar o DB2 silenciosamente usando um arquivo de resposta, execute o comando abaixo no Windows:

```
setup -u <response filename>
```

No Linux, execute o comando:

```
db2setup -r <response filename>
```

3.6 Resumo

Este capítulo cobriu os detalhes da instalação do DB2 Express-C. Esta edição do DB2 está disponível em Linux, Solaris e nas diversas versões do Windows, e pode ser executada nas arquiteturas de 32 bits, 64 bits e Power PC. Após discutir a autoridade de usuário necessária para instalar o DB2 em um sistema, apresentamos uma descrição de uma instalação simples usando a GUI do assistente de instalação do DB2. Em seguida, discutimos as atividades posteriores à instalação, incluindo a execução do DB2 First Steps e a validação da instalação. Finalmente, examinamos como criar e executar uma instalação silenciosa usando os arquivos de resposta do DB2.

3.7 Exercícios

Neste exercício, você instalará o DB2 Express-C e criará o banco de dados SAMPLE (AMOSTRA)

Objetivo

Antes de começar a explorar as funções e ferramentas fornecidas com o DB2 Express-C, você deve instalá-lo em seu sistema. Neste exercício, você irá executar uma instalação básica do DB2 Express-C em Windows. O mesmo assistente de instalação está disponível para Linux, portanto as etapas são muito parecidas nesta outra plataforma.

Procedimento

1. Obtenha as imagens do DB2 Express-C: Faça o download da imagem do DB2 Express-C no site do DB2 Express-C (www.ibm.com/db2/express). Descompacte os arquivos em qualquer diretório que desejar.
2. Localize os arquivos: Navegue até o diretório (ou unidade) contendo os arquivos de instalação do produto DB2.
3. Execute o DB2 Setup Launchpad: Para iniciar o DB2 Setup Launchpad, dê um clique duplo no arquivo `setup.exe`. No Linux, execute o comando `db2setup` na raiz. No Launchpad, clique na opção *Install Product* (Instalar produto) no painel esquerdo da janela.
4. Execute o assistente de configuração do DB2: O assistente de configuração do DB2 verifica se todos os requisitos de sistema foram atendidos e se há alguma instalação do DB2 anterior. Clique em *Install New* (Instalar novo) para iniciar o assistente e clique em *Next* (Avançar).
5. Leia o contrato de licença: Leia e aceite o contrato de licença (selecione o botão *I Accept...* (Eu aceito)) e clique no botão *Next* para continuar.
6. Selecione o tipo de instalação: Neste exercício, selecione a opção *Typical* (Típica, esta é a opção padrão). A opção *Compact* (Compacta) realiza uma instalação básica, enquanto a opção *Custom* (Personalizada) permite personalizar as funções específicas que você deseja instalar. Clique no botão *Next* para continuar.

7. Selecione a instalação, a criação de arquivo de resposta ou ambas: Deixe no padrão para que o DB2 seja instalado e também se crie um arquivo de resposta. Clique no botão *Next* para continuar.
8. Selecione a pasta de instalação: Esta tela permite personalizar a unidade e o diretório onde o código do DB2 será instalado em seu sistema. Verifique se há espaço suficiente para a instalação. Neste exemplo, use a configuração padrão de unidade e diretório (mostrada abaixo):

Drive: C:

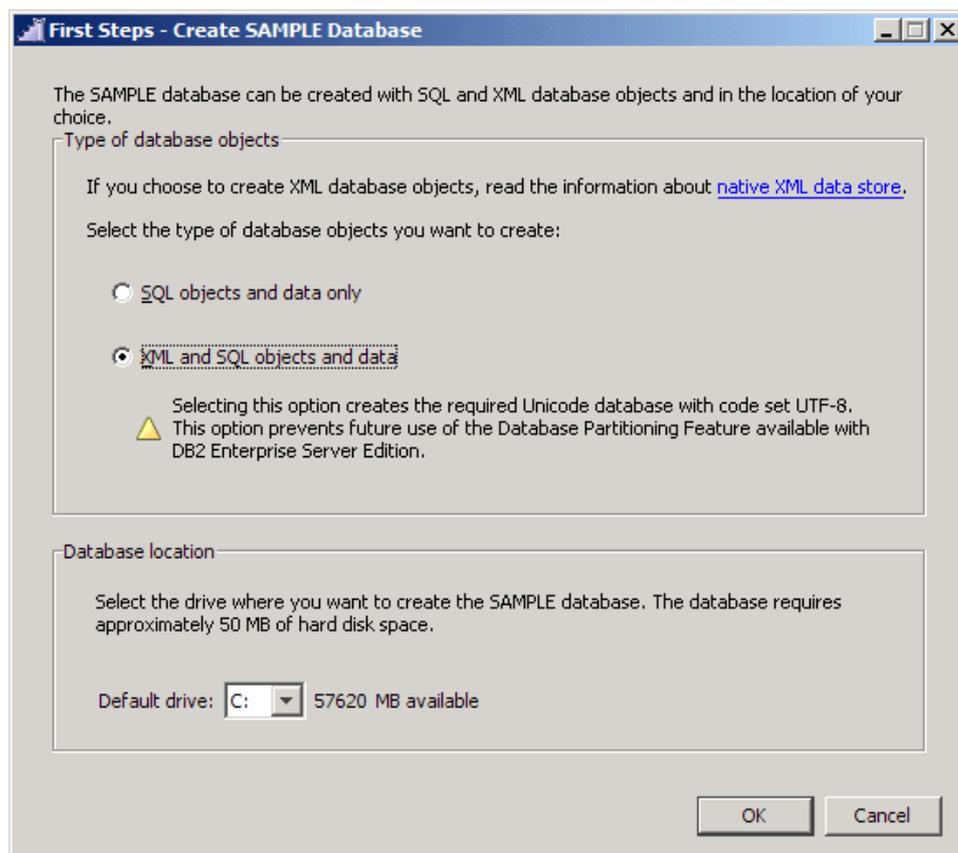
Directory: C:\Program Files\IBM\SQLLIB

Clique no botão *Next* para continuar.

9. Configure as informações do usuário: Após a instalação do DB2 Express-C, alguns processos do DB2 são executados como serviços do sistema. Esses serviços exigem uma conta no sistema operacional para serem executados. No ambiente Windows, é recomendável usar a conta de usuário `db2admin` padrão. Caso a conta de usuário não exista ainda, o DB2 irá criá-la no sistema operacional. Você também pode especificar o uso de uma conta existente, mas esta deve ter autoridade de administrador local. Recomendamos usar os padrões sugeridos. Procure especificar uma senha para a conta. No Linux, use o ID de usuário padrão `db2inst1` do proprietário da instância, `db2fenc1`, para o usuário separado, e `dasusr1` para o usuário do DB2 Administration Server. Clique no botão *Next* para continuar.
10. Configure a instância do DB2: Imagine a instância do DB2 como um contêiner para bancos de dados. É necessário que exista uma instância antes de poder criar um banco de dados dentro dela. Durante a instalação no Windows, é criada automaticamente uma instância denominada `DB2`. Em um ambiente Linux, o nome da instância padrão é `db2inst1`. Abordaremos as instâncias mais adiante neste livro.

Como padrão, a instância do DB2 é configurada para identificar conexões TCP/IP na porta 50000. O protocolo padrão e a porta podem ser alterados, bastando clicar no botão *Configure* (Configurar). Recomendamos usar a configuração padrão neste exemplo. Clique no botão *Next* para continuar.
11. Inicie a instalação: Leia as opções do resumo de instalação selecionado anteriormente. Clique no botão *Finish* (Concluir) para começar a copiar os arquivos para o local de instalação. O DB2 também executa alguns processos de configuração iniciais.
12. First Steps. Após concluir a instalação é exibido outro utilitário de inicialização, denominado First Steps. O First Steps também pode ser iniciado posteriormente, com o comando `db2fs`.

13. O banco de dados **SAMPLE** é um banco de dados que você poderá usar para fins de teste. Ele pode ser criado através do First Steps, clicando no botão *Create SAMPLE database* (Criar banco de dados SAMPLE). Clique neste botão e a janela abaixo será aberta. Selecione a segunda opção (*XML and SQL objects and data*) (Objetos e dados XML e SQL). O banco de dados SAMPLE também pode ser criado com o comando `db2samp1 -xml -sql`.



14. Após alguns minutos, você pode confirmar a criação do banco de dados. Abra a ferramenta DB2 Control Center, selecionando: *Iniciar -> Programas -> IBM DB2 -> DB2COPY1 (Default) -> General Administration Tools -> Control Center*. Você também pode iniciar o Control Center com o comando `db2cc`. Na primeira vez em que abrir o Control Center, uma janela pop-up solicitará que você escolha a visualização do Control Center que deseja usar. Deixe no padrão (Advanced) e clique em *OK*. No painel da esquerda, vá até a pasta *All Databases* (Todos os bancos de dados). Se não conseguir ver o banco de dados SAMPLE nesta pasta, atualize a visualização, selecionando *View -> Refresh* (Visualização -> Atualizar).
15. Reinicie o computador. Esta etapa é opcional. Embora esta etapa não seja mencionada na documentação oficial de instalação do DB2, recomendamos reiniciar o sistema (se possível, pelo menos no Windows) para assegurar que

todos os processos sejam iniciados com sucesso e para liberar qualquer recurso de memória que não tenha sido liberado corretamente.

16. Valide sua instalação do DB2, executando os comandos: **db2level**, **db2licm** e **db2va1**. No menu Iniciar do Windows, abra a Janela de comando do DB2: *Iniciar -> Programas -> IBM DB2 -> DB2COPY1 (Default) -> Command Line Tools -> Command Window*. Na Command Window (Janela de comando) (ou no shell do Linux) digite **db2level** e examine a saída. Faça o mesmo com o comando **db2licm -1**. Em seguida, execute o comando **db2va1**. Se o comando **db2va1** for executado com sucesso, sua instalação está correta! Se houver algum erro, verifique os detalhes no arquivo de log especificado na mensagem de erro. A saída dos três comandos deve ser semelhante às mostradas anteriormente na *Figura 3.8*.

4

Capítulo 4 – Ambiente do DB2

O ambiente do DB2 inclui objetos de bancos de dados e arquivos de configuração diferentes. A *Figura 4.1* fornece uma visão geral dos diversos comandos e ferramentas para trabalhar com o DB2, e também destaca o ambiente do DB2 no lado direito. Esta é a área de foco deste capítulo. O lado esquerdo da figura mostra os diversos comandos do DB2 e as instruções SQL, SQL/XML e XQuery que podem ser executadas para interagir com um servidor de dados DB2. O meio da figura mostra os nomes das diferentes ferramentas usadas para interagir com um servidor de dados DB2.

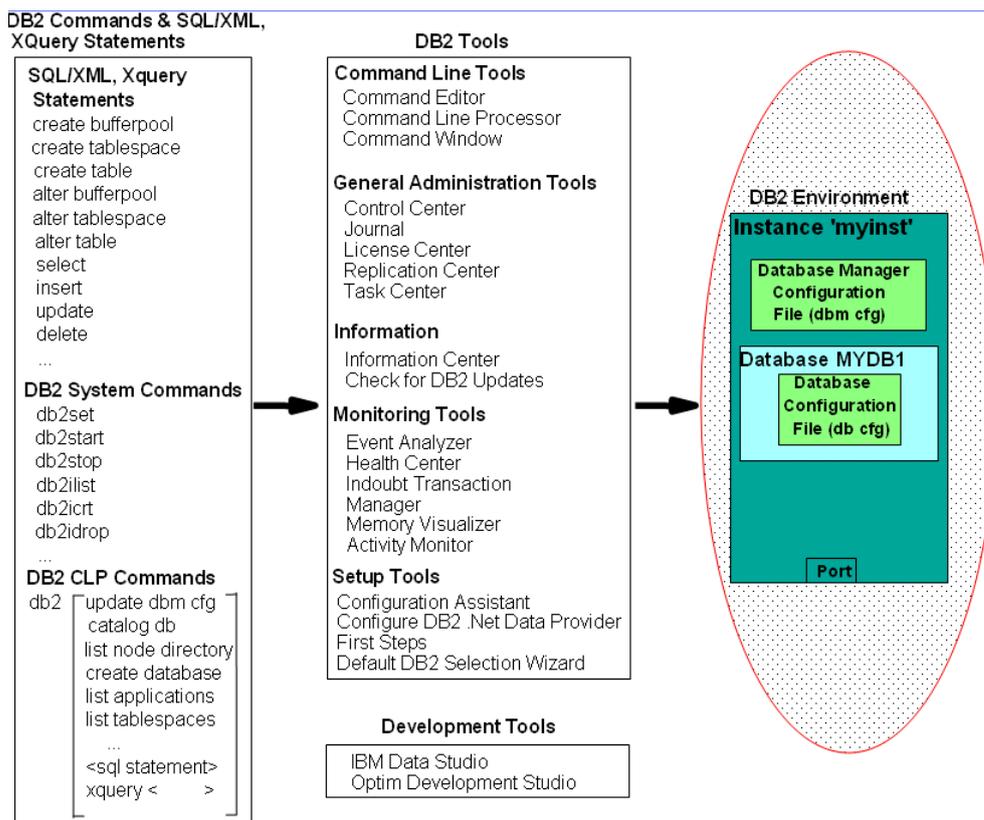


Figura 4.1 – O quadro geral do DB2: ambiente do DB2

Observação:

Assista às apresentações em vídeo sobre o Ambiente do DB2 nestes links:

<http://www.channeldb2.com/video/video/show?id=807741:Video:4029>

<http://www.channeldb2.com/video/video/show?id=807741:Video:4042>

Para descrever o ambiente do DB2, vamos descrever cada elemento componente passo a passo. A *Figura 4.2* mostra uma representação de um servidor de dados DB2 após a instalação do DB2 Express-C 9.7.

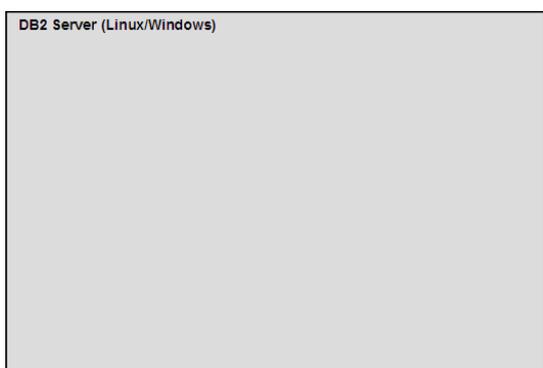


Figura 4.2 – Representação de um servidor DB2 após instalar o DB2 Express-C 9.7

Como parte da instalação no Windows, é criada uma instância padrão denominada **DB2** (**db2inst1** no Linux). Esta é representada pela caixa verde no lado esquerdo da *Figura 4.3*. Uma instância é simplesmente um ambiente independente onde se pode executar aplicativos e criar bancos de dados. Você pode criar diversas instâncias em um servidor de dados e usá-las para finalidades diferentes. Por exemplo, uma instância pode ser usada para conter bancos de dados para uso em produção, outra instância pode ser usada como ambiente de teste de bancos de dados e outra como ambiente de desenvolvimento. Todas estas instâncias são independentes, ou seja, executar operações em uma instância não afeta as outras.

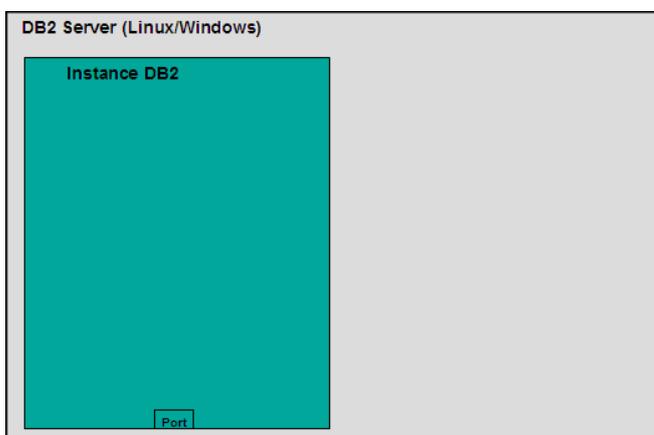


Figura 4.3 – A instância padrão do DB2 criada

Para criar uma nova instância do DB2, use o comando `db2icrt <instance name>`, onde `<instance name>` é substituído por qualquer nome com 8 caracteres. Por exemplo, para criar a instância `myinst`, usamos este comando: `db2icrt myinst`.

A Figura 4.4 mostra uma nova instância denominada `myinst` como uma caixa verde separada no lado direito.

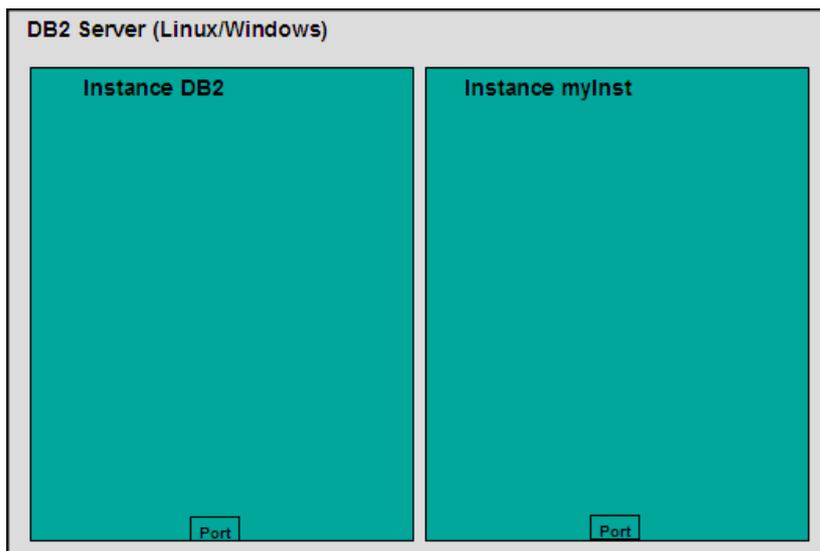


Figura 4.4 – Um servidor DB2 com duas instâncias

Observe que cada instância possui um número de porta exclusivo. Isto ajuda a distinguir entre as instâncias quando você deseja se conectar, a partir de um cliente remoto, a um banco de dados em uma dada instância, via TCP/IP. Ao usar a Janela de comando do DB2 você pode tornar qualquer instância do DB2 ativa, usando este comando de sistema operacional no Windows:

```
set db2instance=myinst
```

Observe que não deve haver espaços em branco antes ou depois do símbolo de igualdade (=). Neste exemplo, se fosse criado um banco de dados a partir da Janela de comando, ele seria criado na instância `myinst`.

Para listar as instâncias presentes no sistema, execute o comando:

```
db2ilist
```

No Linux, uma instância deve corresponder a um usuário do sistema operacional Linux. Portanto, para alternar entre instâncias basta alternar entre usuários. Este usuário é conhecido como proprietário da instância. Você pode sair e fazer login como usuário da outra instância, ou usar o comando `su`.

A Tabela 4.1 mostra alguns comandos úteis no nível de instância.

Comando	Descrição
<code>db2start</code>	Inicia a instância atual
<code>db2stop</code>	Interrompe a instância atual
<code>db2icrt</code>	Cria uma nova instância
<code>db2idrop</code>	Descarta uma instância
<code>db2ilist</code>	Lista as instâncias presentes no sistema
<code>db2 get instance</code>	Lista a instância ativa no momento

Tabela 4.1 – Comandos úteis do DB2 no nível de instância

Alguns dos comandos acima podem, como alternativa, ser executados através do Control Center. Por exemplo, no Control Center, se você expandir a pasta *Instances* e clicar com o botão direito na instância desejada, você pode selecionar *Start* (Iniciar), o que equivale a executar o comando `db2start` na Janela de comando do DB2 ou *Stop* (Parar), equivalente a executar o comando `db2stop` como mostrado na *Figura 4.5*.

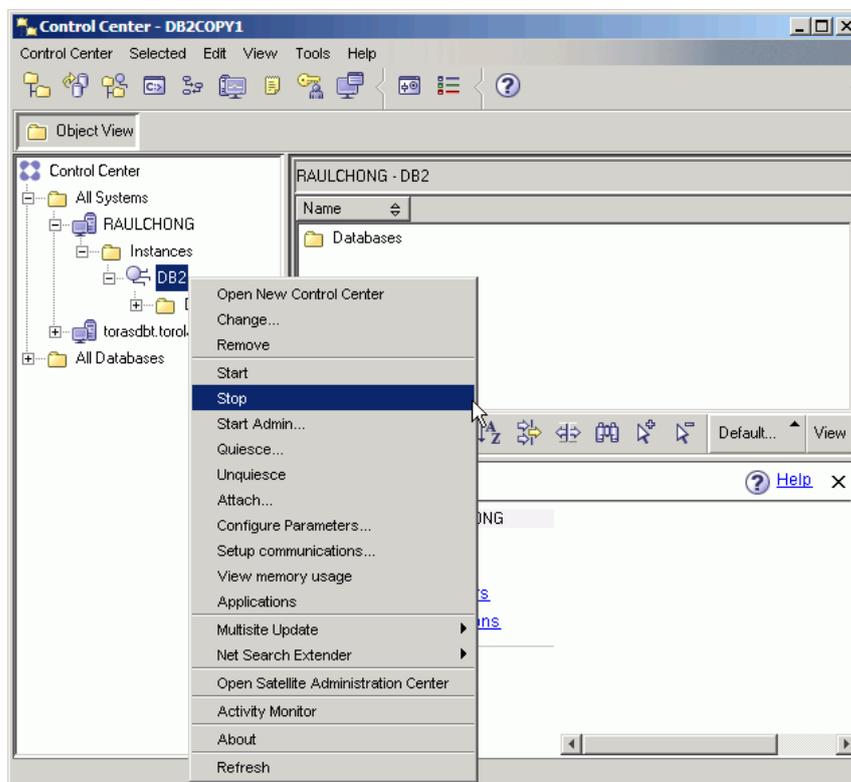


Figura 4.5 – Comandos de instância no Control Center

Para criar um banco de dados na instância ativa, execute este comando na Janela de comando do DB2:

```
db2 create database mydb1
```

Para listar todos os bancos de dados, execute o comando:

```
db2 list db directory
```

É possível criar diversos bancos de dados dentro de qualquer instância. Um banco de dados é uma coleção de objetos como tabelas, visualizações, índices e assim por diante. Os bancos de dados são unidades independentes, portanto não compartilham objetos com outros bancos de dados. A *Figura 4.6* mostra a representação do banco de dados **MYDB1** criado dentro da instância **DB2**.

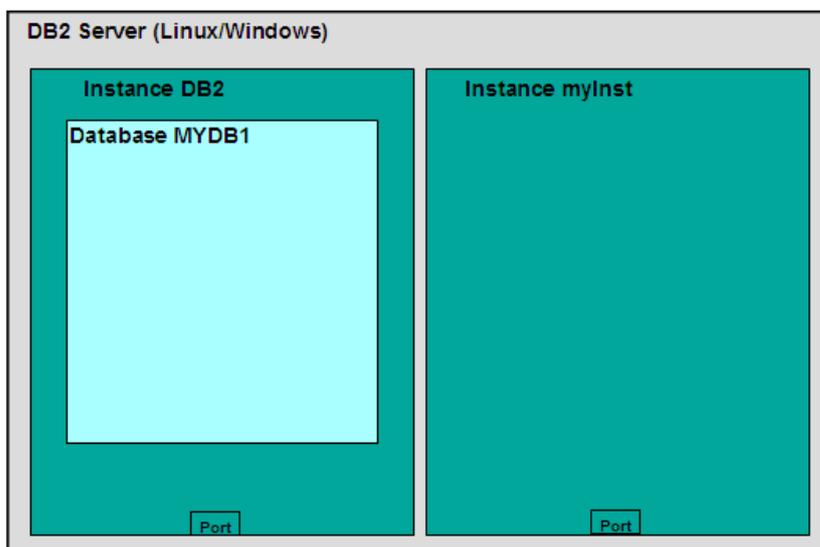


Figura 4.6 – Banco de dados MYDB1 criado na instância DB2

A *Tabela 4.2* exibe alguns comandos que podem ser usados no nível do banco de dados.

Comando/instrução SQL	Descrição
db2 create database	Cria um novo banco de dados
db2 drop database	Descarta um banco de dados
db2 connect to <database_name>	Conecta-se a um banco de dados
db2 create table/create view/create index	Instruções SQL para criar tabelas, visualizações e índices, respectivamente

Tabela 4.2 – Comandos/instruções SQL no nível de banco de dados

Se desejarmos criar outro banco de dados com o mesmo nome (**MYDB1**) porém na instância **myinst**, é necessário executar os seguintes comandos na Command Window (Janela de commando) do DB2:

```
db2 list db directory
set db2instance=myinst
db2 create database mydb1
set db2instance=db2
```

A *Figura 4.7* mostra o novo banco de dados **MYDB1** criado na instância **myinst**.

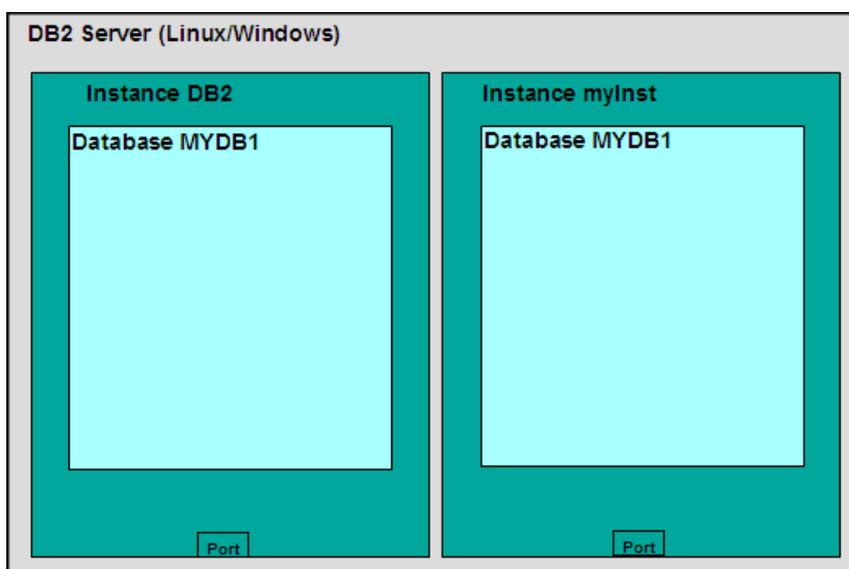


Figura 4.7 – Banco de dados MYDB1 criado na instância myInst

Quando um banco de dados é criado, são criados diversos objetos como padrão: tablespaces, tabelas, um bufferpool e arquivos de log. A criação desses objetos demora um pouco; é por isso que o comando `create database` requer alguns minutos de processamento. A *Figura 4.8* mostra três tablespaces criados como padrão no lado esquerdo da figura. Os tablespaces serão discutidos com mais detalhes no *Capítulo 6, Arquitetura do DB2*; entretanto, no momento, imagine os tablespaces como uma camada lógica entre as tabelas e os recursos físicos, como os discos e a memória.

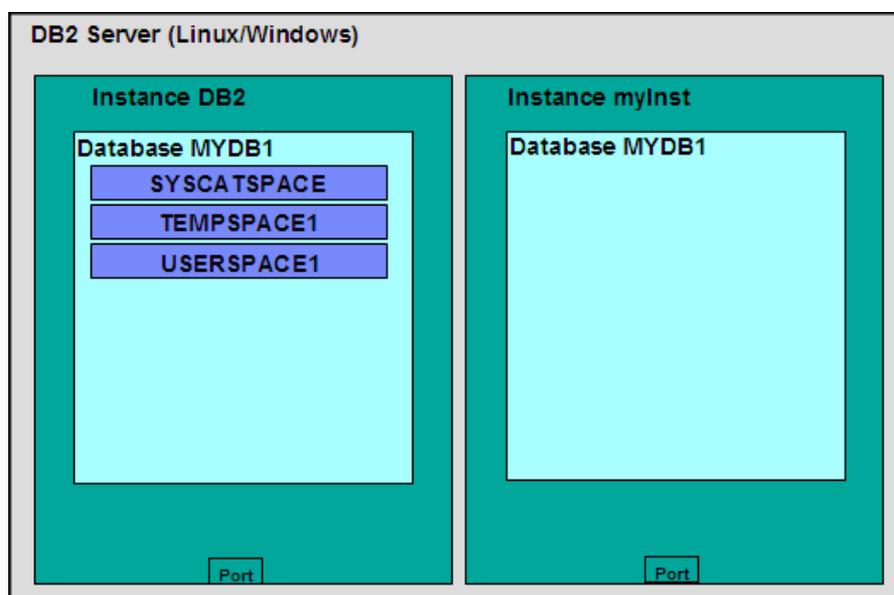


Figura 4.8 – Tablespaces criados como padrão quando um banco de dados é criado

O tablespace **SYSCATSPACE** contém as tabelas do Catálogo do sistema. O Catálogo do sistema é denominado dicionário de dados em outros sistemas de gerenciamento de banco de dados relacional. Basicamente, ele contém as informações de sistema que não devem ser modificadas ou excluídas; caso contrário, o banco de dados não funcionará corretamente. O tablespace **TEMPSPACE1** é usado pelo DB2 quando necessita de espaço adicional para executar algumas operações, como as classificações. O tablespace **USERSPACE1** normalmente é usado para armazenar as tabelas de banco de dados do usuário, caso não seja especificado nenhum espaço em tabela ao criar uma tabela.

Você também pode criar seus próprios tablespaces usando a instrução **CREATE TABLESPACE**. A *Figura 4.9* mostra o tablespace **MYTBLS1** criado dentro do banco de dados **MYDB1** na instância **DB2**. Ao criar um tablespace, você especifica os discos que serão usados e a memória (bufferpool) a ser usada. Portanto, se tiver uma tabela “importante”, ou seja, uma tabela usada com muita frequência, você pode alocar os discos mais rápidos e a maior quantidade de memória, atribuindo um tablespace com estas características.

Na *Figura 4.9* mostramos mais dois objetos que são criados como padrão: Um bufferpool denominado **IBMDEFAULTBP** e os arquivos de log.

Um bufferpool é basicamente um cache de memória usado pelo banco de dados. Você pode criar um ou mais bufferpools, mas sempre deve haver um bufferpool com tamanho de página que corresponda ao tamanho de página dos tablespaces existentes. As páginas e tamanhos de página serão discutidos em mais detalhes no *Capítulo 6, Arquitetura do DB2*.

Os arquivos de log são usados para recuperação. Ao trabalhar com um banco de dados, as informações não são apenas armazenadas nos discos do banco de dados; durante o

trabalho, os arquivos de log armazenam todas as operações executadas com os dados. Imagine que os logs são arquivos temporários onde é executada uma operação de “salvar automaticamente”. Os logs são discutidos em mais detalhes no *Capítulo 11, Backup e recuperação*.

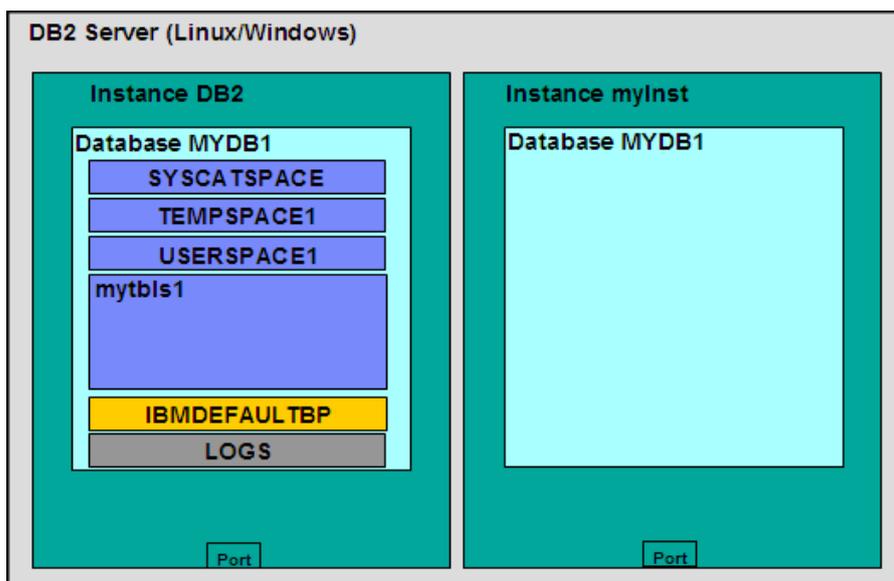


Figura 4.9 – Bufferpool e logs criados como padrão

Já discutimos que as instâncias são ambientes independentes e, portanto, é possível criar um banco de dados com o mesmo nome em diversas instâncias. Assim como as instâncias, os bancos de dados também são unidades independentes. Portanto, os objetos de um banco de dados não têm qualquer relacionamento com os objetos de outro banco de dados. A *Figura 4.10* mostra um tablespace *mytbls1* dentro do banco de dados *MYDB1* e do banco de dados *SAMPLE*, dentro da instância *DB2*. Isto é válido porque os bancos de dados são unidades independentes. Observe que a *Figura 4.10* não mostra os outros objetos padrão do banco de dados *SAMPLE* devido às limitações de espaço da figura.

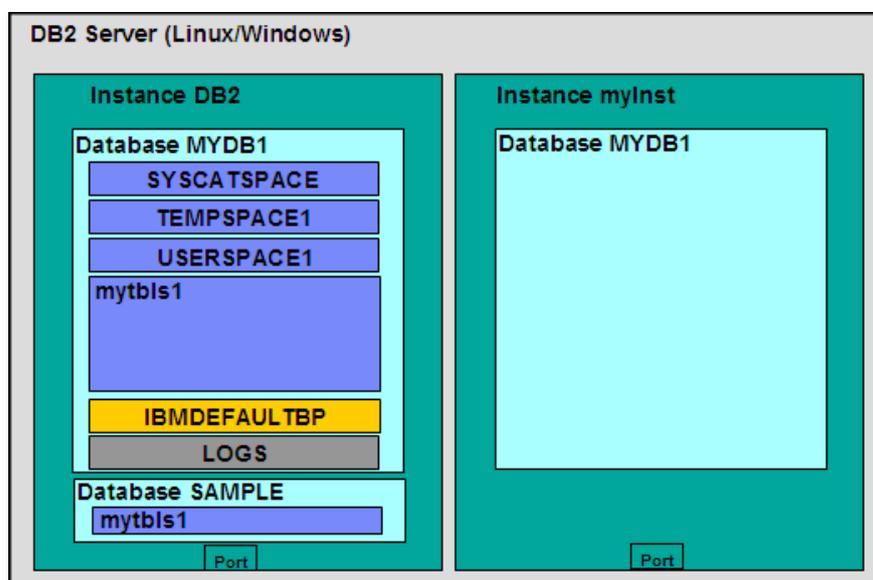


Figura 4.10 – Tablespaces com o mesmo nome em diferentes bancos de dados.

Depois de criar um tablespace, você pode criar objetos dentro do tablespace, como tabelas, visualizações e índices. Isso é ilustrado na *Figura 4.11*.

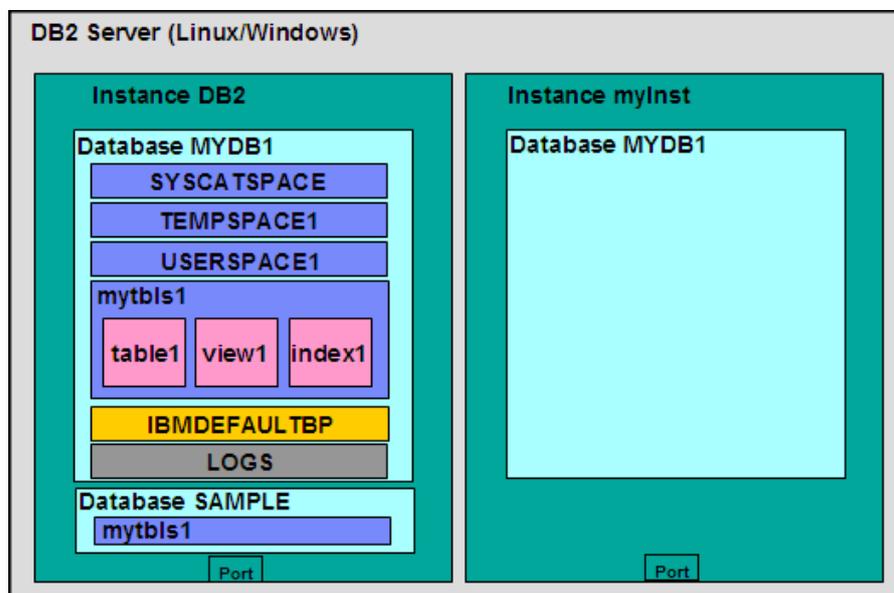


Figura 4.11 – Tabelas, visualizações, índices criados dentro do tablespace

4.1 Configuração do DB2

Os parâmetros do DB2 podem ser configurados com a ferramenta Configuration Advisor. Para acessar o Configuration Advisor através do Control Center, clique com o botão direito

em um banco de dados e selecione *Configuration Advisor*. O Configuration Advisor fornecerá uma lista de parâmetros do DB2 que devem ser alterados e os valores sugeridos para cada um, com base em suas respostas a algumas perguntas sobre os recursos do sistema e carga de trabalho. Se você desejar mais detalhes sobre a configuração do DB2, continue lendo este livro; caso contrário, use o Configuration Advisor e você já está pronto para trabalhar com o DB2!

Um servidor do DB2 pode ser configurado em quatro níveis diferentes:

- Variáveis de ambiente
- Arquivo de configuração de gerenciador de banco de dados (dbm cfg)
- Arquivo de configuração de banco de dados (db cfg)
- Registro de perfil do DB2

Isto também é ilustrado na *Figura 4.12*. Na figura, observe onde estão localizadas as caixas. Por exemplo, as variáveis de ambiente são definidas no nível de sistema operacional do servidor, enquanto os parâmetros do arquivo de configuração do gerenciador de banco de dados são definidos no nível de instância. Os parâmetros de configuração do banco de dados são definidos no nível de banco de dados e o registro de perfil do DB2 é definido no nível do sistema operacional ou de instância.

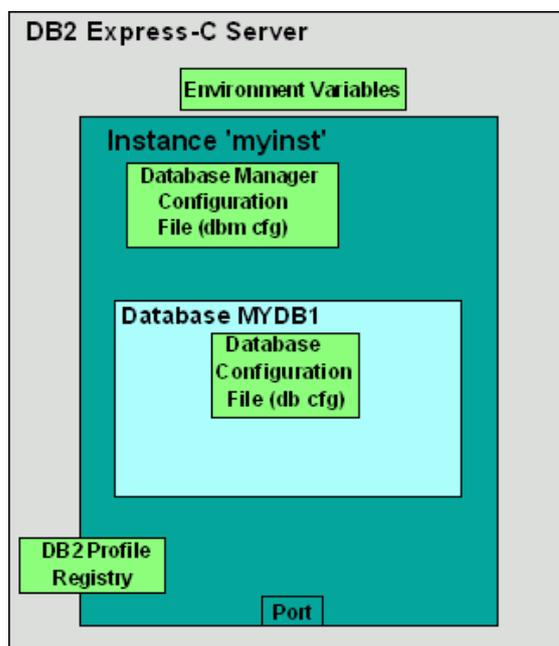


Figura 4.12 – Configuração do DB2

4.1.1 Variáveis de ambiente

As variáveis de ambiente são definidas no nível de sistema operacional. Uma das principais variáveis de ambiente é `DB2INSTANCE`. Esta variável indica a instância ativa na qual você está trabalhando e onde seus comandos do DB2 seriam aplicados. Por

exemplo, para definir a instância ativa para *myinst* no Windows, execute este comando de sistema operacional:

```
set db2instance=myinst
```

4.1.2 Arquivo de configuração de gerenciador de banco de dados (dbm cfg)

O arquivo de configuração de gerenciador de banco de dados (dbm cfg) inclui parâmetros que afetam a instância e todos os bancos de dados nela contidos. O arquivo de configuração de gerenciador de banco de dados pode ser exibido ou modificado através da linha de comando ou do Control Center do DB2.

Para trabalhar com o dbm cfg usando o Control Center, selecione o objeto instância na pasta de instâncias do Control Center, clique com o botão direito para exibir o menu pop-up e selecione *Configure Parameters* (Configurar parâmetros). Isto é mostrado na *Figura 4.13*.

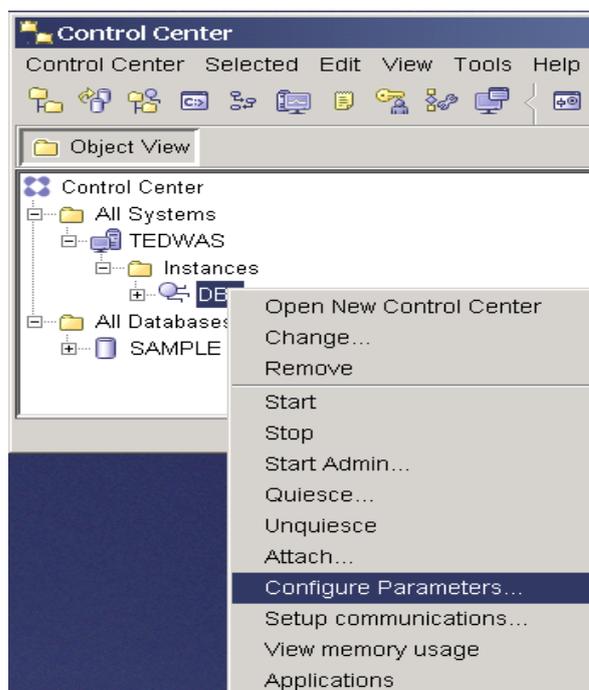


Figura 4.13 – Configuração do dbm cfg no Control Center.

Após selecionar *Configure Parameters*, será exibida a tela mostrada na *Figura 4.14* com a lista de parâmetros do dbm cfg.

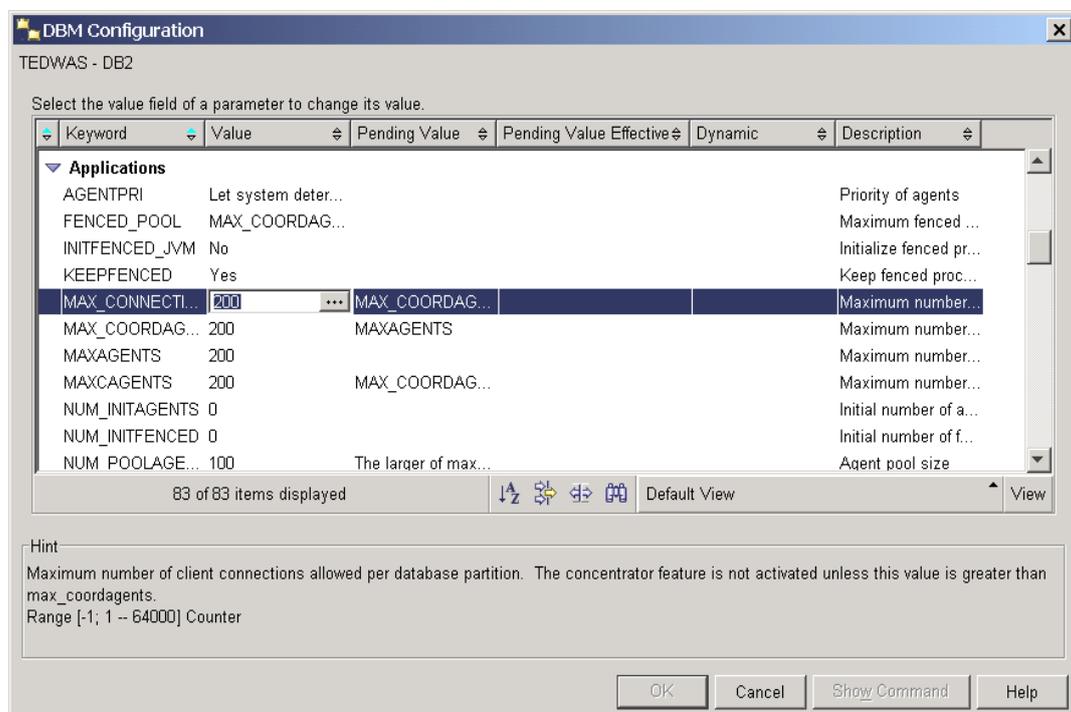


Figura 4.14 – A caixa de diálogo dbm cfg

Muitos parâmetros são dinâmicos, ou seja, as alterações têm efeito imediatamente. Entretanto, as alterações em alguns parâmetros podem exigir que a instância seja interrompida e reiniciada. Isto pode ser feito na Linha de comando, com os comandos **db2stop** e **db2start**.

Antes de interromper uma instância, é necessário desconectar todos os aplicativos. Se você desejar forçar a interrupção da instância, use o comando **db2stop force**.

Uma instância também pode ser interrompida através do Control Center, clicando no objeto instância e selecionando *Stop* (Parar) ou *Start* (Iniciar).

A *Tabela 4.3* mostra alguns comandos úteis para gerenciar o dbm cfg a partir da linha de comando.

Comando	Descrição
db2 get dbm cfg	Recupera informações sobre o dbm cfg
db2 update dbm cfg using <parameter_name> <value>	Atualiza o valor de um parâmetro do dbm cfg

Tabela 4.3 – Comandos para manipular o dbm cfg

4.1.3 Arquivo de configuração de banco de dados (db cfg)

O arquivo de configuração de banco de dados (db cfg) inclui os parâmetros que afetam um banco de dados específico. O arquivo de configuração de banco de dados pode ser exibido ou modificado através da linha de comando ou do Control Center do DB2.

Para trabalhar com o db cfg usando o Control Center, selecione o objeto banco de dados na pasta de bancos de dados do Control Center, clique com o botão direito para exibir o menu pop-up e selecione *Configure Parameters*. Isto é mostrado na *Figura 4.15*.

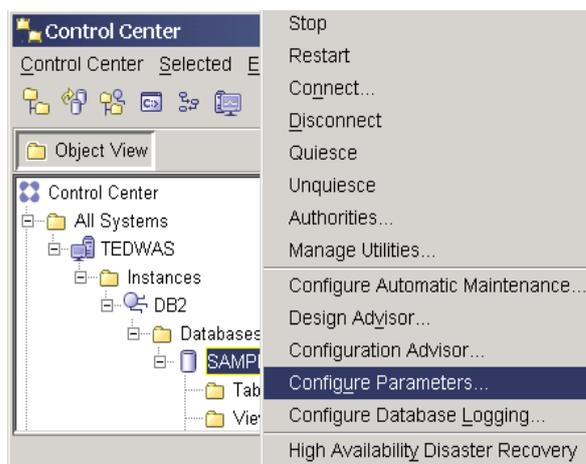


Figura 4.15 – Configuração do db cfg no Control Center.

Após selecionar *Configure Parameters*, será exibida a tela mostrada na *Figura 4.16* com a lista de parâmetros do db cfg.

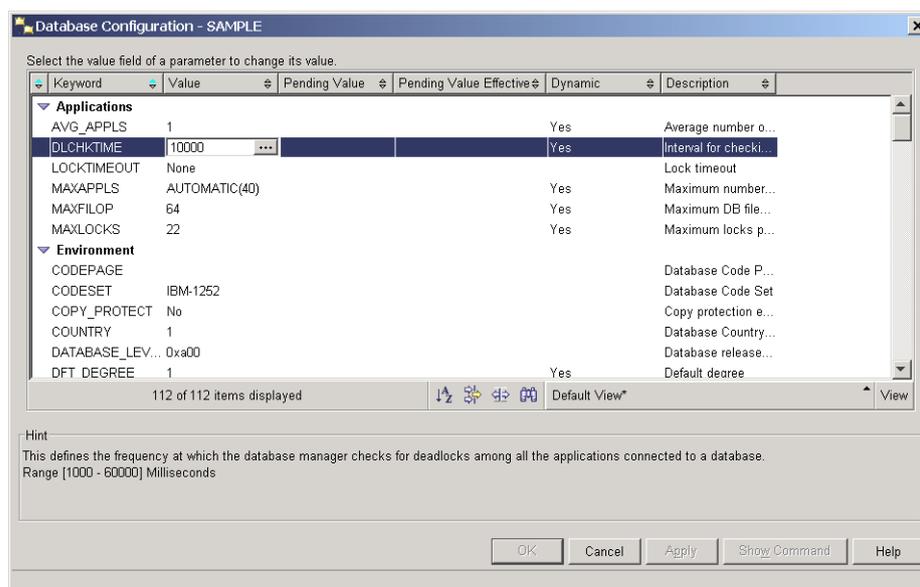


Figura 4.16 –A caixa de diálogo db cfg

A Tabela 4.4 mostra alguns comandos úteis para gerenciar o db cfg a partir da linha de comando.

Comando	Descrição
<code>get db cfg for <database_name></code>	Recupera informações sobre o db cfg para um dado banco de dados
<code>update db cfg for <database_name> using <parameter_name> <value></code>	Atualiza o valor de um parâmetro do db cfg

Tabela 4.4 – Comandos para manipular o db cfg

4.1.4 Registro de perfil do DB2

As variáveis do registro de perfil do DB2 incluem parâmetros que podem ser específicos à plataforma e definidos globalmente (afetando todas as instâncias) ou no nível de instância (afetando uma instância específica).

A Tabela 4.5 mostra alguns comandos úteis para manipular o registro de perfil do DB2

Comando	Descrição
<code>db2set -all</code>	Lista todas as variáveis do registro de perfil do DB2 que estão definidas no momento
<code>db2set -lr</code>	Lista todas as variáveis do registro de perfil do DB2
<code>db2set <parameter>=<value></code>	Atribui um parâmetro com um dado valor

Tabela 4.5 – Comandos para manipular o registro de perfil do DB2

A Tabela 4.6 mostra algumas das variáveis de uso mais comum do registro do DB2

Variável do registro	Descrição
DB2COMM	Especifica os gerenciadores de comunicação iniciados quando o gerenciador de banco de dados é iniciado.
DB2_EXTSECURITY	No Windows, evita o acesso não autorizado ao DB2, bloqueando os arquivos de sistema do DB2
DB2_COPY_NAME	Armazena o nome da cópia do DB2 em uso no momento. Para alternar para uma outra cópia do DB2 instalada, execute o comando <code>installpath\bin\db2envar.bat</code> . Esta variável não pode ser usada para esta finalidade.

Tabela 4.6 – Variáveis do registro de perfil do DB2 comumente usadas

Por exemplo, para permitir a comunicação usando TCP/IP, defina a variável de registro DB2COMM para TCPIP, como mostrado abaixo:

```
db2set db2comm=tcpip
```

4.2 O DB2 Administration Server (obsoleto)

O DB2 Administration Server (DAS) é um processo daemon executado no servidor DB2 para permitir que os clientes remotos administrem o servidor DB2 graficamente. O DAS é necessário somente quando se usar as ferramentas gráficas do DB2, seja local ou remotamente. Há somente um DAS por cada computador físico, como mostrado na *Figura 4.16*.

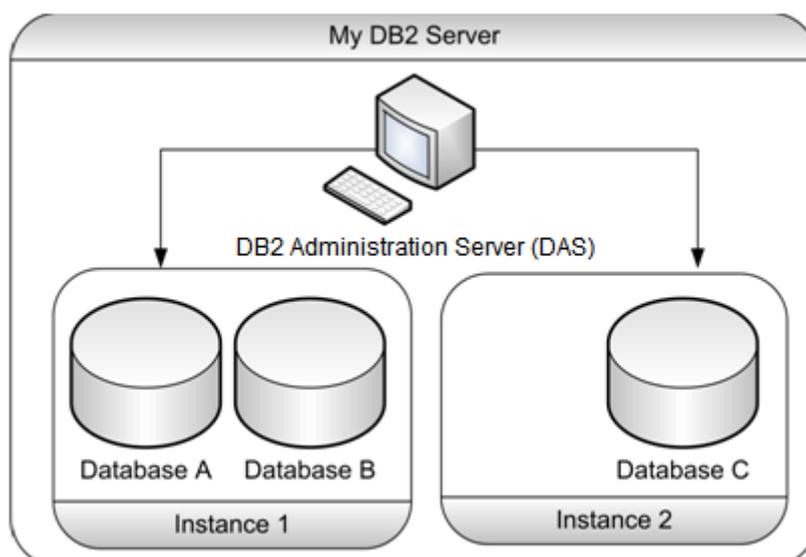


Figura 4.16 – O DB2 Administration Server (DAS)

4.3 Resumo

Neste capítulo, exploramos o ambiente DB2, cobrindo o conceito e a criação de instâncias e banco de dados, juntamente com seus comandos comuns. A partir daí, examinamos os outros aspectos principais das instâncias, incluindo os tablespaces: os três tipos de tablespace disponíveis e as tabelas, visualizações e índices que podem ser criados dentro do tablespace; os bufferpools e os logs.

Finalmente, discutimos a estrutura de configuração do DB2 e como ela pode ser alterada em quatro locais diferentes, através de: variáveis de ambiente; arquivo de configuração de gerenciador de banco de dados; arquivo de configuração de banco de dados, e registro de perfil do DB2.

4.4 Exercícios

Os exercícios desta seção permitirão que você explore os conceitos discutidos neste capítulo e apresentarão a você algumas ferramentas do DB2.

Parte 1: Criar um novo banco de dados usando o assistente Create Database

Nesta parte, você criará um banco de dados usando o assistente Create Database no Control Center.

Procedimento

1. No painel Object View (Visualização de objeto) no lado esquerdo do Control Center, clique com o botão direito na pasta *All Databases* (Todos os bancos de dados) e selecione *Create Database -> With Automatic Maintenance* (Criar banco de dados -> Com manutenção automática). Isto inicia o assistente Create Database.
2. Especifique o nome e local do banco de dados na página *Name* (Nome) do assistente. Use os valores abaixo:

```
Database Name:           EXPRESS
Default Drive (Windows): C:
Default Path: (Linux):   /home/db2inst1
Alias:                   This will default to EXPRESS if left blank
Comment:                 This is optional and can be left blank
```

Clique no botão *Next* para continuar com a próxima página do assistente.



Observação: No Windows, como padrão você pode criar um banco de dados somente em uma unidade e não em um caminho. Se desejar criar um banco de dados em um caminho, defina a variável de registro do DB2 DB2_CREATE_DB_ON_PATH

3. Na página *Storage* (Armazenamento), não faça qualquer alteração e clique em *Next*.
4. Na página *Maintenance* (Manutenção), deixe a opção padrão (*Yes, I can specify an offline ...*) e clique em *Next*.
5. Especifique a janela de tempo de manutenção off-line na página *Timing* (Tempo) do assistente. Configure a janela para iniciar à 01:00 toda segunda-feira até quinta-feira, com duração de 6 horas. Clique no botão *Next* para continuar.
6. Configure a notificação na página *Mail Server* (Servidor de email) do assistente. O DB2 pode avisar alguém ou enviar um email automaticamente se detectar um problema ou condição de falha. Se desejar configurar isto, indique um servidor SMTP disponível para ser usado pelo DB2 para enviar email. Neste exercício não temos um servidor SMTP, portanto deixe em branco e clique em *Next*.

Revise as opções selecionadas na página *Summary* (Resumo) do assistente. Clique no botão *Finish* (Concluir) para iniciar o processo de criação do banco de dados. A criação de banco de dados normalmente demora alguns minutos, sendo exibido um indicador de progresso durante este período.

Parte 2: Trabalhando com instâncias, bancos de dados e configurações

Nesta parte, você irá criar uma nova instância, banco de dados e alterar os parâmetros de configuração em um servidor DB2 no Windows. Você pode fazê-lo do Control Center ou na Janela de comando do DB2. Fornecemos as instruções usando a Janela de comando do DB2.

Procedimento

1. Abra a janela de comando, selecionando *Iniciar -> Programas -> IBM DB2 -> DB2COPY1 (Default) -> Command Line Tools -> Command Window*. Como alternativa, a maneira mais rápida de iniciá-la é selecionar *Iniciar -> Executar* e digitar **db2cmd**.
2. Na Janela de comando do DB2, crie uma nova instância denominada **newinst**

```
db2icrt newinst
```
3. Alterne para a instância **newinst** e verifique se esta é, de fato, sua instância atual. Em seguida, inicie a instância.

```
set db2instance=newinst (Dica: sem espaços antes ou depois do sinal "=" !)  
db2 get instance (Isto confirma que newinst é a instância atual)  
db2start
```
4. Crie um banco de dados **newdb** com valores padrão na instância **newinst**. Isto demora alguns minutos, pois o DB2 cria objetos internos no banco de dados e realiza algumas configurações iniciais.

```
db2 create database newdb
```

5. Conecte-se e desconecte-se do novo banco de dados *newdb*. Em seguida, descarte-o.

```
db2 connect to newdb
db2 terminate
db2 drop db newdb
```

6. Interrompe a instância atual *newinst*

```
db2stop
```

7. Liste todas as instâncias no seu servidor

```
db2ilist
```

8. Alterne para a instância do DB2 e verifique se realmente alternou

```
set db2instance=db2
db2 get instance
```

9. Descarte a instância *newinst*

```
db2idrop newinst
```

10. Encontre o valor atual do parâmetro FEDERATED do dbm cfg. Ele deve ter valor NO como padrão

```
db2 get dbm cfg
```

Dica: No Linux, você pode usar: `db2 get dbm cfg | grep FEDERATED`

11. Altere o valor do parâmetro FEDERATED de dbm cfg para YES e confirme se a alteração ocorreu.

```
db2 update dbm cfg using FEDERATED YES
```

Como FEDERATED não é um parâmetro dinâmico, as alterações não entram em efeito até você interromper e iniciar a instância. Entretanto, para interromper a instância, você deve assegurar que não haja conexões. Uma maneira de fazer isto é executar os comandos:

```
db2 force applications all
db2 terminate
```

Reinicie a instância e confirme o novo valor de FEDERATED:

```
db2stop
db2start
db2 get dbm cfg
```

12. Conecte-se ao banco de dados SAMPLE com o ID de usuário e senha que está usando no sistema operacional

```
db2 connect to sample user <userID> using <password>
```

13. Verifique quantos aplicativos estão sendo executados em sua instância atual

```
db2 list applications
```

14. Abra outra Janela de comando do DB2 e conecte-se novamente ao banco de dados SAMPLE, sem especificar um ID de usuário e senha. Em seguida, verifique quantas conexões existem agora.

```
db2 connect to sample
db2 list applications
```

15. Force a desconexão de uma das janelas de comando do DB2. Isto fornece um exemplo de como um DBA pode encerrar à força o trabalho de um determinado usuário (que provavelmente está sobrecarregando os recursos do sistema)

```
db2 force application (<application handle for db2bp.exe>)
```

O handle do aplicativo é um número ou 'alça' do aplicativo que você deseja forçar. Você pode obter este número na saída do comando `db2 list applications`. O aplicativo db2bp.exe representa a Janela de comando do DB2.

16. Confirme se a conexão de uma das janelas de comando do DB2 foi encerrada à força. Se não souber qual das duas Janelas de comando do DB2 foi encerrada, execute esta instrução em ambas.

```
db2 select * from staff
```

A Janela de comando do DB2 que foi encerrada deve retornar uma mensagem de erro com o código SQL1224N. A outra Janela de comando do DB2 deve retornar a saída (resultado) da consulta.

17. Descarte e recrie o DAS, e o reinicie.

```
db2admin stop
db2admin drop
db2admin create
db2admin start
```

18. Observe o valor atual do valor DB2COMM do registro

```
db2set -all
```

19. Desconfigure a variável DB2COMM do registro e confirme se isto foi realizado

```
db2set db2comm=
db2stop
```

(Dica: você receberá um erro em db2stop se tiver alguma conexão. O que você faria? Consulte uma etapa anterior para resolver isto)

```
db2start
db2set -all
```

20. Defina a variável DB2COMM do registro do DB2 para `tcpip` e `npipe` em sua instância, e confirme o novo valor

```
db2set db2comm=tcpip,npipe
db2stop
db2start
db2set -all
```

21. Verifique o valor atual do parâmetro LOGSEND db cfg e o altere para um valor de 5, confirmando o novo valor

```
db2 connect to sample
db2 get db cfg
db2 update db cfg using LOGSECOND 5
db2 get db cfg
```

5

Capítulo 5 – Ferramentas do DB2

new in
V9.7

Neste capítulo, descrevemos algumas ferramentas que podem ser usadas com o DB2. A partir do DB2 9.7, a maioria das ferramentas descritas neste capítulo passa a ser obsoleta, portanto ainda são suportadas, mas não serão mais aperfeiçoadas, e podem não ser incluídas no produto em versões futuras. O IBM Data Studio é a substituição dessas ferramentas.

O círculo vermelho na *Figura 5.1* mostra a área de foco deste capítulo.

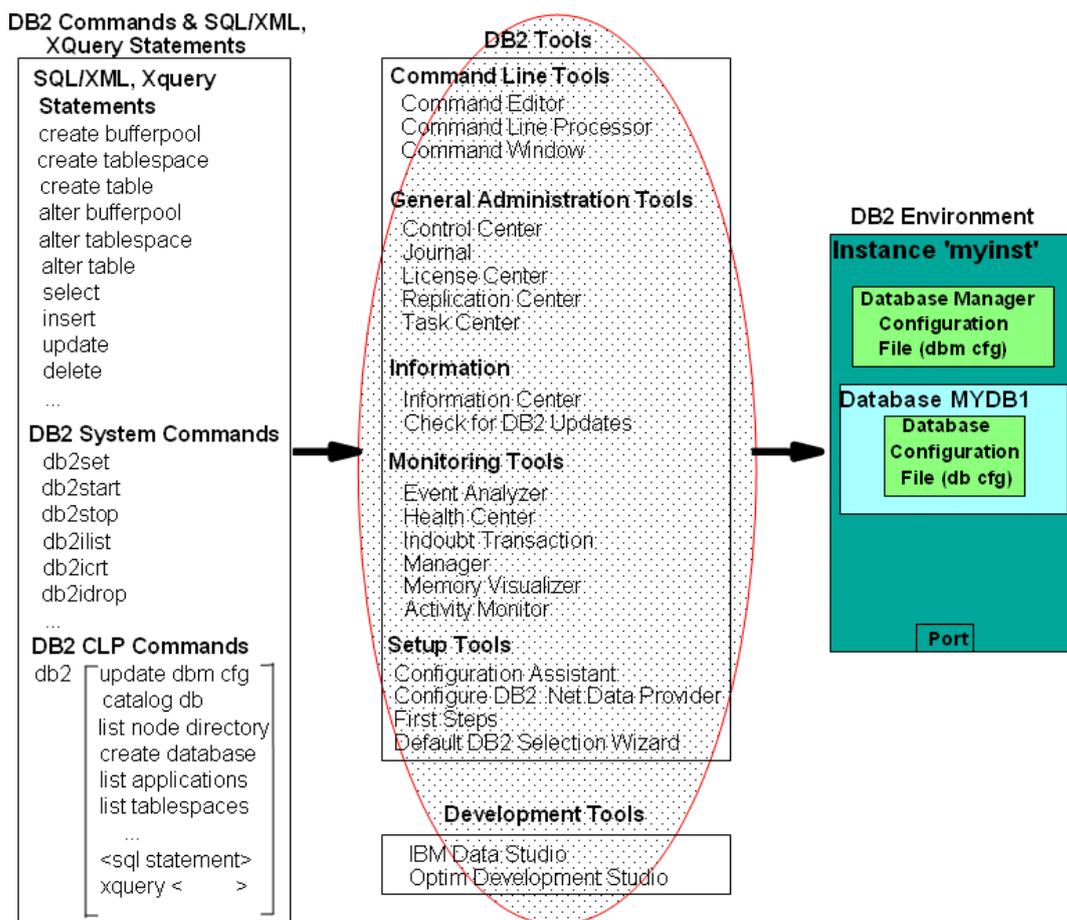


Figura 5.1 – O quadro geral do DB2: ferramentas do DB2

Observação:

Assista às apresentações em vídeo sobre as Ferramentas do DB2 nestes links:
<http://www.channeldb2.com/video/video/show?id=807741:Video:4202>
<http://www.channeldb2.com/video/video/show?id=807741:Video:4182>

A Figura 5.2 lista todas as Ferramentas do DB2 disponíveis nos atalhos do menu Iniciar do IBM DB2. A maioria dessas ferramentas é a mesma em Linux ou Windows.

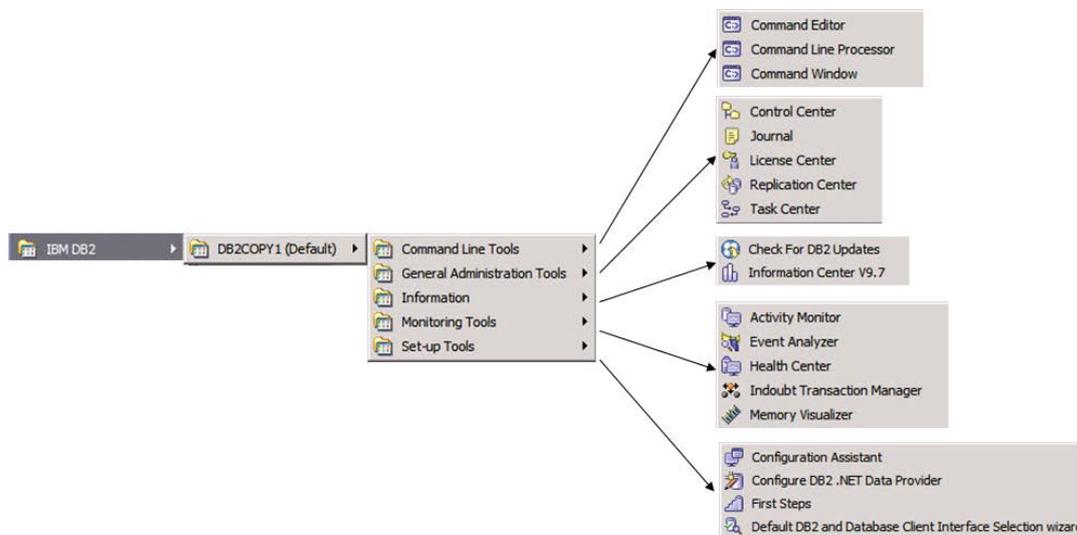


Figura 5.2 – Ferramentas do DB2 no menu Iniciar do IBM DB2

A Tabela 5.1 fornece uma lista de comandos de atalho que podem ser usados para iniciar algumas das ferramentas mais populares, seja em Linux ou Windows. Ela também lista quais ferramentas estão obsoletas no DB2 9.7.

Nome da ferramenta	Comando	Obsoleta?
Command Editor	db2ce	Sim
Processador de linha de comando	db2	Não
Command Window (somente nas plataformas Windows)	db2cmd	Não
Control Center	db2cc	Sim
Task Center	db2tc	Sim
Health Center	db2hc	Sim
Configuration Assistant	db2ca	Sim
First Steps	db2fs	Não

Tabela 5.1 – Comandos de atalho para iniciar algumas ferramentas do DB2

new in
V9.7

5.1 IBM Data Studio

Com o DB2 9.7, o IBM Data Studio é a principal ferramenta a ser usada para administração de banco de dados e desenvolvimento de banco de dados com o DB2. O Data Studio é gratuito. Pode ser executado em Linux e Windows e faz parte do [portfólio de produtos de Gerenciamento de dados integrado da IBM](#). O desenvolvimento do Data Studio segue um cronograma que não está atrelado às versões do DB2. Entretanto, os lançamentos dos produtos são coordenados com a maior frequência possível. Por exemplo, o DB2 9.7 e o IBM Data Studio 2.2 foram lançados no mesmo dia em junho de 2009.

A *Figura 5.3* mostra a aparência do IBM Data Studio.

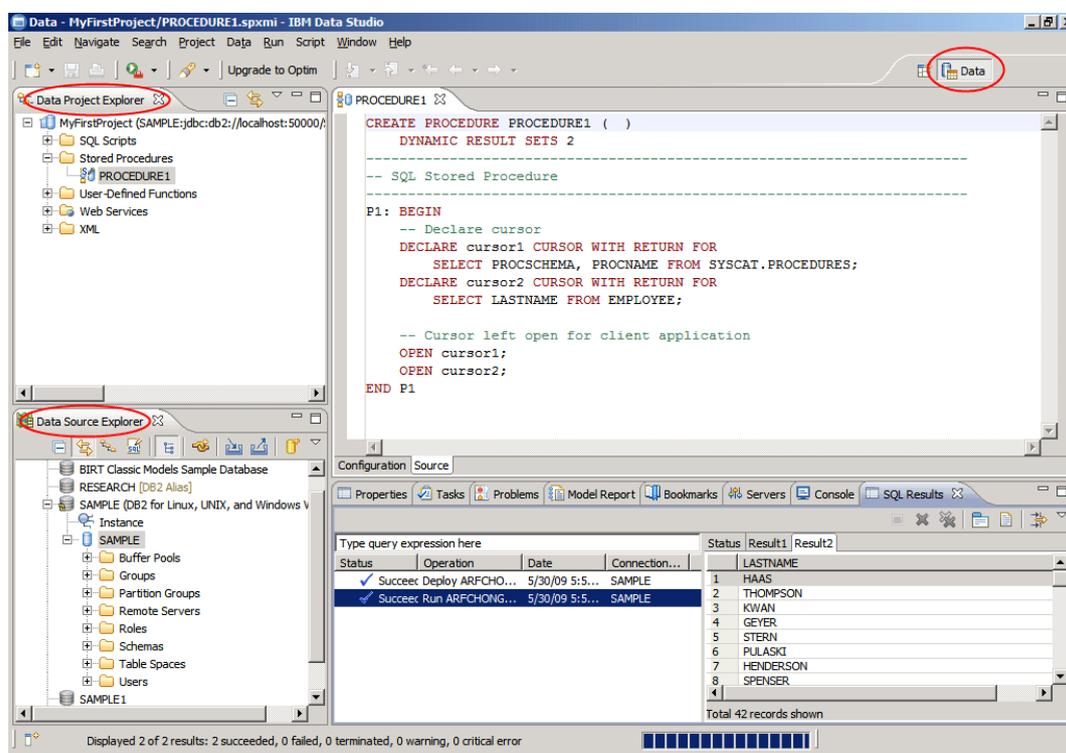


Figura 5.3 – IBM Data Studio

Se você tem alguma familiaridade com o Eclipse, perceberá que o Data Studio é baseado em Eclipse. Normalmente, o trabalho no Data Studio ocorre dentro da janela de perspectiva Data (Dados, realçada na figura no canto superior direito). Você também pode alternar para a perspectiva Java, caso esteja desenvolvendo um programa em Java. Há duas visualizações realçadas na figura:

- Data Project Explorer (alto à esquerda)
- Data Source Explorer (embaixo à esquerda)

A visualização Data Project Explorer é usada pelos desenvolvedores de bancos de dados para trabalhar com scripts SQL, XQuery, procedimentos armazenados, UDFs e serviços de dados de Web.

A visualização Data Source Explorer é usada pelos administradores de bancos de dados para gerenciar as instâncias e bancos de dados do DB2. Usando esta visualização, você pode executar a maior parte das funções disponíveis anteriormente no Control Center.

Na figura, a visualização com o título PROCEDURE1 é um editor do procedimento que está realçado no Project Explorer. Dependendo da tarefa sendo executada, os editores ou outras janelas são exibidos, permitindo criar código ou executar mais configurações.

O IBM Data Studio é fornecido em duas edições:

- Pacote independente
- Pacote IDE

Este pacote independente é bem menor que o pacote IDE, mas não suporta desenvolvimento de serviços de dados de Web e não pode ser estendido a outros produtos IBM baseados em Eclipse, como por exemplo o InfoSphere Data Architect. Excetuando-se essas diferenças, a interface e os recursos são os mesmos.

Com o IBM Data Studio, você também pode trabalhar com outros servidores de dados, como o Informix. As empresas que trabalham com diversos servidores de dados e têm uma pequena equipe de DBA ou desenvolvimento de banco de dados, agora têm a conveniência de poder trabalhar e gerenciar todos eles usando uma única ferramenta.

Observação:

Para obter mais informações sobre o Data Studio, consulte o e-book gratuito [Getting Started with IBM Data Studio for DB2](#) (Primeiros passos com o IBM Data Studio para DB2), que é parte desta série de livros.

5.2 Control Center (obsoleto)

Antes do DB2 9.7, a principal ferramenta do DB2 para a administração de banco de dados era o Control Center, como ilustrado na *Figura 5.4*.

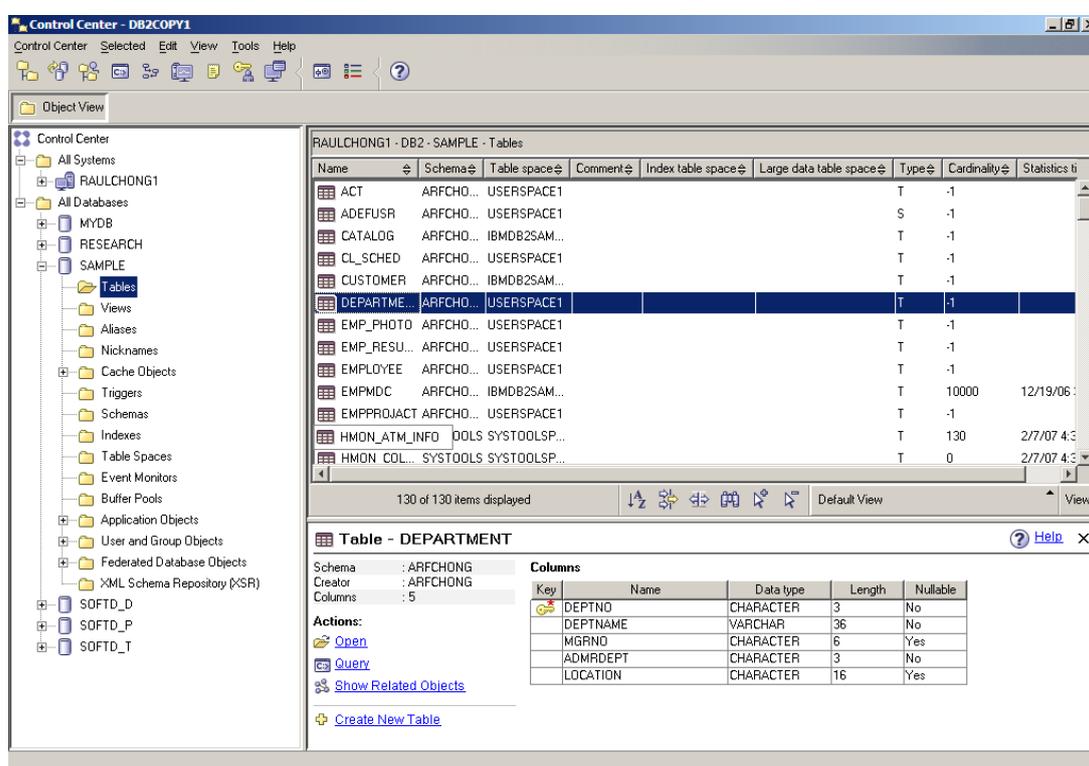


Figura 5.4 – O Control Center do DB2

O Control Center é uma ferramenta de administração centralizada que permite:

- Visualizar os sistemas, instâncias, bancos de dados e objetos de banco de dados;
- Criar, modificar e gerenciar bancos de dados e objetos de banco de dados;
- Inicializar outras ferramentas gráficas do DB2

O painel à esquerda fornece uma hierarquia visual dos objetos de banco de dados do(s) sistema(s), fornecendo uma “pasta” para Tabelas, Visualizações, etc. Ao clicar duas vezes em uma pasta (por exemplo, a pasta Tables, mostrada na Figura 5.4), o painel no alto à direita listará todos os objetos relacionados; neste caso, todas as tabelas associadas ao banco de dados **SAMPLE**. Se você selecionar uma dada tabela no painel do alto à direita, o painel de baixo à direita fornece informações mais específicas sobre essa tabela.

Clique com o botão direito nas diferentes pastas ou objetos da árvore de objetos para abrir os menus aplicáveis à pasta ou ao objeto em questão. Por exemplo, clicar com o botão direito em uma instância e selecionar *Configure parameters* (Configurar parâmetros) permite que você visualize e atualize o arquivo de configuração do gerenciador do banco de dados. De um modo semelhante, ao clicar com o botão direito em um banco de dados e escolher *Configure parameters*, você poderá visualizar e atualizar o arquivo de configuração do banco de dados. O ambiente do DB2 e os parâmetros de configuração são discutidos em maiores detalhes no *Capítulo 4, O ambiente do DB2*.

Na primeira vez em que inicializar o Control Center, você será solicitado a escolher que visualização deseja usar. A escolha da visualização determina quais tipos de opções e objetos de banco de dados são expostos. A *Figura 5.5* exibe a caixa de diálogo Control Center View (Visualização do Control Center).

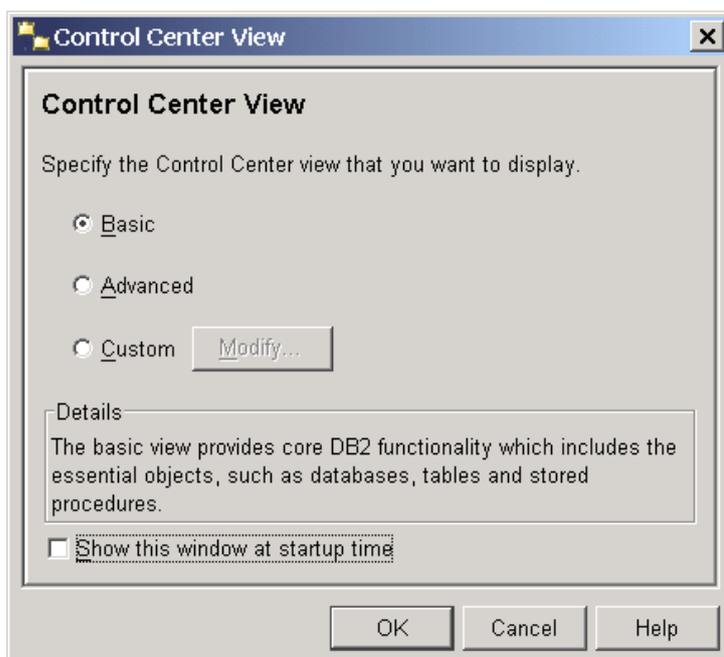


Figura 5.5 – A caixa de diálogo Control Center View do DB2

A visualização básica fornece a funcionalidade principal do DB2; a visualização avançada exibe mais opções e funções. A visualização personalizada permite personalizar as funções, opções e objetos específicos que você vê.

Para chamar novamente a caixa de diálogo Control Center View, selecione *Tools -> Customize Control Center* (Ferramentas -> Personalizar o Control Center) como mostrado na *Figura 5.6*.

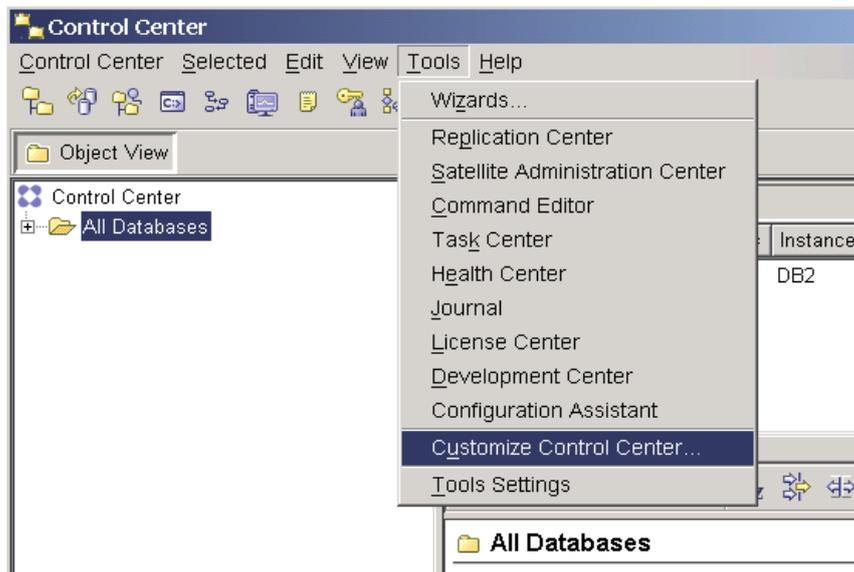


Figura 5.6 – Personalização do Control Center

5.2.1 Como iniciar o Control Center

Há diversas maneiras de iniciar o Control Center:

- Navegando no menu *Iniciar* do Windows
- Executando `db2cc` na linha de comando
- Clicando no ícone do Control Center  na barra de ferramentas de qualquer uma das outras ferramentas de GUI do DB2
- No ícone do DB2 na bandeja de sistema do Windows, como mostrado na *Figura 5.7* (Clique com o botão direito no ícone verde do DB2 e selecione a opção de menu DB2 Control Center)

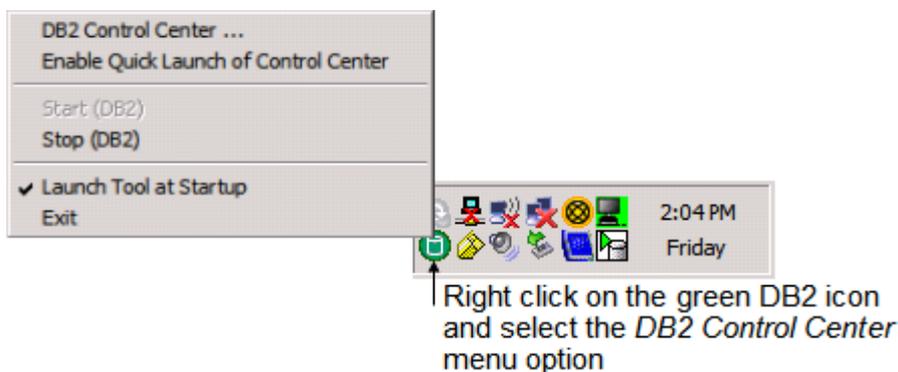


Figura 5.7 – Inicialização do DB2 Control Center a partir da bandeja de sistema do Windows

5.3 Command Editor (obsoleto)

Usando o Command Editor do DB2, você pode executar comandos do DB2, instruções SQL e XQuery, analisar o plano de execução de uma instrução e exibir ou atualizar os conjuntos de resultados da consulta. A *Figura 5.8* exibe o Command Editor com uma descrição de seus elementos.

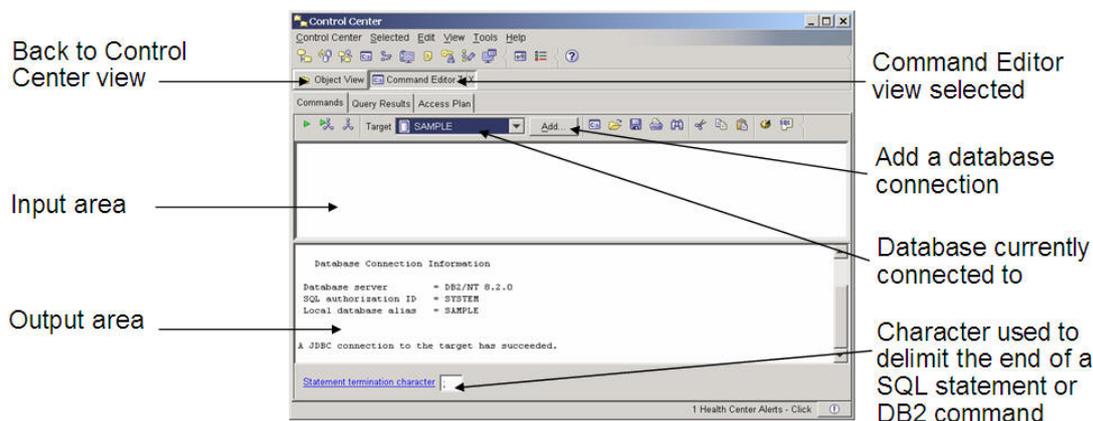


Figura 5.8 – Command Editor do DB2

Você pode inserir diversas instruções na área de entrada, desde que cada instrução termine com um caractere de terminação. Se pressionar o botão executar (o primeiro botão à esquerda na *Figura 5.9*), as instruções serão executadas uma após a outra. Se você destacar explicitamente uma instrução específica, somente esta será executada. É necessário que haja uma conexão com banco de dados para executar qualquer instrução SQL; entretanto, uma das instruções pode ser de conexão.

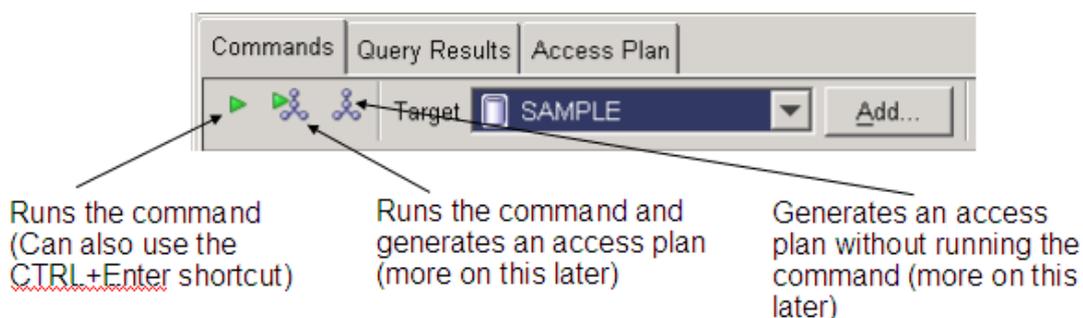


Figura 5.9 – O Command Editor – guia Commands

5.3.1 Inicialização do Command Editor

Você pode iniciar o Command Editor de diversas maneiras:

- No menu Iniciar do Windows: *Iniciar -> Programas -> IBM DB2 -> DB2COPY1 (Default) -> Command Line Tools -> Command Editor*
- Em uma linha de comando, digite `db2ce`
- No menu Tools (Ferramentas) no Control Center

- Incorporado no Control Center
 - Clique com o botão direito do mouse no ícone do banco de dados **SAMPLE** no painel da árvore de objetos do Control Center e selecione o item de menu *Query* (Consulta)
 - Sempre que é selecionado um objeto que pode ser consultado (banco de dados, tabela, etc.), você pode iniciar o Command Editor clicando no link *Query* no painel de detalhes do objeto no Control Center
- No Control Center, clique no ícone do Command Editor  na barra de ferramentas do Control Center, como mostrado na *Figura 5.10*



Figura 5.10 – O ícone do Command Editor no Control Center

5.3.2 Como adicionar uma conexão com banco de dados

Para adicionar uma conexão com banco de dados, clique no botão Add (Adicionar) (Consulte a *Figura 5.8*). Aparecerá a caixa de diálogo mostrada na *Figura 5.11*.

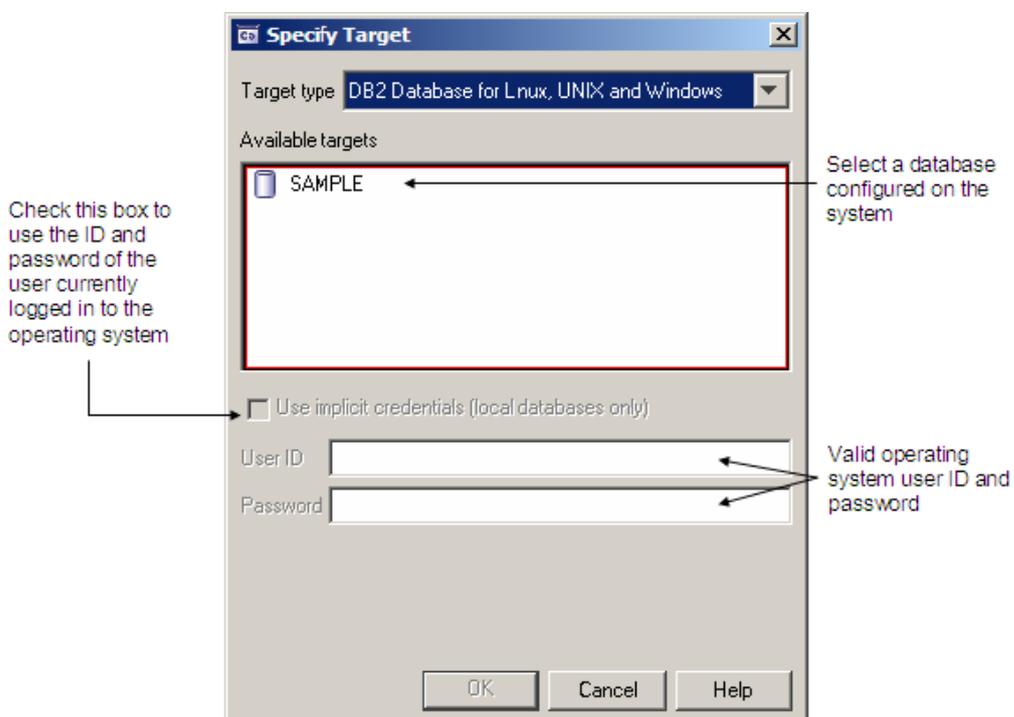


Figura 5.11 – Adicionar uma conexão com banco de dados

5.4 Assistente SQL Assist (obsoleto)

Se você não estiver familiarizado com a linguagem SQL e deseja usar um assistente para gerar o código SQL, o assistente SQL Assist está disponível no Command Editor para ajudá-lo. Como mostrado na *Figura 5.12*, vá até o Command Editor e clique no último ícone, com o símbolo SQL, para iniciá-lo. Este ícone aparece somente após você conectar-se a um banco de dados.

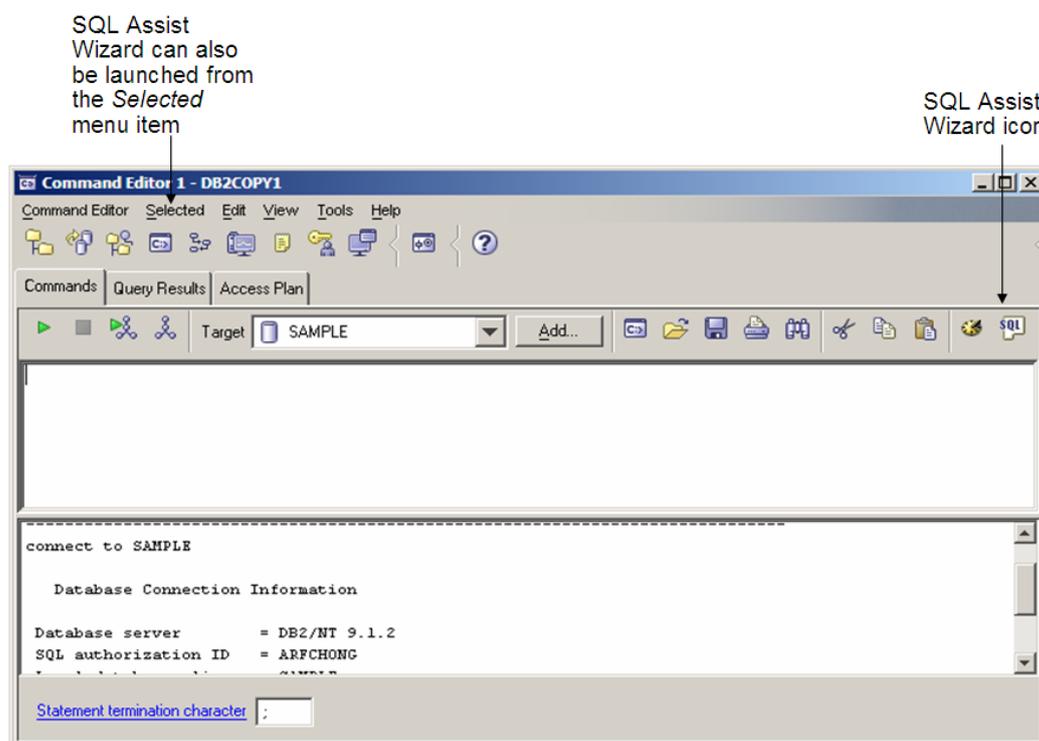


Figura 5.12 – Como chamar o assistente SQL Assist

A *Figura 5.13* mostra o assistente SQL Assist. Ele é bem simples de usar. Primeiro, indique o tipo de instrução SQL para a qual necessita de assistência (SELECT, INSERT, UPDATE, DELETE). Dependendo da instrução escolhida, aparecerão opções diferentes. À medida que seleciona opções diferentes no assistente, você verá na parte inferior da janela como a instrução SQL é construída.

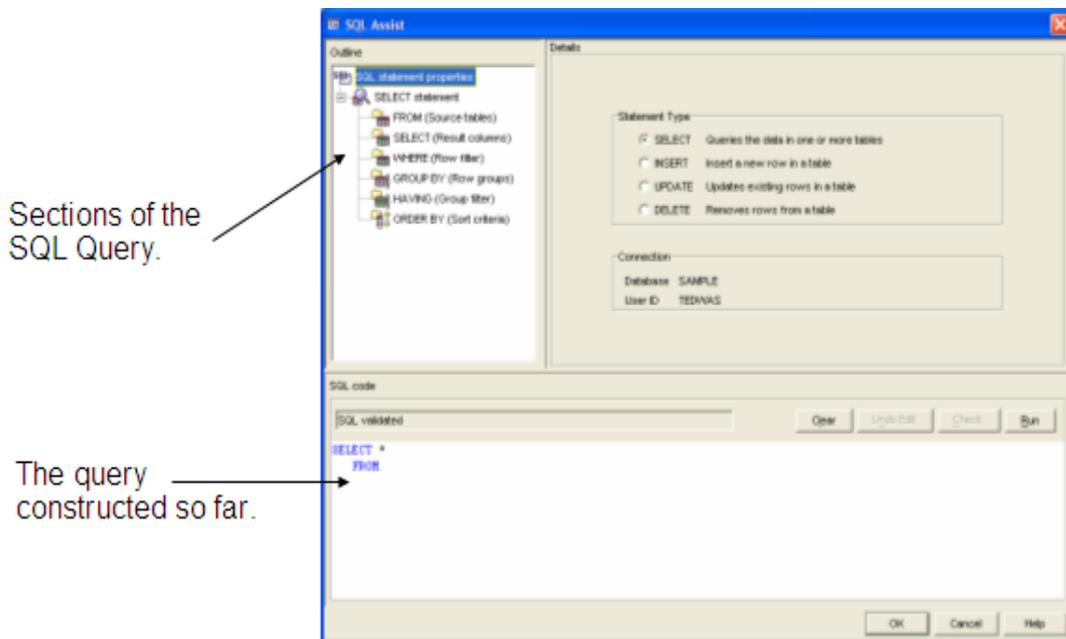


Figura 5.13 – O assistente SQL Assist

5.5 Botão Show SQL (obsoleto)

A maioria das ferramentas de GUI e assistentes do DB2 permite verificar o comando ou instrução SQL em si, criados como resultado do uso da ferramenta ou assistente para executar uma ação. Para ver isto, clique no botão *Show SQL* (Mostrar SQL) na ferramenta com a qual está trabalhando, como mostrado na *Figura 5.14* e na *Figura 5.15*



Figura 5.14 – O botão Show SQL

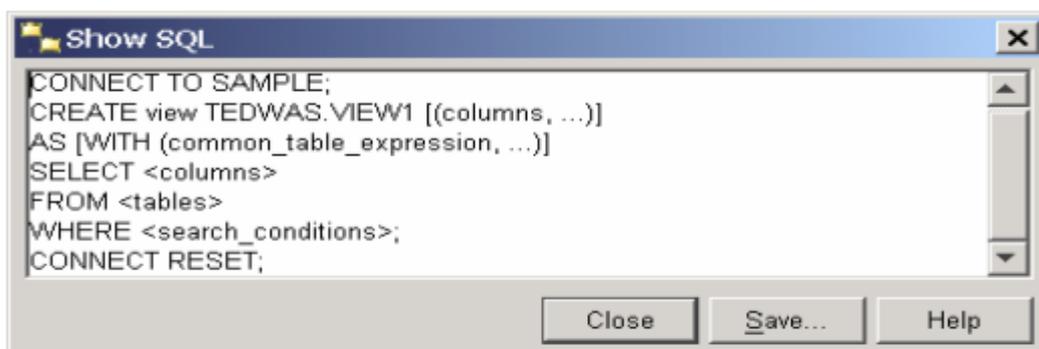


Figura 5.15 – A saída (resultado) de um botão Show SQL

A capacidade de verificar as instruções e comandos SQL é muito útil ao aprendizado da sintaxe SQL e para salvar os comandos ou instruções em um arquivo para uso futuro. Você também pode criar scripts através da reutilização desses comandos e instruções gerados.

5.6 Task Center (obsoleto)

A ferramenta de GUI do Task Center permite criar tarefas: um conjunto de operações tais como executar comandos DB2, comandos do sistema operacional ou scripts. As ações subsequentes podem ser executadas caso a tarefa falhe ou tenha êxito. Por exemplo, se uma tarefa envolvendo o backup de um banco de dados importante às 3:00 da manhã é bem-sucedida, é possível enviar um email ao DBA fornecendo essas informações. Por outro lado, caso a tarefa de backup falhe, o Task Center pode avisar ao DBA. A *Figura 5.16* mostra o Task Center.

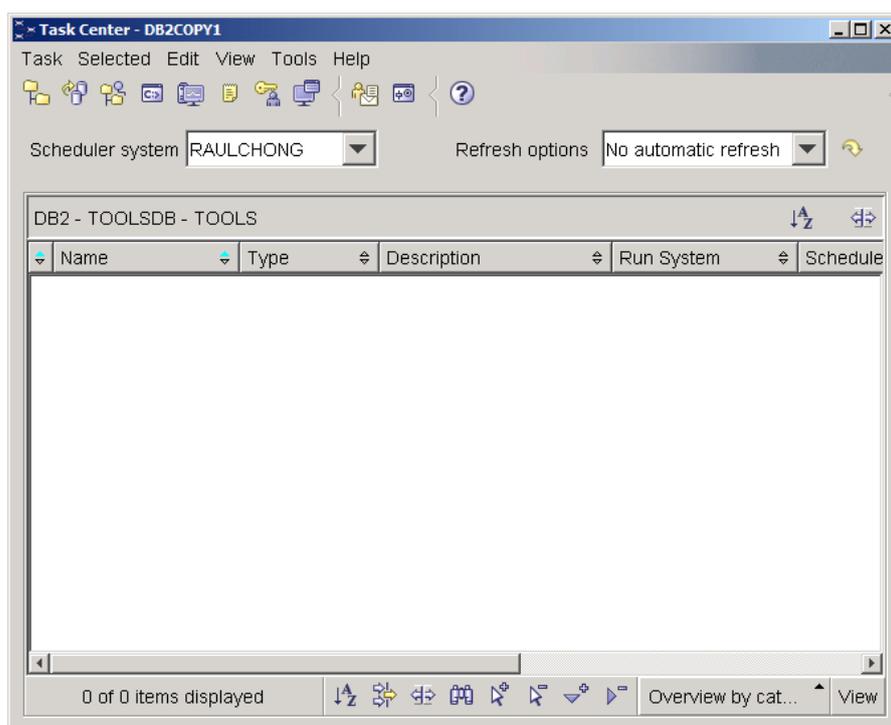


Figura 5.16 – O Task Center

5.6.1 O banco de dados Tools Catalog (obsoleto)

Todos os detalhes de suas tarefas e agendamento de tarefas são armazenados em um banco de dados separado do DB2, denominado Tools Catalog (Catálogo de ferramentas). Este banco de dados deve existir previamente para poder agendar as tarefas. Para criar o banco de dados Tools Catalog, use este comando:

```
CREATE TOOLS CATALOG systools CREATE NEW DATABASE toolsdb
```

No exemplo anterior, *sysools* é o nome do esquema de todas as tabelas do banco de dados, e o nome do banco de dados é *toolsdb*. Falaremos mais sobre os esquemas no *Capítulo 8, Como trabalhar com objetos de banco de dados*.

5.6.1.1 Inicialização do Task Center

Você pode inicializar o Task Center no Control Center, clicando em *Tools -> Task Center* (Ferramentas -> Task Center), como mostrado na *Figura 5.17*. Como alternativa, você pode iniciar esta ferramenta no menu Iniciar do Windows: *Iniciar -> Programas -> IBM DB2 -> DB2COPY1 (Default) -> General Administration Tools -> Task Center*.

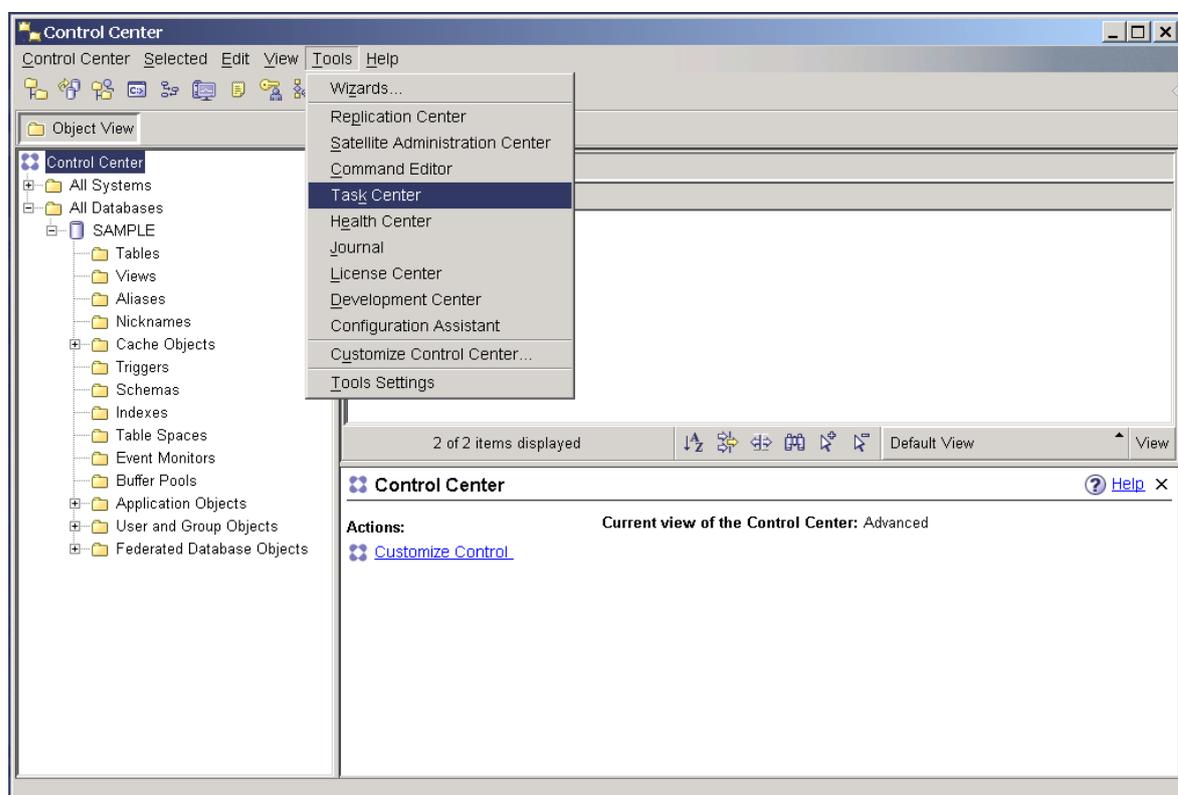


Figura 5.17 – Inicialização do Task Center

5.6.1.2 Agendamento com o Task Center

Qualquer tipo de script, seja ele criado ou não através de uma ferramenta de GUI do DB2, pode ser agendado com o Task Center. As tarefas são executadas no horário de agendamento, no sistema onde foi criado o banco de dados Tools Catalog. Nós o encorajamos a explorar bem o Task Center. A criação de uma tarefa é bem simples.

5.7 Journal (obsoleto)

A ferramenta de GUI Journal do DB2 oferece ao DBA um diário das atividades em formulário on-line. A *Figura 5.18* mostra o Journal e a *Tabela 5.2* exibe as informações que podem ser obtidas no Journal.

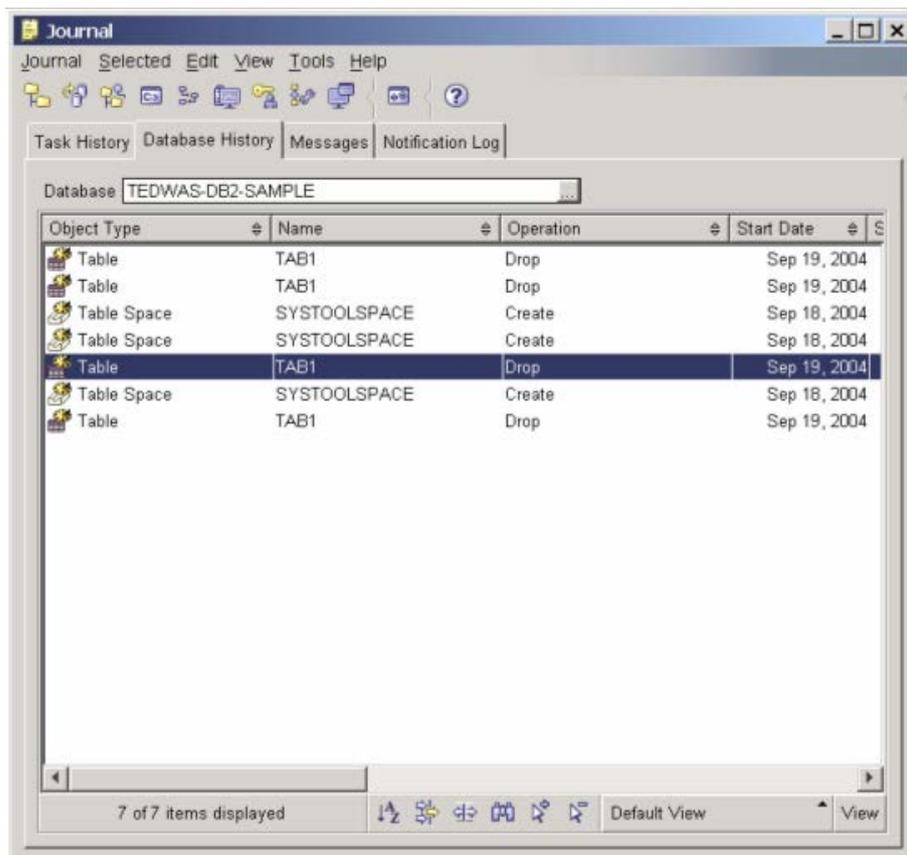


Figura 5.18 –O Journal

Tipo de informação	Descrição
Histórico da tarefa	Todas as tarefas agendadas que foram tentadas e seu status de sucesso
Histórico do banco de dados	Um registro das atividades de banco de dados executadas (backup, restauração, REORGs, etc.)
Mensagem	Histórico das mensagens devolvidas pelas ferramentas do DB2. Isto é útil se você quiser lembrar e comparar mensagens de erro antigas, ou se fechar uma caixa diálogo rápido demais ou acidentalmente.
Log de notificações	Contém mensagens no nível de sistema. Os erros críticos são registrados aqui

Tabela 5.2 – Informações fornecidas no Journal

5.7.1 Inicialização do Journal

Você pode inicializar o Journal no Control Center, clicando em *Tools -> Journal* (Ferramentas -> Journal), como mostrado na *Figura 5.19*. Como alternativa, você pode iniciar esta ferramenta no menu Iniciar do Windows: *Iniciar -> Programas -> IBM DB2 -> DB2COPY1(Default) -> General Administration Tools -> Journal*.

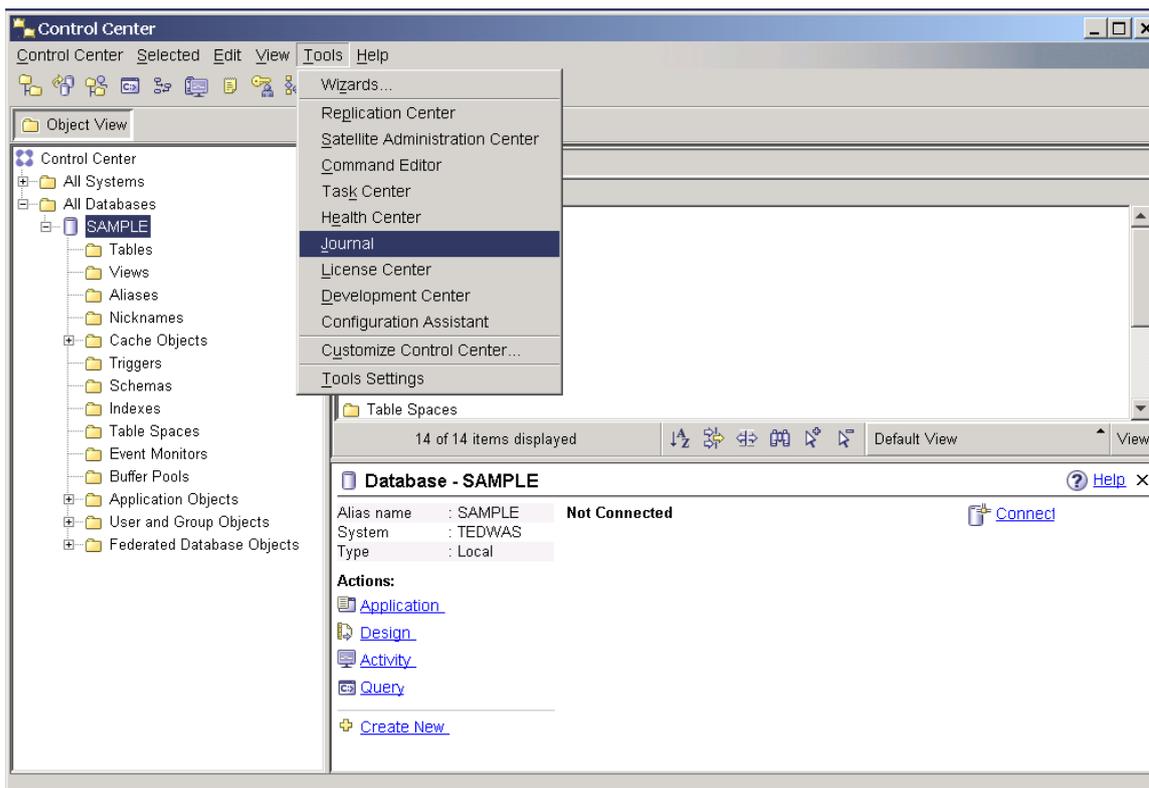


Figura 5.19 – Inicialização do Journal

5.8 Health Monitor (obsoleto)

O Health Monitor é um agente padrão executado dentro do mecanismo DB2, monitorando todos os aspectos da saúde do banco de dados (memória, gerenciamento de espaço, atividades automatizadas definidas previamente, etc.). Quando parte do DB2 estiver operando fora dos parâmetros definidos, uma exceção é gerada e levada à atenção do DBA. Há três tipos de estados de alerta:

- **Atenção:** um estado anormal
- **Aviso:** um estado não crítico que não exige atenção imediata, mas pode indicar uma condição não ideal do sistema
- **Alarme:** uma condição crítica que exige ação imediata

O Health Monitor pode ser ativado ou desativado através do parâmetro de configuração do gerenciador de banco de dados HEALTH_MON.

5.8.1 Health Center (obsoleto)

O Health Center é uma ferramenta gráfica para interagir com o Health Monitor. O Health Center divide os alertas de um sistema pelos níveis de instância, banco de dados e tablespace. A *Figura 5.20* mostra o Health Center.

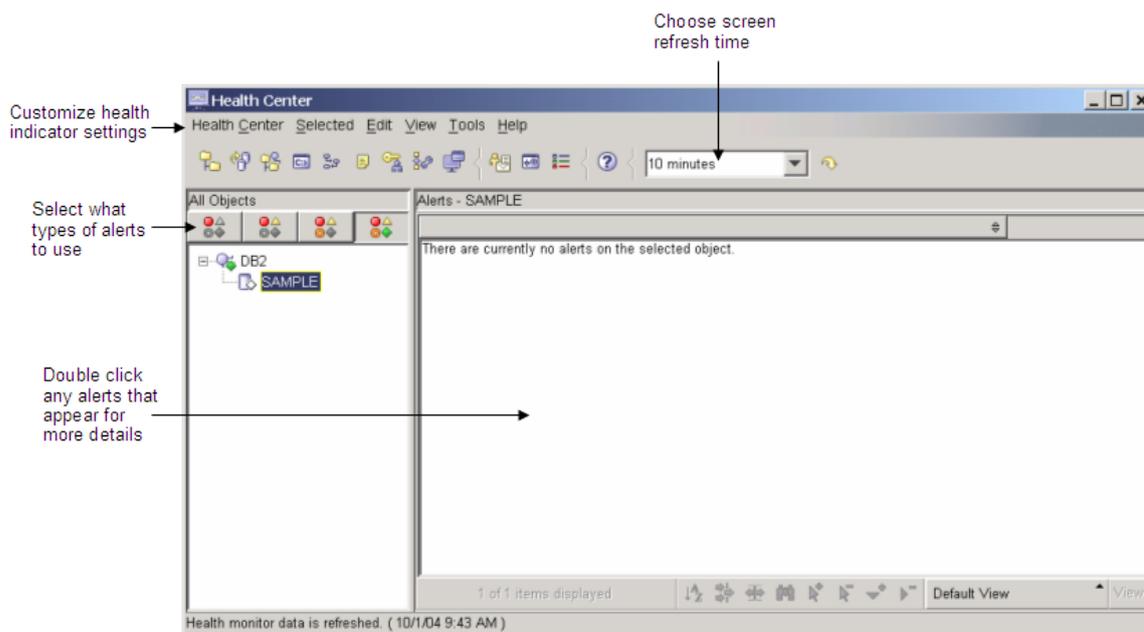


Figura 5.20 – O Health Center

5.8.1.1 Inicialização do Health Center

Você pode iniciar o Health Center no Control Center, selecionando *Tools -> Health Center*. Isto é mostrado na *Figura 5.21*. Você também pode iniciar esta ferramenta em *Iniciar -> Programas-> IBM DB2 -> DB2COPY1(Default) -> Monitoring Tools -> Health Center*

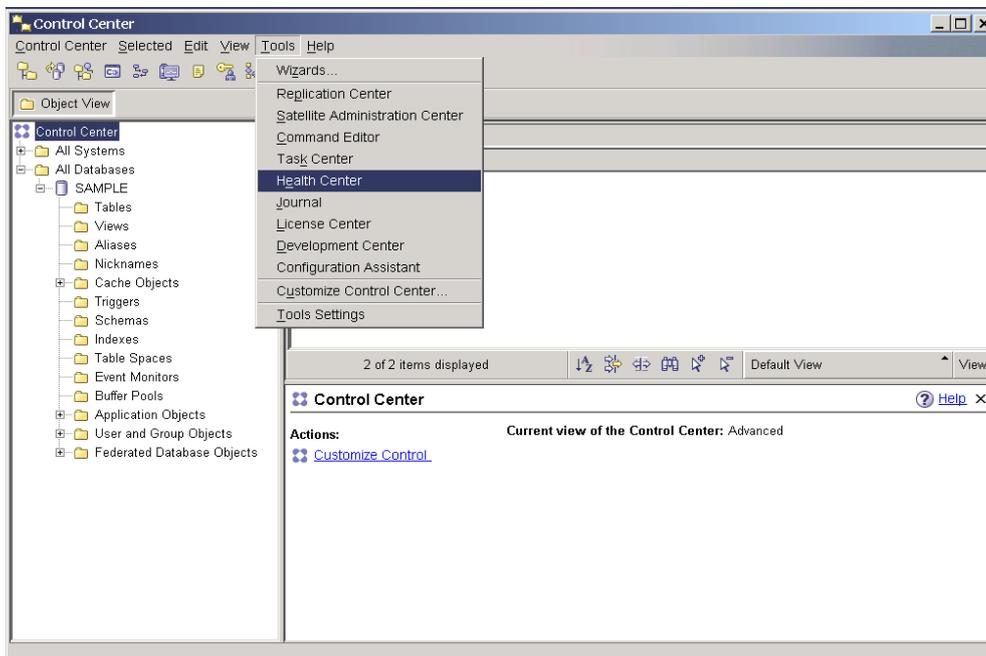


Figura 5.21 – Inicialização do Health Center

5.8.1.2 Configuração das notificações de alerta de saúde

Após iniciar o Health Center, você pode configurar as notificações de Alerta clicando no menu *Health Center* -> *Configure* -> *Alert Notification* como mostrado na *Figura 5.22*. As notificações de alerta permitem inserir os nomes de contato com endereços de email ou número de pager das pessoas que devem ser notificadas caso ocorra um alerta.

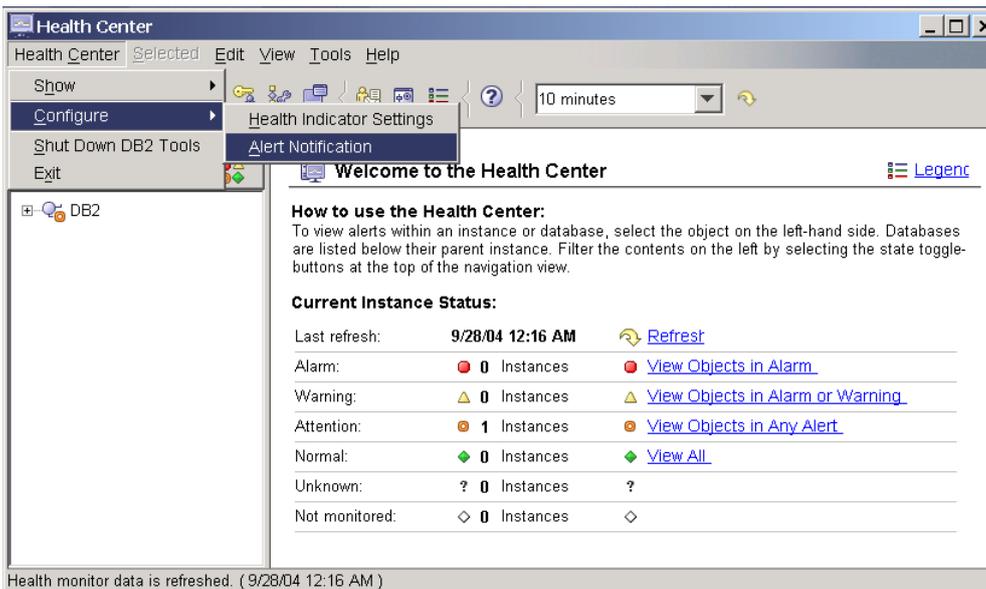


Figura 5.22 – Notificação de alerta**5.9 Gerenciador de memória autoajustável**

O gerenciador de memória autoajustável (STMM) apresentado no DB2 9 é uma das diversas funções de computação autônomas que simplificam a tarefa da configuração de memória, definindo automaticamente os valores de diversos parâmetros de configuração da memória. Quando ativado, o gerenciador de ajuste automático distribui dinamicamente os recursos de memória disponíveis entre os diversos consumidores de memória do banco de dados. O STMM responde às mudanças nas características das cargas de trabalho, ajustando os valores dos parâmetros de configuração da memória e os tamanhos dos bufferpools para otimizar o desempenho. Para ativar o STMM, atualize o parâmetro SELF_TUNING_MEM do db cfg para ON.

Outras funções de computação autônomas, como a manutenção automática e o armazenamento automático, serão discutidas em outras seções deste livro.

5.10 Criação de scripts

É sempre útil poder criar arquivos de script que executem diversos comandos do DB2 ou instruções SQL repetidamente. Por exemplo, um DBA pode executar um dado script todos os dias, para verificar a contagem de linhas de tabelas importantes. Há dois tipos de formatos gerais de script:

- Scripts SQL
- Scripts de sistema operacional (shell).

5.10.1 Scripts SQL

Os scripts SQL incluem instruções de consulta e comandos de banco de dados. Esses scripts são relativamente simples de compreender e independentes de plataforma. Entretanto, não são suportados variáveis ou parâmetros. Por exemplo, os comandos mostrados na *Listagem 5.1* abaixo são salvos em um arquivo denominado `script1.db2`.

```
CONNECT TO EXPRESS;  
CREATE TABLE user1.mytable  
    (   col1 INTEGER NOT NULL,  
        col2 VARCHAR(40),  
        col3 DECIMAL(9,2));  
SELECT * FROM user1.mytable FETCH FIRST 10 ROWS ONLY;  
COMMIT;
```

Listagem 5.1 – Uma amostra de script SQL armazenado no arquivo `script1.db2`

No script acima, todas as instruções são SQL e cada uma é separada por um delimitador de instrução, neste caso, um ponto-e-vírgula. O nome do arquivo não precisa usar a extensão “db2”. Pode ser usada qualquer extensão.

5.9.1.1 Execução de scripts SQL

Um script SQL pode ser executado no Command Editor ou na Janela de comando do DB2 no Windows, ou através de um shell no Linux. Para executar o script da *Listagem 5.1* na Janela de comando do DB2 ou em shell no Linux, use o comando abaixo:

```
db2 -t -v -f script1.db2 -z script1.log
```

ou o equivalente:

```
db2 -tvf script1.db2 -z script1.log
```

Neste comando:

- t** indica que as instruções usam o caractere padrão de terminação (um ponto-e-vírgula)
- v** indica modo detalhado, fazendo o db2 ecoar o comando sendo executado
- f** indica que o nome de arquivo especificado após este sinalizador é o arquivo de script
- z** indica que o nome de arquivo especificado após este sinalizador é usado para anexar a saída de tela para análise posterior (isto é opcional, mas recomendado)

Quando o sinalizador **-t** é usado sem a especificação de um delimitador de linha, supõe-se que o ponto-e-vírgula será o delimitador das instruções. Pode haver situações em que seja necessário outro delimitador. Por exemplo, um script contendo código SQL PL necessita usar um caractere de terminação de instrução diferente do padrão (ponto-e-vírgula), porque estes são usados dentro das definições de objetos SQL PL para terminar as instruções processuais.

Por exemplo, no arquivo de script `functions.db2` mostrado na *Listagem 5.2* abaixo, você pode ver a instrução que ele contém para criar uma função, sendo necessário um ponto-e-vírgula ao final da instrução `SELECT` porque ela faz parte da sintaxe necessária dentro da função. Como terminador da instrução `CREATE FUNCTION`, estamos usando um ponto de exclamação (!). Se usássemos um ponto-e-vírgula como delimitador da instrução, haveria um conflito no script em relação ao usado na instrução `SELECT`, resultando em um relatório de erro emitido pelo DB2.

```
CREATE FUNCTION f1()  
  SELECT ... ;  
  ...  
END!
```

Listagem 5.2 – Conteúdo do arquivo de script `functions.db2`

Para informar ao DB2 o uso de um caractere de terminação de instrução diferente, use o sinalizador **-d**, seguido pelo caractere de terminação desejado, como mostrado abaixo:

```
db2 -td! -v -f functions.db2 -z functions.log
```

Para saber mais sobre os outros sinalizadores que podem ser usados na Janela de comando ou no shell do Linux, use este comando:

```
db2 list command options
```

5.10.2 Scripts de sistema operacional (shell).

Os scripts de sistema operacional fornecem mais flexibilidade e poder que os scripts SQL, pois oferecem a possibilidade de adicionar mais lógica de programação. Eles são dependentes de plataforma, mas suportam parâmetros e variáveis. A *Listagem 5.3* mostra um exemplo de script simples do sistema operacional Windows (shell).

```
set DBPATH=C:
set DBNAME=PRODEXPR
set MEMORY=25
db2 CREATE DATABASE %DBNAME% ON %DBPATH% AUTOCONFIGURE USING
MEM_PERCENT %MEMORY% APPLY DB AND DBM
db2 CONNECT TO %DBNAME% USER %1 USING %2
del schema.log triggers.log app_objects.log
db2 set schema user1
db2 -t -v -f schema.db2 -z schema.log
db2 -td@ -v -f triggers.db2 -z triggers.log
db2 -td@ -v -f functions.db2 -z functions.log
```

Listagem 5.3 – Conteúdo do arquivo de script create_database.bat de sistema operacional

Para executar este script de sistema operacional na linha de comando, você deve executar o seguinte comando no Windows:

```
create_database.bat db2admin ibmdb2
```

onde *db2admin* é o ID de usuário e primeiro parâmetro do script, e *ibmdb2* é a senha e segundo parâmetro do script.

No Windows, o uso da extensão “bat” informa ao sistema operacional que é um arquivo executável em lote. No Linux, seria necessário alterar o modo do arquivo para indicar que este é executável, usando um comando como `chmod +x`. Depois disso, você pode executá-lo da mesma maneira listada acima.

5.11 Considerações sobre o Windows Vista

O recurso Controle de Acesso de Usuário do Windows Vista faz os aplicativos serem iniciados com direitos padrão, mesmo que o ID de usuário seja de um administrador local. Isto significa que qualquer ferramenta ou comando do DB2 iniciado no Vista provavelmente será executado, mas relatando problemas relacionados ao acesso. Para evitar este problema, use o atalho denominado “Command window – Administrator” (Janela de comando - Administrador) criado especificamente no momento da instalação para os usuários do Vista. Nesta janela, você pode executar outros comandos e iniciar outras ferramentas (usando os comandos mostrados na *Tabela 5.1* no início do capítulo). Como alternativa, no menu Iniciar do Windows Vista ou em qualquer atalho do DB2,

encontre a ferramenta do DB2 que você deseja iniciar, clique nela com o botão direito do mouse e selecione a opção *Executar como administrador*.

Se a segurança estendida do DB2 estiver ativada, o que é o padrão (consulte o *Capítulo 10, Segurança de banco de dados* para obter mais detalhes), você também deve assegurar que seu ID de usuário seja membro do grupo DB2ADMNS para poder iniciar ferramentas gráficas como o Control Center.

5,12 Resumo

Neste capítulo, examinamos a grande variedade de ferramentas disponíveis para a administração, configuração e gerenciamento do servidor de dados DB2.

A chegada do IBM Data Studio no DB2 9.7 como principal ferramenta de administração fornece uma nova dimensão ao trabalho de administração e desenvolvimento de banco de dados.

Também discutimos diversas ferramentas de GUI, agora obsoletas: o Control Center, o assistente SQL Assist, o Task Center e o Journal, e o Health Agent e Monitor. Entretanto, as ferramentas Command Line Processor e Command Window continuarão fazendo parte do aplicativo em versões posteriores do DB2 9.7. Além disso, a ferramenta STMM permanece sendo uma parte importante do processo de otimização de banco de dados.

Um componente essencial da caixa de ferramentas de qualquer administrador de banco de dados é o uso de scripts para executar comandos e funções do DB2. Neste capítulo, examinamos em detalhes os scripts SQL e de sistema operacional (shell), particularmente como são compostos, armazenados e executados.

Finalmente, concluímos com um aviso sobre como assegurar que as ferramentas DB2 funcionem sem problemas no Windows Vista.

5,13 Exercícios

Os exercícios desta seção permitirão praticar o trabalho com scripts no DB2.

Parte 1: Como popular o banco de dados EXPRESS usando scripts

Nesta parte, você irá popular o banco de dados EXPRESS (criado anteriormente) usando o Command Editor e dois scripts fornecidos.

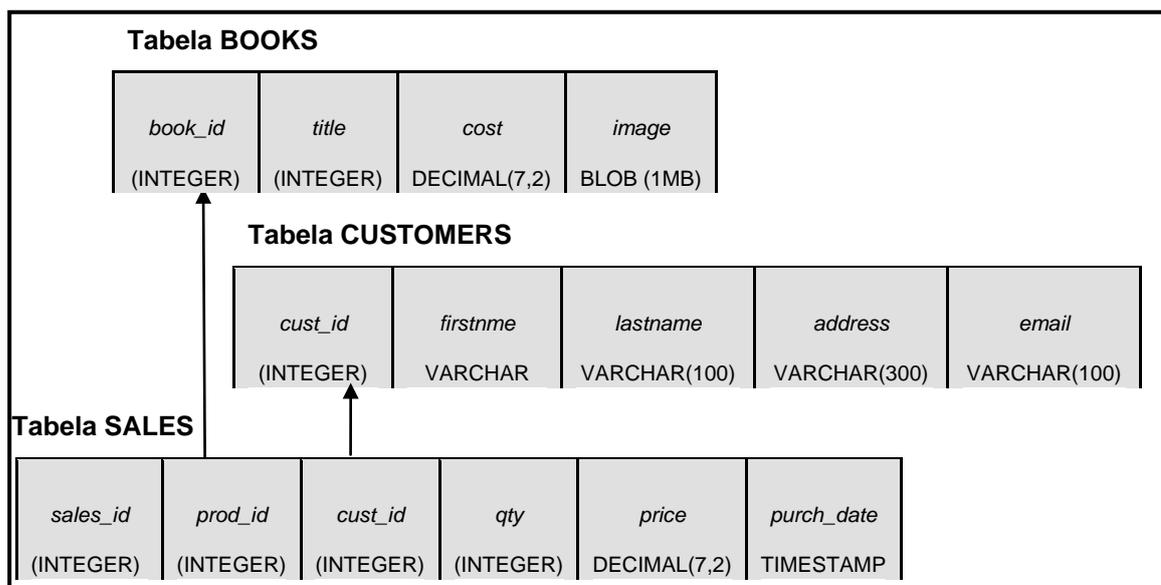
Procedimento

1. Popular o banco de dados EXPRESS com algumas tabelas e dados. Para sua conveniência, foram criados dois scripts, `Lab_Chpt5.db2` e `Lab_Chpt5.dat` para fazer isso para você. O script `Lab_Chpt5.db2` contém os comandos usados para criar as tabelas, e portanto, deve ser executado primeiro. O script `Lab_Chpt5.dat` contém instruções que inserem os dados nas tabelas. Ambos os scripts podem ser encontrados no arquivo `expressc_book_exercises_9.7.zip` que acompanha este livro. Para executá-los, abra o Command Editor. Verifique se o novo banco de dados que você criou está selecionado na lista suspensa da barra de ferramentas. Se o novo

banco de dados não aparecer na lista, use o botão *Add* (Adicionar) para adicionar uma conexão a ele.

2. Clique no menu *Selected* → *Open* no Command Editor e navegue até a pasta onde estão armazenados os scripts. Selecione o arquivo `Lab_Chpt5.db2` e clique no botão *OK*. Agora, o conteúdo do arquivo deve estar sendo exibido na área de entrada do Command Editor. Clique no botão *Run* (Executar) para executar o script. Confirme se não houve mais erros ao executar o script.
3. Repita a etapa (2) para o arquivo `Lab_Chpt5.dat`.

O novo banco de dados criado é para uma livraria bem simples na Internet. A tabela **BOOKS** contém todas as informações sobre os livros do estoque da loja. A tabela **CUSTOMERS** contém informações sobre cada um dos clientes da loja. Finalmente, a tabela **SALES** contém dados de vendas. Sempre que um cliente compra um livro, é feito um registro na tabela **SALES**. O diagrama abaixo mostra o design e o relacionamento entre as tabelas.



Parte 2: Criar um script de instalação para o banco de dados EXPRESS

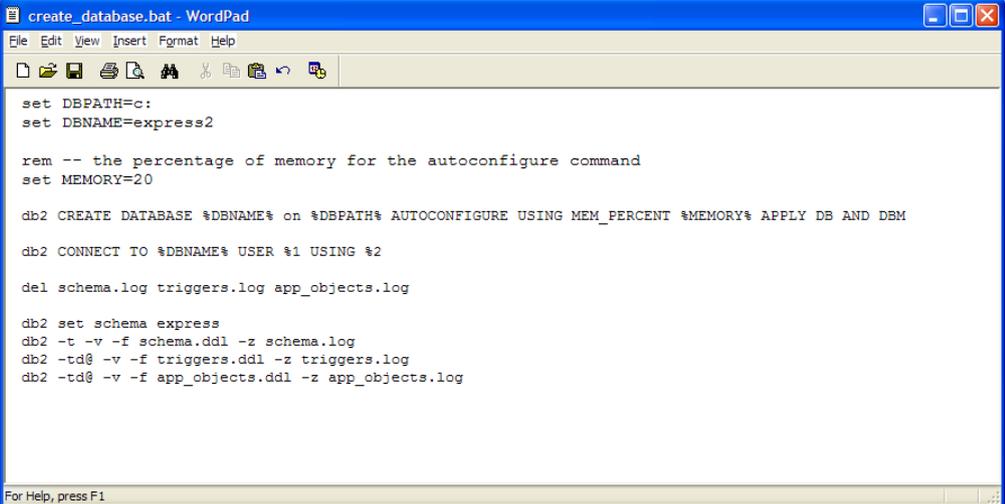
Os scripts são um mecanismo poderoso para criar tarefas repetitivas como coleta de estatísticas de banco de dados, backups e implementação de banco de dados. Os scripts de sistema operacional têm a vantagem de suportar parâmetros de script, tornando-os mais flexíveis. Nesta parte, você criará um script de sistema operacional para reimplementar o banco de dados **EXPRESS** como o banco de dados **EXPRESS2**. O script chamará scripts SQL para os objetos do banco de dados, gerados previamente. Para economizar espaço, este exercício mostra os scripts e comandos específicos à plataforma

Windows. Se você prefere trabalhar com Linux, procure fazer as alterações apropriadas nas instruções abaixo.

Procedimento

1. Abra um editor de texto e insira as informações mostradas abaixo. A maioria das pessoas apresenta erros ao digitar as linhas abaixo. Propositamente, não fornecemos isto como um arquivo separado, para que você erre e aprenda a corrigi-los por si mesmo!

Observe também que pode ser necessário especificar os caminhos corretos para os arquivos `schema.ddl`, `triggers.ddl` e `app_objects.ddl`, que também são fornecidos com o arquivo `expressc_book_exercises_9.7.zip` que acompanha este livro.



```
set DBPATH=c:
set DBNAME=express2

rem -- the percentage of memory for the autoconfigure command
set MEMORY=20

db2 CREATE DATABASE %DBNAME% on %DBPATH% AUTOCONFIGURE USING MEM_PERCENT %MEMORY% APPLY DB AND DBM

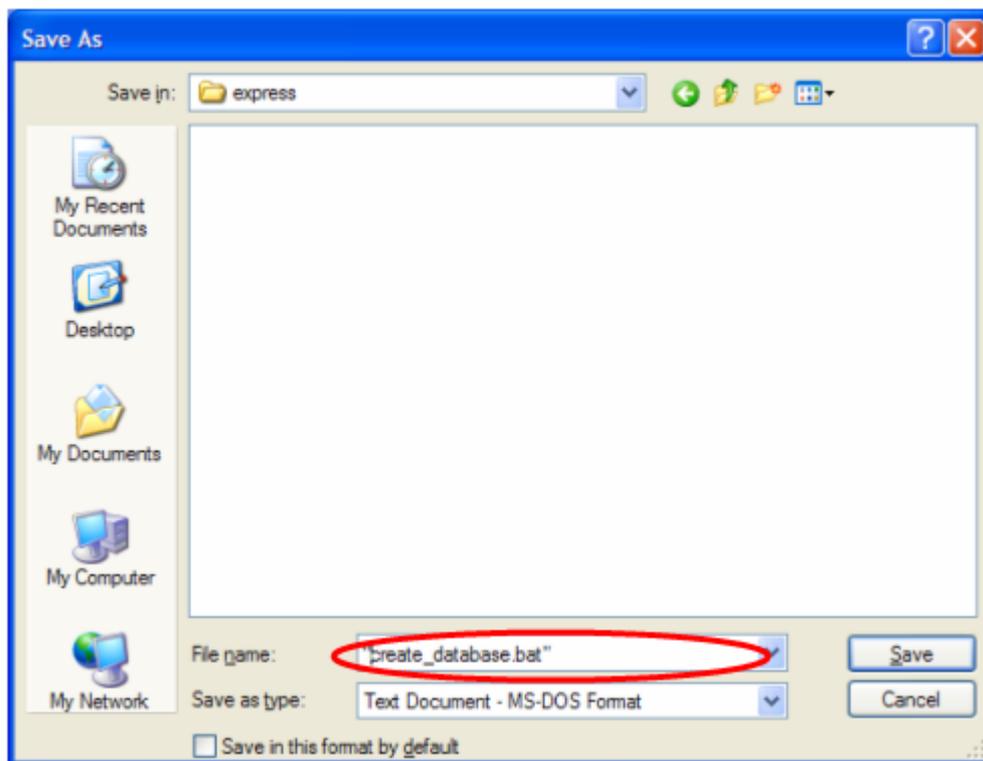
db2 CONNECT TO %DBNAME% USER %1 USING %2

del schema.log triggers.log app_objects.log

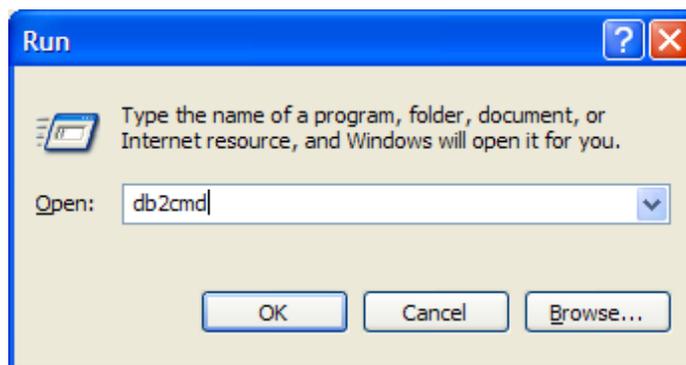
db2 set schema express
db2 -t -v -f schema.ddl -z schema.log
db2 -td@ -v -f triggers.ddl -z triggers.log
db2 -td@ -v -f app_objects.ddl -z app_objects.log
```

2. Salve o arquivo do script em um diretório como `C:\express`, e o nomeie `create_database.bat`.

Observação: Se você usa o Wordpad, na caixa de diálogo *Salvar como*, verifique se escolheu a opção *Formato MS-DOS*. Se salvar o arquivo em um formato diferente, o Wordpad pode introduzir caracteres invisíveis que causarão problemas na execução do script. Além disso, coloque aspas no nome do arquivo, como mostrado na figura abaixo, para assegurar que o Windows não anexe uma extensão `.TXT` a ele.

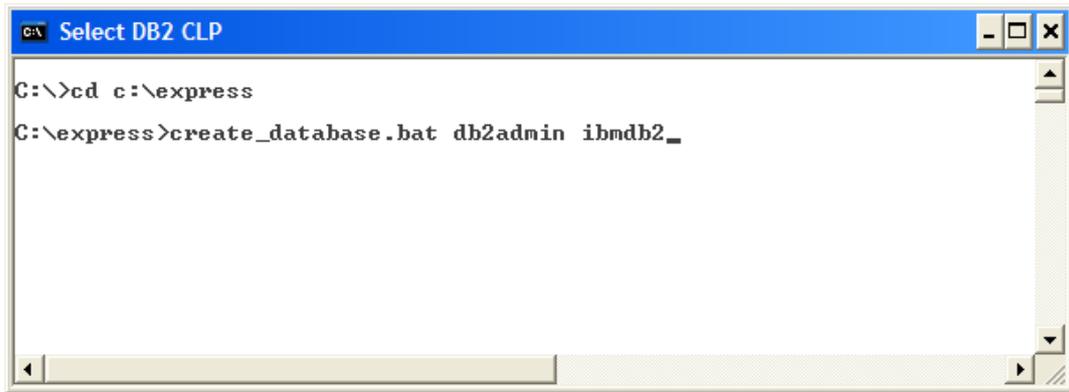


3. Para executar scripts que interagem com o DB2, é necessário ter um ambiente de linha de comando do DB2. Para abrir a Janela de comando, vá para *Iniciar -> Programas -> IBM DB2 -> DB2COPY1 (Default) -> Command Line Tools -> Command Window*. Como alternativa, você pode usar *Iniciar-> Executar*, digitar **db2cmd** e pressionar *enter*, como mostrado abaixo.



4. Em seguida, para executar o script, digite estes comandos na Janela de comando:

```
cd C:\express  
create_database.bat db2admin ibmdb2
```



```
C:\>cd c:\express
C:\express>create_database.bat db2admin ibmdb2_
```

5. Faça uma pausa para se familiarizar com o script que acabou de criar. Você compreende o que está ocorrendo em cada linha?
6. Tente responder as perguntas a seguir:
 - A. Onde a conexão com o banco de dados está sendo estabelecida?
 - B. O que %1 e %2 significam?
 - C. O que a linha de código abaixo faz? Onde ela é usada? Para quê?
`set DBPATH=C:`
 - D. O que a linha de código abaixo faz?
`del schema.log, triggers.log, app_objects.log`
 - E. O que ocorre quando o script é chamado sem qualquer parâmetro?
 - F. Por que os scripts SQL não contêm instruções CONNECT TO? Como eles se conectam ao banco de dados?

**PART II – APRENDENDO O DB2:
ADMINISTRAÇÃO DO BANCO DE
DADOS**

6

Capítulo 6 – Arquitetura do DB2

Neste capítulo, discutiremos rapidamente a arquitetura do DB2. Você aprenderá sobre:

- O modelo de processo do DB2
- O modelo de memória do DB2
- O modelo de armazenamento do DB2

Observação:

Para obter mais informações sobre a arquitetura do DB2, assista ao vídeo:

<http://www.channeldb2.com/video/video/show?id=807741:Video:4482>

6.1 O modelo de processo do DB2

A *Figura 6.1* descreve o modelo de processo do DB2. Nesta figura, os retângulos representam processos e as elipses representam os threads. O principal processo do DB2 é denominado db2sysc. Este processo engloba diversos threads, e o principal também é denominado db2sysc. Este é o thread principal, que dá origem a outros threads. Quando um aplicativo remoto tenta se conectar ao servidor usando uma instrução SQL CONNECT, as escutas remotas do protocolo de comunicação receberão esta solicitação e entrarão em contato com um agente coordenador do DB2 (db2agent). Um agente do DB2 é como um pequeno operário que executa operações em nome do DB2. Quando o aplicativo é local, ou seja, está sendo executado no mesmo servidor do DB2, as etapas são muito semelhantes, porém um agente db2ipccm processa a solicitação em vez do thread db2tccpm. Em alguns casos, como quando o paralelismo está ativado, um db2agent pode dar origem a outros agentes, que aparecem como threads db2agntp. Outros agentes mostrados na figura, como db2pfchr, db2loggr, db2dlock, também podem ser usados para finalidades diferentes. Os processos mais comuns são descritos na *Tabela 6.1*, e os threads mais comuns são descritos na *Tabela 6.2*.

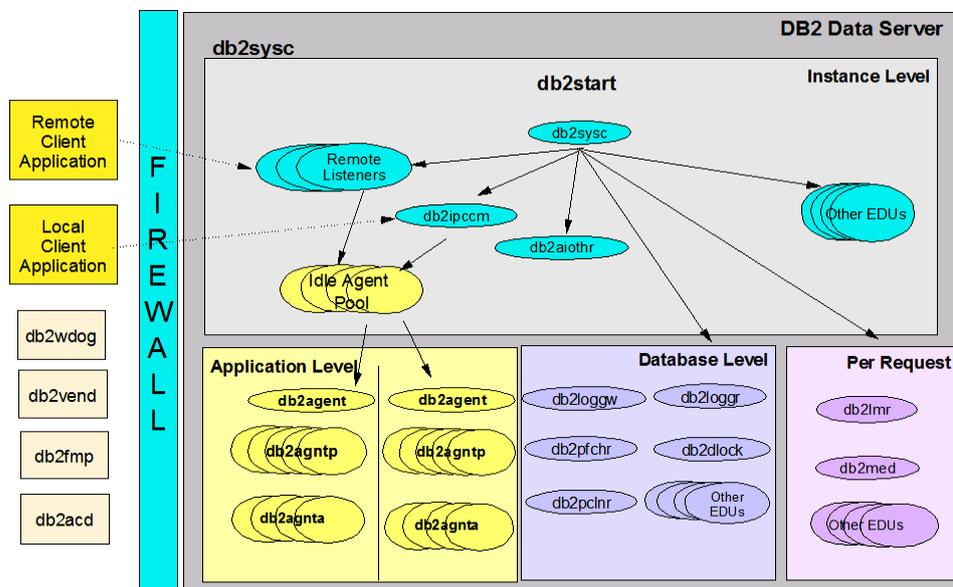


Figura 6.1 – O modelo de processo do DB2

Nome do processo	Descrição
db2sysc (Linux) db2syscs (Win)	O principal controlador ou mecanismo do sistema DB2. A partir do DB2 9.5, há apenas um único processo de mecanismo principal multithread para a partição inteira. Todas as EDUs (Engine Dispatchable Units) são threads dentro deste processo. Sem este processo, o servidor de banco de dados não pode funcionar.
db2acd	O daemon de computação autônomo. Ele é usado para executar tarefas automáticas pelo lado do cliente, como o health monitor, os utilitários de manutenção automática e o admin scheduler. Este processo era chamado anteriormente de db2hmon.
db2wdog	O watchdog do DB2. O watchdog é o pai do processo do mecanismo principal, db2sysc. Ele libera os recursos caso o processo db2sysc seja encerrado de maneira anormal.
db2vend	O processo de fornecedor cercado introduzido no DB2 9.5. Todo código de fornecedores terceirizados é executado neste processo, fora do mecanismo. 3Os aplicativos de terceiros são programas não-IBM que podem interagir com o DB2; por exemplo, o arquivamento de log pode ser gerenciado por um código de fornecedor terceirizado, através da especificação de um parâmetro de rotina de saída de usuário que aponte para este código.

db2fmp	Processos cercados que executam código do usuário no servidor fora do firewall, para procedimentos armazenados e funções definidas pelo usuário. Este processo substitui ambos os processos db2udf e db2dari que foram usados em versões anteriores do DB2.
--------	---

Tabela 6.1 – Processos comuns do DB2

Nome do thread	Descrição
db2sysc	O thread controlador do sistema. Este thread é responsável pela inicialização e desligamento, e pelo gerenciamento da instância sendo executada
db2tcpcm	Escuta de comunicações TCP/IP
db2agent	Agente coordenador que executa operações de banco de dados em nome dos aplicativos (pelo menos 1 por conexão, dependendo do Connection Concentrator estar ativado ou não)
db2agntp	Sub-agente ativo gerado se INTRA_PARALLEL estiver definido como YES. Este thread executa operações de banco de dados para o aplicativo. O db2agent coordenará o trabalho entre os diferentes sub-agentes db2agntp
db2pfchr	Obtenção antecipada de dados de E/S assíncrona do DB2 (NUM_IOSERVERS)
db2pclnr	Gravação de dados de E/S assíncrona do DB2 (NUM_IOCLEANERS)

Tabela 6.2 – Threads comuns do DB2

6.2 O modelo de memória do DB2

O modelo de memória do DB2 consiste em áreas diferentes da memória no nível de instância, nível de banco de dados e nível de aplicativos e agentes, como mostrado na *Figura 6.2*. Neste livro não explicaremos em detalhes cada uma das áreas diferentes da memória, mas apenas forneceremos uma rápida visão geral.

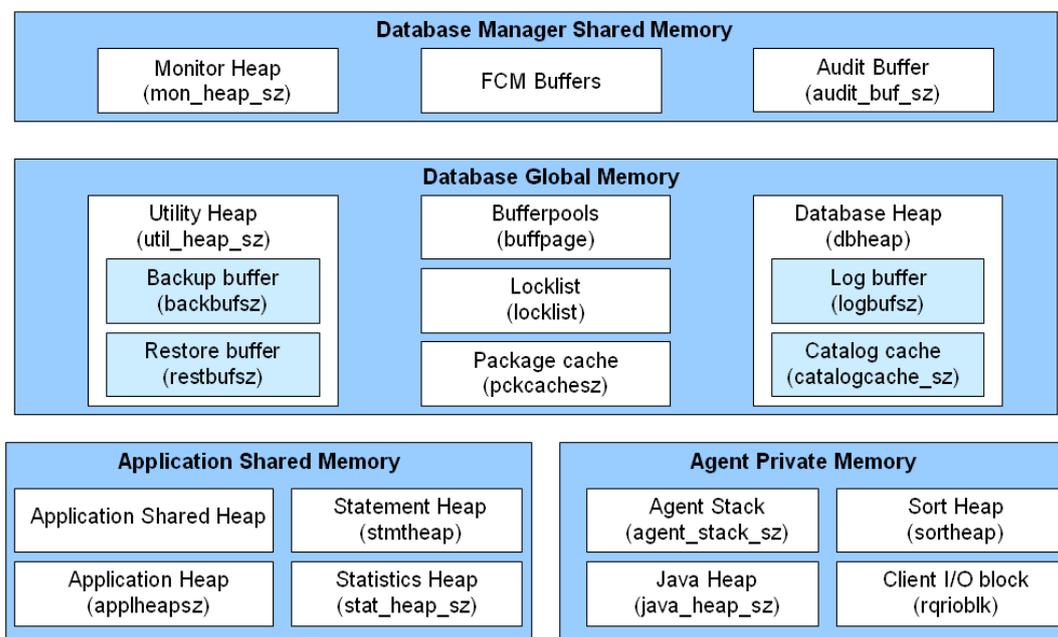


Figura 6.2 – O modelo de memória do DB2

Quando é iniciada uma instância, a memória compartilhada do gerenciador do banco de dados é alocada. Haverá uma destas áreas de memória alocada para cada instância. Isto normalmente não ocupa muito espaço, normalmente menos de 100 MB. Ao conectar-se a um banco de dados pela primeira vez, a memória global do banco de dados é alocada. Haverá uma destas áreas de memória alocada para cada banco de dados ao qual você se conecte. Neste bloco, o bufferpool é uma das partes mais importantes, especialmente para melhorar o desempenho das consultas. O tamanho dos bufferpools determinará o tamanho de toda a memória global de banco de dados.

A memória compartilhada com o aplicativo é uma área na qual os aplicativos e agentes do DB2 podem compartilhar informações, como seções de planos de acesso, compilação de SQL e XQuery, coleta de estatísticas, etc.

A memória particular do agente é aquela usada por cada agente do DB2. Sem usar o concentrador de conexão, cada conexão requer um agente. Normalmente, um agente usa aproximadamente de 3 a 5 MB. Com o concentrador de conexão, diversas conexões podem usar um único agente, reduzindo desta forma a necessidade de mais memória física.

6.3 O modelo de armazenamento do DB2

Nesta seção, descreveremos os conceitos abaixo:

- Páginas e extensões
- Bufferpool
- Tablespace

6.3.1 Páginas e extensões

Uma página é a unidade mínima de armazenamento do DB2. Os tamanhos de página permitidos são: 4K, 8K, 16K e 32K. Uma extensão é um agrupamento de páginas. Em termos de desempenho, seria dispendioso trabalhar com uma única página de cada vez no DB2. Assim, em vez disso, o DB2 trabalha com extensões. O tamanho da página e o tamanho da extensão são definidos ao trabalhar com os bufferpools e tablespaces, como veremos nas próximas seções.

6.3.2 Bufferpools

Um bufferpool é um cache de memória real para dados de tabela e de índice. Ele aumenta o desempenho, reduzindo as E/S sequenciais diretas, e promovendo a leitura (busca prévia) e gravação assíncronas. Isto é, o DB2 antecipa quais páginas serão necessárias e realiza uma pesquisa prévia no disco para o bufferpool, para que fiquem prontas para o uso.

Os bufferpools são alocados em unidades de memória de páginas com 4K, 8K, 16K e 32K. Deve haver pelo menos um bufferpool por banco de dados, e pelo menos um bufferpool correspondente para um tablespace de um dado tamanho de página.

6.3.2.1 Como criar um bufferpool

Para criar um bufferpool, use a instrução **CREATE BUFFERPOOL**. Como alternativa, use o Control Center para clicar com o botão direito do mouse na pasta Buffer Pool dentro de um determinado banco de dados e selecione *Create*, (Criar) como mostrado na *Figura 6.3*

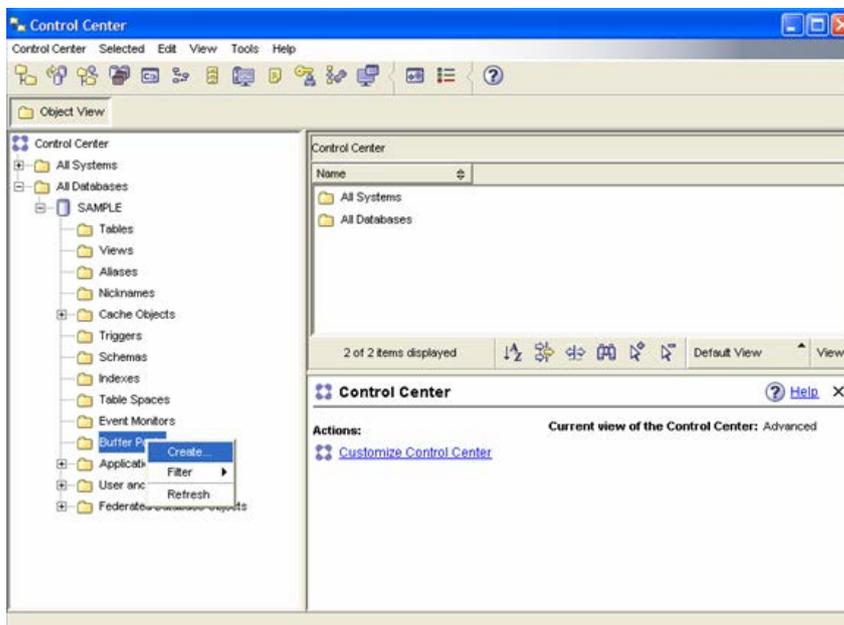


Figura 6.3 – Como criar um bufferpool

Depois de clicar em *Create*, a caixa de diálogo *Create Buffer Pool* (Criar bufferpool) é aberta, como mostrado na *Figura 6.4*

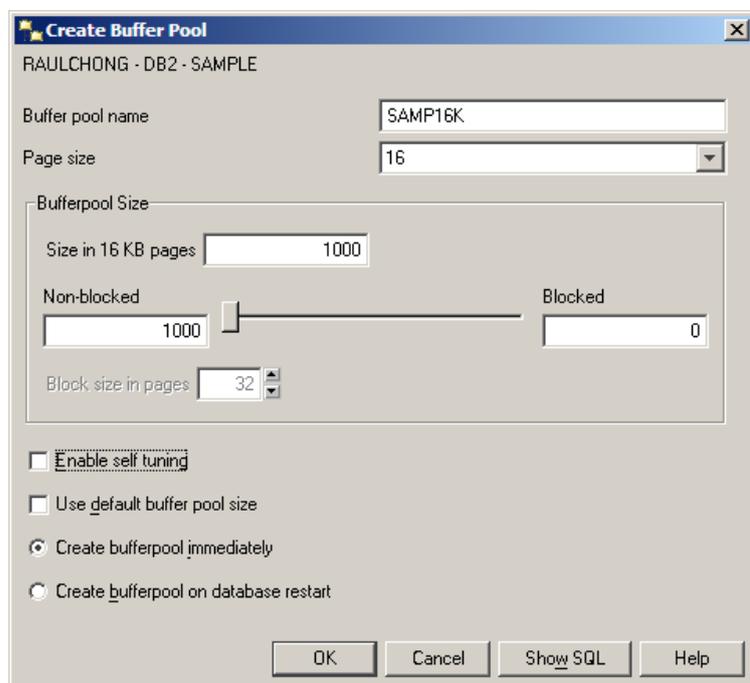


Figura 6.4 – Caixa de diálogo Create a buffer pool

A maioria das entradas na *Figura 6.4* explica-se por si mesma. Os campos *Non-blocked* (Sem bloco) e *Blocked* (Com bloco) referem-se ao número de páginas que devem existir como blocos ou não. Os bufferpools baseados em bloco asseguram que sejam movidas páginas contíguas do disco para o bufferpool de forma que também fiquem contíguas em uma área em bloco; isto pode aumentar o desempenho. O número de páginas não deve ser maior que 98 por cento do número de páginas do bufferpool. Especificar o valor como 0 desativa a E/S em bloco.

Depois de criar o bufferpool, ele é exibido no Control Center, como mostrado na *Figura 6.5*.

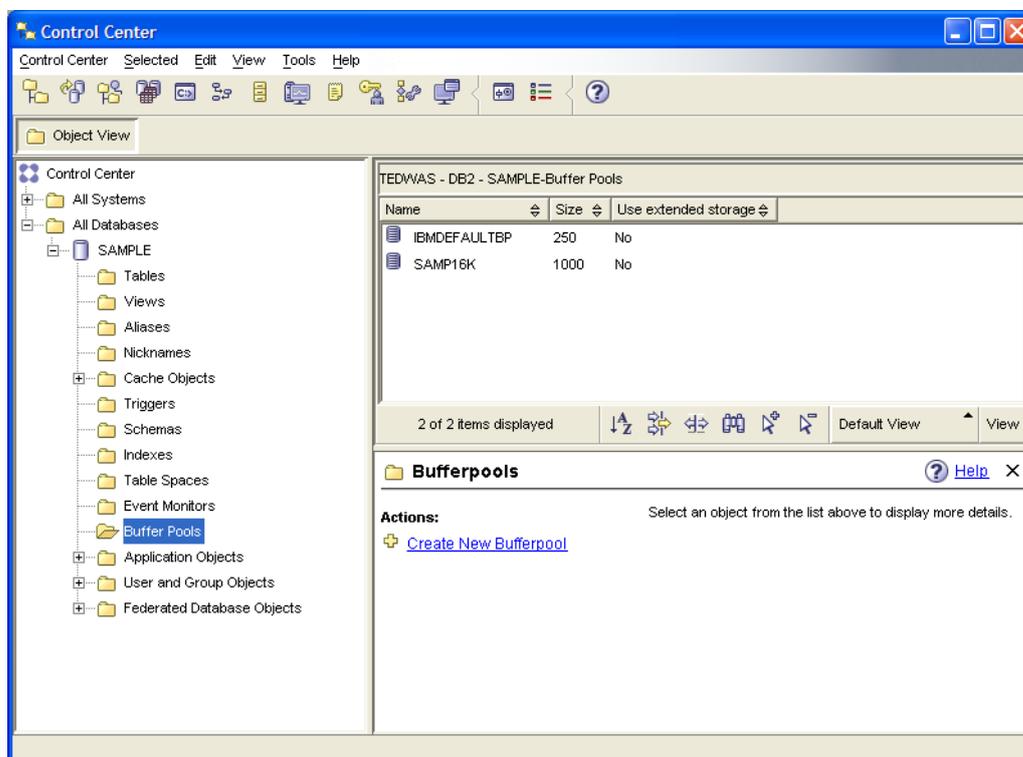


Figura 6.5 – O Control Center após a criação do bufferpool SAMP16K

6.3.3 Tablespaces

Os tablespaces são uma interface lógica entre as tabelas lógicas e a memória física do sistema (bufferpool) e contêineres (discos). Use a instrução **CREATE TABLESPACE** para criar um tablespace onde você pode especificar:

- O tamanho de página do tablespace (4KB, 8KB, 16KB ou 32KB). O tamanho de página deve corresponder a um bufferpool com o mesmo tamanho de página.
- O nome do bufferpool associado a este tablespace.
- Um tamanho de extensão.
- Um tamanho de busca prévia.

6.3.3.1 Tipos de tablespace

Há três tipos de tablespaces:

- Regular
 - Esses são para tabelas de usuário. Por exemplo, o tablespace USERSPACE1 criado como padrão é um tablespace regular.
- Grande

Esses são usados opcionalmente para separar os dados de LOB em seus próprios tablespaces. Ele também é usado para armazenar dados XML para os bancos de dados criados com suporte a pureXML usando o tipo de dados XML nas colunas. O padrão é tablespaces grandes.

- Temporário

Há dois tipos de tablespaces temporários:

- Temporários do sistema

Estes são usados pelo DB2 para operações internas, como classificações. Por exemplo, o tablespace TEMPSPACE1, criado como padrão quando se cria um banco de dados, é um tablespace temporário do sistema. É necessário que haja pelo menos um tablespace temporário do sistema.

- Temporários do usuário

Estes são usados para criar DGTTs (Tabelas temporárias globais declaradas) e CGTTs (Criar tabelas temporárias globais) definidas pelo usuário, que são tabelas temporárias na memória. Elas frequentemente são confundidas com os tablespaces temporários do sistema. Os usuários devem criar um tablespace temporário do usuário antes de poder usar DGTTs ou CGTTs.

6.3.3.2 Gerenciamento de tablespace

Os tablespaces podem ser classificados de acordo com a maneira pela qual são gerenciados. Isso pode ser especificado na instrução **CREATE TABLESPACE**.

Gerenciados pelo sistema

Este tipo de tablespace é conhecido por SMS (System Managed Storage, ou Armazenamento gerenciado pelo sistema). Isso significa que o sistema operacional gerencia o armazenamento. São de fácil gerenciamento, e os contêineres são diretórios do sistema de arquivos. O espaço não é alocado previamente, mas os arquivos crescem dinamicamente. Após especificar os contêineres, estes são fixados no momento da criação e não é possível adicionar outros contêineres posteriormente, a menos que seja realizada uma restauração redirecionada. Ao usar os tablespaces SMS, os dados da tabela, índice e dados de LOB não podem ser distribuídos em tablespaces diferentes.

Gerenciados pelo banco de dados

Este tipo de tablespace é conhecido por DMS (Database Managed Storage, ou Armazenamento gerenciado pelo banco de dados). Isto significa que o DB2 gerencia o armazenamento. O gerenciamento do espaço requer mais intervenção manual de um DBA. Os contêineres podem ser arquivos alocados previamente ou dispositivos brutos. Nos dispositivos brutos, os dados são gravados diretamente sem cache de SO.

Os contêineres podem ser adicionados, descartados ou redimensionados usando a instrução ALTER TABLESPACE. Os tablespaces DMS são melhores para o desempenho,

e os dados da tabela, índice e dados de LOB podem ser divididos em tablespaces separados, o que aumenta o desempenho.

Gerenciados por armazenamento automático

Este tipo de tablespace é gerenciado pelo armazenamento automático, e pode aproveitar a facilidade de uso semelhante a dos tablespaces SMS, mas com o desempenho melhor e a flexibilidade dos tablespaces DMS. Portanto, a partir do DB2 9, este é o tipo de tablespace padrão. Com estes tablespaces, primeiro o usuário especifica um caminho de armazenamento e um grupo lógico de dispositivos de armazenamento que será usado pelo DB2 para gerenciar o espaço. Não são fornecidas definições explícitas de contêiner. Os contêineres são criados automaticamente nos caminhos de armazenamento. O crescimento dos contêineres existentes e a adição de novos contêineres são completamente gerenciados pelo DB2. Quando um caminho de armazenamento não é especificado no comando **CREATE DATABASE**, o caminho do banco de dados é usado como caminho de armazenamento. O caminho do banco de dados é onde reside a definição principal do banco de dados. Se o caminho do banco de dados não for especificado, ele será obtido do parâmetro DFTDBPATH de configuração de gerenciador de banco de dados. No Windows, este só pode ser uma unidade e não um caminho.

Para permitir o armazenamento automático, primeiro é necessário criar um banco de dados com o armazenamento automático ativado (este é o comportamento padrão) e associar a ele um conjunto de caminhos de armazenamento. Se necessário, você pode redefinir os caminhos de armazenamento após a criação, usando a operação RESTORE de banco de dados. Em seguida, você pode criar tablespaces para usar o armazenamento automático (novamente, este é o comportamento padrão).

O tablespace gerenciado por armazenamento automático é bem semelhante aos tablespaces DMS, mas as operações foram automatizadas e são gerenciadas pelo DB2; isto inclui a atribuição e alocação de contêineres e o redimensionamento automático.

Vamos examinar um exemplo de tablespace gerenciado por armazenamento automático. Primeiro, crie o banco de dados com o armazenamento automático ativado, como nestes exemplos:

O armazenamento automático é ativado como padrão:

```
CREATE DATABASE DB1
```

O armazenamento automático é especificado explicitamente:

```
CREATE DATABASE DB1 AUTOMATIC STORAGE YES
```

O armazenamento automático é ativado como padrão, mas os caminhos de armazenamento são indicados. Se o caminho de armazenamento for um diretório, ele deve ser criado previamente:

Exemplo no Windows:

```
CREATE DATABASE DB1 ON C:/, C:/storagepath1, D:/storagepath2
```

Observe que o primeiro item da lista é uma unidade, pois ela representa um caminho de banco de dados que só poderá ser uma unidade e não um caminho no Windows. Este item também será usado como um dos caminhos de armazenamento. Assim, o caminho do banco de dados é C:, e os caminhos de armazenamento consistem em C:, C:\storagepath1 e D:\storagepath2, onde os dois últimos diretórios devem ser criados previamente.

Exemplo no Linux:

```
CREATE DATABASE DB1 ON /data/path1, /data/path2
```

O armazenamento automático é explicitamente desativado:

```
CREATE DATABASE DB1 AUTOMATIC STORAGE NO
```

Em seguida, crie o tablespace com o armazenamento automático ativado, como nestes exemplos:

O armazenamento automático para tablespaces também é ativado como padrão:

```
CREATE TEMPORARY TABLESPACE TEMPTS
```

O armazenamento automático para o tablespace é especificado explicitamente:

```
CREATE TABLESPACE TS2 MANAGED BY AUTOMATIC STORAGE
```

O armazenamento automático é especificado implicitamente, o tamanho inicial é alocado, juntamente com o valor do aumento e o tamanho máximo até o qual poderá aumentar.

```
CREATE TABLESPACE TS1  
  INITIALSIZE 500 K  
  INCREASESIZE 100 K  
  MAXSIZE 100 M
```

6.3.3.3 Como os dados são armazenados nos tablespaces

Como padrão, o DB2 gravará no disco extensões em faixas pelos contêineres. Por exemplo, se você tem um tablespace de 4K com tamanho de extensão 8, usando 3 contêineres brutos em um tablespace DMS, isso significa que 32K de dados (4K x 8 páginas por extensão = 32K) serão gravados em um disco antes de começar a gravar no próximo. Isso é mostrado na *Figura 6.6*. Observe que as tabelas não compartilham extensões.

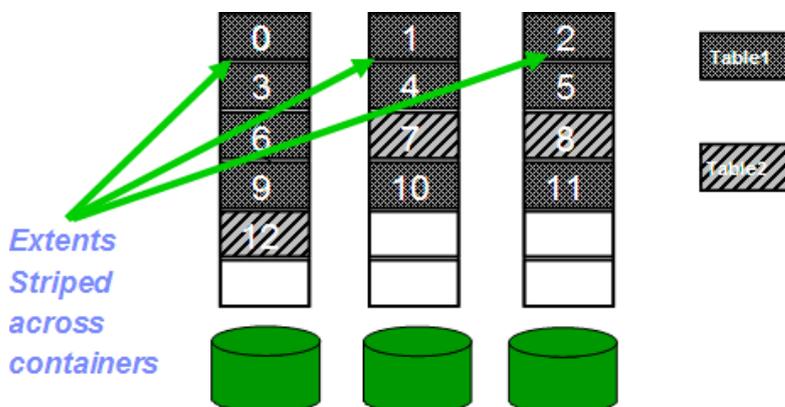


Figura 6.6 – Gravação de dados em tablespaces

6.3.3.4 Criação de um tablespace usando o Control Center

Para criar um tablespace no Control Center, clique com o botão direito na pasta *Table Spaces* dentro de um dado banco de dados e selecione *Create* (Criar) como mostrado na *Figura 6.7*. O assistente *Create table space* (Criar tablespace) será exibido, como mostrado na *Figura 6.8*.

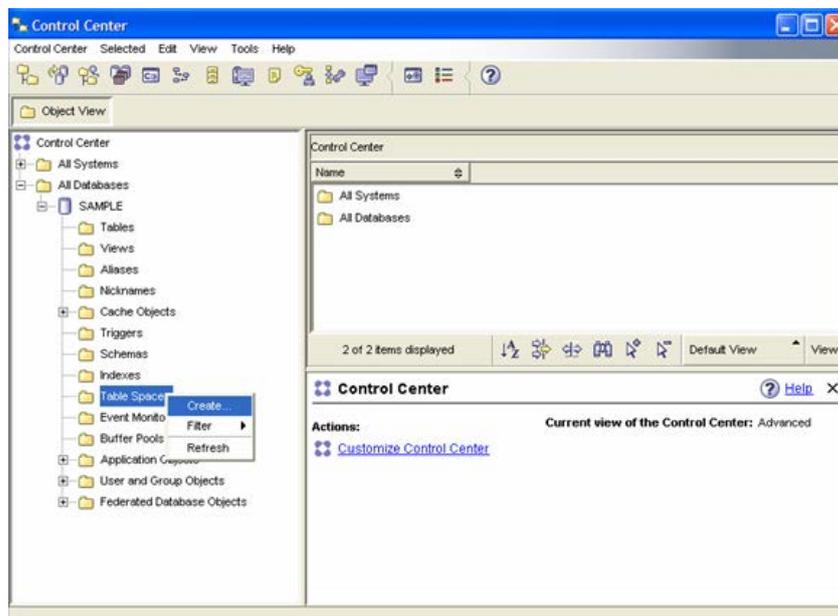


Figura 6.7 – Criação de um tablespace do Control Center

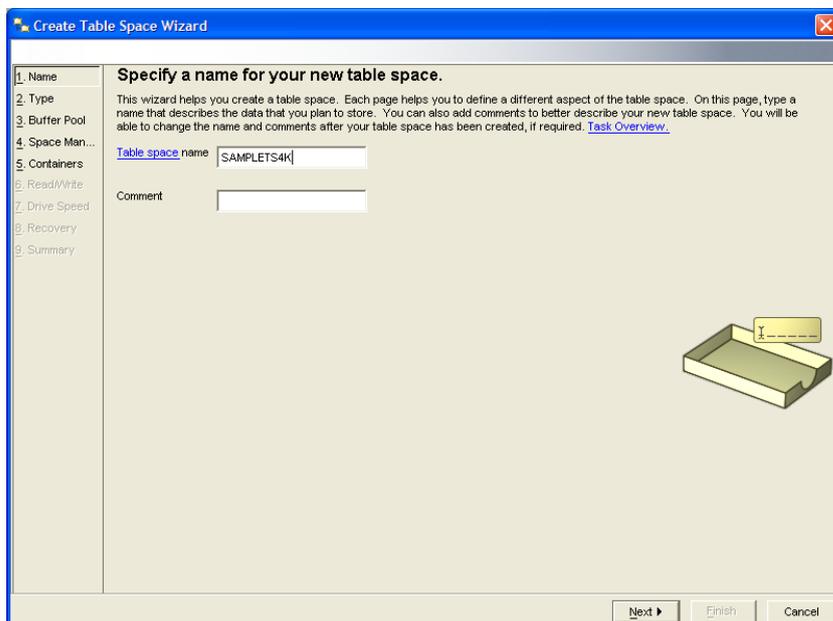


Figura 6.8 – Assistente Create Table Space

O assistente mostrado na *Figura 6.8* irá guiá-lo pelas etapas da criação do tablespace.

6.4 Resumo

Neste capítulo exploramos os três aspectos principais da arquitetura do DB2: o modelo de processo, o modelo de memória e o modelo de armazenamento. No modelo de processo, examinamos os processos e threads comuns, incluindo o db2sysc, sem o qual o DB2 não poderia funcionar.

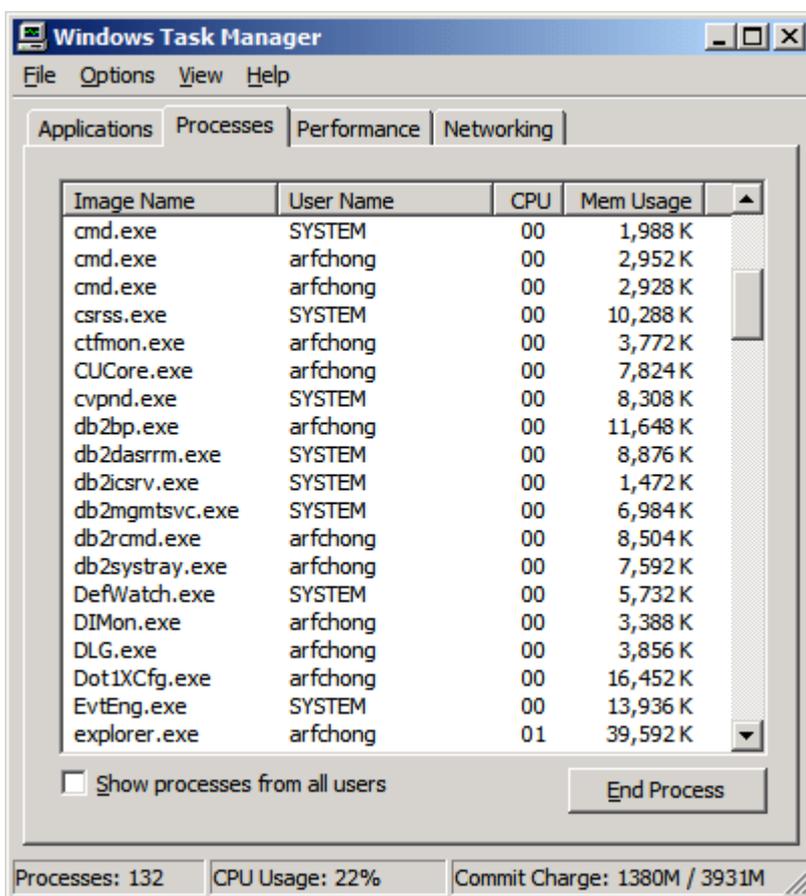
O modelo de armazenamento foi discutido em detalhes, abrangendo os três aspectos mais importantes: páginas e extensões, o bufferpool (incluindo detalhes de criação) e os tablespaces. Finalmente, analisamos os diversos tipos de tablespaces, juntamente com detalhes sobre como são gerenciados (SMS, DMS, automático) e como criar um novo tablespace usando o Control Center (Centro de Controle).

6.5 Exercícios

Este exercício o ajudará a compreender o modelo de processo, modelo de memória e modelo de armazenamento do DB2 no Windows. Você irá analisar os diferentes processos e threads, monitorar o uso de memória e praticar a criação de banco de dados usando armazenamento automático e caminhos de armazenamento no Windows. De um modo ideal, os caminhos de armazenamento devem ser criados em todos os discos (unidades), mas como seu computador pode não estar configurado com diversos discos, este exercício usa somente sua unidade C:\.

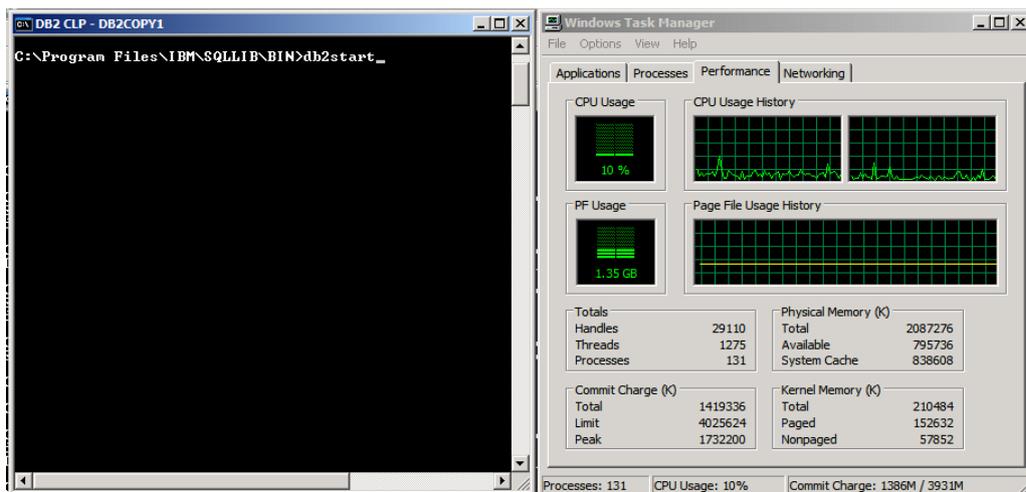
Procedimento:

1. Vejamos alguns processos no Windows. Primeiramente, abra a Janela de comando do DB2 (*Iniciar -> Executar -> db2cmd*) e assegure que sua instância seja interrompida, usando este comando: **db2stop force**
2. Abra o Gerenciador de Tarefas do Windows, selecione a guia *Processos*; verifique se a caixa *Mostrar processos de todos os usuários* está selecionada. Clique na coluna Nome da imagem para classificar segundo esta coluna; em seguida, procure o processo db2sysc.exe, como mostrado na figura abaixo.

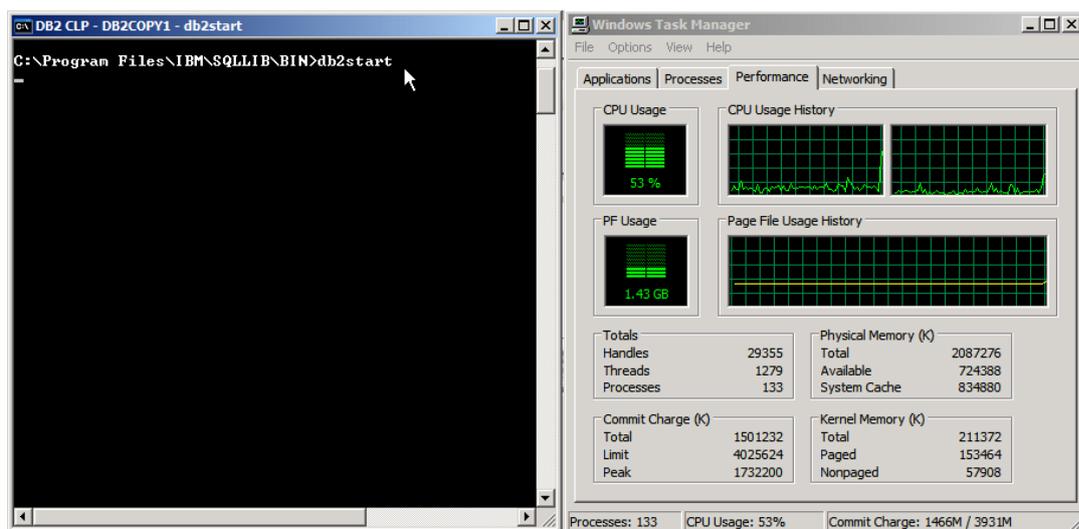


3. Você não deve encontrar o processo db2syscs.exe porque, na etapa um, solicitamos que interrompesse a instância do DB2.
4. Inicie a instância do DB2 com este comando: **db2start**, e repita a etapa anterior. Agora você consegue encontrar o processo db2syscs.exe?
5. Agora vamos examinar o consumo de CPU e memória. Siga as etapas abaixo:
 - G. Assegure que não haja nada em execução no seu sistema, fechando todos os aplicativos

- H. Abra uma nova Janela de comando do DB2 e execute: **db2stop force**
- I. No Gerenciador de Tarefas, selecione a guia Desempenho
- J. Mantenha o Gerenciador de Tarefas e a Janela de comando do DB2 abertos lado a lado, como mostrado abaixo. Anote a quantidade de memória física disponível e o uso de CPU.



- K. Execute **db2start** e ao mesmo tempo, monitore o consumo de CPU e de memória logo que executar este comando. Você deve ver um rápido salto em uso de CPU, e uma redução da memória disponível em cerca de 50 MB a 70 MB. Esta é a quantidade de memória que a instância do DB2 consome. Se você executar **db2stop force** novamente, a memória deve retornar ao seu valor anterior.



6. Repita a etapa anterior; desta vez, entretanto, monitore o que ocorre após conectar-se ao banco de dados SAMPLE. Execute estes comandos na Janela de comando do DB2:

```
db2start
db2 connect to sample
```

Logo que se conectar ao banco de dados SAMPLE, você verá uma redução na quantidade de memória física disponível. Isto se deve ao fato de que, logo ao se conectar a um banco de dados, a memória global do banco de dados (bufferpool, cache de catálogo, etc.) é alocada.

7. Repita a etapa anterior mas, desta vez, monitore o que ocorre após criar um bufferpool de qualquer tamanho. Procure não exceder a memória física do seu computador. Se o fizer, o DB2 não alocará o bufferpool imediatamente, mas adiará a criação até a desativação do banco de dados. Além disso, um pequeno bufferpool de sistema será usado, e o DB2 continuará usando-o até haver memória suficiente. Por exemplo, para criar um bufferpool de cerca de 160 MB, execute este comando enquanto estiver conectado ao banco de dados SAMPLE:

```
db2 create bufferpool mybp immediate size 5000 pagesize 32k
```

8. Crie um banco de dados **mydb1** usando armazenamento automático, e onde o caminho do banco de dados seja a unidade C:, e os caminhos de armazenamento sejam C:, C:\mystorage1, C:\mystorage2. Execute o comando abaixo na Janela de comando do DB2:

```
db2 create database mydb1 on C:\, C:\mystorage1, C:\mystorage2
```

Se você executou o comando acima, provavelmente recebeu uma mensagem de erro porque primeiro é necessário criar os diretórios C:\mystorage1, C:\mystorage2. Crie-os e tente novamente!

7

Capítulo 7 – Conectividade com o cliente

Este capítulo abrange a configuração necessária para a conexão de um cliente DB2 a um servidor DB2 usando TCP/IP. Observe que o servidor DB2 é fornecido com um componente de cliente, portanto um servidor DB2 também pode se comportar como cliente para se conectar a outro servidor DB2. Há diversas maneiras de configurar a conectividade de um cliente DB2; entretanto, neste capítulo discutiremos somente o método mais fácil, que é usando o Configuration Assistant (Assistente de configuração).

Observação:

Para obter mais informações sobre a conectividade de cliente do DB2, assista ao vídeo: <http://www.channeldb2.com/video/video/show?id=807741:Video:4222>

A partir do DB2 9.7, o Configuration Assistant está obsoleto; entretanto, ele ainda poderá ser usado e é incluído no produto.

7.1 Diretórios do DB2

Os diretórios do DB2 são arquivos binários que armazenam informações sobre os bancos de dados aos quais você pode se conectar a partir de sua máquina. Há quatro diretórios:

- Diretório de banco de dados do sistema
- Diretório de banco de dados local
- Diretório do nó
- Diretório DCS

A verificação e atualização do conteúdo de todos esses diretórios pode ser executada através da ferramenta de GUI do Configuration Assistant.

7.1.1 Diretório de banco de dados do sistema

Este diretório é como o índice de um livro. Ele mostra todos os bancos de dados aos quais você pode se conectar, sejam eles locais ou remotos. Para um banco de dados local, o ponteiro será para o *Diretório de banco de dados local*. Para um banco de dados remoto, o ponteiro será para o *Diretório do nó*. Para verificar o conteúdo deste diretório, execute o comando:

```
list db directory
```

7.1.2 Diretório de banco de dados local

Este diretório contém informações sobre bancos de dados aos quais você pode se conectar e que residem em sua máquina. Para verificar o conteúdo deste diretório, execute o comando:

```
list db directory on <drive/path>
```

7.1.3 Diretório do nó

Este diretório inclui informações sobre como se conectar a um dado banco de dados remoto. Por exemplo, se for usado o protocolo TCP/IP, uma entrada de nó TCP/IP incluiria o endereço IP do servidor no qual reside o banco de dados DB2 ao qual você está tentando se conectar, e a porta da instância onde este banco de dados reside. Para verificar o conteúdo deste diretório, execute o comando:

```
list node directory
```

7.1.4 Diretório DCS

Este diretório aparecerá somente se você tiver instalado o software DB2 Connect para conectar-se ao DB2 for z/OS (mainframe) ou ao DB2 for i5/OS. Para verificar o conteúdo deste diretório, execute o comando:

```
list dcs directory
```

7.2 Configuration Assistant (obsoleto)

Usado a ferramenta de GUI do Configuration Assistant, você pode configurar facilmente a conectividade entre um cliente DB2 e um servidor DB2. Para iniciar o Configuration Assistant no Windows, selecione: *Iniciar -> Programas -> IBM DB2 -> DB2COPY1 -> Setup Tools -> Configuration Assistant*.

Na linha de comando, inicie a ferramenta com o comando `db2ca`. A *Figura 7.1* mostra o Configuration Assistant.

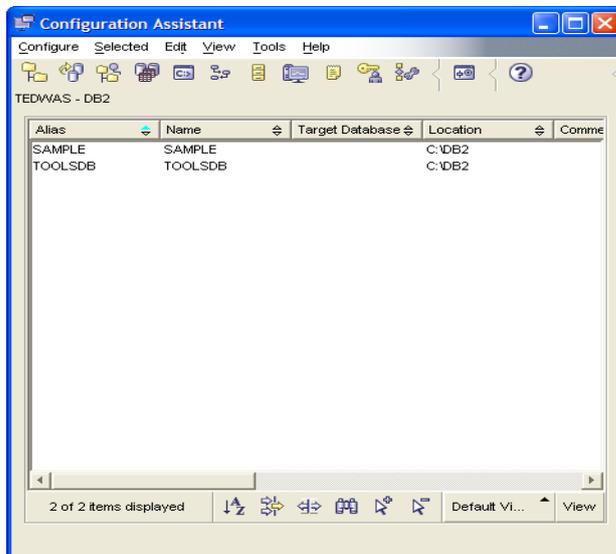


Figura 7.1 – O Configuration Assistant

7.2.1 Configuração necessária no servidor

Há duas configurações que devem ser feitas no servidor:

1) DB2COMM

Esta variável de registro do DB2 determina que protocolo de comunicação as escutas devem monitorar nas solicitações de clientes. Normalmente, o protocolo usado com mais frequência é o TCP/IP. Alterar este parâmetro requer que a instância seja reiniciada. Para verificar e alterar o valor de DB2COMM no Configuration Assistant, selecione *Configure -> DB2 Registry* (Configurar -> Registro do DB2) como mostrado na *Figura 7.2* e na *Figura 7.3*.

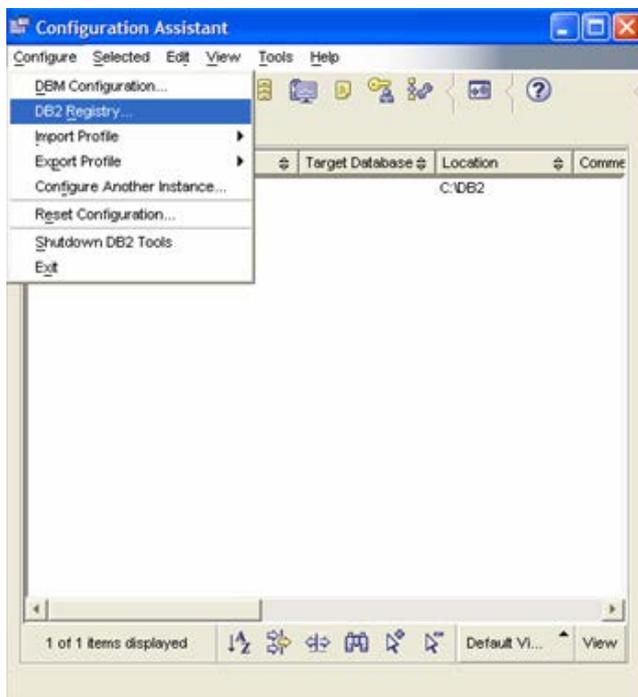


Figura 7.2 – Como acessar o DB2 Registry

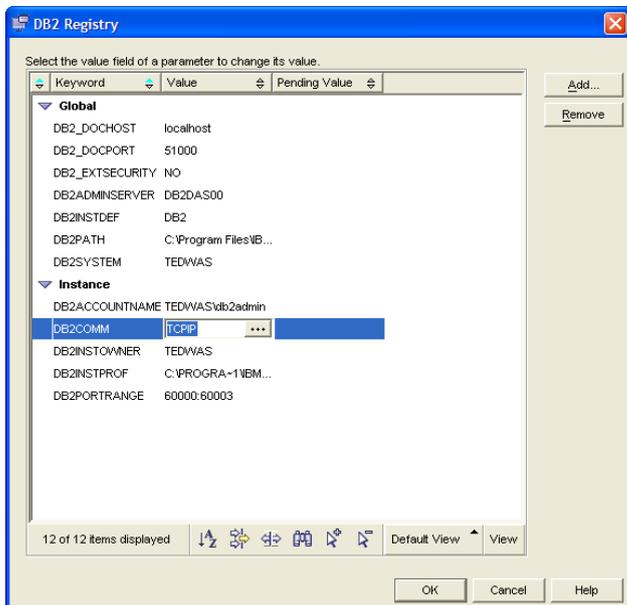


Figura 7.3 – Confirmação do valor da variável DB2COMM do DB2 Registry

2) SVCENAME

Este parâmetro de configuração de gerenciador de banco de dados deve ser configurado com o nome do serviço (como definido no arquivo de serviços de TCP/IP) ou com o número da porta a ser usada quando você deseja acessar os bancos de dados desta instância. No Configuration Assistant, selecione *Configure -> DBM configuration* como mostrado na *Figura 7.4*

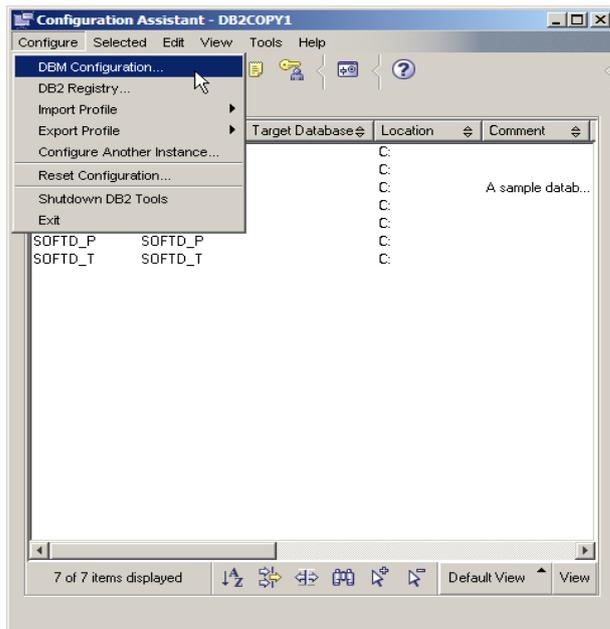


Figura 7.4 – Verificação do dbm cfg no Configuration Assistant

Após entrar na janela DBM Configuration, encontre a seção Communications (Comunicação) e procure SVCENAME. Você pode alterar o valor de uma cadeia ou mesmo um número de porta, se necessário. Isto é mostrado na *Figura 7.5*.

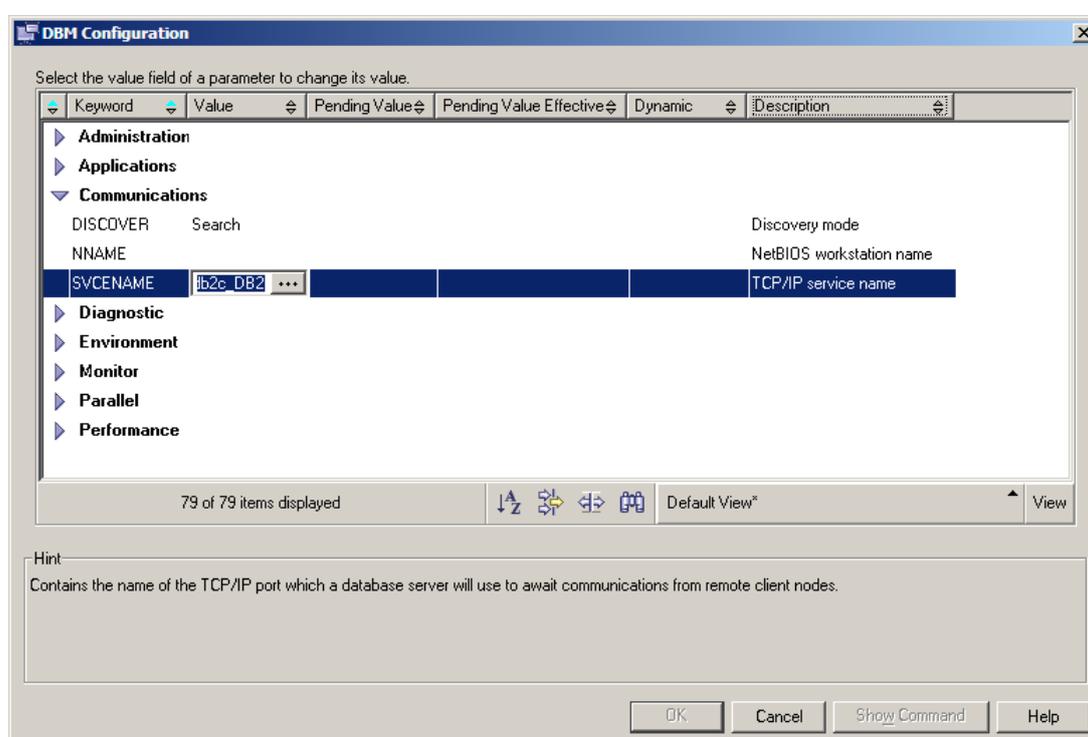


Figura 7.5 –Verificação do parâmetro de dbm cfg SVCENAME

7.2.2 Configuração necessária no cliente

No cliente, primeiro é necessário conhecer estas informações:

- O nome do banco de dados ao qual deseja se conectar
- O número da porta da instância do DB2 no servidor onde reside o banco de dados. Você também pode usar um nome de serviço, desde que haja uma entrada correspondente no arquivo de serviços de TCP/IP
- O ID de usuário e a senha do sistema operacional para se conectar ao banco de dados. Este ID de usuário deve ter sido definido anteriormente no servidor

As informações acima podem ser inseridas a partir do cliente DB2, usando o Configuration Assistant. Primeiro, inicie o assistente *Add Database* (Adicionar banco de dados), clicando em *Selected -> Add Database Using Wizard* (Selecionado -> Adicionar banco de dados usando o assistente) como mostrado na *Figura 7.6*

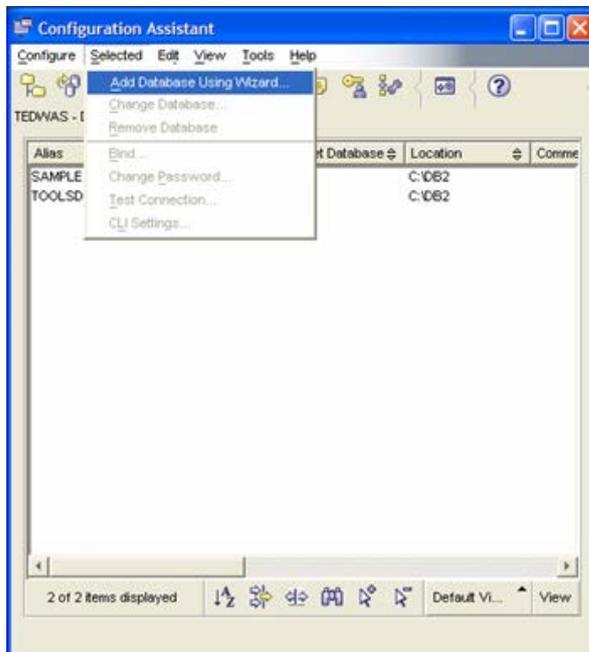


Figura 7.6 – Como chamar o assistente Add Database

Você também pode chegar a este assistente clicando com o botão direito no espaço em branco do Configuration Assistant e selecionando *Add Database Using Wizard*. A Figura 7.7 mostra o assistente Add Database.

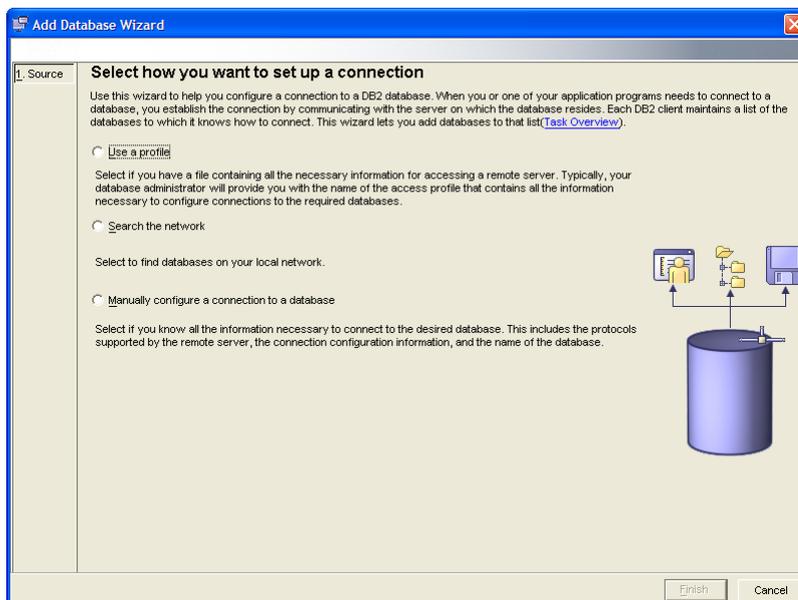


Figura 7.7 –Assistente Add Database

No assistente Add Database, há três opções:

1. Use a Profile (Usar um perfil)

Pode haver situações onde seja necessário configurar diversos clientes para se conectarem ao mesmo servidor DB2. Nestas situações, é conveniente realizar todas as configurações de um cliente e armazenar essas configurações em um arquivo de “perfil”. Com este arquivo, você pode carregar todas as informações diretamente para os outros clientes. Na *Figura 7.7*, ao escolher *Use a Profile* (Usar um perfil), as informações seriam carregadas a partir de um “perfil” existente. Mais detalhes descrevendo como criar perfis de cliente e servidor serão fornecidos mais adiante neste capítulo.

2. Search the network (Pesquisar a rede)

Este método, também conhecido como *Discovery* (Descoberta), faz o DB2 pesquisar um dado servidor, instância e banco de dados na rede. Para que este método funcione, o DAS deve estar sendo executado em cada servidor DB2 onde os bancos de dados deverão ser descobertos. Com este método, há duas maneiras de realizar a pesquisa:

- Pesquisa:

Pesquisa a rede inteira. Isto não é recomendado se sua rede for grande e tiver muitos hubs, pois a recuperação de dados de todos os sistemas iria demorar muito

- Conhecido:

Pesquisa a rede em busca de um servidor conhecido em um endereço fornecido por você.

Os dois métodos são ilustrados na *Figura 7.8*

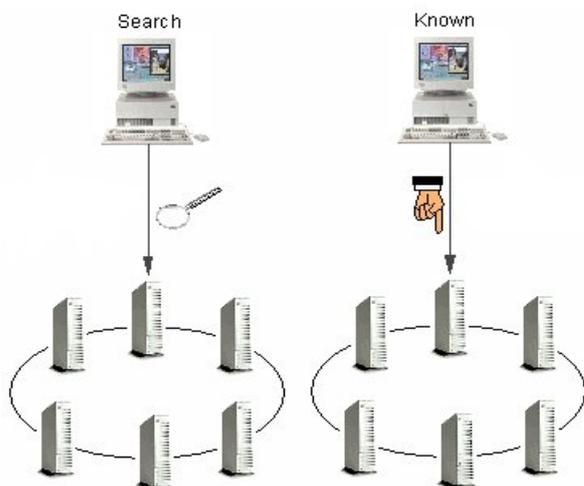


Figura 7.8 – Os métodos de pesquisa (ou descoberta) Pesquisar e Conhecido

Pode haver circunstâncias onde um administrador não deseje que os clientes pesquem a rede em busca de bancos de dados com informações confidenciais. Isto pode ser evitado no nível de DAS, da instância ou do banco de dados. A *Figura 7.9* fornece detalhes sobre isto.

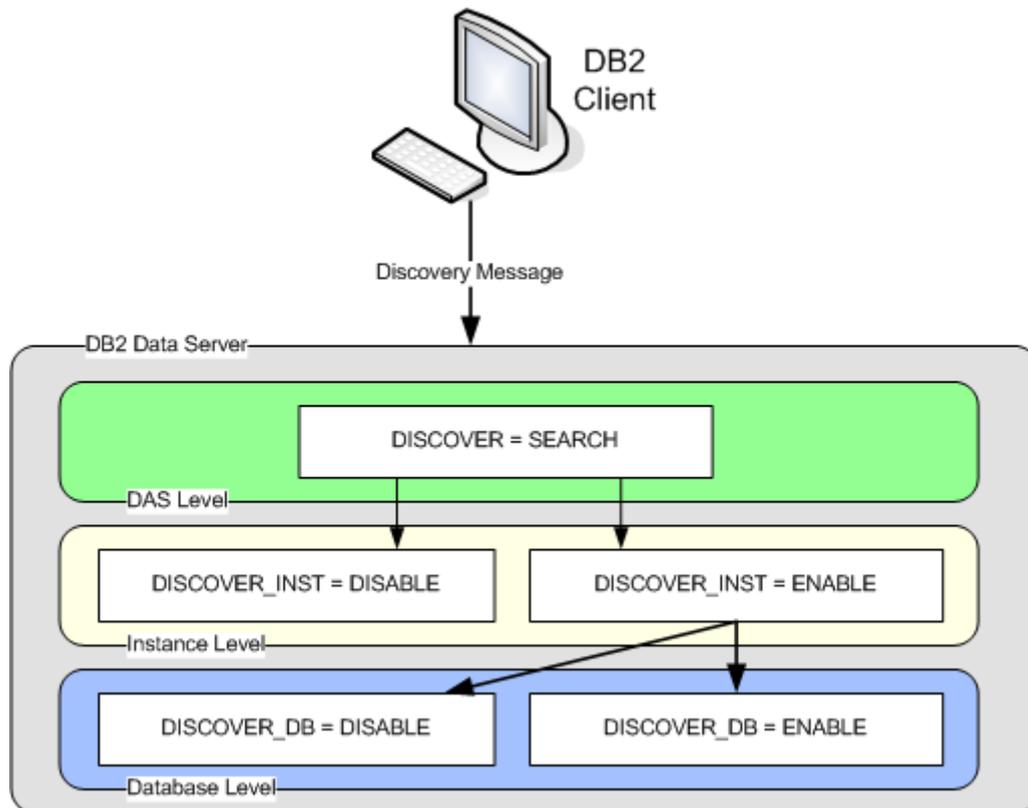


Figura 7.9 – Configuração de parâmetros para permitir a descoberta

A *Figura 7.9* exibe os níveis diferentes onde você pode ativar ou desativar a descoberta. No nível de DAS, você pode fornecer ao parâmetro DISCOVER o valor SEARCH ou KNOWN. No nível de instância, o parâmetro de configuração de gerenciador de banco de dados DISCOVER_INST pode ser definido como DISABLE ou ENABLE. Finalmente, no nível de banco de dados, o parâmetro DISCOVER_DB também pode ser definido como ENABLE ou DISABLE. Configurar esses parâmetros de acordo fornece granularidade à descoberta de bancos de dados.

3. Configurar manualmente uma conexão a um banco de dados

Usando este método, você adiciona manualmente as informações de nome do host, números de porta e banco de dados ao Configuration Assistant, que em seguida irá gerar comandos de catálogo para executar a configuração de conectividade. O Configuration Assistant não verifica se as informações estão corretas. Você saberá que estão incorretas se não conseguir se conectar a um servidor. Além disso, assegure que o ID de usuário e a senha fornecidos para se conectar ao banco de dados remoto estão corretos. Como

padrão, a autenticação é realizada no servidor DB2 com o qual você está tentando se conectar; portanto, é necessário fornecer um ID de usuário e uma senha definidos neste servidor.

7.2.3 Como criar perfis de cliente e servidor

Se você for configurar um grande número de servidores ou clientes em vez de definir cada um individualmente, você pode configurar apenas um e exportar um perfil do mesmo, para em seguida aplicar o perfil aos outros clientes/servidores. Isto economiza muito tempo de administração ao configurar o ambiente.

Para criar um perfil personalizado no Configuration Assistant, clique no menu *Configure* (Configurar) e, em seguida, selecione *Export Profile* -> *Customize* (Exportar perfil -> Personalizar), como mostrado na *Figura 7.10*

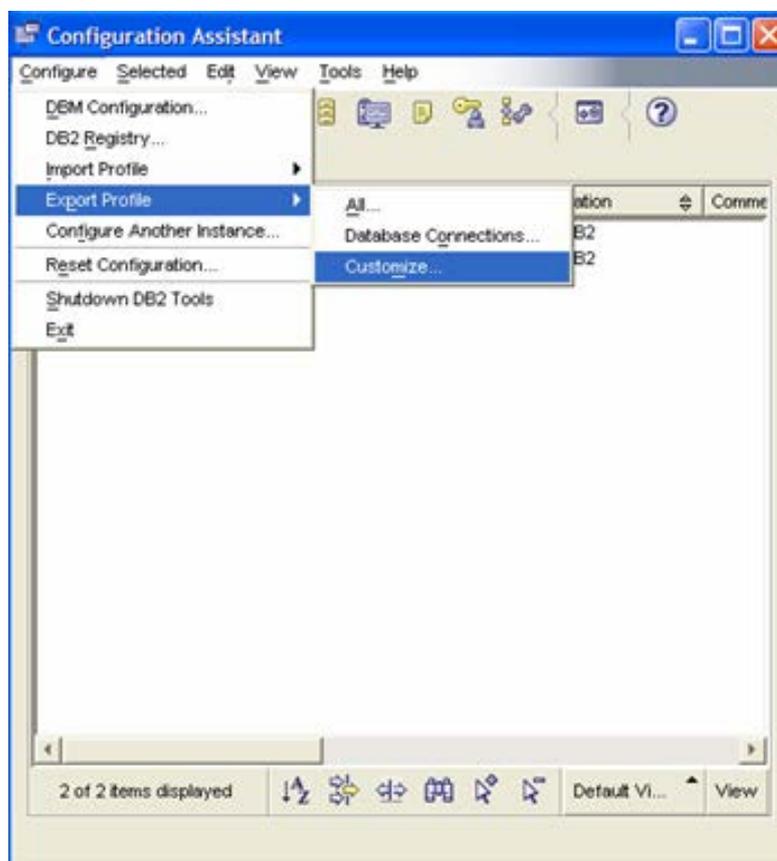


Figura 7.10 – Exportação de um perfil

A *Figura 7.11* mostra os campos que precisam ser preenchidos para exportar um perfil

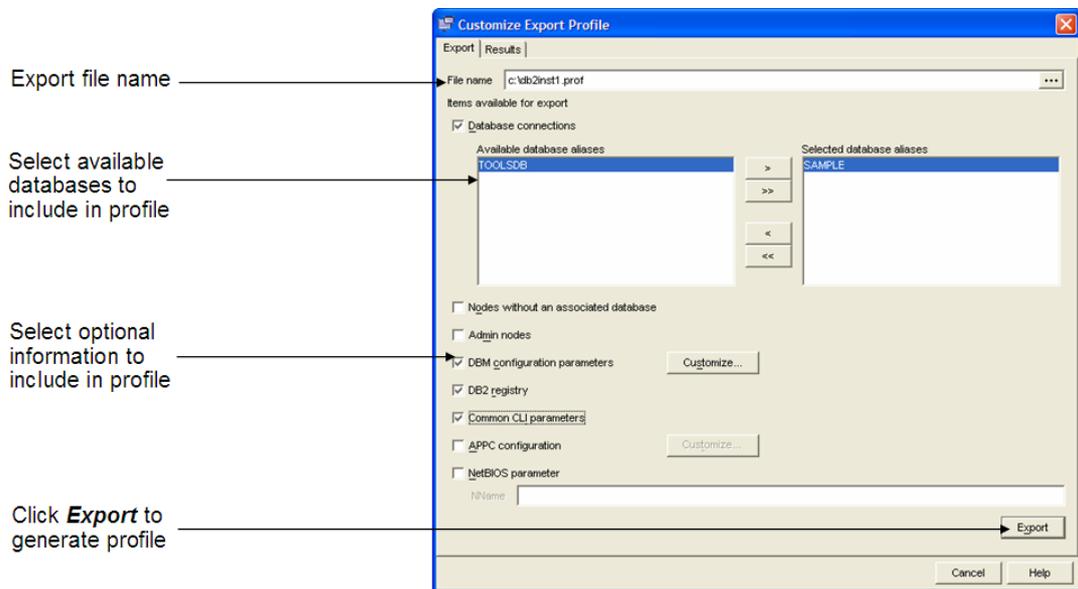


Figura 7.11 – Caixa de diálogo Customize Export Profile

A *Figura 7.12* exibe os resultados após clicar em “Export” (Exportar) na caixa de diálogo Customize Export Profile (Personalizar perfil de exportação).

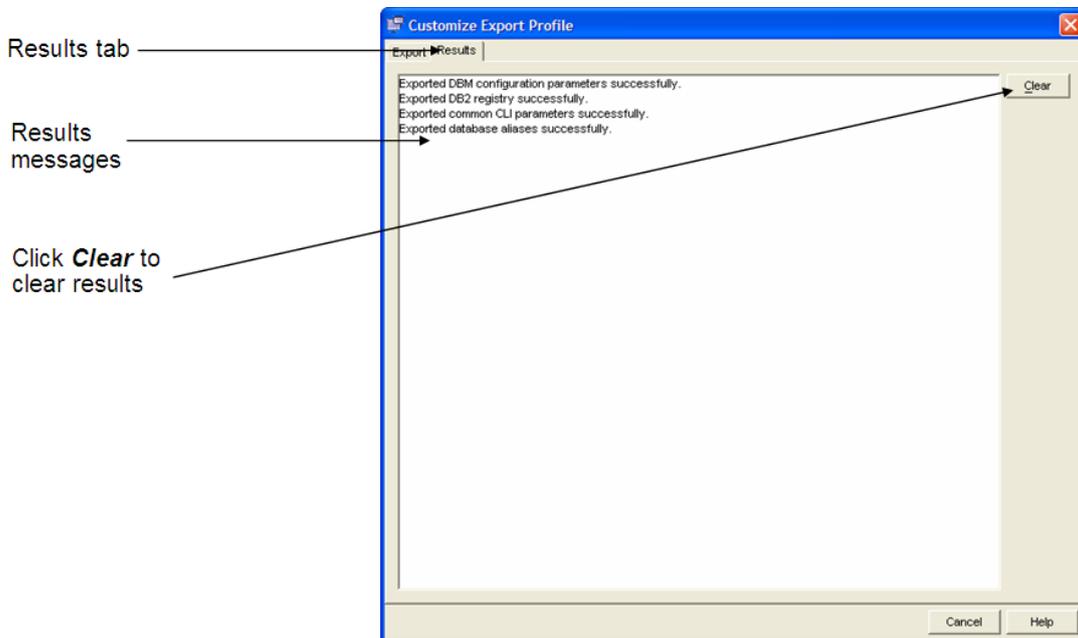


Figura 7.12 – Resultados de Export Profile

Para importar um perfil personalizado no Configuration Assistant, clique no menu *Configure* e, em seguida, selecione *Import Profile -> Customize* (Importar perfil -> Personalizar), como mostrado na *Figura 7.13*

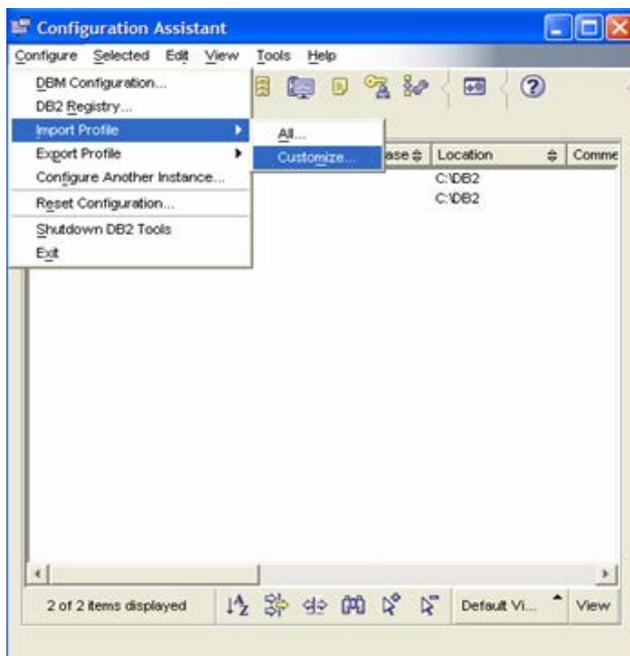


Figura 7.13 – Importação de um perfil

A Figura 7.14 mostra os campos que precisam ser preenchidos para importar um perfil

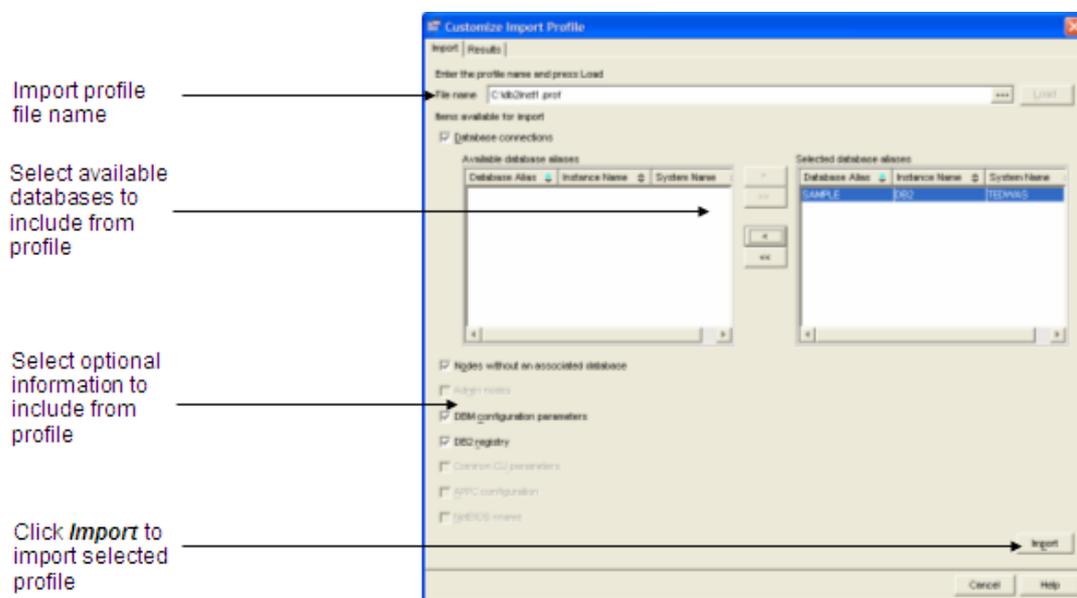


Figura 7.14 – Caixa de diálogo Customize Import Profile

7.3 Resumo

A conexão de um cliente de dados a um servidor é um aspecto essencial do gerenciamento de bancos de dados relacionais. Neste capítulo, examinamos a conectividade de cliente, começando pela finalidade e conteúdo do banco de dados e diretórios de nó associados ao DB2.

Em seguida, discutimos o uso da GUI do Configuration Assistant para configurar a conexão entre cliente e servidor, incluindo o que necessita ser configurado em ambos os lados da conexão.

Além disso, analisamos como o assistente Add Database também pode ser usado para fazer uma conexão ao servidor através de um de três métodos: usar um perfil armazenado, pesquisar a rede (também conhecido por Descoberta) ou manualmente, digitando as informações do servidor. Em seguida, a criação de perfil de cliente e servidor foi abordada em mais detalhes.

7.4 Exercícios

O Configuration Assistant pode ser usado para configurar de forma fácil e rápida as conexões com os bancos de dados remotos. Neste exercício, você irá catalogar um banco de dados residente em um servidor DB2 remoto (representado pela estação de trabalho do seu colega, usando ambos os modos Search e Discover). Após o banco de dados ser catalogado, você será capaz de acessá-lo como se estivesse em seu sistema local. O DB2 realiza todos os processos de comunicação de maneira oculta.

Este exercício supõe que você está trabalhando dentro de uma rede. Se este não for o caso, você sempre pode usar seu próprio computador como ambas as máquinas cliente e servidor, e seguir as instruções de configuração abaixo para conectar-se ao seu próprio sistema.

Procedimento

1. Solicite ao seu colega de grupo de trabalho (ou instrutor) as informações a seguir:
2. Informações do banco de dados remoto:

(PR)	Protocolo	<u>TCPIP</u>
(IP)	Endereço IP ou nome do host	_____
(PN)	Número da porta da instância	_____
(DB)	Nome do banco de dados:	<u>SAMPLE</u>

Dicas:

Para obter o nome do host no Windows, digite **hostname** em uma janela de comando

Para obter o endereço IP no Windows, digite **ipconfig** em uma janela de comando

3. Abra o Configuration Assistant. (Dica: ele é acessível através do menu Iniciar).

4. Abra o menu *Selected* e selecione *Add Database Using Wizard*.
5. Na página *Source* (Fonte) do assistente, selecione a opção *Manually Configure a Connection to a Database* (Configurar uma conexão com o banco de dados manualmente). Clique no botão *Next* para passar para a próxima página do assistente.
6. Na página *Protocol* (Protocolo) do assistente, selecione a opção TCP/IP. Clique no botão *Next* para passar para a próxima página do assistente.
7. Na página *TCP/IP* do assistente, digite o nome do host ou endereço IP completo que anotou na etapa (1). Digite o número da porta anotado na etapa (1). Clique no botão *Next* para passar para a próxima página do assistente.
8. Observação: A opção de *Service Name* (Nome do serviço) pode ser usada se você tiver um item no arquivo de serviços *local* com número de porta definido que corresponda à porta sendo rastreada pela instância do servidor remoto. Ao usar esta opção, o DB2 irá procurar no arquivo de serviços da máquina *local*, e não no servidor. Você deve adicionar uma entrada neste arquivo caso deseje usar esta opção.
9. Na página *Database* (Banco de dados) do assistente, digite o nome do banco de dados definido no servidor remoto que anotou na etapa (1) no campo *Database name* (Nome do banco de dados). Observe como o campo *Database Alias* (Alias do banco de dados) é preenchido automaticamente com o mesmo valor. O alias do banco de dados é um nome que os aplicativos locais usarão para se conectar a este banco de dados. Como você já possui um banco de dados local com nome *SAMPLE* definido, o DB2 não permitirá que você catalogue outro banco de dados com o mesmo nome. Assim, você deve usar um alias diferente. Para este exemplo, altere o alias do banco de dados para *SAMPLE1*. Você pode digitar um comentário opcional sobre este banco de dados, se desejar. Clique no botão *Next* para passar para a próxima página do assistente.
10. Na página *Data Source* do assistente, você pode registrar opcionalmente este novo banco de dados (fonte de dados) como uma fonte de dados ODBC. Isto o registra automaticamente no ODBC Manager do Windows para você. Neste exemplo, desmarque a opção *Register this database for ODBC* (Registrar este banco de dados para ODBC), uma vez que você não usará ODBC. Clique no botão *Next* para passar para a próxima página do assistente.
11. Na página *Node Options* do assistente, especifique o sistema operacional do servidor onde o banco de dados remoto está localizado. Como todas as estações de trabalho neste laboratório usam Microsoft Windows, verifique se o item *Windows* da lista suspensa está selecionado. O campo Instance name (Nome da instância) deve estar configurado para *DB2*. Caso não esteja, defina seu valor para *DB2*. Clique no botão *Next* para passar para a próxima página do assistente.
12. Esta página *System Options* (Opções do sistema) do assistente oferece a oportunidade de assegurar que o sistema e o nome do host estejam corretos e de

confirmar a configuração do sistema operacional. Clique no botão *Next* para passar para a próxima página do assistente.

13. A página *Security Options* (Opções de segurança) do assistente permite especificar onde deseja que ocorra a autenticação de usuários, e que método você deseja usar. Selecione a opção *Use authentication value in server's DBM Configuration* (Usar o valor de autenticação na configuração de DBM do servidor). Isto usará o método especificado pelo parâmetro `AUTHENTICATION` do arquivo de configuração da instância remota. Clique no botão *Finish* (Concluir) para catalogar o banco de dados remoto e fechar o assistente. Aparecerá uma caixa de confirmação. Clique no botão *Test Connection* (Testar conexão) para assegurar que você pode se conectar ao banco de dados com sucesso. Além disso, assegure que o nome de usuário e a senha fornecidos sejam válidos e *definidos no servidor remoto* (já que é provável que o parâmetro `AUTHENTICATION` do servidor esteja definido com o valor `SERVER`). Caso a conexão de teste tenha êxito, você conseguiu catalogar o servidor remoto com sucesso. Caso contrário, volte ao assistente e verifique se foram especificados todos os valores corretos. (Clique no botão *Change* (Alterar) para retornar às configurações do assistente).
14. Abra o Control Center e tente visualizar as diferentes tabelas do banco de dados remoto recém-catalogado.
15. Volte ao Configuration Assistant e tente catalogar um banco de dados diferente, desta vez usando a opção *Search the Network* (Pesquisar na rede). Siga as etapas do assistente da mesma maneira que fez na configuração manual da conexão. Observe que, nas redes maiores, a descoberta pesquisada pode demorar muito para retornar resultados.

8

Capítulo 8 – Como trabalhar com objetos de banco de dados

Este capítulo discute objetos de banco de dados como esquemas, tabelas, visualizações, índices, sequências e assim por diante. Alguns objetos de aplicativos avançados de bancos de dados como disparadores, funções definidas pelo usuário (UDFs) e procedimentos armazenados são discutidos no *Capítulo 14, Introdução ao desenvolvimento de aplicativos DB2*.

Observação:

Para obter mais informações sobre como trabalhar com objetos de bancos de dados, assista ao vídeo: <http://www.channeldb2.com/video/video/show?id=807741:Video:4242>

8.1 Esquemas

Os esquemas são espaços de nome para diversos objetos de banco de dados. Eles são usados principalmente para:

- Fornecer uma indicação da propriedade do objeto ou seu relacionamento com um aplicativo
- Agrupar de maneira lógica os objetos relacionados

Todos os objetos do DB2, exceto os sinônimos públicos, têm um nome completamente qualificado em duas partes; o esquema é a primeira metade deste nome, como mostrado abaixo:

```
<schema_name>.<object_name>
```

Um nome de objeto totalmente qualificado deve ser exclusivo. Ao se conectar a um banco de dados e criar ou fazer referência a um objeto sem especificar o esquema, o DB2 utiliza o ID de usuário usado para se conectar ao banco de dados como nome do esquema. Por exemplo, ao se conectar ao banco de dados *SAMPLE* com o usuário *arfchong* e criar a tabela *artists* usando a instrução **CREATE TABLE** abaixo:

```
CREATE TABLE artists ...
```

o nome totalmente qualificado da tabela criada é *arfchong.artists*.

Você pode usar a instrução `set schema` para definir o esquema de uma sessão. A *Listagem 8.1* fornece um exemplo.

```
connect to sample user arfchong using mypsw
select * from staff ## This looks for arfchong.staff
set schema db2admin
select * from staff ## This looks for db2admin.staff
```

Listagem 8.1 – Um exemplo de como usar a instrução `set schema`

Um “sistema de concurso” pode ser usado para ilustrar o uso dos esquemas. Digamos que uma empresa está promovendo um concurso no qual os participantes precisam criar suas próprias tabelas e executar algumas operações em SQL. Todos os participantes recebem o mesmo ID de usuário para se conectar ao banco de dados e o mesmo script para criar tabelas, onde todos os objetos não estão qualificados, ou seja, não possuem um nome de esquema. Após os concorrentes entrarem no sistema do concurso, o sistema gera o nome do esquema com base em um carimbo de data/hora após a conexão. Dessa forma, o concorrente A trabalhará em uma tabela com o mesmo nome que o concorrente B, mas com esquemas diferentes, e assim não haverá um conflito em seus trabalhos.



8.2 Sinônimos (ou aliases) públicos

Uma novidade do DB2 9.7 é o conceito de sinônimo público, também conhecido como alias público. Os sinônimos públicos permitem fazer referência a objetos sem a necessidade de especificar um esquema. A *Listagem 8.2* fornece um exemplo.

```
connect to sample user arfchong using mypsw
create public synonym raul for table arfchong.staff
select * from raul
select * from arfchong.raul ## Error
connect to sample user db2admin using psw
select * from raul
```

Listagem 8.2 – Um exemplo de sinônimo público

Na *Listagem 8.2*, primeiro você se conecta ao usuário **arfchong** e cria o sinônimo público **raul** que faz referência à tabela **arfchong.staff**. O sinônimo em si não utiliza um esquema. Se você tentar usar um, receberá uma mensagem de erro. Outros usuários, como **odb2admin** no exemplo da *Listagem 8.2*, também podem usar o sinônimo **raul**, que é público.

No exemplo, se a palavra-chave **public** não fosse usada, o sinônimo criado seria um sinônimo privado. Na *Listagem 8.3*, vamos examinar o mesmo exemplo usando um sinônimo privado.

```
connect to sample user arfchong using mypsw
create synonym raul for table arfchong.staff
select * from raul
select * from arfchong.raul ## OK, it also works
connect to sample user db2admin using psw
select * from raul ## Error, cannot find db2admin.raul
select * from arfchong.raul ## OK, this works
```

Listagem 8.3 – Um exemplo de sinônimo privado

Observe que na *Listagem 8.3*, como o sinônimo é privado, não é possível fazer referência a ele, ao se conectar como outro usuário, sem especificar o esquema.

8.3 Tabelas

Uma tabela é uma coleção de dados relacionados organizados de maneira lógica em colunas e linhas. A *Listagem 8.4* abaixo fornece um exemplo de como criar uma tabela usando a instrução **CREATE TABLE**.

```
CREATE TABLE artists
(artno          SMALLINT    not null,
 name           VARCHAR(50) with default 'abc',
 classification CHAR(1)     not null,
 bio            CLOB(100K)  logged,
 picture        BLOB(2M)    not logged compact
)
IN mytbls1
```

Listagem 8.4 – Um exemplo de instrução CREATE TABLE

Nas seções a seguir, descrevemos as partes principais desta instrução **CREATE TABLE**

8.3.1 Tipos de dados

A *Figura 8.1*, obtida do Information Center, lista os tipos de dados suportados no DB2

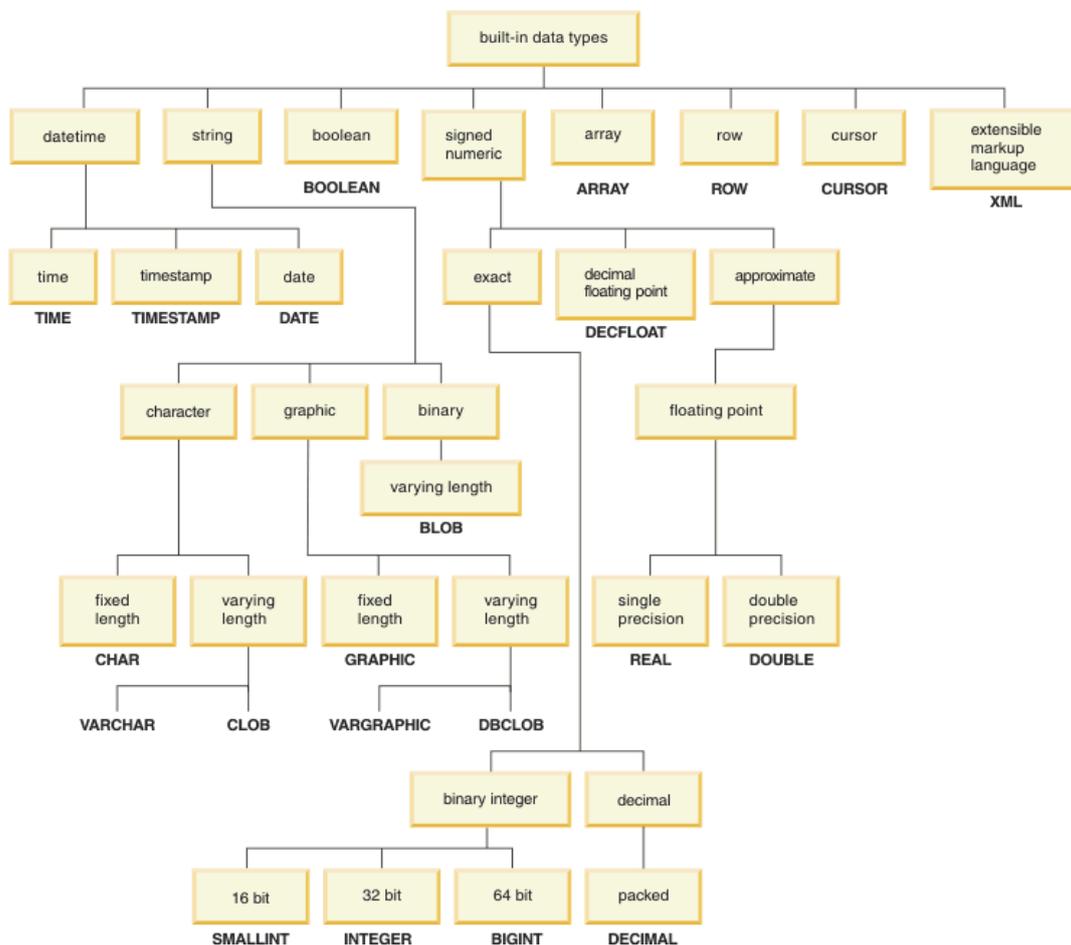


Figura 8.1 – Tipos de dados internos do DB2

Os tipos de dados mostrados na *Figura 8.1* são descritos em detalhes na documentação do DB2; sendo a maioria comum ou muito semelhante entre sistemas de gerenciamento de bancos de dados relacionais, portanto não os descreveremos aqui. Por outro lado, alguns tipos de dados como os objetos grandes (LOBs) podem não ser tão intuitivos para os novos usuários.

Os tipos de dados Objeto grande são usados para armazenar grandes cadeias de texto, grandes cadeias binárias ou arquivos, como mostrado na *Figura 8.2*

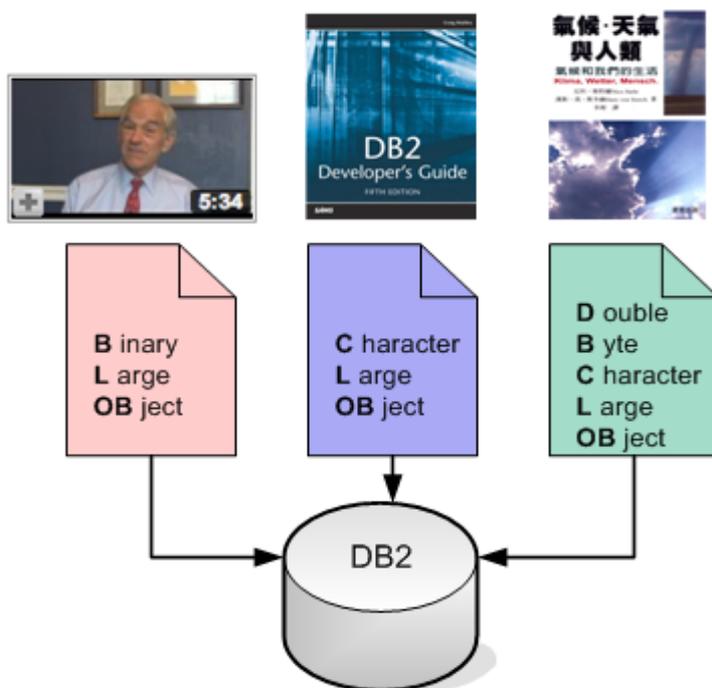


Figura 8.2 – Tipos de dados LOBs

Estes grandes objetos binários normalmente são abreviados para aumentar a clareza: um objeto binário grande é um BLOB, um objeto grande de caractere é um CLOB e um objeto grande de caracteres de byte duplo também é conhecido por DBCLOB.

A *Figura 8.1* também lista os novos tipos de dados adicionados com o DB2 9.7:

**new in
V9.7**

- BOOLEAN
- ARRAY
- ROW
- CURSOR

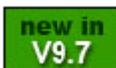
Estes fazem parte dos tipos de dados do servidor de banco de dados Oracle, e agora são suportados no DB2. Os tipos de dados do servidor de banco de dados Oracle são discutidos em mais detalhes mais adiante neste capítulo.

8.3.1.1 Tipos definidos pelo usuário

O DB2 permite definir seus próprios tipos de dados, usando os UDTs (tipos definidos pelo usuário). Os UDTs podem ser classificados como:

- Tipo distinto
- Tipo estruturado
- Tipo de referência
- Tipo matriz

- Tipo linha
- Tipo cursor



Os tipos referência, matriz, linha e cursor são novos no DB2 9.7 e são usados com as rotinas SQL PL. Os tipos de dados distintos definidos pelo usuário são baseados nos tipos de dados internos. Estes UDTs são úteis quando:

- Existe uma necessidade de estabelecer contexto para os valores
- Existe uma necessidade de que o DB2 reforce o uso de tipos de dados, usando “strong typing”

As instruções SQL na *Listagem 8.5* ilustram um exemplo de como e quando se deve usar UDTs distintos:

```
CREATE DISTINCT TYPE POUND AS INTEGER WITH COMPARISONS
CREATE DISTINCT TYPE KILOGRAM AS INTEGER WITH COMPARISONS
CREATE TABLE person
  (f_name    VARCHAR(30),
   weight_p  POUND NOT NULL,
   weight_k  KILOGRAM NOT NULL )
```

Listagem 8.5 – Um exemplo dos tipos de dados distintos

Neste exemplo, são criados dois UDTs distintos: POUND e KILOGRAM. Ambos são construídos com base no tipo de dado interno INTEGER. As cláusulas WITH COMPARISONS definidas como parte da sintaxe indicam que também são criadas funções de conversão com o mesmo nome dos tipos de dados.

A tabela `person` utiliza dois novos UDTs nas colunas `weight_p` e `weight_k`, respectivamente. Se agora executarmos a instrução abaixo:

```
SELECT F_NAME FROM PERSON
WHERE weight_p > weight_k
```

Você receberá uma mensagem de erro, pois estão sendo comparadas duas colunas com tipos de dados diferentes. Mesmo que `weight_p` e `weight_k` usem respectivamente os tipos de dados POUND e KILOGRAM, ambos criados com base no tipo de dados INTEGER, este tipo de comparação torna-se impossível com a criação dos UDTs. É exatamente isso que se deseja pois, na vida real, não faria sentido uma comparação entre libras e quilogramas.

No próximo exemplo, desejamos comparar a coluna `weight_p` com um inteiro; entretanto, estes dois tipos de dados são diferentes, e portanto você irá receber um erro, a menos que use uma função de conversão.

Como pode ver na instrução abaixo, usamos a função de conversão POUND() para tornar possível esta comparação. Como já indicamos antes, a função de conversão POUND() foi criada com o UDT ao usar a cláusula WITH COMPARISONS na instrução CREATE DISTINCT TYPE.

```
SELECT F_NAME FROM PERSON
WHERE weight_p > POUND(30)
```



8.3.1.2 Tipos de dados de servidor de banco de dados Oracle

Os tipos de dados abaixo usados no servidor de banco de dados Oracle agora são suportados com o servidor de dados DB2: NUMBER, VARCHAR2, TIMESTAMP(n), "DATE", BOOLEAN, INDEX BY, VARRAY, Row Type, Ref Cursor. Para que eles funcionem, primeiro é necessário ativar a variável de registro DB2_COMPATIBILITY_VECTOR como abaixo:

```
db2set DB2_COMPATIBILITY_VECTOR=FF
db2stop
db2start
```

Após ativar esta variável de registro, os novos bancos de dados poderão suportar esses tipos de dados. Alguns desses tipos podem ser usados somente no contexto onde é usado SQL PL.

Observação:

Se você for usar uma edição do DB2 que suporte o recurso de compatibilidade com SQL (descrito no *Capítulo 2*), você também pode usar esses tipos de dados onde o PL/SQL é usado. Se for este o caso, o valor a ser definido para a variável de registro DB2_COMPATIBILITY_VECTOR deverá ser 'FFF' em vez de 'FF', como mostrado no exemplo acima.



8.3.1.3 Conversão implícita ou "weak typing"

Muitas linguagens dinâmicas, como Ruby on Rails ou PHP, permitem a conversão implícita. Isto era um problema no DB2 devido aos seus requisitos de "strong typing". Com o DB2 9.7, as regras foram relaxadas, de modo a ser permitida a conversão implícita ou "weak typing". O que isto significa, por exemplo, é que agora você pode atribuir ou comparar cadeias a tipos numéricos, como mostrado abaixo:

```
create table t1 (coll int)
select * from t1 where coll = '42'
```

No exemplo, a cadeia '42' agora pode ser comparada à coluna do tipo inteiro *coll*.

Além disso, com o DB2 9.7, você também pode especificar marcadores de parâmetros sem tipo e NULLs sem tipo em mais situações. Antes, era necessário convertê-los explicitamente para um determinado tipo de dados. Por exemplo, a instrução abaixo agora funcionará:

```
select ?, NULL, myUDF(?, NULL) from t1
```

8.3.1.4 Valores nulos

Um valor nulo representa um estado desconhecido. A instrução CREATE TABLE pode definir uma coluna usando a cláusula NOT NULL para assegurar que a coluna contenha valores de dados conhecidos. Você também pode especificar um valor padrão para a coluna caso seja declarado NOT NULL. A próxima instrução fornece exemplos deste comportamento:

```
CREATE TABLE staff (  
    ID          SMALLINT NOT NULL,  
    NAME        VARCHAR(9),  
    DEPT        SMALLINT NOT NULL with default 10,  
    JOB         CHAR(5),  
    YEARS       SMALLINT,  
    SALARY      DECIMAL(7,2),  
    COMM        DECIMAL(7,2) with default 15  
)
```

Neste exemplo, as colunas *ID* e *DEPT* são definidas como NOT NULL. A coluna *DEPT* também inclui um valor padrão de 10 caso não seja fornecido nenhum valor.

8.3.2 Colunas de identificação

Uma coluna de identificação é uma coluna numérica que gera automaticamente um valor numérico exclusivo para cada linha inserida. Pode haver apenas uma coluna de identificação por tabela.

Há duas maneiras de gerar valores para uma coluna de identificação, dependendo de como esta foi definida:

- **Gerados sempre:** os valores sempre são gerados pelo DB2. Os aplicativos não podem fornecer um valor explícito.
- **Gerados como padrão:** os valores podem ser fornecidos explicitamente por um aplicativo ou, caso não seja fornecido nenhum valor, o DB2 gera um valor. O DB2 não pode assegurar a exclusividade. Esta opção destina-se à propagação de dados, e para descarregar e recarregar uma tabela.

Vejamos agora o exemplo abaixo:

```
CREATE TABLE subscriber(  
    subscriberID INTEGER GENERATED ALWAYS AS  
        IDENTITY (START WITH 100 INCREMENT BY 100),  
    firstname VARCHAR(50),  
    lastname  VARCHAR(50) )
```

No exemplo, a coluna *subscriberID* é um INTEGER definido como coluna de identificação que sempre é gerada. O valor gerado começa em 100 e será incrementado de 100.

8.3.3 Objetos de sequência

Embora os objetos de sequência sejam independentes das tabelas, eles são mencionados nesta seção porque funcionam de maneira semelhante às colunas de identificação. A diferença é que os objetos de sequência geram números exclusivos por todo o banco de dados, enquanto as colunas de identificação geram números exclusivos dentro de uma tabela. As instruções abaixo fornecem um exemplo:

```
CREATE TABLE t1 (salary int)

CREATE SEQUENCE myseq
  START WITH 10
  INCREMENT BY 1
  NO CYCLE

INSERT INTO t1 VALUES (nextval for myseq)

INSERT INTO t1 VALUES (nextval for myseq)

INSERT INTO t1 VALUES (nextval for myseq)

SELECT * FROM t1

SALARY
-----
      10
      11
      12
  3 record(s) selected.

SELECT prevval for myseq FROM sysibm.sysdummy1

1
-----
      12
  1 record(s) selected
```

Listagem 8.6 – Um exemplo de sequências

PREVVAL fornece o valor atual da sequência, enquanto NEXTVAL fornece o valor seguinte. O exemplo acima também usa SYSIBM.SYSDUMMY1. Esta é uma tabela de catálogo de sistema contendo uma coluna e uma linha. Ela pode ser usada em situações onde a consulta requer a saída de um único valor. As tabelas de catálogo de sistema são descritas na próxima seção.

8.3.4 Tabelas de catálogo do sistema

Cada banco de dados possui suas próprias tabelas de catálogo do sistema e visualizações. Estas armazenam *metadados* sobre os objetos do banco de dados. Se

estas tabelas de sistema forem corrompidas, seu banco de dados ficará inutilizado. Você pode consultar estas tabelas como se fossem qualquer tabela comum de banco de dados. São usados três esquemas para identificar as tabelas de catálogo do sistema:

- SYSIBM: tabelas básicas, otimizadas para uso pelo DB2
- SYSCAT: visualizações baseadas nas tabelas SYSIBM, otimizadas para facilitar o uso
- SYSSTAT: estatísticas do banco de dados

A seguir, alguns exemplos de visualização de catálogo:

- SYSCAT.TABLES
- SYSCAT.INDEXES
- SYSCAT.COLUMNS
- SYSCAT.FUNCTIONS
- SYSCAT.PROCEDURES

8.3.5 Tabelas temporárias globais declaradas (DGTTs)

As tabelas temporárias declaradas são aquelas criadas na memória e usadas por um aplicativo, e que são descartadas automaticamente quando o aplicativo é encerrado. Essas tabelas podem ser acessadas somente pelo aplicativo que as criou e não existem em qualquer tabela de catálogo do DB2. O acesso a essas tabelas fornece um desempenho muito eficiente, pois não há contenção de catálogo, bloqueio de linhas, registro de log padrão (o log é opcional) ou verificação de autoridade. Também há suporte a índice para essas tabelas temporárias, ou seja, pode-se criar qualquer índice padrão em uma tabela temporária. Você também pode executar o RUNSTATS nessas tabelas.

As tabelas temporárias declaradas residem dentro de um tablespace temporário do usuário, que deve ser definido antes da criação de qualquer tabela temporária declarada. As instruções na *Listagem 8.7* fornecem um exemplo sobre como criar três tabelas temporárias declaradas:

```
CREATE USER TEMPORARY TABLESPACE apptemps
    MANAGED BY SYSTEM USING ('apptemps');
```

```
DECLARE GLOBAL TEMPORARY TABLE employees
    LIKE employee NOT LOGGED;
```

```
DECLARE GLOBAL TEMPORARY TABLE tempdept
    (deptid CHAR(6), deptname CHAR(20))
    ON COMMIT DELETE ROWS NOT LOGGED;
```

```
DECLARE GLOBAL TEMPORARY TABLE tempprojects
```

```
AS ( fullselect ) DEFINITION ONLY
ON COMMIT PRESERVE ROWS NOT LOGGED
WITH REPLACE IN TABLESPACE apptemps;
```

Listagem 8.7 – Como trabalhar com DGTTs

Ao criar uma tabela temporária declarada, seu esquema é SESSION; este esquema deve ser especificado ao fazer referência à DGTT. O ID de usuário usado para criar uma tabela temporária terá todos os privilégios sobre a tabela. Cada aplicativo que cria uma tabela temporária terá sua própria cópia independente como mostrado na *Figura 8.3*.



Figura 8.3 – Escopo das tabelas temporárias globais declaradas

A *Listagem 8.8* ilustra as limitações de escopo das tabelas temporárias globais declaradas. Ela supõe que os tablespaces temporários do usuário já foram criados.

Na Command Window #1 do DB2:

```
db2 connect to sample
db2 declare global temporary table mydgtt (col1 int, col2 varchar(10)) on
commit preserve rows
db2 insert into session.mydgtt values (1,'hello1'),(2,'hello2'),
(3,'hello3')
db2 select * from session.mydgtt
```

```
COL1          COL2
-----
1 hello1
2 hello2
3 hello3
3 record(s) selected.
```

Na Command Window #2 do DB2:

```
db2 connect to sample
db2 select * from session.mydgtt
SQL0204N "SESSION.MYDGTT" is an undefined name.  SQLSTATE=42704
```

Listagem 8.8 – Como trabalhar com DGTTs - escopo

Como você pode ver na *Listagem 8.8*, ao tentar usar *SESSION.MYDGTT* na segunda sessão (DB2 Command Window #2), você recebe uma mensagem de erro porque a DGTT não foi definida nesta sessão. Observe que a definição da DGTT utiliza a cláusula ON COMMIT PRESERVE ROWS, porque trabalhar na Janela de comando do DB2 fará a alocação como padrão a cada instrução inserida.



8.3.6 Criar tabelas temporárias globais (CGTTs)

Embora as DGTTs permitam a criação de uma tabela temporária, a definição da tabela não pode ser compartilhada entre conexões ou sessões diferentes. Sempre que uma sessão é estabelecida, é necessário executar a instrução DECLARE GLOBAL TEMPORARY TABLE. Por outro lado, com CGTTs (Criar tabelas temporárias globais), é necessário criar a definição da tabela temporária somente uma vez, pois esta é armazenada permanentemente no catálogo do DB2. Isto significa que outras conexões podem simplesmente usar a tabela sem ser necessário criá-la novamente. Embora a estrutura de tabela possa ser usada imediatamente, os dados de cada conexão são independentes uns dos outros, e desaparecerão após o encerramento da conexão. Como exemplo, vejamos a *Listagem 8.9*. Ela supõe que o tablespace temporário do usuário já foi criado.

Na Command Window #1 do DB2:

```
db2 connect to sample
db2 create global temporary table mycgtt (col1 int, col2 varchar(10)) on
commit preserve rows
db2 insert into mycgtt values (1,'hello1'),(2,'hello2'), (3,'hello3')
db2 select * from mycgtt
```

```
COL1          COL2
-----
          1 hello1
          2 hello2
          3 hello3
3 record(s) selected.
```

Na Command Window #2 do DB2:

```
db2 connect to sample
db2 select * from mycgtt
```

```
COL1          COL2
-----
0 record(s) selected.
```

Listagem 8.9 – Como trabalhar com CGTTs - escopo

Na *Listagem 8.9*, vimos que dentro de DB2 Command Window #2 (outra sessão ou conexão), não é necessário criar CGTT novamente, basta fazer referência; entretanto, não é retornada nenhuma linha porque os dados pertenciam à primeira sessão.

8.4 Visualizações

Uma visualização é uma representação dos dados nas tabelas. Os dados da visualização não são armazenados separadamente, mas são obtidos quando a visualização é chamada. São suportadas visualizações aninhadas, ou seja, visualizações criadas com base em outras visualizações. Todas as informações sobre a visualização são mantidas nas seguintes visualizações de catálogo do DB2: SYSCAT.VIEWS, SYSCAT.VIEWDEP e SYSCAT.TABLES. A *Listagem 8.10* fornece um exemplo de como criar e usar uma visualização.

```
CONNECT TO MYDB1;

CREATE VIEW MYVIEW1
  AS SELECT ARTNO, NAME, CLASSIFICATION
  FROM ARTISTS;

SELECT * FROM MYVIEW1;
```

Output:

```
ARTNO      NAME                CLASSIFICATION
-----
10         HUMAN                A
20         MY PLANT              C
30         THE STORE             E
...
```

Listagem 8.10 – Como trabalhar com as visualizações

8.5 Índices

Um índice é um conjunto ordenado de chaves que apontam, cada uma, para uma linha de uma tabela. Um índice permite a exclusividade, e isso também aumenta o desempenho. Algumas das características que podem ser definidas nos índices:

- A ordem do índice pode ser crescente ou decrescente
- As chaves de índice podem ser exclusivas ou não
- Diversas colunas podem ser usadas no índice (isto é chamado de índice composto)

- Se o índice e os dados físicos forem agrupados em sequências de indexação semelhantes, eles são um índice de cluster

Por exemplo:

```
CREATE UNIQUE INDEX artno_ix ON artists (artno)
```

8.5.1 Design Advisor

O Design Advisor é uma ferramenta excelente para aconselhá-lo sobre o projeto ideal para seu banco de dados, dada uma carga de trabalho de SQL específica. O design advisor pode ajudá-lo a projetar seus índices, MQTs (Tabelas de consulta materializada), clustering multidimensional (MDC) e o recurso de particionamento do banco de dados. O Design Advisor é chamado através do Control Center; clique com o botão direito em um banco de dados e selecione *Design Advisor* como mostrado na *Figura 8.4*.

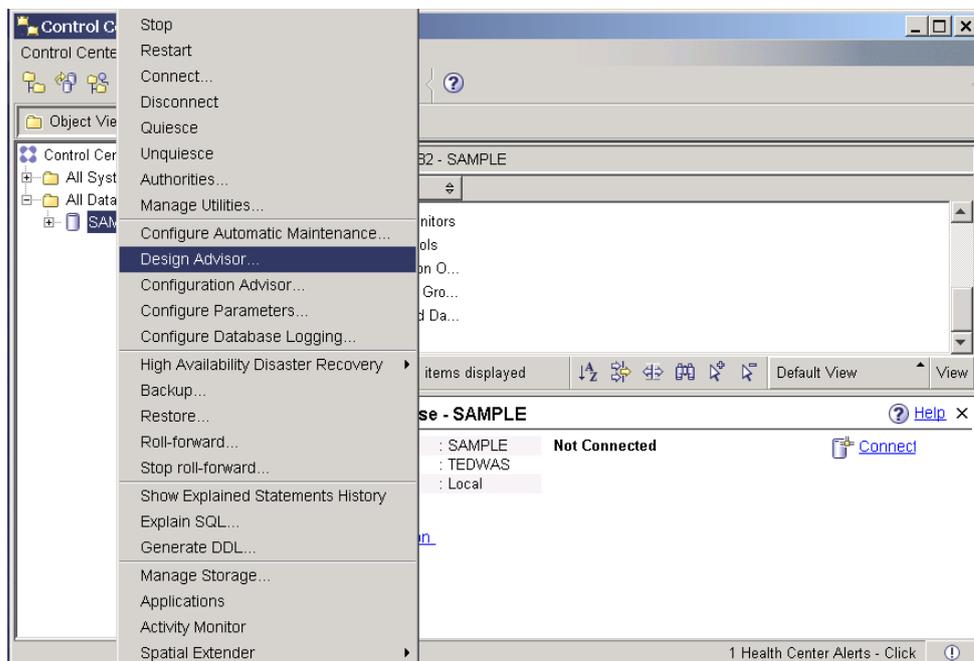


Figura 8.4 – Como chamar o Design Advisor no Control Center

A *Figura 8.5* mostra o Design Advisor. Siga as etapas neste assistente para obter as recomendações de projeto do DB2.

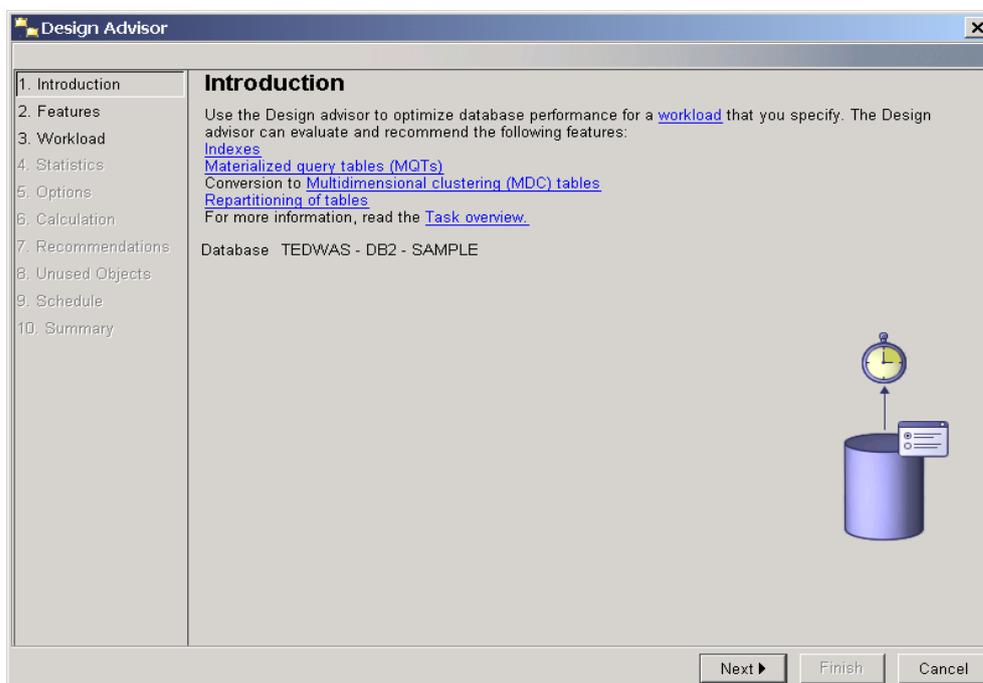


Figura 8.5 – O Design Advisor

8.6 Integridade das referências

A integridade das referências permite que seu banco de dados gerencie os relacionamentos entre as tabelas. Você pode estabelecer relacionamentos do tipo pai-filho entre as tabelas, como mostrado na *Figura 8.6*. Na figura há duas tabelas, DEPARTMENT e EMPLOYEE, relacionadas pelo número do departamento. A coluna WORKDEPT da tabela EMPLOYEE pode conter somente números de departamentos que já existam na tabela DEPARTMENT. Isto ocorre porque neste exemplo, a tabela DEPARTMENT é a tabela-pai e EMPLOYEE é a tabela-filho ou dependente. A figura também mostra a instrução CREATE TABLE para a tabela EMPLOYEE, necessária para estabelecer o relacionamento.

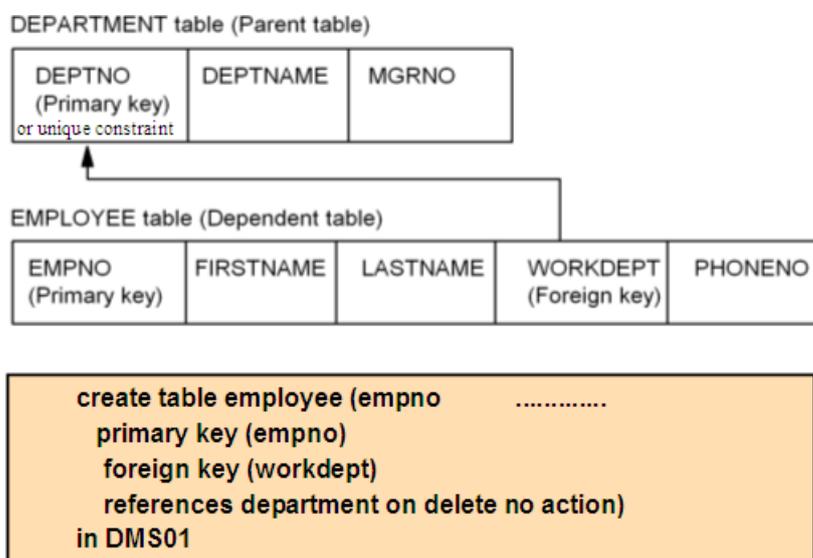


Figura 8.6 – Um exemplo de integridade das referências entre tabelas

Na integridade das referências, frequentemente são usados os conceitos listados abaixo, na Tabela 8.1.

Conceito	Descrição
Tabela-pai	Uma tabela de dados controladora, na qual existe a chave-pai
Tabela dependente	Uma tabela que depende dos dados da tabela-pai. Ela também contém uma chave estrangeira. Para que uma linha exista em uma tabela dependente, já deve existir uma linha correspondente dentro de uma tabela-pai.
Chave primária	Define a chave-pai da tabela-pai. Não pode conter valores nulos (NULL) e os valores devem ser exclusivos. Uma chave primária consiste em uma ou mais colunas dentro de uma tabela.
Chave estrangeira	Tem como referência a chave primária da tabela-pai

Tabela 8.1 – Principais conceitos de integridade das referências

Os dados das tabelas podem estar relacionados aos dados de uma ou mais tabelas com integridade das referências. Também é possível impor restrições sobre os valores dos dados, para que se adequem a uma determinada propriedade ou regra de negócios. Por exemplo, se uma coluna de uma tabela armazena o sexo de uma pessoa, a restrição pode impor que os valores permitidos sejam somente “M” para masculino e “F” para feminino.

new in
V9.7

8.7 Evolução do esquema

Quando mudam as necessidades de negócios, a infraestrutura de tecnologia de informação (TI) e os sistemas de apoio também devem mudar. No mundo dos bancos de dados, isto significa que é necessário criar novas tabelas, as tabelas existentes devem ser descartadas ou modificadas, a lógica do disparador deve ser alterada e assim por diante. Embora essas alterações pareçam simples, na realidade podem ser difíceis e complexas.

São necessárias longas janelas de manutenção e deve-se executar procedimentos intrincados e arriscados. Um dos motivos dessas mudanças costumarem ser difíceis antes é que o DB2 exigia que todos os objetos fossem consistentes o tempo todo; assim, as ações que afetassem objetos que dependiam do objeto sendo alterado eram não permitidas ou fariam os objetos dependentes serem descartados. A *Figura 8.7* fornece um cenário de exemplo.

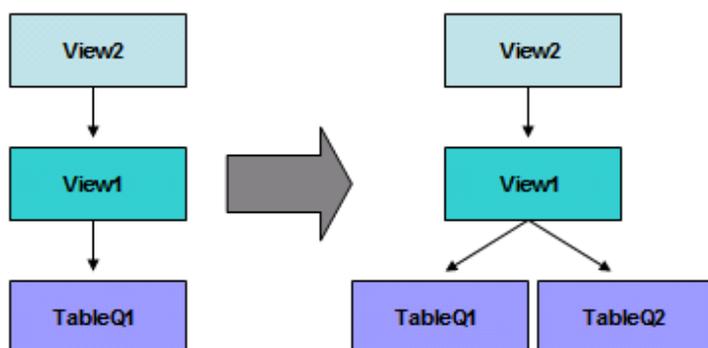


Figura 8.7 – Cenário de exemplo de evolução de esquema

Como exemplo, na *Figura 8.7*, a definição de *view1* precisa ser alterada de modo que não seja apenas baseada na *TableQ1* com as informações financeiras do primeiro trimestre da empresa, mas na *TableQ1* e na *TableQ2* com as informações do primeiro e do segundo trimestres, respectivamente. Normalmente seria necessário descartar a *view1* e recriá-la com a nova definição. Entretanto, *view2* depende de *view1*. Antes do DB2 9.7, o DB2 não permitiria descartar *view1* devido à *view2* dependente. Seria necessário descartar *view2* primeiro, para só então descartar *view1* e, em seguida, reconstruir ambas as visualizações. Com o DB2 9.7, as regras foram relaxadas. Agora, as alterações que afetam os objetos dependentes são permitidas. O objeto dependente (*view2* no exemplo), deve ser revalidado antes de ser usado, mas isso é feito automaticamente. Isto é conhecido por **revalidação automática**. O parâmetro `AUTO_REVAL` do `db cfg` é usado para ativar ou desativar a revalidação automática, e determinar quando a revalidação irá ocorrer. Por exemplo, se você definir este parâmetro como `DEFERRED_FORCE`, a revalidação será adiada até o objeto inválido ou os objetos dependentes serem acessados, mas serão permitidas instruções `CREATE` (com um aviso) em um objeto dependente que ainda não exista.

Outras alterações que afetam o modelo de dependência incluem a implementação de recursos como a sintaxe `CREATE OR REPLACE` para visualizações, funções, procedimentos, disparadores, aliases e assim por diante. Por exemplo:

```
create or replace procedure p1 begin ... end
```

Com esta sintaxe, se um objeto (por exemplo, o procedimento P1) não existisse, ele seria criado. Se ele existisse anteriormente, seria substituído. Este segundo comportamento é o que é importante para a dependência dos objetos. Quando P1 é substituído, os objetos dependentes de P1 são revalidados automaticamente. Uma situação semelhante ocorre

para os recursos novos, como `RENAME COLUMN`, e para alterações de tipos de dados com `ALTER COLUMN`, que também foi aprimorado para suportar mais alterações de tipos de dados.

Um conceito relacionado conhecido como *invalidação suave* permite que os usuários descartem um objeto mesmo que esteja sendo usado por outras transações sendo executadas. Qualquer transação nova terá acesso negado ao objeto descartado.

8.8 Resumo

Neste capítulo, examinamos os objetos de banco de dados no DB2: o que são, como são criados e como usá-los. Introduzimos os esquemas de banco de dados e os comparamos aos novos sinônimos públicos, além de como os sinônimos públicos se comparam com os privados.

Em seguida, discutimos as tabelas e seus elementos em detalhes: tipos de dados (internos e definidos pelo usuário), colunas de identificação, objetos de sequência e tabelas temporárias globais. A estes, seguiu-se um exame das visualizações, índices e do uso da GUI do Design Advisor para aumentar a capacidade de acesso e recuperação dos dados dentro de uma tabela.

Por fim, examinamos a integridade das referências para definir os relacionamentos entre as tabelas, e o novo conceito de evolução de esquema, que permite que os objetos de dados sejam alterados sem complicações desnecessárias.

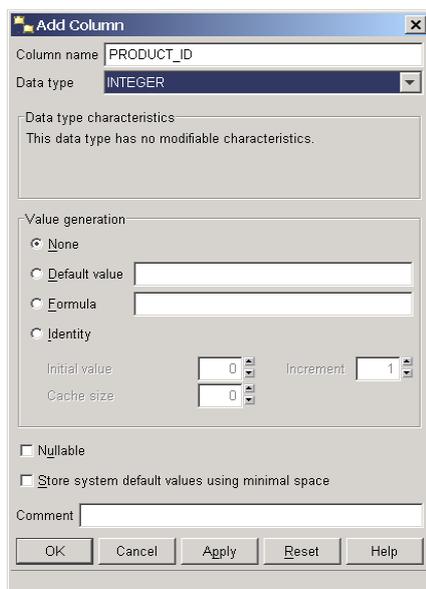
8.9 Exercícios

Até aqui, você usou as tabelas existentes no banco de dados `SAMPLE` para ilustrar os conceitos. Em algum momento, será necessário criar suas próprias tabelas em um banco de dados. Neste exercício, você usará o assistente *Create Table* para criar duas novas tabelas no banco de dados `SAMPLE`.

Procedimento

1. Inicie o assistente *Create Table* como mostrado anteriormente no capítulo. (*Control Center* -> *All Databases* -> *SAMPLE* -> (clique com o botão direito) *Tables object* -> *Create ...*)
2. Defina o nome da tabela, as definições de coluna e quaisquer restrições. A tabela será usada para armazenar informações sobre o material de escritório usado por um projeto no banco de dados `SAMPLE`. A cada vez que for comprado material, será adicionada uma linha a esta tabela. A tabela terá seis colunas:
 - `product_id`: identificador exclusivo do item sendo comprado
 - `description`: descrição do item
 - `quantity`: a quantidade comprada
 - `cost`: o custo do item
 - `image`: uma imagem do item (se disponível)
 - `project_num`: o projeto para o qual este produto foi comprado

3. Na primeira página do assistente, a do nome do esquema, digite o ID de usuário com o qual fez login e use o nome de tabela a seguir: **SUPPLIES**. Você pode digitar um comentário, opcionalmente. Clique no botão *Next* para passar para a próxima página do assistente.
4. Nesta página, você pode adicionar colunas à tabela. Clique no botão *ADD* para adicionar colunas



5. Digite o nome da coluna **product_id** e selecione o tipo de dados: **INTEGER**. Desmarque *Nullable* (anulável) e clique no botão *Apply* (Aplicar) para definir a coluna.
6. Repita esta etapa para as colunas restantes da tabela, usando as opções mostradas na tabela abaixo. Após todas as colunas serem adicionadas (aplicadas), clique no botão *OK* e a lista de colunas que você acabou de criar deve ser resumida. Clique no botão *Next* para passar para a próxima página do assistente.

Nome da coluna	Atributos
product_id (preenchido)	INTEGER, NOT NULL
description	VARCHAR, comprimento 40, NOT NULL
quantity	INTEGER, NOT NULL
cost	DECIMAL, precisão 7, escala 2, NOT NULL
image	BLOB, 1MB, NULLABLE, NOT LOGGED
project_num	CHAR, comprimento 6, NOT NULL

Observação: A opção NOT LOGGED pode ser especificada ao declarar as colunas LOB. Ela é obrigatória para as colunas maiores que 1 GB. Ela também é recomendada para LOBs maiores que 10 MB, pois as alterações em grandes colunas podem preencher o arquivo de log rapidamente. Mesmo que a instrução NOT LOGGED seja executada, as alterações nos arquivos LOB durante uma transação ainda podem ser revertidas. Observe também que a coluna image é a única definida como NULLABLE. Por que você acha que esta coluna foi definida assim?

7. Neste ponto, todas as informações obrigatórias para criar uma tabela foram fornecidas. Ao saltar os outros painéis do assistente, você opta pelos valores padrão dessas opções. Sempre é possível adicionar chaves e restrições após a criação de uma tabela.
8. Adicione uma restrição à tabela para restringir os valores da coluna *quantity*. Na página *Constraint* (Restrição) do assistente, clique no botão *ADD*. No campo *Check Name* (Verificar nome), digite: *valid_quantities*. No campo *Check Condition* (Verificar condição), digite: *quantity > 0*

Clique no botão *OK*. Você deve ver um resumo da restrição que acabou de adicionar na página *Constraint* do assistente. Clique no botão *Next* para passar para a próxima página do assistente.

9. Você pode continuar percorrendo o assistente, alterando os outros parâmetros da tabela. Como alternativa, salte para a página *Summary* (Resumo) ou simplesmente clique no botão *Finish* (Concluir) para criar a tabela.
10. No Control Center, clique na pasta *Tables* no banco de dados **SAMPLE** do painel da árvore de objetos. A tabela que acabou de criar agora deve aparecer na lista. Pode ser necessário atualizar a visualização do Control Center, para poder ver as alterações.
11. Agora vamos testar a conversão implícita, usando a tabela **STAFF** no banco de dados **SAMPLE**. Tente o seguinte:

```
C:\>db2 describe table staff
```

Observe que a coluna ID é definida como SMALLINT

```
C:\>db2 select * from staff where id = '350' --> Observe que '350' é uma cadeia de texto
```

```
C:\>db2 select * from staff where id = 350 --> Observe que 350 é um número
```

Em ambos os casos, a saída seria:

ID	NAME	DEPT	JOB	YEARS	SALARY	COMM
350	Gafney	84	Clerk	5	43030.50	188.00

Na primeira instrução SELECT usando '350' como uma cadeia de texto, o DB2 executa a conversão implícita para um número (SMALLINT).

9

Capítulo 9 – Utilitários de movimentação de dados

As ferramentas ou comandos descritos nesta seção são usados para mover os dados dentro do mesmo banco de dados, ou entre bancos de dados da mesma ou de diferentes plataformas. A *Figura 9.1* fornece uma visão geral dos utilitários de movimentação de dados.

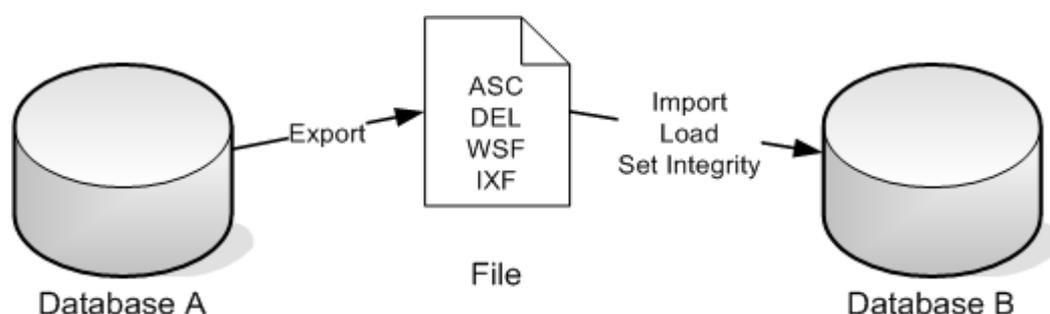


Figura 9.1 – Utilitários de movimentação de dados

Na *Figura 9.1* há dois bancos de dados, A e B. Usando o utilitário EXPORT, você pode exportar os dados de uma tabela para um arquivo. O arquivo pode ser de um dos formatos abaixo:

- ASC = ASCII
- DEL = ASCII delimitado
- WSF = formato de planilha
- IXF = Integrated Exchange Format

Os arquivos ASC e DEL são arquivos de texto, e podem ser abertos e lidos em qualquer editor de texto. WSF é um formato que pode ser usado para movimentar dados para planilhas como o Excel ou o Lotus® 1-2-3. IXF é um formato que não apenas inclui os dados, mas também a DDL (Linguagem de definição de dados) da tabela em questão. O formato IXF é conveniente porque, quando é necessário reconstruir a tabela, isso pode ser feito diretamente a partir de um arquivo com exportação formatada em IXF; isto não é possível caso você use os outros formatos.

Após exportar os dados para um arquivo, o utilitário IMPORT pode ser usado para que os dados do arquivo sejam importados por outra tabela. Para os formatos ASC, DEL e WSF, a tabela deve existir previamente, o que não é necessário para o formato IXF. Outro método para carregar os dados em uma tabela é usar o utilitário LOAD. O utilitário LOAD é mais rápido, pois vai diretamente até as páginas do banco de dados sem interagir com o mecanismo do DB2; entretanto, este método não verifica as restrições, e os disparadores não são acionados. Para assegurar a consistência dos dados carregados usando LOAD, frequentemente o comando SET INTEGRITY é usado logo em seguida.

As seções abaixo descrevem os utilitários EXPORT, IMPORT e LOAD em mais detalhes.

Observação:

Para obter mais informações sobre como trabalhar com utilitários de movimentação de dados, assista ao vídeo:

<http://www.channeldb2.com/video/video/show?id=807741:Video:4262>

9.1 Utilitário EXPORT

Como já discutimos anteriormente, o utilitário EXPORT é usado para extrair dados de uma tabela para um arquivo. Nos bastidores, o que está realmente sendo executado é uma operação SELECT de SQL. O exemplo abaixo exporta, para o arquivo `employee.ixf` de formato IXF, 10 linhas da tabela `employee`.

```
EXPORT TO employee.ixf OF IXF
  SELECT * FROM employee
  FETCH FIRST 10 ROWS ONLY
```

Sugerimos que você tente fazer o exemplo acima. A tabela de funcionários faz parte do banco de dados `SAMPLE`; portanto, primeiro você precisa se conectar a este banco de dados criado em um capítulo anterior.

Se você prefere trabalhar com ferramentas de GUI, o utilitário EXPORT também pode ser chamado do Control Center, como mostrado na *Figura 9.2*.

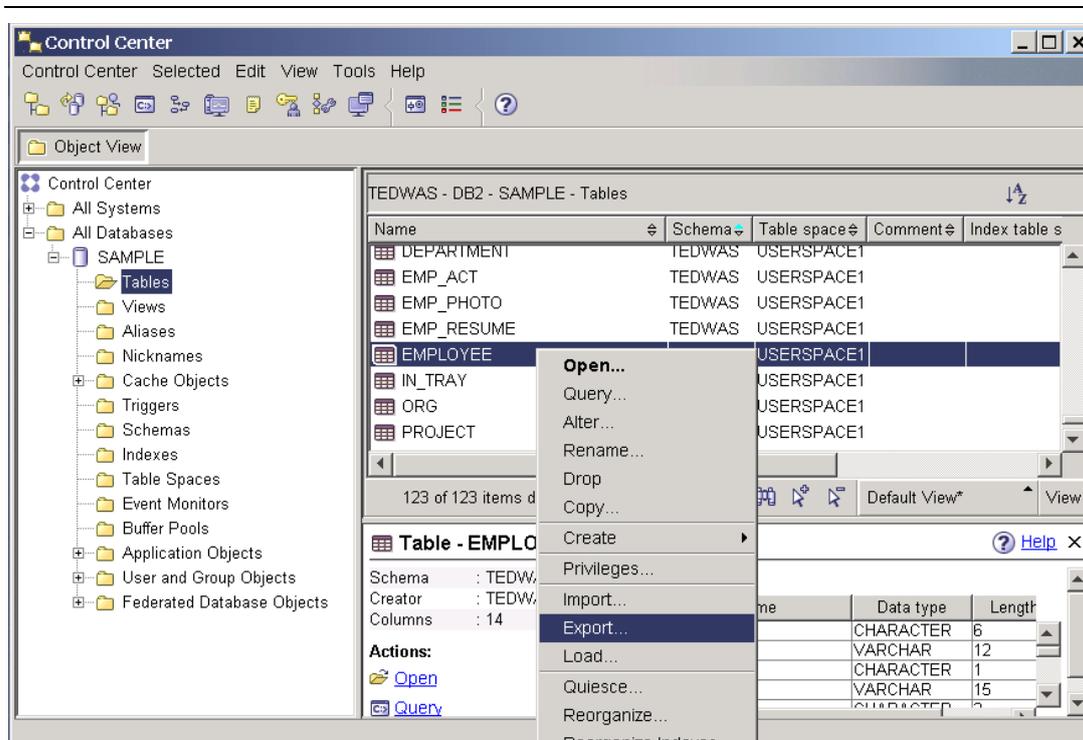


Figura 9.2 – Inicialização da caixa de diálogo EXPORT

Como mostrado na figura, primeiro é necessário clicar uma vez na tabela de funcionários para selecioná-la e, em seguida, clicar com o botão direito do mouse na tabela para obter um menu suspenso, no qual você pode selecionar a opção Export (Exportar). Após escolher esta opção, é aberto um assistente. Basta seguir as etapas fornecidas pelo assistente para concluir a operação.

9.2 Utilitário IMPORT

Como já discutimos anteriormente, o utilitário IMPORT é usado para carregar dados de um arquivo para uma tabela. Nos bastidores, o que está realmente sendo executado é uma operação INSERT de SQL. Como uma operação de INSERT está sendo executada, todos os disparadores são acionados, todas as restrições são cumpridas imediatamente e o bufferpool do banco de dados é usado. O exemplo abaixo carrega todos os dados do arquivo com formatação IXF `employee.ixf` para a tabela `employee_copy`. Sugerimos que você tente fazer o exemplo, mas é necessário primeiro que o utilitário EXPORT tenha sido executado na seção anterior.

```
IMPORT FROM employee.ixf OF IXF
  REPLACE_CREATE
  INTO employee_copy
```

A opção `REPLACE_CREATE` é uma das diversas opções disponíveis com o utilitário IMPORT. Esta opção substitui o conteúdo da tabela `employee_copy` se esta já existia

anteriormente à execução do utilitário IMPORT, ou cria a tabela e carrega os dados, caso a tabela ainda não exista.

Se você prefere trabalhar a partir do Control Center, inicie o utilitário IMPORT selecionando qualquer tabela, clique com o botão direito na mesma e selecione a opção Import exibida na *Figura 9.3*

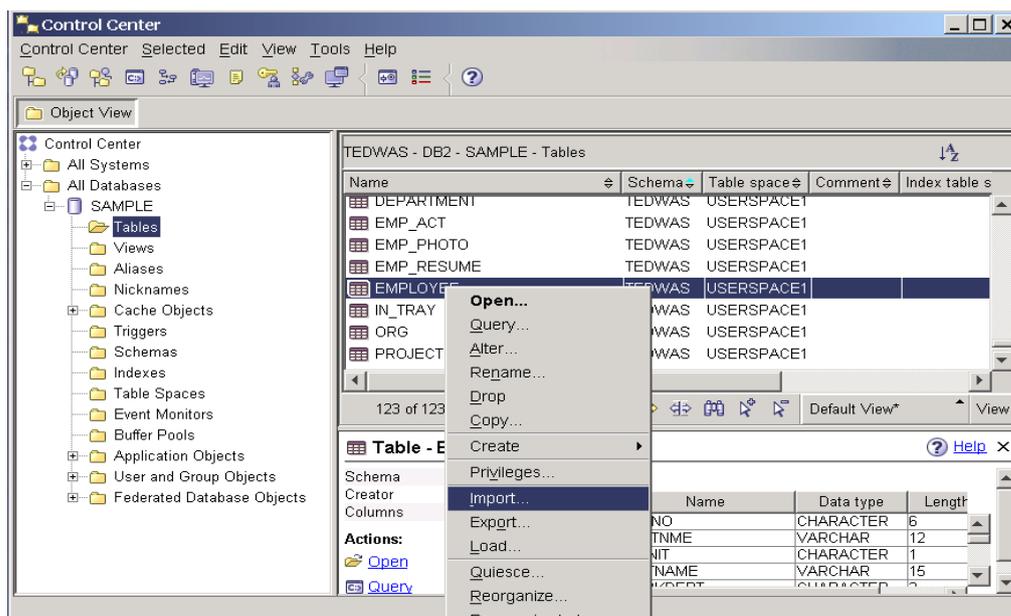


Figura 9.3 – Inicialização da caixa de diálogo IMPORT

9.3 Utilitário LOAD

O utilitário LOAD é uma maneira mais rápida de carregar dados de um arquivo para uma tabela. Como discutido antes, o utilitário LOAD não passa pelo mecanismo DB2; portanto, os disparadores não são ativados, o bufferpool não é usado e as restrições podem ser impostas, mas somente como uma etapa separada. Por outro lado, uma operação LOAD é mais rápida que o IMPORT, pois ela executa um carregamento de dados de baixo nível, acessando diretamente as páginas de dados do disco. Ele funciona em três fases: LOAD, BUILD e DELETE.

O exemplo abaixo carrega todos os dados do arquivo com formatação IXF `employee.ixf` para a tabela `employee_copy`. A opção REPLACE é uma das muitas opções disponíveis com LOAD. Neste caso, ela é usada para substituir (REPLACE) todo o conteúdo da tabela `employee_copy`.

```
LOAD FROM employee.ixf OF IXF
  REPLACE INTO employee_copy
```

Após executar o comando acima, o tablespace onde a tabela reside pode ter sido colocado em estado CHECK PENDING. Isto significa que é necessário executar o

comando SET INTEGRITY para verificar a consistência dos dados. O exemplo a seguir mostra como fazer isso:

```
SET INTEGRITY FOR employee_copy
    ALL IMMEDIATE UNCHECKED
```

Se você prefere trabalhar no Control Center, você pode iniciar os utilitários LOAD e SET INTEGRITY como mostrado na *Figura 9.4* e *9.5*, respectivamente.

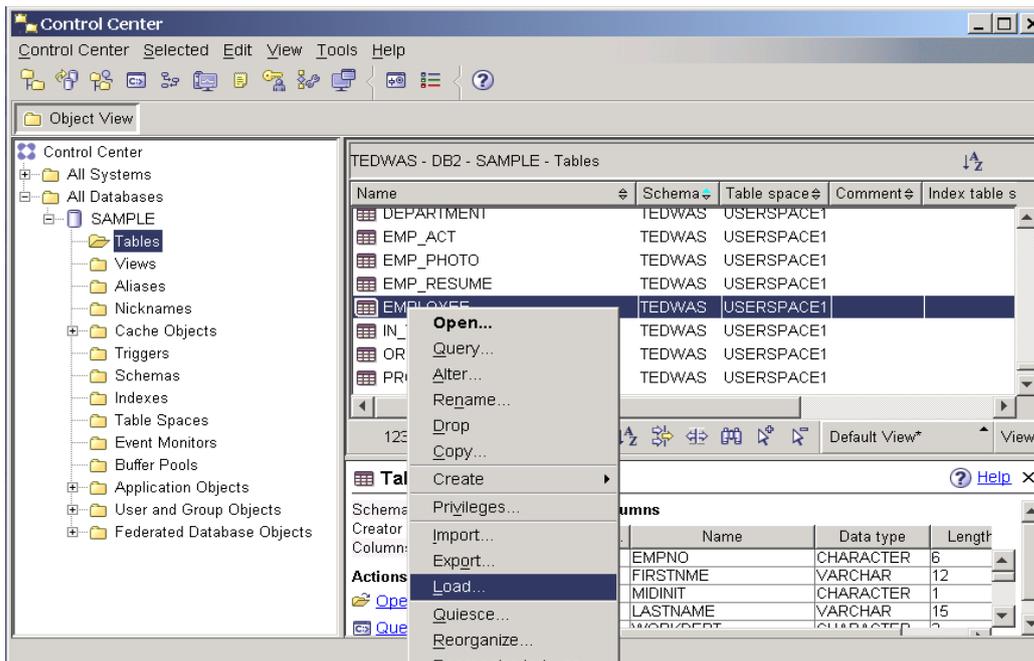


Figura 9.4 – Inicialização do utilitário LOAD

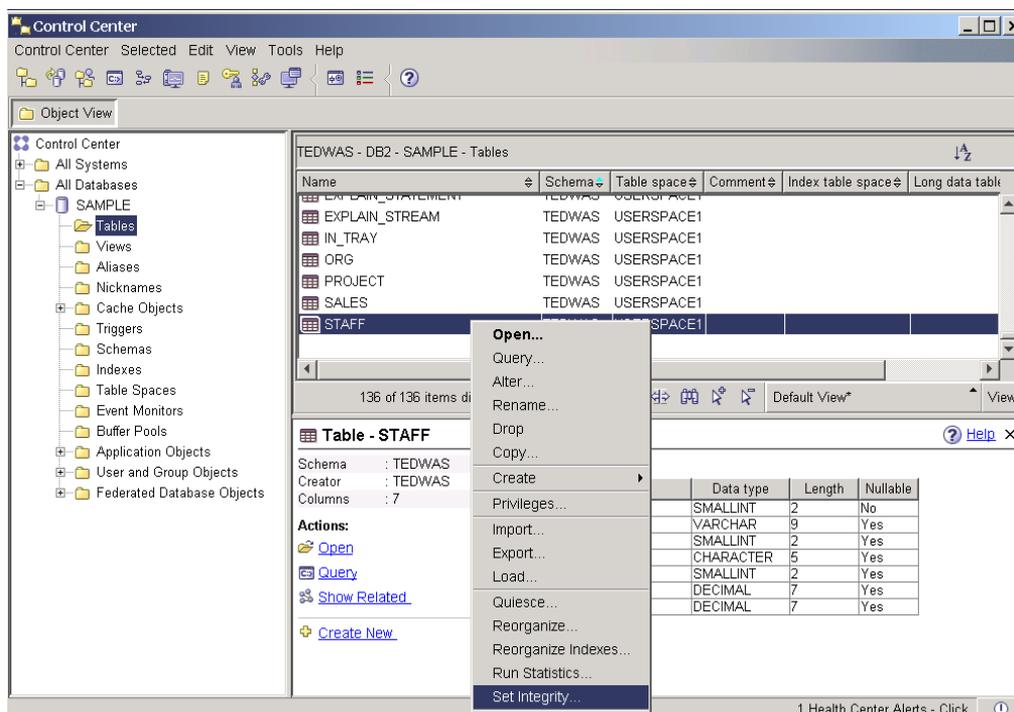


Figura 9.5 – Inicialização do assistente SET INTEGRITY

9.4 O utilitário db2move

Os utilitários EXPORT, IMPORT e LOAD trabalham em uma tabela de cada vez. Embora você possa criar um script para gerar os comandos acima para cada tabela de um banco de dados, outro utilitário, chamado **db2move**, pode fazê-lo para você. O utilitário **db2move** pode trabalhar somente com arquivos IXF, e os nomes de arquivo serão gerados automaticamente pelo **db2move**. Os exemplos abaixo mostram como executar o **db2move** com as opções export e import, respectivamente, usando o banco de dados **SAMPLE**.

```
db2move sample export
db2move sample import
```

O Control Center não possui uma opção para **db2move**.

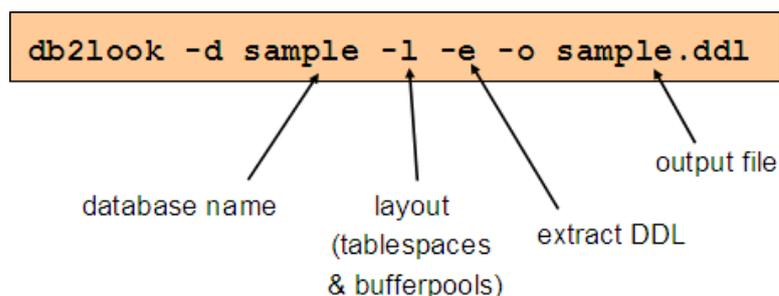
9.5 O utilitário db2look

Enquanto os utilitários EXPORT, IMPORT, LOAD e **db2move** permitem mover dados de uma tabela para outra, seja dentro de um banco de dados ou entre diversos bancos de dados, o utilitário **db2look** pode ser usado para extrair as expressões DDL, estatísticas de banco de dados e características de tablespace de um banco de dados, e armazená-las em um arquivo de script que, mais tarde, poderá ser executado em outro sistema. Por exemplo, se você deseja clonar um banco de dados de um servidor DB2 sendo executado em Linux para um servidor DB2 sendo executado em Windows, primeiro execute o utilitário **db2look** no servidor DB2 do Linux para obter a estrutura do banco de dados e

armazenar esta estrutura em um arquivo de script. Em seguida, copie este arquivo de script para o servidor DB2 do Windows e execute-o para começar a compilar o banco de dados clonado. Neste ponto, a estrutura do banco de dados foi clonada. A próxima etapa seria executar o utilitário `db2move` com a opção `export` no servidor DB2 do Linux, copiar todos os arquivos gerados para o servidor DB2 do Windows e, em seguida, executar o `db2move` com a opção `import` ou `load`. Depois de fazer isto, seu banco de dados foi completamente clonado de um servidor para outro em uma plataforma diferente.

O cenário acima pode ser necessário ao trabalhar com bancos de dados em diferentes plataformas, como o Linux e o Windows. Se ambos os servidores estão sendo executados na mesma plataforma, provavelmente seriam usados os comandos `backup` e `restore` (restauração), o que torna este processo mais fácil e mais simples. Os comandos `backup` e `restore` são discutidos em mais detalhes em um capítulo posterior deste livro.

O exemplo a seguir extrai os layouts do tablespace e do bufferpool, juntamente com as expressões DDL, do banco de dados `SAMPLE`, e os armazena no arquivo `sample.ddl`. Sugerimos que você execute o comando abaixo e analise o arquivo de texto da saída, `sample.ddl`.



O comando `db2look` tem opções demais para serem descritas neste livro. Entretanto, você pode usar o sinalizador `-h` para obter uma descrição rápida das opções disponíveis:

```
db2look -h
```

O utilitário `db2look` também pode ser chamado do Control Center, como mostrado na *Figura 9.6*

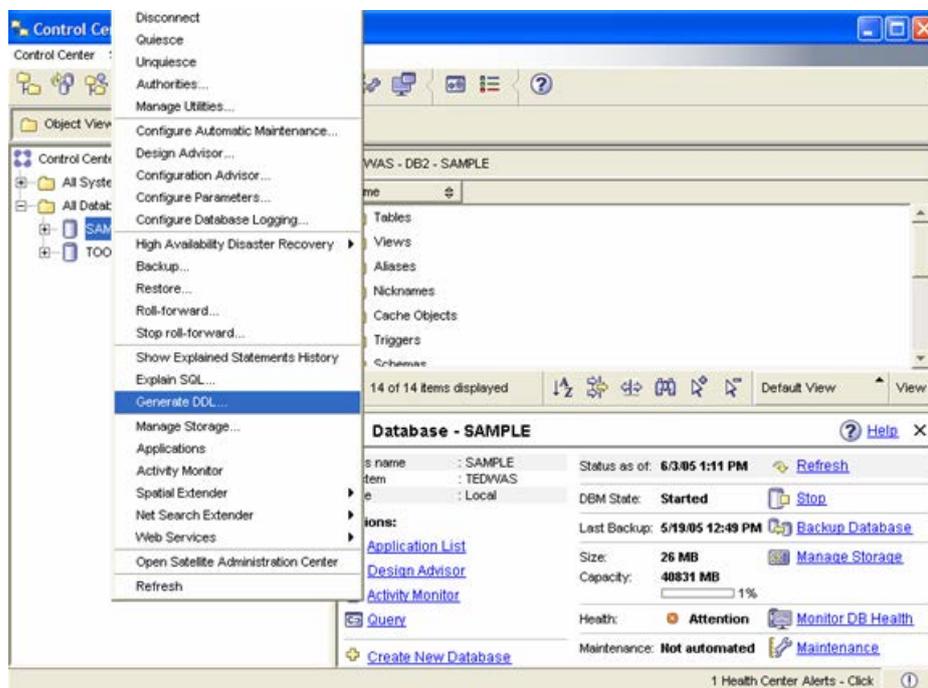


Figura 9.6 – Extração de DDL no Control Center

Na *Figura 9.6*, selecione o banco de dados do qual deseja obter o DDL, clique com o botão direito do mouse e selecione *Generate DDL* (Gerar DDL). A janela *Generate DDL* é aberta exibindo diversas opções de extração, como mostrado na *Figura 9.7*.

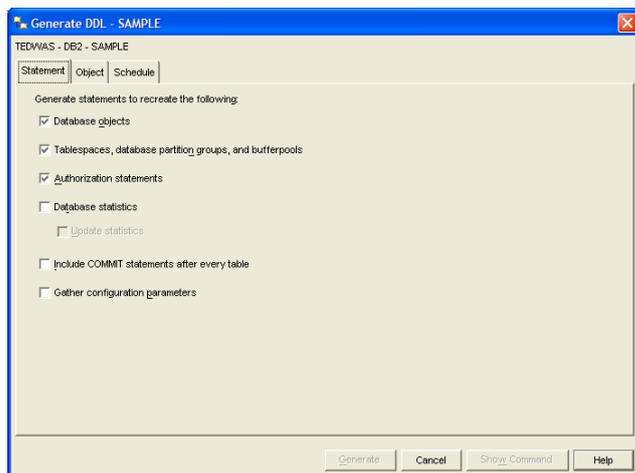


Figura 9.7 – Extração de DDL no Control Center

9.6 Resumo

Neste capítulo, discutimos as diversas funções de exportação e importação do DB2. Começando com um exame dos diversos formatos de exportação (ASC, DEL, WSF e IXF), passamos para uma análise aprofundada do utilitário EXPORT. Os utilitários de importação IMPORT e LOAD foram discutidos em seguida, juntamente com a necessidade da instrução SET INTEGRITY ao usar o LOAD.

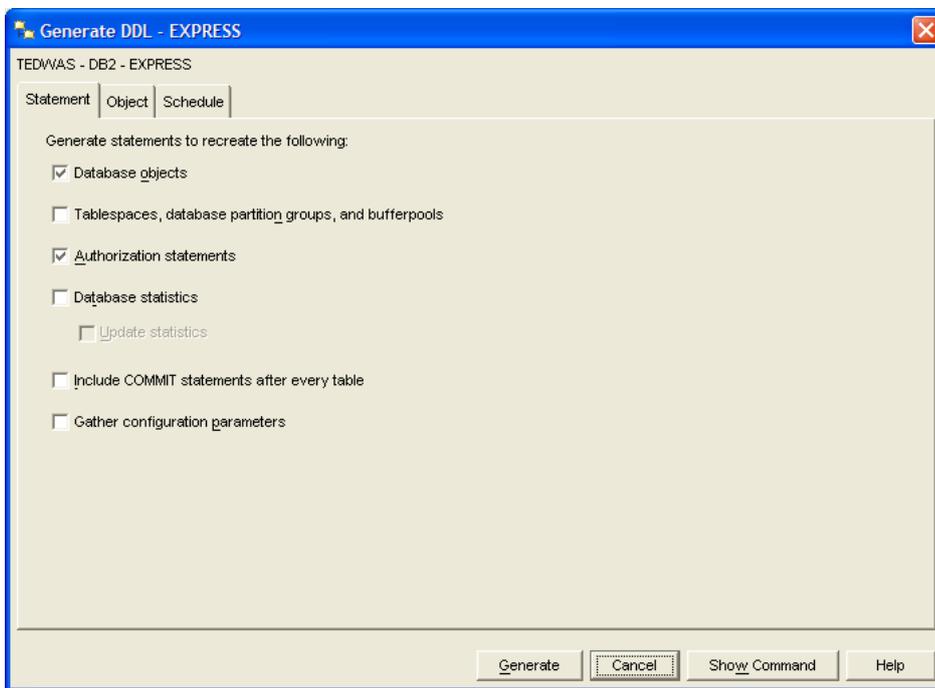
O comando db2move fornece um meio de simplificar o processo de transferência de exportação e importação. Um comando mais complexo, db2look, permite extrair e armazenar todos os elementos de banco de dados necessários para recriar o banco de dados inteiro de maneira independente, se desejado.

9.7 Exercícios

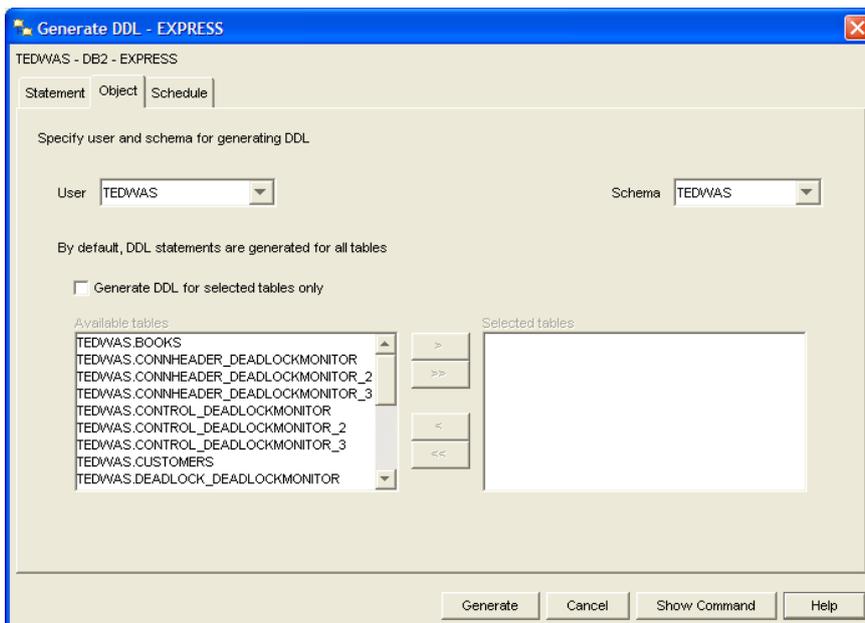
Ao clonar um banco de dados, sua meta deve ser recriar o banco de dados da maneira direta e reproduzível possível. Isto normalmente é feito usando scripts SQL, que podem ser executados imediatamente após o DB2 ser instalado. Neste exercício, você extrairá as definições de objeto do banco de dados **EXPRESS** (criado em um exercício anterior) usando o Control Center.

Procedimento

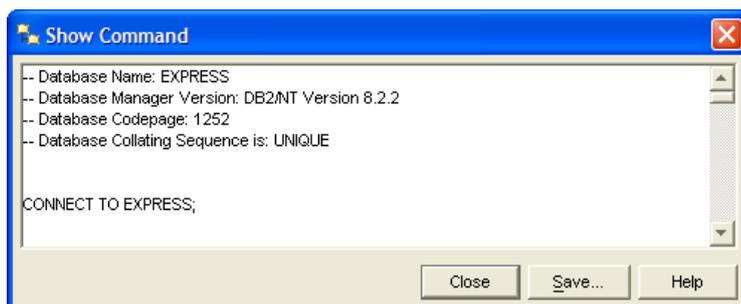
1. Abra o Control Center.
2. Clique com o botão direito do mouse no banco de dados **EXPRESS** na árvore de objetos e selecione o item de menu *Generate DDL* (Gerar DDL). Isto abrirá a janela de diálogo *Generate DDL*.
3. Na janela *Generate DDL*, especifique as opções para o DDL, como mostrado abaixo. Se você tivesse criado objetos adicionais em seu ambiente, como tablespaces, bufferpools, etc., iria selecioná-los aqui. Como você não criou esses tipos de objeto, desmarque a caixa. As estatísticas de banco de dados não foram incluídas porque o ambiente de produção provavelmente contém um conjunto de estatísticas diferente do ambiente de desenvolvimento. Da mesma maneira, os parâmetros de configuração provavelmente serão diferentes também. Em seu próprio ambiente, se tudo for configurado exatamente da maneira que será implementado, você pode incluir essas opções adicionais.



4. Vá para a guia *Object* (Objeto). Você será capaz de escolher especificamente os objetos para os quais deseja gerar o DDL. Neste caso, selecione o usuário e o esquema que estava usando para criar todos os seus objetos e gere o DDL para todos os objetos desse esquema. Clique no botão *Generate* (Gerar) para começar a geração de DDL.



5. Verifique o DDL resultante. O resultado da etapa anterior é um único script, contendo todas as instruções SQL dos objetos escolhidos. Agora, você irá organizar este script em agrupamentos lógicos.
6. Crie um diretório denominado `C:\express` no sistema de arquivos e salve o arquivo DDL, gerado neste novo diretório, em um arquivo denominado `schema.ddl`. (Clique no botão **Save (Salvar)**)



7. Abra o arquivo que acabou de salvar no Command Editor. (Dica: No Command Editor, selecione *File -> Open (Arquivo -> Abrir)*)
8. Embora apenas desejássemos o DDL para as tabelas, observe que o DDL dos outros objetos do banco de dados também foi incluído. Mova todas as instruções **CREATE TRIGGER** para um novo arquivo separado, denominado `triggers.ddl`. Mesmo tendo criado somente um disparador, geralmente a melhor prática é separar os objetos por tipos.
9. Por enquanto, recomendamos remover tudo:
 - Instruções de banco de dados **CONNECT TO**
 - Instruções **DISCONNECT**Você deve ter dois scripts neste momento:
`C:\express\schema.ddl` contendo o DDL para tabelas, visualizações, índices e restrições.
`C:\express\triggers.ddl` contendo o DDL para disparadores
10. Limpe o script para sua implementação:
 - Remova os comentários desnecessários (por ex., `-- CONNECT TO...`)
 - Separe as funções e procedimentos em seus próprios arquivos (isto é útil quando há diversas funções e procedimentos). Você pode desejar agrupá-los por função ou aplicação (por ex. `billing.ddl`, `math.ddl`, `stringfunc.ddl`, etc.)
11. Você também deve ter percebido que um caractere especial está sendo usado para delimitar o final dos disparadores, das funções e dos procedimentos (`@`). Isto é necessário para delimitar o final da instrução **CREATE <object>** em contraste com o final de uma instrução processual dentro do objeto.

10

Capítulo 10 – Segurança de banco de dados

Este capítulo discute como a segurança é tratada no DB2. A *Figura 10.1* fornece uma visão geral básica.

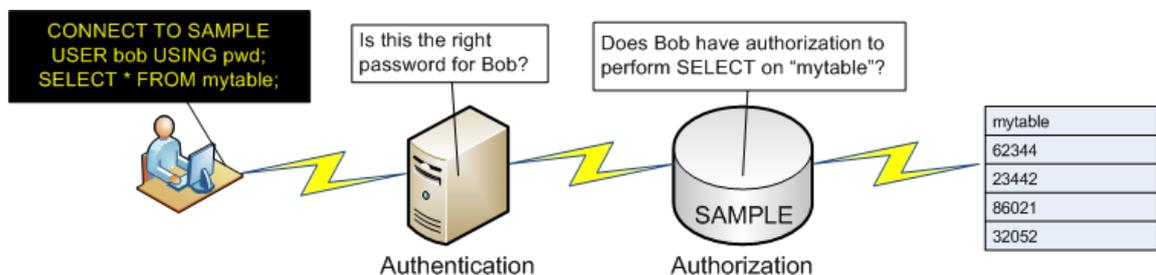


Figura 10.1 – Visão geral da segurança no DB2

Como mostrado na *Figura 10.1*, a segurança do DB2 consiste em duas partes:

- **Autenticação**

É o processo através do qual a identidade do usuário é validada. A autenticação é realizada por um recurso de segurança fora do DB2, através de um plugin de segurança. O plugin de segurança padrão baseia-se na segurança do sistema operacional, mas você também pode usar plugins para o Kerberos, LDAP ou compilar seu próprio plugin de autenticação personalizado. Ao usar o plugin de autenticação padrão baseado em sistema operacional, o ID de usuário e a senha são enviados ao servidor de banco de dados (por exemplo, como parte de uma instrução de conexão). O servidor de banco de dados, por sua vez, chama a autenticação do sistema operacional para validar o ID de usuário e a senha.

- **Autorização**

Neste estágio, o DB2 verifica se o usuário autenticado pode executar a operação solicitada. As informações de autorização são armazenadas em um catálogo do DB2 e em um arquivo de configuração do DBM.

Por exemplo, na *Figura 10.1*, o usuário *bob* conecta-se ao banco de dados **SAMPLE** com esta instrução:

```
CONNECT TO sample USER bob USING pwd
```

bob e *pwd* são passados ao sistema operacional ou recurso de autenticação externa para realizar a aprovação de autenticação, confirmando que um usuário de nome *bob* já foi definido e que a senha fornecida corresponde a esse usuário. Se esta parte for bem-sucedida, o sistema operacional retornará o controle de segurança ao DB2. Em seguida, quando o usuário *bob* executar uma instrução como:

```
SELECT * FROM mytable
```

O DB2 assume o controle da segurança para realizar a verificação de autorização e confirmar se o usuário *bob* tem privilégios para usar `SELECT` na tabela *mytable*. Se a verificação de autorização falhar, o DB2 retornará uma mensagem de erro; caso contrário, a instrução será executada em relação a *mytable*.

Observação:

Para obter mais informações sobre como trabalhar com a segurança do DB2, assista ao vídeo: <http://www.channeldb2.com/video/video/show?id=807741:Video:4267>

10.1 Autenticação

Embora a autenticação em si seja executada pelo sistema operacional através do plugin de segurança padrão (ou outro recurso externo de segurança), o DB2 decide em que nível esta autenticação ocorre.

O parâmetro `AUTHENTICATION` de configuração do banco de dados, definido no servidor do DB2, apresenta vários valores possíveis. Por exemplo, quando o parâmetro é definido como `SERVER` (o padrão), a autenticação é realizada pelo sistema operacional ou recurso externo de segurança no servidor. Entretanto, se `AUTHENTICATION` for definido como `CLIENT`, a autenticação é realizada pelo sistema operacional ou recurso externo de segurança no cliente. Isto é mostrado na *Figura 10.2*.

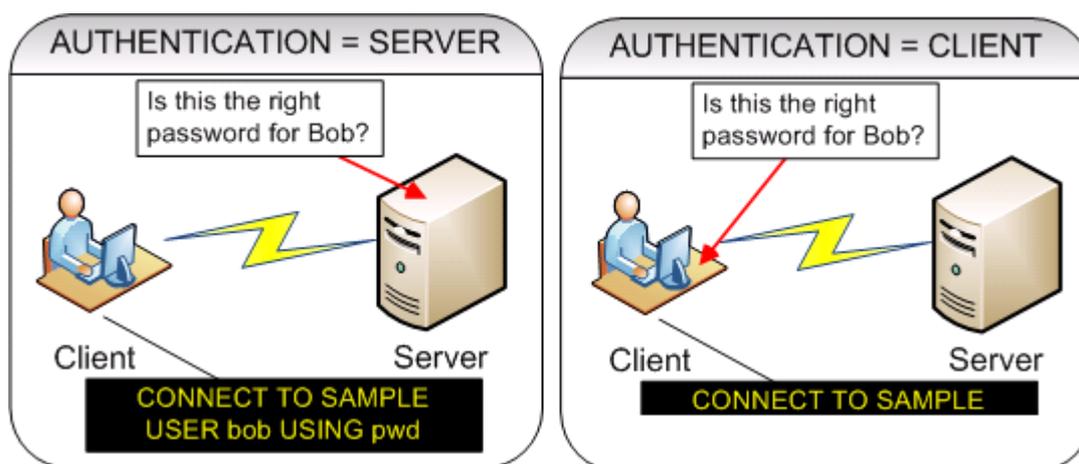


Figura 10.2 – Onde ocorre a autenticação

O parâmetro AUTHENTICATION pode ser definido como qualquer um dos valores listados na *Tabela 10.1*

Comando	Descrição
SERVER (padrão)	Autenticação ocorre no servidor
CLIENT	Autenticação ocorre no cliente
SERVER_ENCRYPT	Mesmo caso de SERVER, exceto pelas IDs de usuário e senhas serem criptografadas
KERBEROS	Autenticação ocorre usando um mecanismo de segurança Kerberos
SQL_AUTHENTICATION_DATAENC	A autenticação no servidor e as conexões devem usar criptografia de dados
SQL_AUTHENTICATION_DATAENC_CMP	Como acima, exceto pela criptografia de dados ser usada somente quando disponível
GSSPLUGIN	A autenticação usa um mecanismo de segurança com plugin externo GSS baseado em API

Tabela 10.1 – Valores válidos do parâmetro AUTHENTICATION

10.2 Autorização

A autorização consiste nos privilégios, autoridades, funções e credenciais de controle de acesso baseado em rótulo (LBAC), que são armazenados nas tabelas de sistema do DB2 e gerenciados pelo DB2.

Um privilégio permite que um usuário execute um único tipo de operação em relação ao banco de dados, como CREATE, UPDATE, DELETE, INSERT, etc.

Uma função permite agrupar diferentes privilégios que você pode conceder a um usuário, grupo ou outras funções.

Uma autoridade é uma função predefinida que consiste em diversos privilégios.

As credenciais de acesso baseado em rótulo (LBAC) incluem políticas e rótulos que suportam acesso granular de determinados usuários a linhas e colunas específicas. O LBAC não é incluído no DB2 Express-C, mas você pode ler mais a respeito no *Capítulo 2*.

10.2.1 Privilégios

A *Figura 10.3* mostra alguns dos diferentes privilégios no DB2.

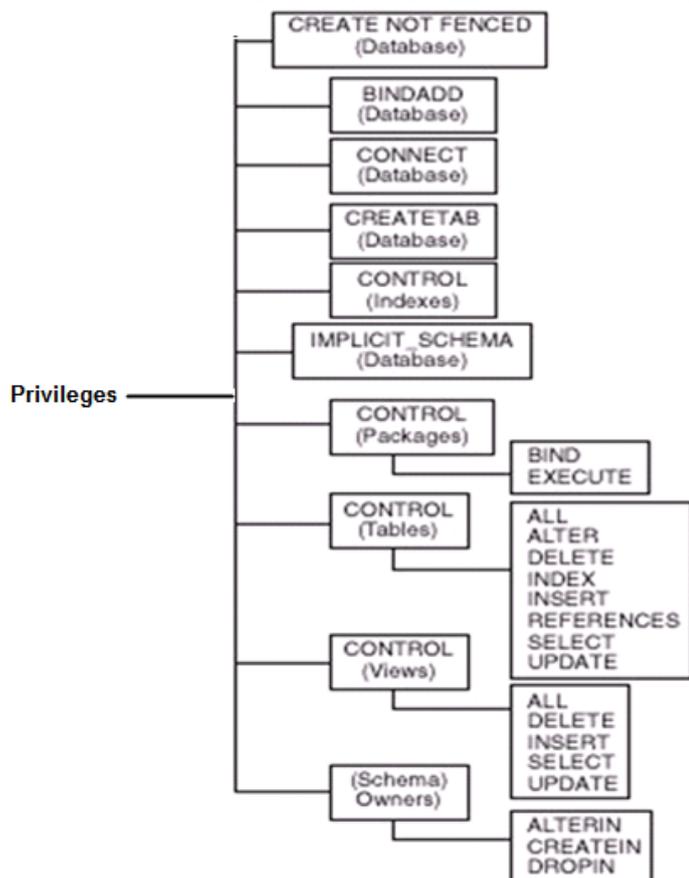


Figura 10.3 – Listagem de alguns privilégios do DB2

Quando um usuário ou grupo recebe privilégios CONTROL (de controle), isto significa que ele também pode conceder o privilégio a outro usuário ou grupo. Consulte o Information Center do DB2 para obter mais detalhes sobre os outros privilégios diferentes.

10.2.2 Autoridades

As autoridades estão classificadas em dois grupos:

- Autoridades em nível de instância: Estas autoridades podem operar no nível de instância. Por exemplo, SYSADM.
- Autoridades em nível de banco de dados: Estas autoridades podem operar no nível de banco de dados. Por exemplo, DBADM.

10.2.2.1 Autoridades em nível de instância

A *Tabela 10.2* lista as autoridades em nível de instância

Autoridade	Descrição
SYSADM	Gerencia a instância como um todo
SYSCTRL	Administra uma instância de gerente de banco de dados
SYSMAINT	Realiza manutenção em bancos de dados dentro de uma instância
SYSMON	Monitora a instância e seus bancos de dados

Tabela 10.2 – Autoridades em nível de instância

Para conceder a autoridade de SYSADM, SYSCTRL, SYSMAINT ou SYSMON a um grupo, pode-se atribuir a um grupo de sistema operacional os parâmetros SYSADM_GROUP, SYSCTRL_GROUP, SYSMAINT_GROUP e SYSMON_GROUP, respectivamente, do DBM CFG.

Por exemplo, para conceder autoridade de SYSADM ao grupo de sistema operacional *myadmns*, você pode executar este comando:

```
update dbm cfg using SYSADM_GROUP myadmns
```

Cada instância do DB2 possui suas próprias definições de grupo de autoridade. No Windows, como padrão, estes parâmetros são vazios, o que significa que o grupo Administrador local terá autoridade de SYSADM. No DB2 9.7, o grupo DB2ADMNS (se a segurança estendida estiver ativada) e a conta LocalSystem também terão autoridade de SYSADM. O ID de autorização para a conta LocalSystem é SYSTEM. No Linux, o grupo proprietário da instância é o grupo SYSADM padrão.

new in
V9.7

A *Figura 10.4*, extraída do Information Center do DB2, mostra as diferentes autoridades em nível de instância e as diferentes funções que podem realizar. Como ilustrado na figura, uma autoridade SYSADM inclui todas as funções SYSCTRL e mais. Uma autoridade SYSCTRL inclui todas as funções SYSMAINT e mais, e assim por diante.

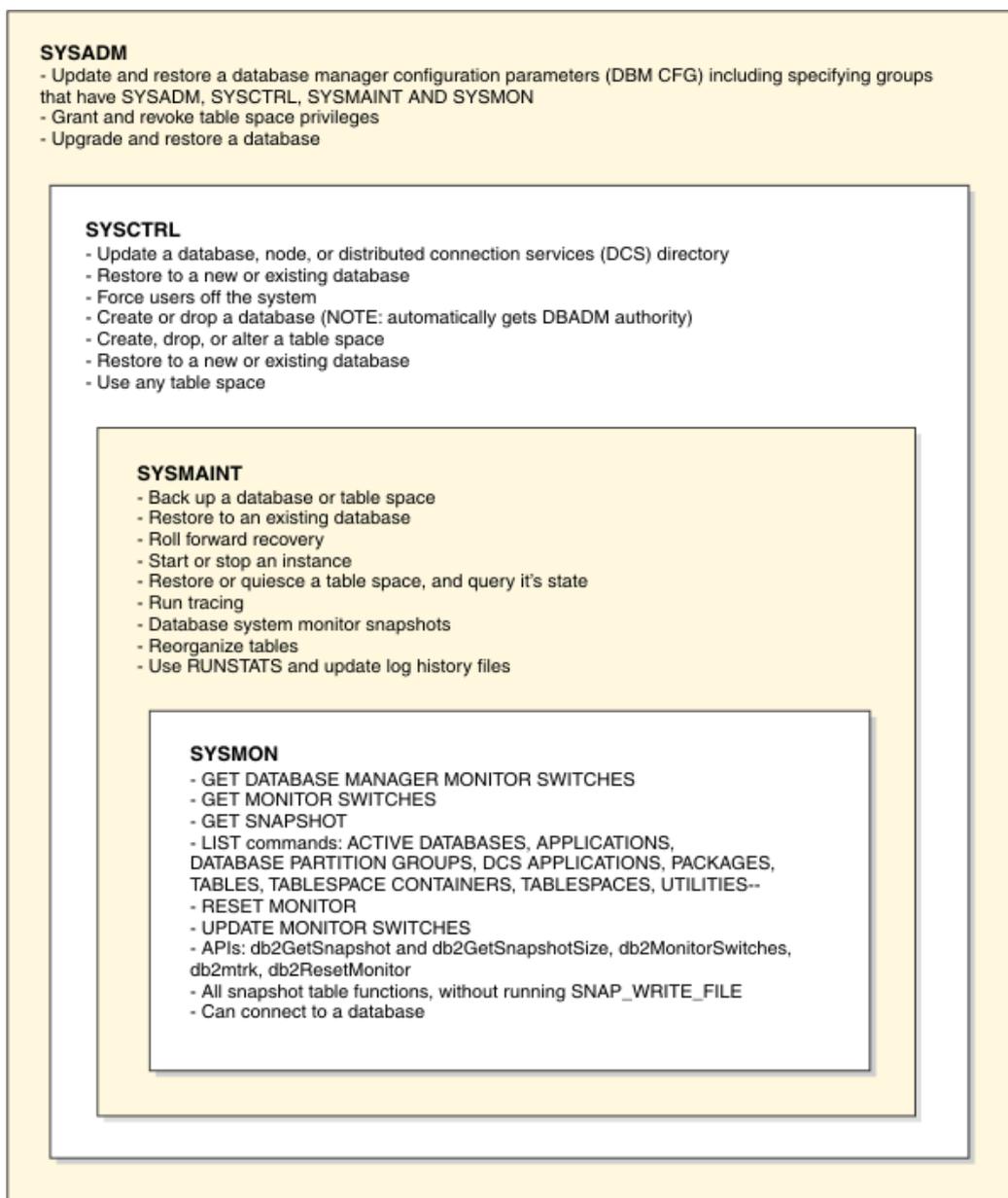


Figura 10.4 – Autoridades no nível de instância e suas funções

10.2.2.2 Autoridades no nível de banco de dados

A *Tabela 10.3* lista as autoridades no nível de banco de dados.

Autoridade	Descrição
SECADM	Gerencia a segurança dentro de um banco de dados
DBADM	Administra um banco de dados
 ACCESSCTRL	Concede e revoga autoridades e privilégios (além da autoridade SECADM, DBADM, ACCESSCTRL e DATAACCESS. Observe que a autoridade SECADM é necessária para conceder e revogar essas autoridades)
 DATAACCESS	Fornece a capacidade de acessar dados em um banco de dados.
SQLADM	Monitora e ajusta as consultas SQL
WLMADM	Gerencia as cargas de trabalho
EXPLAIN	Usuários que necessitam explicar os planos de consulta (a autoridade EXPLAIN não dá acesso aos dados em si)

Tabela 10.3 – Autoridades em nível de banco de dados

Para conceder autoridade em nível de banco de dados, use a instrução GRANT. Por exemplo, para conceder DBADM no banco de dados **SAMPLE** ao usuário *bob*, use:

```
connect to sample
grant DBADM on database to user bob
```

No exemplo acima, primeiro é necessário se conectar ao banco de dados, neste caso, o banco de dados **SAMPLE**, e em seguida, você pode conceder DBADM a um usuário. Para conceder a autoridade DBADM e qualquer outra autoridade em nível de banco de dados, você precisa ser um SECADM.

Observe que um DBADM não pode criar tablespaces, mesmo que sejam objetos dentro de um banco de dados, porque um tablespace trata de contêineres (discos) e bufferpools (memória), que são recursos físicos do sistema. Isto pode ser feito por um SYSADM.

A *Figura 10.5*, tirada do Information Center do DB2, mostra as diferentes autoridades em nível de banco de dados e as diferentes funções que podem realizar.

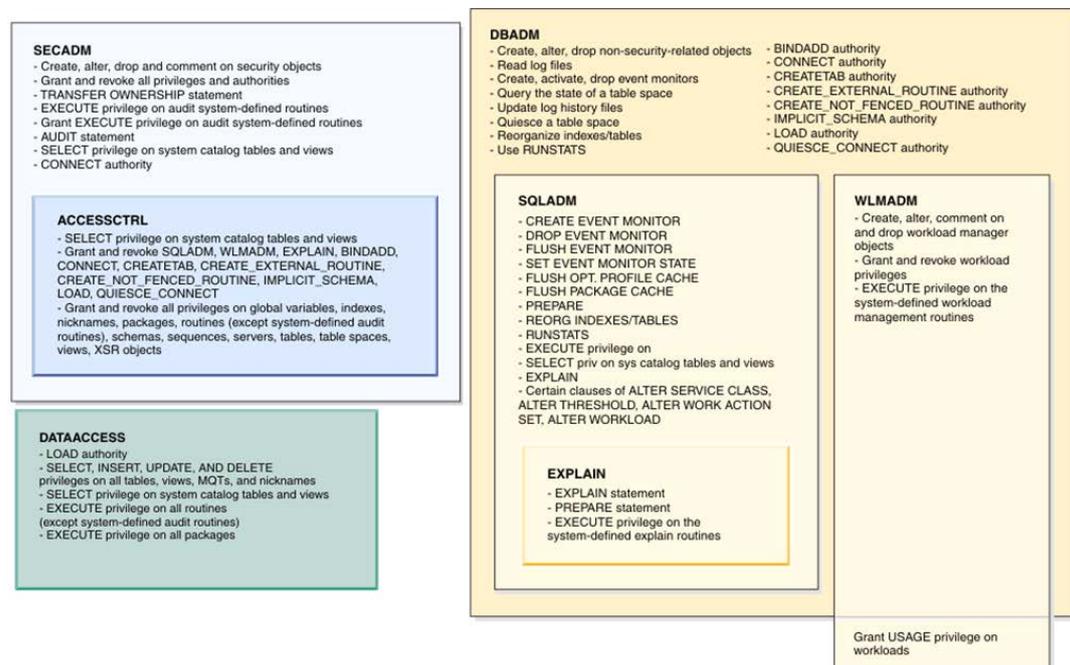


Figura 10.5 – Autoridades no nível de banco de dados e suas funções

Observação:

No DB2 9.7, para fornecer melhor privacidade de dados e conformidade com o controle, o modelo de autorização foi atualizado para separar claramente as obrigações do administrador do sistema, do administrador do banco de dados e do administrador de segurança.

Em geral, o escopo funcional de diversas autoridades foi reduzido em comparação com as versões anteriores do DB2. Por exemplo, um SYSADM não tem mais direitos de acesso aos dados de qualquer banco de dados. Um DBADM não tem mais direitos de acesso aos dados do banco de dados que administra. Por outro lado, o SECADM ganhou mais funcionalidade, como a capacidade de conceder e revogar autoridades e privilégios a usuários, funções e grupos.

Também foram criadas novas autoridades para permitir uma maior granularidade e controle da segurança do sistema. Isto também reduz o risco de exposição dos dados, ao não conceder aos usuários mais do que necessitam para fazer seu trabalho.

A atualização do DB2 9.7.2 também inclui melhorias na auditoria que permitem rever as atividades passadas do banco de dados. Por exemplo, se você necessita analisar como uma dada solicitação que ocorreu há alguns anos afetou algumas tabelas, agora você pode usar as informações de auditoria do banco de dados para obter o que necessita para essa análise.

10.2.2.3 Como possibilitar que o SYSADM e DBADM funcionem da mesma maneira que as versões do DB2 anteriores à 9.7

Se você desejar que um SYSADM se comporte da mesma maneira que nas versões do DB2 anteriores ao DB2 9.7, deve considerar dois casos:

- Se o SYSADM for o criador do banco de dados, ele recebe automaticamente as autoridades de DATAACCESS, ACCESSCTRL, SECADM e DBADM para esse banco de dados. Isto fornece ao SYSADM as mesmas capacidades das versões do DB2 anteriores à 9.7.
- Se o SYSADM não for o criador do banco de dados, é necessário que um SECADM conceda (GRANT) a ele as autoridades DBADM com DATAACCESS e ACCESSCTRL (que é o padrão) nesse banco de dados, para que ele obtenha as mesmas capacidades das versões anteriores do DB2 (exceto SECADM).

Alguns casos a considerar em relação a um SECADM:

- O SECADM padrão é o criador do banco de dados.
- Se um usuário com autoridade SECADM conceder SECADM a um usuário com autoridade SYSADM, o SYSADM pode conceder SECADM a outros usuários.
- Se um usuário com autoridade SECADM conceder DBADM a um usuário, o DBADM também recebe DATAACCESS e ACCESSCTRL, como padrão.

Se você estiver migrando de um banco de dados DB2 9.5, as capacidades de SYSADM e DBADM não se alteram, porque o DB2 concede automaticamente as autoridades DBADM, DATAACCESS e ACCESSCTRL ao grupo SYSADM, durante a migração. O DB2 também concede automaticamente DATAACCESS e ACCESSCTRL a cada ID de autorização que apresente DBADM após a migração. Além disso, o DB2 concede SECADM automaticamente ao ID de usuário que esteja fazendo a migração, caso não haja ID de autorização do tipo USER com autoridade SECADM no banco de dados. O SYSADM perde sua capacidade implícita de conceder ou revogar DBADM e SECADM, o que agora pode ser realizado somente pelo SECADM.

10.2.3 Funções

As funções permitem que um administrador de segurança atribua privilégios ou autoridades a diversos usuários ou grupos. As funções são muito semelhantes aos grupos, mas são definidas dentro do DB2 e, portanto, fornecem algumas vantagens. Por exemplo, os privilégios e autoridades concedidos às funções são sempre usados ao criar objetos como visualizações ou disparadores, o que não é o caso dos grupos. Por outro lado, não é possível atribuir autoridades em nível de instância como o SYSADM a uma função, somente privilégios e autoridades em nível de bancos de dados; enquanto para um grupo, todos os privilégios e autoridades podem ser atribuídos.

Para trabalhar com funções, é necessário seguir diversas etapas:

1. Primeiro, um administrador de segurança (SECADM) deve criar uma função usando um comando como

```
CREATE ROLE TESTER
```

2. Em seguida, um DBADM deve conceder privilégios ou autoridades à função. Por exemplo, para conceder privilégio SELECT sobre as tabelas **STAFF** e **DEPT** no banco de dados **SAMPLE** à função **TESTER**, execute:

```
GRANT SELECT ON TABLE STAFF TO ROLE TESTER
GRANT SELECT ON TABLE DEPT TO ROLE TESTER
```

3. Em seguida, o administrador de segurança concede a função **TESTER** aos usuários RAUL e JIN:

```
GRANT ROLE TESTER TO USER RAUL, USER JIN
```

4. Em seguida, se JIN fosse sair do departamento TEST, o administrador de segurança revogaria a função **TESTER** do usuário JIN:

```
REVOKE ROLE TESTER FROM USER JIN
```

10.3 Considerações sobre privilégio de grupo

Se você decidir usar grupos em vez de funções, leve em consideração o seguinte:

- Quando são concedidos privilégios a um grupo, os membros do grupo recebem privilégios implícitos herdados através da associação com o grupo.
- Quando um usuário é removido de um grupo, ele perde os privilégios implícitos ao grupo mas ainda retém qualquer privilégio anterior que tenha sido concedido explicitamente. Os privilégios concedidos explicitamente a um usuário devem ser revogados explicitamente.

10.4 O grupo PUBLIC

O DB2 define um grupo interno denominado PUBLIC (público). Qualquer usuário identificado pelo serviço de autenticação do sistema operacional ou da rede é implicitamente um membro do grupo PUBLIC. Quando é criado um banco de dados, alguns privilégios são concedidos automaticamente ao PUBLIC:

- CONNECT,
- CREATETAB,
- IMPLICIT_SCHEMA,
- BINDADD

Para adicionar segurança, recomendamos revogar todos os privilégios do grupo PUBLIC, como mostrado abaixo:

```
REVOKE CONNECT ON DATABASE FROM PUBLIC
REVOKE CREATETAB ON DATABASE FROM PUBLIC
REVOKE IMPLICIT_SCHEMA ON DATABASE FROM PUBLIC
REVOKE BINDADD ON DATABASE FROM PUBLIC
```

10.5 As instruções GRANT (conceder) e REVOKE (revogar)

As instruções GRANT e REVOKE são parte do padrão SQL e são usadas para dar ou remover privilégios de um usuário, grupo ou função. O usuário que executa este comando deve ter pelo menos autoridade ACCESSCTRL. Veja abaixo alguns exemplos dessas instruções:

Para conceder o privilégio SELECT sobre a tabela T1 ao usuário USER1:

```
GRANT SELECT ON TABLE T1 TO USER user1
```

Para conceder todos os privilégios sobre a tabela T1 ao grupo GROUP1:

```
GRANT ALL ON TABLE T1 TO GROUP group1
```

Para revogar todos os privilégios sobre a tabela T1 do grupo GROUP1:

```
REVOKE ALL ON TABLE T1 FROM GROUP group1
```

Para conceder privilégio EXECUTE sobre o procedimento p1 ao usuário USER1:

```
GRANT EXECUTE ON PROCEDURE p1 TO USER user1
```

Para revogar o privilégio EXECUTE sobre o procedimento p1 do usuário USER1:

```
REVOKE EXECUTE ON PROCEDURE p1 FROM USER user1
```

10.6 Autorização e verificação de privilégio

A maneira mais simples de verificar a autorização e os privilégios é através do Control Center. A *Figura 10.6* ilustra como inicializar a caixa de diálogo Table Privileges (Privilégios de tabela) a partir do Control Center para a tabela **EMPLOYEE**.

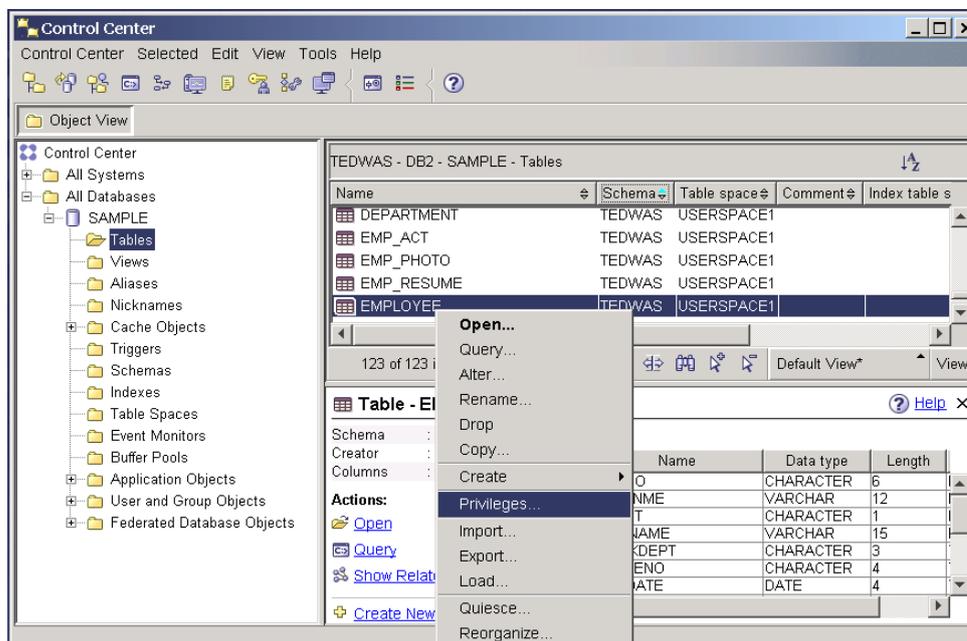


Figura 10.6 – Inicialização da caixa de diálogo Table Privileges

Como mostrado pela Figura 10.6, selecione a tabela desejada, clique com o botão direito nela e selecione *Privileges*. Depois de selecionada, a caixa de diálogo *Table Privileges* aparece como mostrado na Figura 10.7. A figura também explica os campos e elementos diferentes da caixa de diálogo.

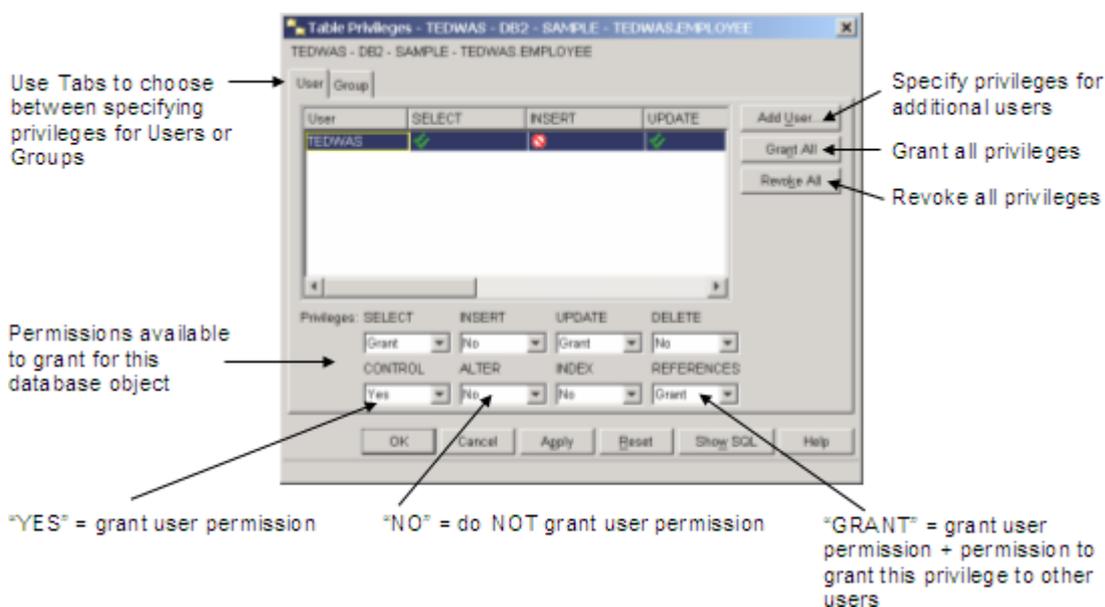


Figura 10.7 – A caixa de diálogo Table Privileges

Como alternativa, você pode consultar as visualizações de catálogo SYSCAT do DB2, que contêm a informações de autorização. Por exemplo, se você deseja saber se o usuário **DB2ADMIN** possui o privilégio SELECT sobre a tabela T2 e quem concedeu este privilégio, execute uma consulta como esta:

```
SELECT grantor, grantee, selectauth
   FROM syscat.tabauth
  WHERE tabname = 'T2'
```

GRANTOR	GRANTEE	SELECTAUTH
ARFCHONG	DB2ADMIN	Y

No exemplo acima, o usuário **ARFCHONG** concedeu o privilégio SELECT ao usuário **DB2ADMIN**.

10.7 Segurança estendida no Windows

Para impedir o acesso através do sistema operacional Windows aos arquivos e diretórios do DB2 (como aqueles nos quais o DB2 armazena informações de instância), o DB2 ativa, como padrão, a segurança estendida no momento da instalação. A segurança estendida cria dois grupos:

- **DB2ADMNS**: este grupo e os administradores locais terão acesso completo a todos os objetos do DB2, através do sistema operacional.
- **DB2USERS**: este grupo terá acesso de leitura e gravação a todos os objetos do DB2 através do sistema operacional.



Com o DB2 9.7, os membros do grupo DB2ADMNS terão autoridade de SYSADM no DB2 automaticamente, caso a segurança estendida seja ativada e o parâmetro de configuração de banco de dados SYSADM_GROUP não seja definido.

10.8 Resumo

Este capítulo cobriu os aspectos de segurança do DB2, começando com uma discussão abrangente sobre as diferenças e a importância da autenticação e da autorização. A partir daí, examinamos os diversos níveis de autoridade que fornecem segurança à instância e ao banco de dados.

Em seguida, cobrimos o novo conceito de funções e como elas podem ser usadas para o seu benefício, no que diz respeito à segurança e às limitações de estabelecer a segurança através dos grupos. Foi discutido especificamente o grupo PUBLIC, junto com sugestões sobre como protegê-lo para que os usuários gerais sejam bloqueados no servidor de dados.

Além disso, as instruções GRANT e REVOKE foram analisadas e, finalmente, vimos como usar o Control Center e as tabelas de catálogo do sistema para verificar a autorização e os níveis de privilégio.

10.9 Exercícios

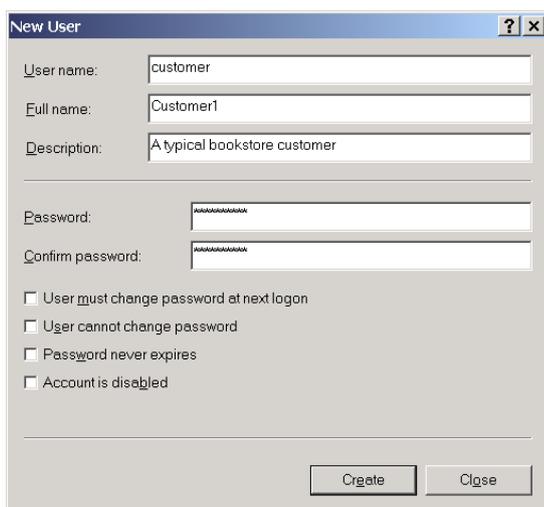
Até o momento, você vem usando a conta de administrador de instância (SYSADM) para executar todos os comandos de banco de dados. Esta conta tem acesso amplo a todos os utilitários, dados e objetos de banco de dados. Portanto, é muito importante proteger esta conta para evitar a perda de dados acidental ou deliberada. Na maioria dos casos, basta criar contas ou grupos de usuários diferentes, com um conjunto de permissões limitado. Neste exercício, você irá criar uma nova conta de usuário e atribuir privilégios específicos a ela.

Parte 1 – Como trabalhar com privilégios

Nesta parte do exercício, você irá praticar como conceder e revogar privilégios aos usuários, usando o Control Center.

Procedimento

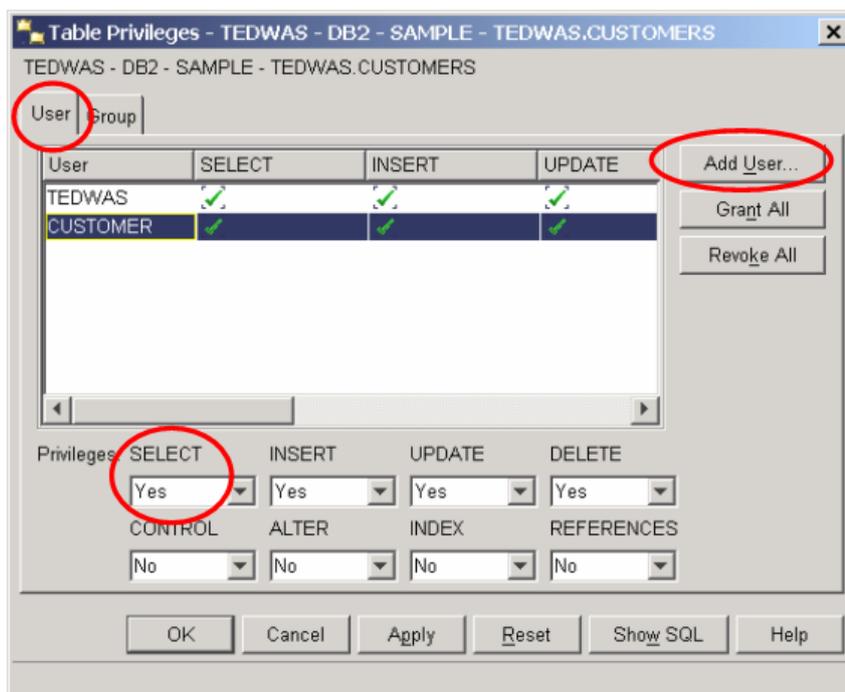
1. Abra o console de Gerenciamento do computador do Windows, clicando com o botão direito no ícone *Computador* na área de trabalho e selecionando o item de menu *Gerenciar*.
2. Expanda a seleção *Ferramentas do sistema* na árvore do painel esquerdo da janela e, em seguida, expanda a pasta *Usuários e grupos locais*. Clique com o botão direito na pasta *Usuário* e selecione o item *Novo usuário*.
3. Na caixa de diálogo *Novo usuário*, digite as informações abaixo: no campo *Nome de usuário*, digite *customer* e no campo *Nome completo*, digite *Customer1*. No campo *Descrição*, digite *A typical bookstore customer* (Um cliente típico de livraria). Nos campos *Senha* e *Confirmar senha*, digite *ibmdb2ibm*. Remova a marca da opção *O usuário deve alterar a senha no próximo logon*, clique no botão *Criar* para criar o novo usuário e, em seguida, clique no botão *Fechar* para fechar a janela.



The screenshot shows a 'New User' dialog box with the following fields and options:

- User name: customer
- Full name: Customer1
- Description: A typical bookstore customer
- Password: [masked]
- Confirm password: [masked]
- User must change password at next logon
- User cannot change password
- Password never expires
- Account is disabled
- Buttons: Create, Close

4. Abra o Control Center e selecione a visualização avançada. Para alternar para a visualização avançada, selecione no menu *Tools ->Customize Control Center* (Ferramentas -> Personalizar Control Center). Em seguida, selecione a opção *Advanced* (Avançada) e clique no botão *OK*.
5. Expanda a árvore de objetos no painel esquerdo do Control Center para *All Databases -> EXPRESS -> Tables* (*Todos os bancos de dados -> EXPRESS -> Tabelas*).
6. Conceda os privilégios necessários ao usuário recém-criado. Na lista de tabelas do banco de dados *EXPRESS*, clique com o botão direito na tabela *CUSTOMERS* e selecione o item *Privileges* (Privilégios) para exibir a caixa de diálogo *Table Privileges* (Privilégios de tabela).
7. Clique no botão *Add User* (Adicionar usuário) e selecione o usuário *customer* que acabou de criar. Clique no botão *OK* para fechar a caixa de diálogo *Add User* (Adicionar usuário).
8. Você poderá ver que o usuário *customer* foi adicionado à lista de usuários, mas nenhum privilégio foi atribuído. Para conceder os privilégios *SELECT*, *INSERT*, *UPDATE* e *DELETE* ao usuário, altere cada caixa suspensa para *Yes* (Sim). Um cliente da Internet deve poder visualizar/adicionar/atualizar/excluir os dados de sua própria conta. Não concedemos ao usuário as outras permissões porque ele não necessita delas. Clique no botão *OK* para fechar a caixa de diálogo *Table Privileges* e aceitar as alterações feitas.



9. Repita as etapas 7 a 9 para as tabelas *BOOKS* e *SALES*. Para a tabela *BOOKS*, conceda somente o privilégio SELECT, porque o cliente não deve ser capaz de modificar nada nos dados de inventário da loja. Na tabela *SALES*, conceda somente os privilégios SELECT e INSERT. O cliente NÃO deve ter os privilégios DELETE ou UPDATE porque somente os funcionários da loja devem ter acesso para modificar as transações de vendas.
10. Conecte-se ao banco de dados usando o ID de usuário *customer* criado acima, usando a Command Window (Janela de commando) do DB2 da forma a seguir:

```
db2 connect to express user customer using ibmdb2ibm
```

Tente selecionar (SELECT) os dados da tabela *customers*. O que ocorre? Tente excluir (DELETE) ou atualizar (UPDATE) os dados da tabela *SALES*. O que ocorre?

Parte 2 – Como trabalhar com as autoridades SYSADM, DBADM e SECADM

Nesta parte do exercício, você praticará como atribuir as autoridades SYSADM e DBADM e compreenderá como estas autoridades funcionam.

Procedimento

1. Siga as mesmas etapas da parte 1 para criar um novo usuário: *mysysadm*
2. Crie o grupo *mysysadmgrp* do Windows. Siga as mesmas etapas para criar um usuário, mas em vez de clicar na pasta *User* (Usuário), clique com o botão direito do mouse na pasta *Group* (Grupo) e selecione *New Group* (Novo grupo). No campo Group name (Nome do grupo), digite *mysysadmgrp*. Na seção Members (Membros), clique em *Add* (Adicionar) para adicionar um novo membro e digite *mysysadm*. Clique no botão *Check Names* (Verificar nomes) para confirmar se digitou o nome do membro corretamente. Se o fez, clique em OK. Em seguida, clique em *Create* (Criar) e, em seguida, clique em *Close* (Fechar).
3. Até aqui, você criou o usuário *mysysadm* e o grupo *mysysadmgrp*, ao qual pertence o usuário *mysysadm*. Isso tudo foi feito no sistema operacional Windows. Agora, precisamos informar ao DB2 que desejamos que o grupo *mysysadmgrp* seja o grupo SYSADM, usando este comando na Command Window (Janela de commando) do DB2:

```
db2 update dbm cfg using SYSADM_GROUP mysysadmgrp
```

Como o parâmetro SYSADM_GROUP não é dinâmico, é necessário parar e reiniciar a instância. A opção para forçar a interrupção em *db2stop* irá assegurar que todas as conexões anteriores a *db2stop* sejam removidas.

```
db2stop force  
db2start
```

4. Conecte-se ao banco de dados SAMPLE com o usuário *mysysadm* na Command Window (Janela de commando) do DB2 e execute uma instrução SELECT * na

tabela STAFF. Observe que é necessário usar o esquema correto, que foi usado ao criar esta tabela. No exemplo abaixo, usamos **arfchong** como esquema.

```
db2 connect to sample user mysysadm using ibmdb2ibm
db2 select * from arfchong.staff
```

Você deve receber uma mensagem de erro como esta. Por quê? Você não é o SYSADM?

```
SQL0551N  "MYSYSADM" does not have the required authorization or
privilege to perform operation "SELECT" on object "ARFCHONG.STAFF".
SQLSTATE=42501
```

A partir do DB2 9.7, o SYSADM não possui autoridade DBADM como padrão, é por isso que você recebeu a mensagem de erro.

- Com o usuário do Windows que foi usado para criar o banco de dados **SAMPLE**, conecte-se ao banco de dados. No exemplo, este usuário é **ARFCHONG**. Em seguida, conceda DBADM sem DATAACCESS ao usuário **mysysadm** e tente usar SELECT novamente em STAFF, como **mysysadm**. Funcionou? Por quê?

```
db2 connect to sample user arfchong using ibmdb2ibm
db2 grant dbadm without dataaccess on database to user mysysadm
db2 connect to sample user mysysadm using ibmdb2ibm
db2 select * from arfchong.staff
```

Como você viu, ainda receberá a mesma mensagem de erro, mesmo após **mysysadm** receber a autoridade DBADM. este comportamento é esperado porque incluímos a cláusula WITHOUT DATAACCESS, que significa que a autoridade DATAACCESS não foi incluída e, portanto, **mysysadm** ainda não possui autoridade para acessar os dados. Isso mostra a você um exemplo de como é possível restringir o acesso do DBADM aos dados.

- Agora, vamos conceder DATAACCESS ao **mysysadm** e tentar usar SELECT novamente.

```
db2 connect to sample user arfchong using ibmdb2ibm
db2 grant DATAACCESS on database to user mysysadm
db2 connect to sample user mysysadm using ibmdb2ibm
db2 select * from arfchong.staff
```

Agora SELECT deve funcionar! Este exercício mostra o novo comportamento do SYSADM e DBADM a partir do DB2 9.7. A ideia principal a tirar deste exercício é que agora há uma separação entre o acesso aos dados e o que o SYSADM e o DBADM podem fazer.

- Restaure o valor de SYSADM_GROUP para NULL, para que o grupo do Administrador local e a conta LocalSystem possam se tornar SYSADM novamente:

```
db2 update dbm cfg using sysadm_group NULL
db2stop force
db2start
```


11

Capítulo 11 – Backup e recuperação

Neste capítulo discutiremos o log do banco de dados DB2, como fazer uma cópia total ou parcial do banco de dados com o utilitário BACKUP e como recuperar os dados com o utilitário RESTORE.

Observação:

Para obter mais informações sobre log, backup e recuperação, assista ao vídeo: <http://www.channeldb2.com/video/video/show?id=807741:Video:4282>

11.1 Log do banco de dados

Ao trabalhar com um editor de texto, sempre que deseja assegurar que o documento seja salvo, você clica no botão *salvar*. No universo dos bancos de dados, a instrução COMMIT faz exatamente isso. Sempre que uma instrução COMMIT é executada, você assegura que quaisquer alterações dos dados serão salvas em algum lugar.

De um modo semelhante, ao trabalhar com um documento de texto, às vezes aparece no canto inferior direito uma mensagem rápida informando "salvando automaticamente". No universo dos bancos de dados, isto também ocorre, pois cada operação realizada com os dados, como UPDATE, INSERT ou DELETE, será salva em algum lugar ao ser realizada.

Esse "algum lugar" dos parágrafos anteriores refere-se aos logs do banco de dados. Os logs do banco de dados são armazenados em disco e usados para registrar as ações das transações. Se houver uma queda do sistema ou do banco de dados, os logs serão usados para reproduzir e refazer as transações salvas, durante uma recuperação.

A *Figura 11.1* fornece uma visão geral gráfica do que ocorre em termos de log quando você trabalha com um banco de dados.

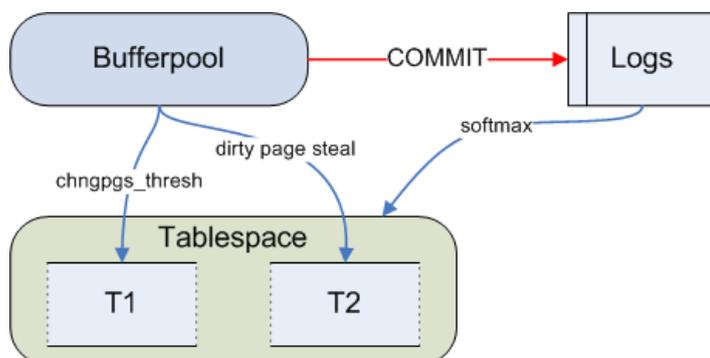


Figura 11.1 – Log do banco de dados

Na *Figura 11.1*, vemos um tablespace e os logs. Ambos residem nos discos, embora seja recomendável que não fiquem no mesmo disco. Quando ocorre uma operação UPDATE, por exemplo, as páginas da(s) linha(s) em questão serão levadas ao bufferpool (memória). As alterações de atualização são realizadas no bufferpool, e os valores antigo e novo serão armazenados nos arquivos de log, às vezes imediatamente e às vezes quando o buffer de log fica cheio. Se for executada uma operação COMMIT após a UPDATE, os valores antigo e novo serão armazenados nos arquivos de log imediatamente. Este processo é repetido para diversas outras operações de SQL executadas no banco de dados. As páginas do bufferpool só serão “externalizadas” e gravadas no disco do tablespace quando forem atendidas determinadas condições, como atingir o limite de página de alterações especificado pelo parâmetro CHNGPGS_THRES. O parâmetro CHNGPGS_THRES indica o percentual do bufferpool com páginas “suja”, isto é, páginas contendo alterações.

Do ponto de vista do desempenho, não faz sentido executar duas gravações para cada operação de COMMIT: Uma para gravar nos logs e a outra para gravar no disco do tablespace. É por isso que a “externalização” dos dados para o disco do tablespace ocorre somente quando é atingido o limite de parâmetros como CHNGPGS_THRES.

11.2 Tipos de logs

Há dois tipos de logs:

- **Logs primários**

Estes são pré-alocados e o número de logs primários disponíveis é determinado pelo parâmetro de configuração de banco de dados LOGPRIMARY.

- **Logs secundários**

Estes são alocados dinamicamente, conforme necessário ao DB2. O número máximo de logs secundários é definido pelo parâmetro de configuração de banco de dados, LOGSECOND. A alocação dinâmica de um log é dispendiosa; portanto, para as operações diárias, mantenha-se dentro da alocação de log primário. Os arquivos de log secundário são excluídos quando todas as conexões a um banco de dados são encerradas.

É possível haver um número de arquivos de logs infinito caso LOGSECOND seja definido com valor -1; entretanto, isto não é recomendado, pois você pode ficar sem espaço no sistema de arquivos.

11.3 Tipos de logs

Há dois tipos de logs: log circular (padrão) e log de arquivamento.

11.3.1 Log circular

O padrão é o log circular, que é ativado quando ambos os parâmetros LOGARCHMETH1 e LOGARCHMETH2, de configuração de banco de dados, são definidos como OFF. Esses parâmetros indicam o método usado para arquivar os arquivos de logs; se você desativá-los, isto significa que não deseja arquivar os arquivos de logs, que é como funciona o log circular. A *Figura 11.2* descreve um exemplo que ilustra o log circular.

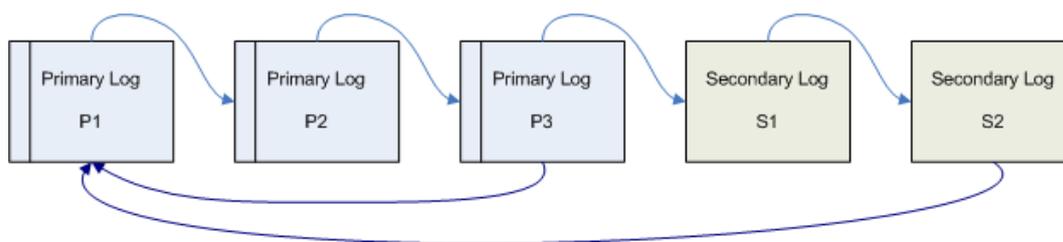


Figura 11.2 – Como trabalhar com logs primários e secundários

Na *Figura 11.2* há 3 logs primários; portanto, podemos supor que o valor do parâmetro LOGPRIMARY é 3. Para manter a simplicidade, este exemplo mostra somente uma transação sendo executada. Conforme a transação é realizada, primeiro o arquivo de log P1 começa a ficar cheio, seguido de P2. Se ocorrer uma operação commit e as informações forem externalizadas posteriormente para o disco do tablespace, é possível sobrescrever P1 e P2, pois as informações não são mais necessárias à recuperação de queda (que será discutida em mais detalhes mais adiante neste capítulo). Por outro lado, se a transação for tão longa que utilize P1, P2, P3 e ainda necessite de mais espaço no log porque ainda não foi salva nem externalizada, um log secundário (S1, na figura) é alocado dinamicamente. Se a transação continuar, mais arquivos de logs secundários serão alocados até a alocação máxima de logs de LOGSECOND. Se forem necessários ainda mais logs, o usuário receberá uma mensagem de erro indicando que foi atingida uma condição de log cheio e a transação será revertida. Como alternativa, você pode configurar o DB2 usando o parâmetro de configuração BLK_LOG_DSK_FUL para continuar gravando nos logs a cada 5 minutos enquanto algumas transações são colocadas em espera. Isso dá ao DBA algum tempo para encontrar mais espaço, para que a transação possa continuar.

O log circular permite a recuperação após uma queda, mas se você desejar recuperar seus dados para um ponto específico no tempo, o momento mais próximo disponível será quando você realizou o último backup off-line.

11.3.2 Log de arquivamento

No log de arquivamento, também conhecido como log com retenção, os arquivos de log não são sobrescritos e sim mantidos, seja on-line ou off-line. Os logs de arquivamento on-line permanecem com os logs ativos que ainda são necessários para a recuperação após a uma queda. Os logs de arquivamento off-line são movidos para outras mídias como fita, e isto pode ser feito através de rotinas USEREXIT, do Tivoli Storage Manager ou de qualquer outro produto de arquivamento de terceiros.

Para ativar o log de arquivamento, configure os parâmetros LOGARCHMETH1 ou LOGARCHMETH2 (ou ambos) de configuração de banco de dados com um valor diferente de OFF. Outra maneira de ativar isso é definir o parâmetro de configuração LOGRETAIN como RECOVERY. Isto fará com que LOGARCHMETH1 seja definido automaticamente como LOGRETAIN. Entretanto, o parâmetro LOGRETAIN está obsoleto, e foi deixado principalmente para manter a compatibilidade com versões anteriores do DB2.

O log de arquivamento normalmente é usado em sistemas de produção. Como os logs são mantidos, isto permite a recuperação do banco de dados de volta até o momento do arquivo de log mais antigo. Com o log de arquivamento, um DBA pode se recuperar de erros causados por pessoas. Por exemplo, caso um usuário de um sistema comece a executar inadvertidamente uma transação incorreta que dure dias, quando o problema for detectado, o DBA pode restaurar o sistema de volta ao momento antes do problema ser introduzido. Entretanto, pode ser necessária alguma manipulação manual para a transação ser re-executada corretamente.

O log de arquivamento é necessário para a recuperação do tipo rolar para a frente e para o backup on-line. A *Figura 11.3* descreve o processo de log de arquivamento.

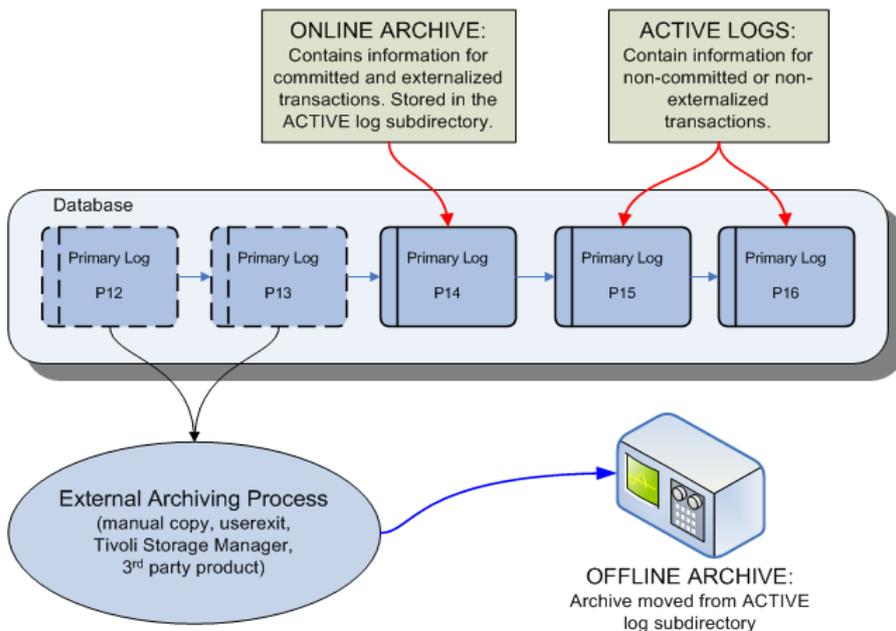


Figura 11.3 – Log de arquivamento

11.4 Log do banco de dados no Control Center

Você pode configurar o log do banco de dados no Control Center, clicando com o botão direito do mouse no banco de dados em questão e selecionando *Configure Database Logging* (Configurar log do banco de dados). Isso é ilustrado na *Figura 11.4*.

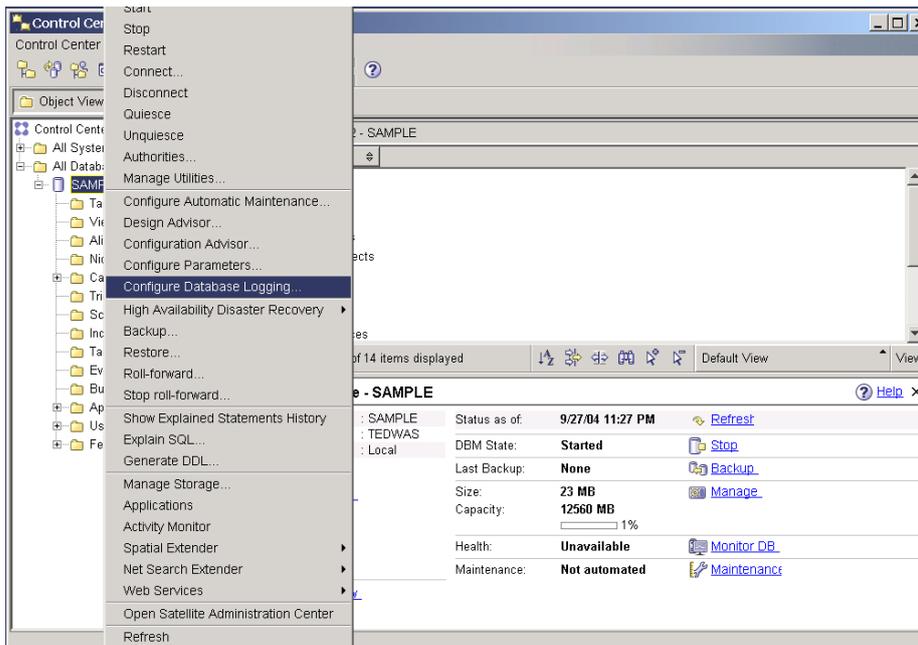


Figura 11.4 – Configuração de log de banco de dados no Control Center.

A *Figura 11.5* mostra o assistente Database Logging, onde você pode selecionar o log circular ou de arquivamento.

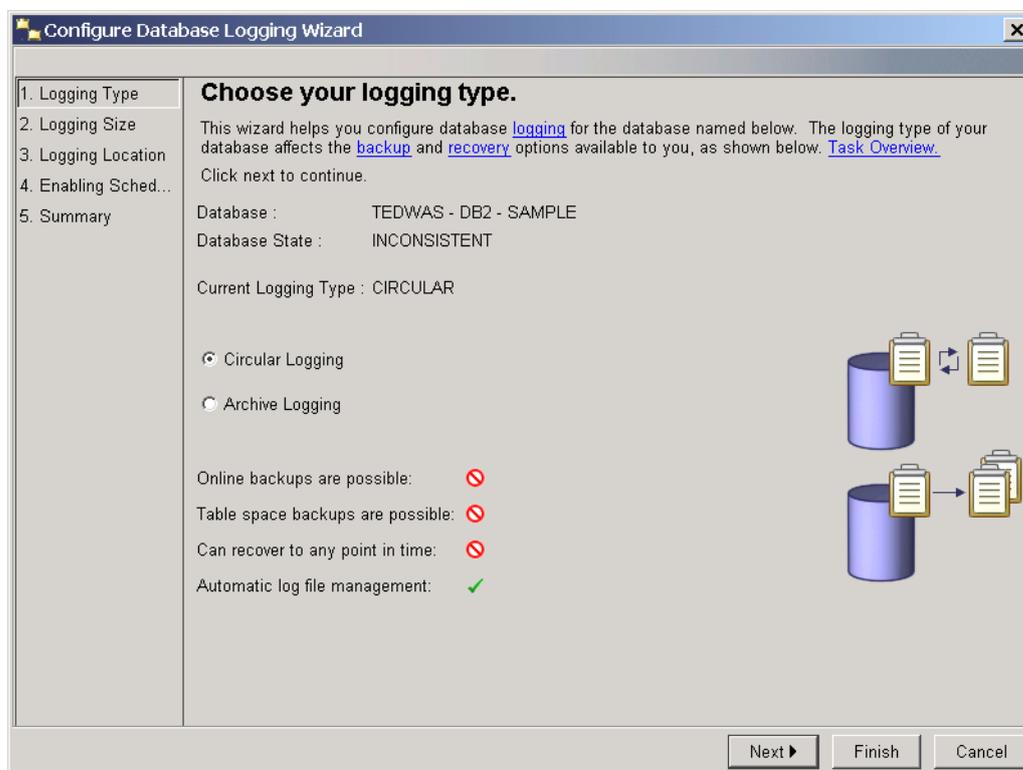


Figura 11.5 – Assistente Database Logging

11.5 Parâmetros de log

Há diversos parâmetros do DB CFG relacionados ao log. A *Tabela 11.1* lista os principais parâmetros.

Parâmetro	Descrição
logbufsz	A quantidade de memória a ser usada como buffer para os registros de log antes de gravá-los em disco.
logfilsz	O tamanho de cada arquivo de log configurado, em número de páginas de 4KB
logprimary	O número de arquivos de log primários com tamanho logfilsz que serão criados
logsecond	O número de arquivos de log secundários criados e usados para a recuperação, se necessário.
newlogpath	Inicialmente, os arquivos de log de banco de dados ativos e on-line são criados no diretório do banco de dados, no subdiretório SQLOGDIR. Você pode alterar esta localização, alterando o valor deste parâmetro de configuração para apontar para um diretório diferente ou para um dispositivo.
mirrorlogpath	Para proteger os logs do caminho do log primário de falha de disco ou exclusão acidental, você pode especificar que seja mantido um

	conjunto de logs idêntico em um caminho de log secundário (ou espelho).
logarchmeth1 logarchmeth2	Especifica uma localização diferente do caminho do log ativo para armazenar os arquivos de log de arquivamento. Se ambos os parâmetros forem especificados, cada arquivo de log é arquivado duas vezes. Isto significa que você terá duas cópias do log arquivado em duas localizações diferentes. Os valores possíveis são OFF (que significa que o log circular é ativado), LOGRETAIN, USEREXIT, DISK, TSM, VENDOR
loghead	O nome do arquivo de log ativo no momento
softmax	Limita o custo da recuperação de queda
overflowlogpath	Especifica uma localização para o DB2 procurar arquivos de log necessários para uma operação de rolar para a frente. Semelhante à opção OVERFLOW LOG PATH do comando ROLLFORWARD.
blk_log_dsk_ful	Definido para evitar que erros de disco cheio sejam gerados quando o DB2 não puder criar um novo arquivo de log no caminho de log ativo. Em vez disso, o DB2 irá tentar criar o arquivo de log a cada cinco minutos, até conseguir. O SQL desbloqueado e somente de leitura poderá continuar.
max_log	Percentual de espaço máximo de log ativo por uma transação
num_log_span	Número de arquivos de log ativos por 1 UOW ativo
mincommit	Número de operações de commit para grupo antes de ser gravado em disco

Tabela 11.1 – Parâmetros de log

11.6 Backup do banco de dados

O comando backup do DB2 permite fazer uma cópia instantânea do banco de dados no momento em que o comando é executado. A sintaxe mais simples que se pode usar para executar este comando é:

```
BACKUP DATABASE <dbname> [ TO <path> ]
```

A maioria dos comandos e utilitários pode ser executada on-line ou off-line. On-line implica que outros usuários podem estar conectados e realizando operações no banco de dados enquanto você executa o comando. Off-line significa que não há nenhum outro usuário conectado ao banco de dados enquanto você executa sua operação. Para permitir uma operação on-line, adicione a palavra-chave ONLINE à sintaxe do comando; caso contrário, o comando irá supor que está sendo executado off-line, como padrão.

Por exemplo, se deseja fazer backup do banco de dados **SAMPLE** no caminho C:\BACKUPS, você pode executar este comando na Command Window (Janela de comando) do DB2 ou no shell do Linux:

```
db2 BACKUP DB sample TO C:\BACKUPS
```

Observe que o diretório C:\BACKUPS deve existir antes de executar o comando. Verifique também se não há conexões com o banco de dados ao executar o comando acima; caso

contrário, você receberá uma mensagem de erro, já que um backup off-line não pode ser executado quando houver conexões.

Para descobrir se há conexões aos bancos de dados de uma instância, execute este comando na janela de comando do DB2 ou no shell do Linux:

```
db2 list applications
```

Para forçar o encerramento de todas as conexões a todos os bancos de dados de uma instância, execute este comando na Command Window (Janela de comando) do DB2 ou no shell do Linux:

```
db2 force applications all
```

Este último comando não deve ser executado em um ambiente de produção com muitos usuários; caso contrário, você receberá muitas chamadas de colegas de trabalho irritados! Além disso, lembre-se que o último comando é executado de forma assíncrona. Isto significa que ao tentar executar o comando de backup logo a seguir, ele pode não funcionar ainda. Aguarde alguns segundos e repita o comando de backup, caso tenha recebido uma mensagem de erro da primeira vez.

Após uma execução bem-sucedida do comando de backup, é criado um novo arquivo contendo a imagem de backup do banco de dados. O nome deste arquivo segue a convenção mostrada na *Figura 11.6*.

Linux/UNIX/Windows

Alias Instance Year Day Minute Sequence
 DBALIAS.0.DB2INST.NODE0000.CATN0000.20060314131259.001
 Type Node Catalog Node Month Hour Second

Figura 11.6 – Convenção de nomeação de imagem de backup

O tipo “0” significa que o backup é completo. O tipo “3” significa que é um backup de tablespace. O nó está fixo em NODE0000 para bancos de dados sem particionamento, que é o caso de todas as edições do DB2, exceto o DB2 Enterprise Edition com recurso DPF. O nó de catálogo também está fixo, em CATN0000. Consulte os manuais do DB2 para obter mais detalhes.

Quando forem realizados e armazenados diversos backups no mesmo caminho, o carimbo de data/hora no final do nome do arquivo é usado para distinguir entre as imagens de backup. Como veremos na próxima seção, o comando RESTORE pode usar este carimbo de data/hora para restaurar um backup específico.

11.7 Recuperação do banco de dados

Uma recuperação de banco de dados implica na restauração do banco de dados usando um arquivo de backup e/ou os logs. Se você restaurar apenas a partir do backup, irá recriar o banco de dados como ele existia no momento em que foi feito o backup.

Se o log de arquivamento foi ativado antes do backup, além de restaurar usando uma imagem de backup, você também pode usar os logs. Como veremos na próxima seção, uma recuperação de rolagem para a frente permite a restauração de um backup e, em seguida, a aplicação (rolagem para a frente) dos logs até o final dos mesmos, ou até um momento específico.

Observe que o termo "recuperação" é usado com frequência nesta seção, mas o comando usado para a recuperação é denominado RESTORE.

11.7.1 Tipos de recuperação

Há três tipos de recuperação:

- **Recuperação de queda ou reinício**

Suponha que você está trabalhando em um computador de mesa executando transações importantes em um banco de dados do DB2. De repente, há uma falta de luz ou outra pessoa desliga acidentalmente o cabo da tomada: o que ocorre com o banco de dados?

Na próxima vez em que ligar o computador e iniciar o DB2, a recuperação de queda será executada automaticamente. Na recuperação de queda, o DB2 executa automaticamente o comando RESTART DATABASE e lê e refaz/desfaz as transações com base nos logs ativos. Quando este comando for concluído, você tem a garantia de que seu banco de dados está em um estado consistente, ou seja, o que quer que tenha sido alocado será salvo e o que não foi alocado será revertido.

- **Recuperação de versão ou imagem**

Este tipo de recuperação implica que você está restaurando somente a partir de uma imagem de backup; portanto, seu banco de dados será colocado no estado em que estava no momento em que foi feito o backup. Qualquer transação executada no banco de dados após o backup ter sido feito será perdida.

- **Recuperação com rolagem para a frente**

Com este tipo de recuperação, você não apenas usa o RESTORE a partir de uma imagem de backup mas também executa o comando ROLLFORWARD para aplicar os logs sobre o backup, para poder recuperar até um determinado instante no tempo. Este tipo de recuperação reduz a perda de dados.

11.7.2 Restauração do banco de dados

Use o comando RESTORE para recuperar o banco de dados a partir de uma imagem de backup. A sintaxe abaixo é a mais simples que pode ser usada com este comando:

```
RESTORE DATABASE <dbname> [from <path>] [taken at <timestamp>]
```


diversos parâmetros de banco de dados relacionados ao log. Para cada tipo de log, discutimos quando e porque é usado, e como configurá-lo no Control Center.

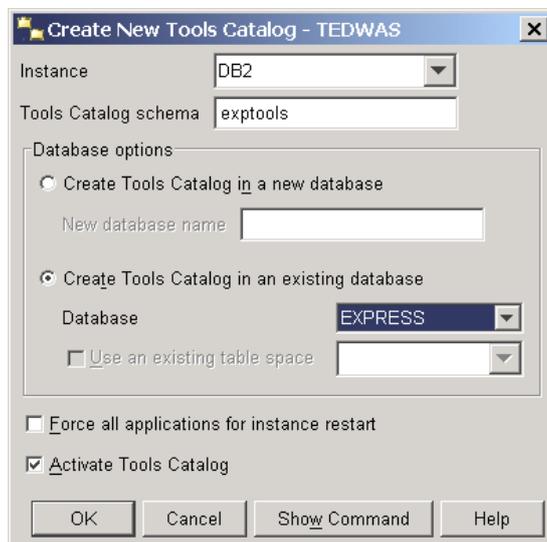
Também vimos como executar as atividades de backup e recuperação usando a linha de comando do DB2, incluindo uma visão detalhada dos três tipos de recuperação de banco de dados: queda, versão e rolagem para a frente.

11.10 Exercícios

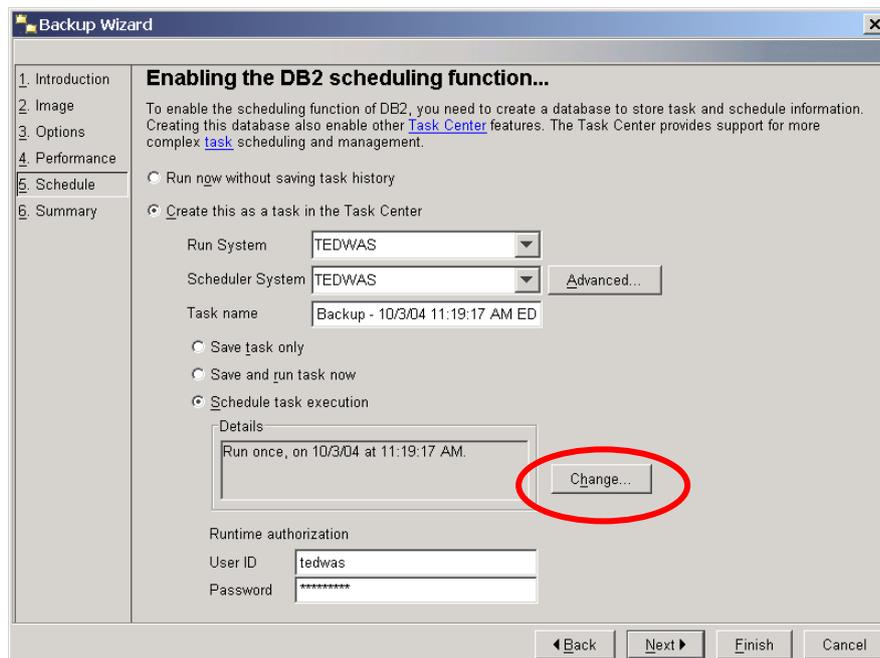
Embora o DB2 seja capaz de automatizar diversas atividades de manutenção de banco de dados, às vezes você deseja personalizar o momento em que determinadas atividades devem ocorrer. Neste exercício, você irá criar uma programação de backup noturno personalizada para o banco de dados **EXPRESS**.

Procedimento

1. Na árvore de objetos do Control Center, navegue até *Control Center* -> *All Databases* (Control Center -> Todos os bancos de dados). Clique com o botão direito no banco de dados **EXPRESS** e selecione o item *Backup*. Isto inicia o *assistente de Backup*.
2. A página *Introduction* (Introdução) do assistente resume o estado atual do banco de dados, incluindo a hora do último backup e o método de log. Clique no botão *Next* para passar para a próxima página do assistente.
3. Na página *Image* (Imagem) do assistente, selecione o destino da imagem de backup. Normalmente, você deve selecionar uma unidade física diferente daquela onde o banco de dados existente está armazenado. Por enquanto, crie uma nova pasta no sistema de arquivos denominado `C:\db2backup` e especifique essa pasta como localização do backup. No assistente, selecione o item *File System* (Sistema de arquivos) na lista suspensa *Media Type* (Tipo de mídia). Clique no botão *Add*, selecione a pasta que acabou de criar e clique no botão *OK*. Clique no botão *Next* para passar para a próxima página do assistente.
4. Você pode explorar as páginas *Options* (Opções) e *Performance* (Desempenho), mas as opções padrão normalmente são suficientes porque o DB2 executa o backup do banco de dados automaticamente da maneira ideal. Navegue até a página *Schedule* (Programação) quando terminar de explorar.
5. Na página *Schedule*, selecione ativar o recurso de agendamento, caso isto ainda não tenha sido feito. Selecione o sistema no qual irá criar o catálogo de ferramentas e crie um novo catálogo de ferramentas. Especifique um esquema para o catálogo de ferramentas e selecione criá-lo no banco de dados **EXPRESS** já existente. O catálogo de ferramentas contém metadados sobre todas as tarefas programadas. Clique no botão *OK* para continuar. Clique no botão *Next* para passar para a próxima página do assistente após a criação do catálogo de ferramentas.



6. Na página *Schedule*, selecione criar uma programação para a execução de tarefas. Programe o backup para ser executado todos os dias, começando à 01:00 hora. Clique no botão *Next* para passar para a próxima página.





7. Na página *Summary* (Resumo), você pode verificar as tarefas agendadas que serão criadas. Após revisar as alterações, clique no botão *Finish* (Concluir) para criar a tarefa.
8. Inicie o Task Center (Centro de Tarefas) para visualizar ou modificar a tarefa de backup que acabou de criar.

12

Capítulo 12 – Tarefas de manutenção

Este capítulo discute algumas das tarefas necessárias para uma boa manutenção do banco de dados. A direção geral no DB2 é automatizar a maioria dessas tarefas. A edição DB2 Express-C, como todas as atuais edições do DB2, inclui esses recursos automatizados. Esta capacidade de auto-gerenciamento é um enorme benefício para pequenas e médias empresas que não podem contratar um DBA em tempo integral para gerenciar o servidor de dados. Por outro lado, se for contratado um DBA, ele terá mais tempo livre para executar atividades avançadas que agregam valor aos resultados financeiros da empresa.

Observação:

Para obter mais informações sobre as tarefas de manutenção, assista ao vídeo:

<http://www.channeldb2.com/video/video/show?id=807741:Video:4302>

12.1 REORG, RUNSTATS, REBIND

Há três tarefas de manutenção principais no DB2, conforme ilustrado na *Figura 12.1*: REORG, RUNSTATS e REBIND.

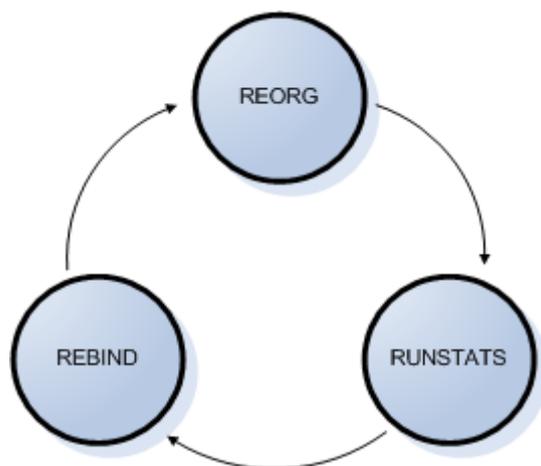


Figura 12.1 – Tarefas de manutenção: REORG, RUNSTATS, REBIND

A *Figura 12.1* mostra que as tarefas de manutenção são executadas de maneira circular. Se um REORG é executado, também é recomendável executar um RUNSTATS, seguido de um REBIND. Após algum tempo, as tabelas de um banco de dados serão modificadas, devido às operações UPDATE, DELETE e INSERT. Nesse momento, o ciclo se inicia novamente com REORG.

12.1.1 O comando REORG

Com o tempo, ao executar operações INSERT, UPDATE e DELETE (inserir, atualizar e excluir) no banco de dados, os dados começam a ficar cada vez mais fragmentados nas páginas do banco de dados. O comando REORG recupera o espaço desperdiçado e reorganiza os dados para tornar a recuperação mais eficiente. As tabelas modificadas com frequência são as mais beneficiadas com o comando REORG. Você pode usar o REORG também em índices, e um REORG pode ser executado on-line ou off-line.

O REORG off-line é mais rápido e eficiente, mas não permite o acesso à tabela, enquanto o REORG on-line permite acesso à tabela mas pode consumir muitos recursos do sistema; ele funciona melhor com tabelas pequenas.

Sintaxe:

```
REORG TABLE <tablename>
```

Exemplo:

```
REORG TABLE employee
```

O comando REORGCHK pode ser usado antes de um REORG para determinar se uma tabela ou índice necessita de correção.

12.1.2 O comando RUNSTATS

O DB2 Optimizer é o “cérebro” do DB2. Ele encontra os caminhos de acesso mais eficientes para localizar e recuperar os dados. O Optimizer é um sistema consciente dos custos, e utiliza estatísticas dos objetos de banco de dados armazenados nas tabelas de catálogo para maximizar o desempenho do banco de dados. Por exemplo, as tabelas de catálogo possuem estatísticas sobre quantas colunas estão presentes em uma tabela, quantas linhas existem, quantos e que tipo de índices estão disponíveis para uma tabela, e assim por diante.

As informações de estatística não são atualizadas dinamicamente. Isto é proposital, pois não é desejável que o DB2 atualize as estatísticas constantemente, para cada operação realizada no banco de dados. Isto teria um efeito negativo sobre o desempenho do banco de dados inteiro. Em vez disso, o DB2 fornece o comando RUNSTATS para atualizar tais estatísticas. Ele é essencial para manter atualizadas as estatísticas do banco de dados. O Optimizer do DB2 pode fazer alterações radicais no caminho de acesso, se achar que uma tabela possui 1 linha em vez de 1 milhão de linhas. Quando as estatísticas do banco de dados estão atualizadas, o DB2 pode escolher um plano de acesso melhor. A frequência

da coleta de estatísticas deve ser determinada pela frequência com que os dados são alterados na tabela.

Sintaxe:

```
RUNSTATS ON TABLE <schema.tablename>
```

Exemplo:

```
RUNSTATS ON TABLE myschema.employee
```

12.1.3 BIND / REBIND

Após executar com sucesso um comando RUNSTATS, nem todas as consultas usarão as estatísticas mais recentes. Os planos de acesso SQL estático são determinados ao executar pela primeira vez o comando BIND; portanto, as estatísticas usadas nesse momento podem não ser iguais às atuais. A *Figura 12.2* ajuda a ilustrar esta ideia.

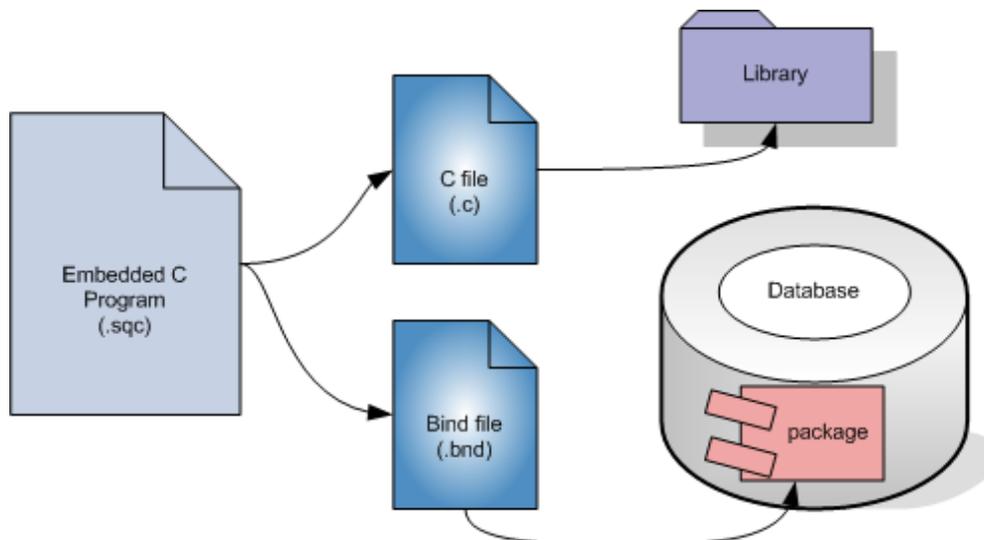


Figura 12.2 – Processo bind de SQL estático

Na *Figura 12.2*, um programa C integrado (armazenado como arquivo com extensão “.sqc”) é pré-compilado. Após a pré-compilação são gerados dois arquivos: um arquivo “.c” contendo o código em C com todo o SQL transformado em comentários, e um arquivo “.bnd” contendo todas as instruções SQL. O arquivo C com a extensão “.c” é compilado normalmente com um compilador C, criando uma “biblioteca” como mostrado no lado superior direito da figura. O arquivo “.bnd” é associado de um modo semelhante, gerando um pacote que é armazenado no banco de dados. Associar equivale a compilar as instruções SQL, onde o melhor plano de acesso é determinado com base nas estatísticas disponíveis no momento; em seguida, elas são armazenadas no pacote.

Agora, o que ocorre se forem inseridas 1 milhão de linhas em uma tabela usada no SQL deste programa C integrado? Após a inserção, se for executado um RUNSTATS, as estatísticas serão atualizadas. Entretanto, o pacote não será atualizado automaticamente para recalculer o caminho de acesso com base nas estatísticas mais recentes. O comando

`db2rbind` pode ser usado para associar novamente os pacotes existentes, para levar as últimas estatísticas em consideração.

Sintaxe:

```
db2rbind database_alias -l <logfile>
```

Exemplo:

Para reassociar todos os pacotes do banco de dados **SAMPLE** e armazenar o log de saída no arquivo `mylog.txt`, execute este comando:

```
db2rbind sample -l mylog.txt
```

12.1.4 Tarefas de manutenção do Control Center (Centro de Controle)

No Control Center, você pode executar REORG e RUNSTATS. A *Figura 12.3* mostra como fazer.

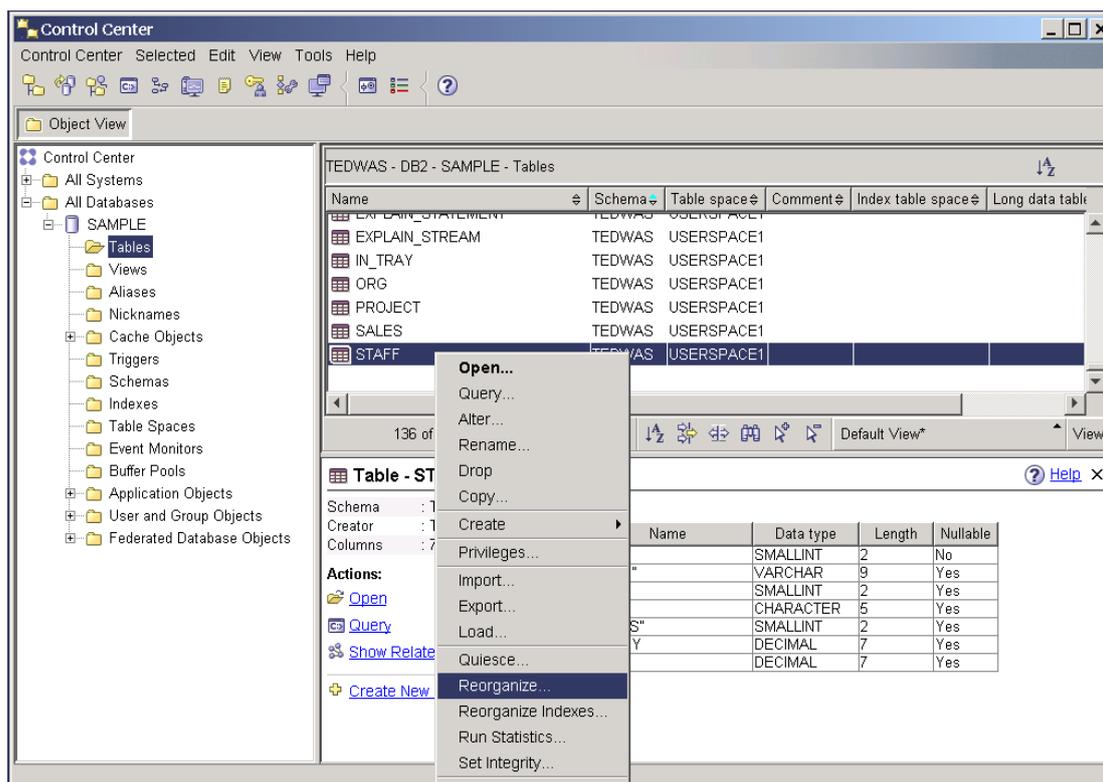


Figura 12.3 – REORG e RUNSTATS do Control Center

Selecione a tabela na qual deseja realizar as operações, clique com o botão direito na mesma e selecione Reorganize (para REORG) ou Run Statistics (para RUNSTATS).

12.1.4.1 A visualização operacional do banco de dados

Ao selecionar um banco de dados, a visualização operacional no lado inferior direito do Control Center fornecerá informações sobre o banco de dados, como seu tamanho, quando foi feito o último backup, se a manutenção automática está definida, etc. Esta visualização permite identificar rapidamente as necessidades de manutenção do banco de dados. A *Figura 12.4* mostra estas informações.

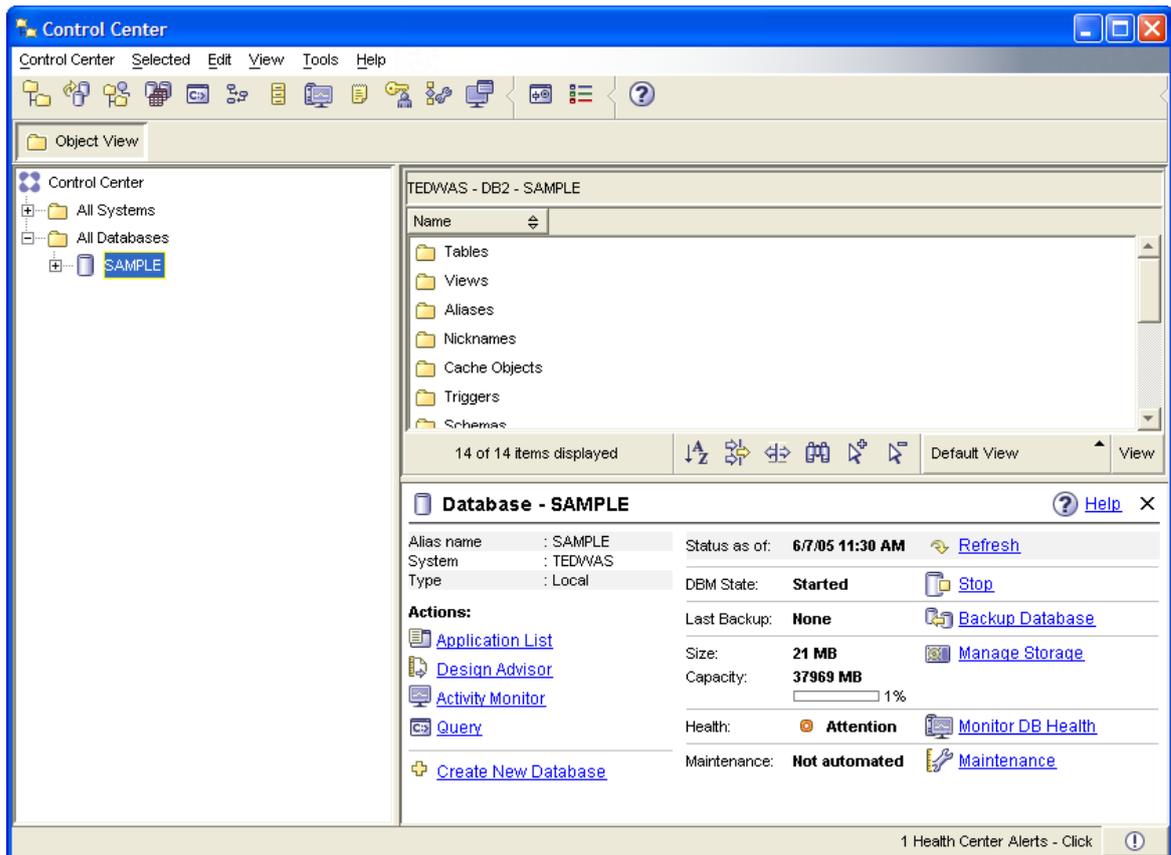


Figura 12.4 – A visualização operacional do banco de dados no Control Center

12.2 Opções de manutenção

Há três maneiras de realizar tarefas de manutenção:

- Manutenção manual

As atividades de manutenção são realizadas manualmente, quando surge a necessidade.

- Criar scripts para realizar manutenção

Você pode criar scripts com os comandos de manutenção e programá-los para serem executados regularmente.

- Manutenção automatizada

O DB2 cuida automaticamente da manutenção para você (REORG, RUNSTATS, BACKUP).

Nesta seção, nos concentramos na manutenção automática.

A manutenção automática consiste em:

- O usuário define uma *janela de manutenção* na qual as tarefas podem ser executadas com um mínimo de interrupções. Por exemplo, se o sistema tem sua menor atividade nos domingos de 2:00 às 4:00 da manhã, este horário funcionaria como janela de manutenção.
- Há duas janelas de manutenção: uma para operações on-line e outra para operações off-line.
- O DB2 executa automaticamente as operações de manutenção somente quanto necessário, durante as janelas de manutenção

No Control Center, inicie o assistente *Configure Automated Maintenance* (Configurar manutenção automática) como mostrado na *Figura 12.5*.

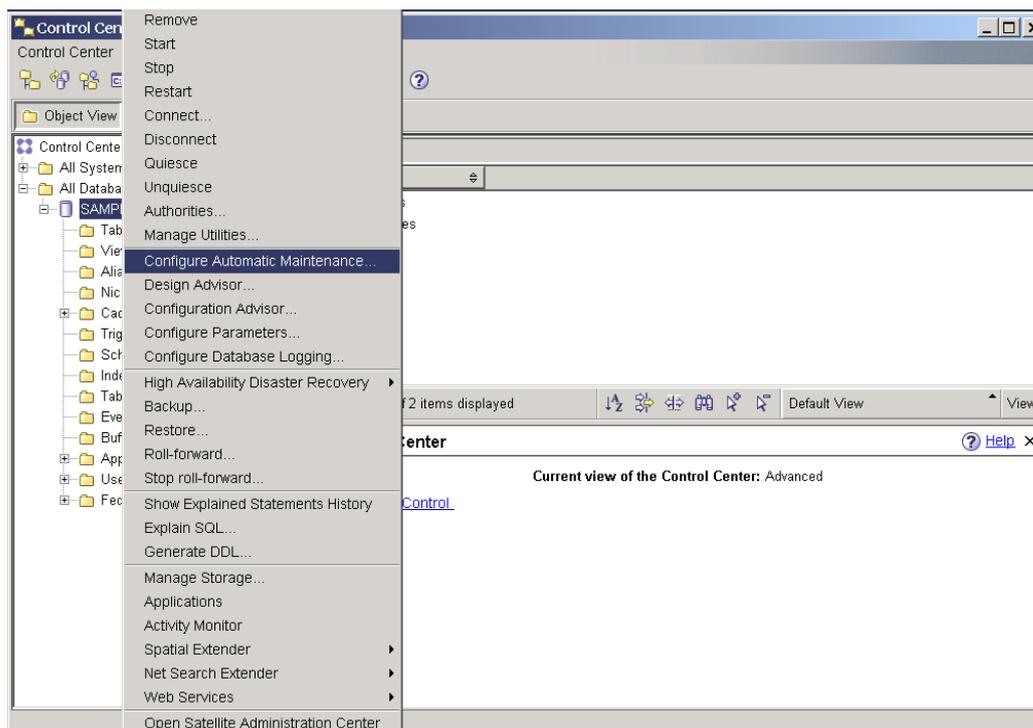


Figura 12.5 – Inicialização do assistente Configure Automated Maintenance

A *Figura 12.6* mostra o assistente *Configure Automated Maintenance*.

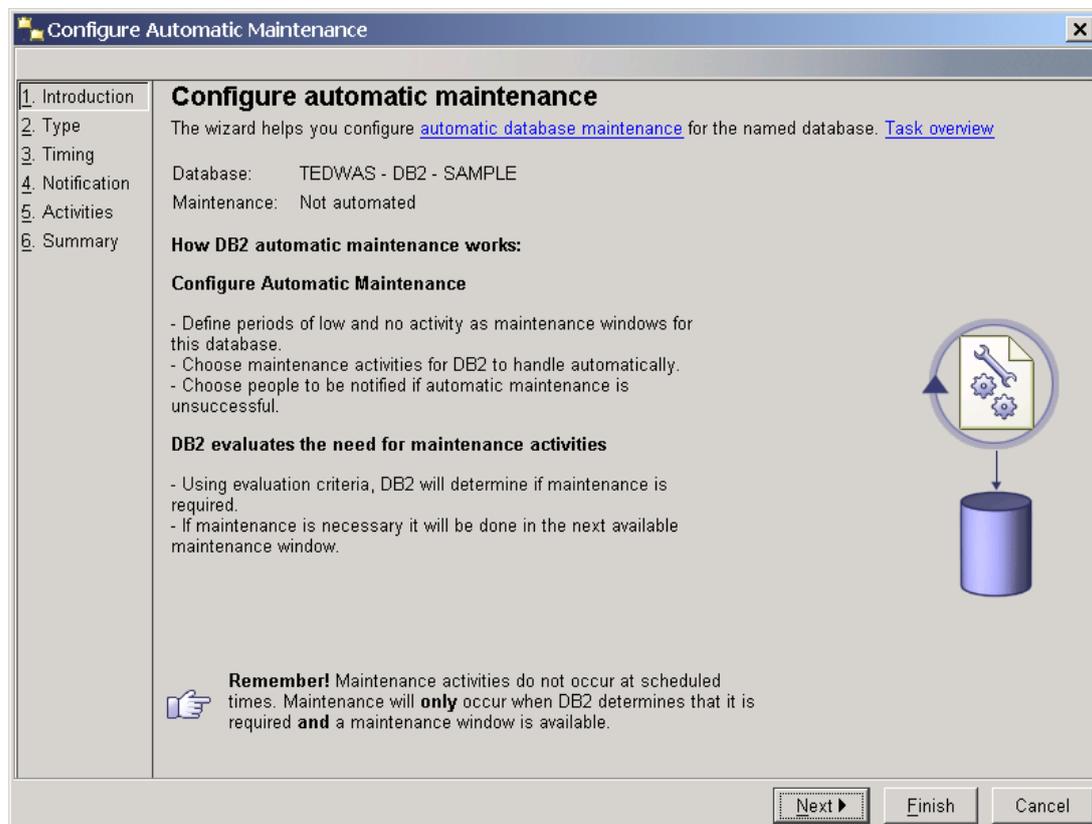


Figura 12.6 – O assistente Configure Automated Maintenance

12.3 Resumo

Este capítulo examinou a importância da manutenção em seus bancos de dados, incluindo a função do ciclo de REORG, RUNSTATS e REBIND. O comando REORG, como o nome indica, reorganiza seus dados para eliminar a fragmentação e acelerar a recuperação de dados. O comando RUNSTATS atualiza as informações estatísticas usadas pela ferramenta de otimização do DB2 para aumentar o desempenho dos dados. O processo BIND ou REBIND atualiza os pacotes do banco de dados com os caminhos de acesso mais recentes.

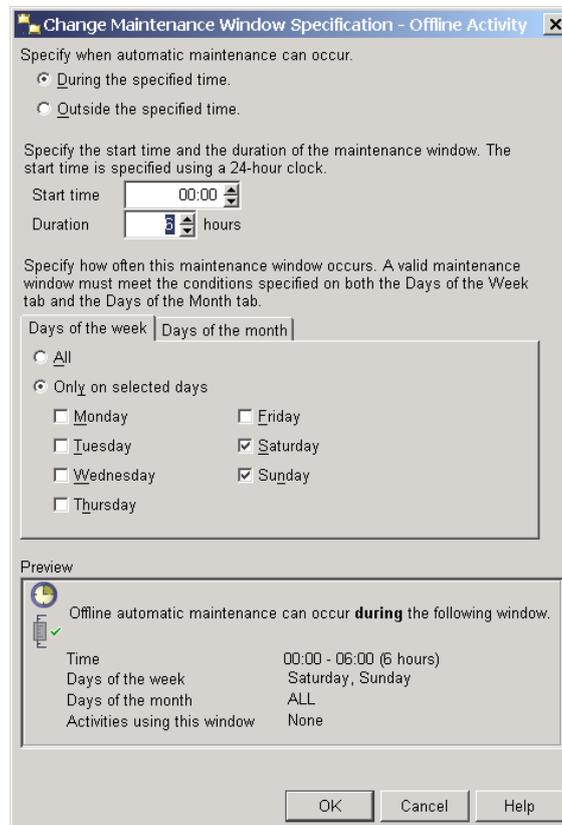
Também examinamos as ferramentas gráficas fornecidas no Control Center do DB2 para executar as atividades de manutenção nos modos manual, com script e automático.

12.4 Exercícios

Neste exercício, você irá configurar a manutenção automática no banco de dados **SAMPLE** do DB2.

Procedimento

1. Na árvore de objetos do Control Center, clique com o botão direito do mouse no banco de dados **SAMPLE** e selecione o item de menu *Configure Automatic Maintenance* (Configurar manutenção automática). Isto abrirá o assistente *Configure Automatic Maintenance*.
2. A página *Introduction* do assistente exibe a configuração atual da manutenção automatizada. Se você criou o banco de dados com a opção de manutenção automatizada, ela já está configurada. Você pode usar este assistente para reconfigurar as opções de manutenção automatizada. Clique no botão *Next* para passar para a próxima página do assistente.
3. A página *Type* (Tipo) do assistente solicita que você escolha entre desativar totalmente a manutenção automática ou alterar suas configurações de manutenção automatizada. Selecione a opção para alterar as configurações atuais de manutenção automatizada. Clique em *Next*.
4. A página *Timing* (Tempo) do assistente solicita que você especifique as janelas de manutenção. Configure o banco de dados para ficar off-line nas noites de sábado e domingo, da meia-noite até 06:00 AM, como mostrado abaixo. Clique no botão *Change* (Alterar) ao lado do painel de visualização da janela de manutenção off-line e selecione os horários desejados. Após especificar as informações necessárias, clique no botão *OK* para retornar ao assistente. Deixe a janela on-line sem alterações (a manutenção on-line pode ocorrer a qualquer momento). Clique no botão *Next* para continuar.



5. Na página *Notification* (Notificação) do assistente, configure um contato para o caso de falha em uma atividade de manutenção automatizada. Salte esta etapa no momento. Clique no botão *Next* para continuar.
6. Na página *Activities* (Atividades) do assistente, você pode selecionar a automatização individualmente ou optar por não automatizar atividades específicas, além de escolher se deseja ser notificado sobre atividades em particular. Neste exemplo, assegure que todas as caixas de *Automate* (Automatizar) estejam selecionadas e as caixas de *Notify* (Notificar) estejam desmarcadas. Clique no botão *Next* para continuar.
7. Antes de prosseguir para a próxima página do assistente, você deve configurar a localização de backup do banco de dados. O ideal é armazenar os backups em unidades físicas diferentes, para o caso de falha em disco. Da página *Activities*, selecione a opção *Backup database* (Fazer backup do banco de dados) e clique no botão *Configure Settings* (Definir configurações).
8. Na guia *Backup Criteria* (Critérios de backup) da janela *Configure Settings*, selecione a opção *Balance Database Recoverability with Performance* (Equilibrar capacidade de recuperação e desempenho do banco de dados). Na guia *Backup Location* (Local do backup), selecione o local do backup existente e clique no botão *Change* (Alterar). Especifique um local diferente para a realização do

backup (verifique se existe espaço suficiente na unidade). Na guia *Backup Mode* (Modo de backup), assegure que a opção *Offline Backup* (Backup off-line) esteja selecionada. Clique no botão *OK* para fechar a caixa de diálogo *Backup Criteria*. Clique no botão *Next* para continuar.

9. A página *Summary* do assistente *Configure Automated Maintenance* contém um resumo das seleções feitas. Clique no botão *Finish* (Concluir) para aceitar e implementar as alterações.

13

Capítulo 13 – Concorrência e bloqueios

Este capítulo explica como permitir que múltiplos usuários acessem um banco de dados ao mesmo tempo, sem que um interfira na ação do outro e mantendo os dados consistentes. Nós iremos mostrar os conceitos de transação, concorrência e bloqueio (locking).

Observação:

Para obter mais informações sobre concorrência e bloqueio, assista ao vídeo:
<http://www.channeldb2.com/video/video/show?id=807741:Video:4322>

13.1 Transações

Uma transação ou unidade de trabalho consiste em uma ou mais instruções SQL, as quais, quando executadas, podem ser consideradas como uma única unidade; o que significa que, se uma instrução da transação falhar, a transação inteira falha e todas as instruções que foram executadas antes do ponto de falha são revertidas (rolled back). A *Figura 13.1* fornece um exemplo de transação.

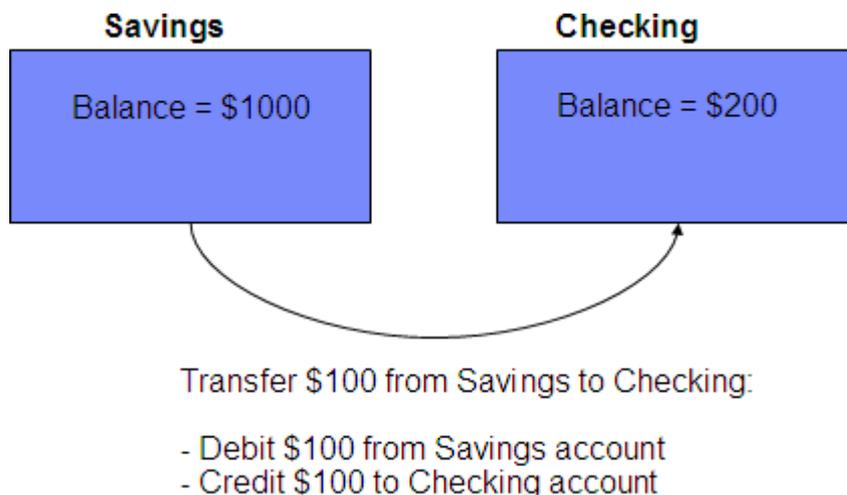


Figura 13.1 – Um exemplo de transação

Na *Figura 13.1*, por exemplo, você deseja transferir 100 dólares de sua conta de poupança para sua conta corrente. A seguinte sequência de eventos é necessária para que esta transação se complete:

- Debitar US\$100 da conta de poupança
- Creditar US\$100 na conta corrente

Se a sequência de eventos acima não fosse tratada como uma única unidade de trabalho, uma transação, imagine o que ocorreria se houvesse falta de luz após o débito na conta de poupança mas antes do crédito na conta corrente. Você perderia US\$100!

13.2 Concorrência

Podemos definir concorrência como o fato de que muitos usuários podem acessar, ao mesmo tempo. O DB2 foi projetado como um banco de dados multiusuário. O acesso aos dados deve ser coordenado adequadamente e com transparência, usando um mecanismo que assegure a integridade e a consistência dos dados. Considere a *Figura 13.2* como exemplo.

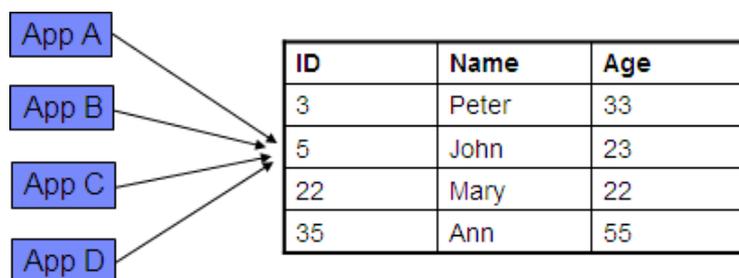


Figura 13.2 – Um exemplo de concorrência e da necessidade de controle de acesso aos dados

Na *Figura 13.2* há quatro aplicativos, App A, App B, App C e App D, tentando acessar a mesma linha (row 2) de uma tabela. Se não houver nenhum controle de concorrência, todos os aplicativos podem executar operações com a mesma linha. Supondo que todos os aplicativos estejam atualizando a coluna Age (Idade) da linha 2 com valores diferentes, o aplicativo que realizar a última atualização provavelmente será o “vencedor” nesta situação. Neste exemplo, fica óbvio que é necessário algum tipo de controle de concorrência para assegurar resultados consistentes. Este controle de concorrência é baseado no uso de bloqueios (locks).

Os conceitos de bloqueio e concorrência andam juntos. O bloqueio temporário impede que outros aplicativos executem suas operações até que a operação seja concluída. Quanto mais bloqueios houver em um sistema, menor a possibilidade de concorrência. Por outro lado, quando menos bloqueios houver em um sistema, maior a possibilidade de concorrência.

Os bloqueios são adquiridos automaticamente conforme a necessidade para suportar uma transação, e liberados quando a transação é encerrada (através do uso de um comando COMMIT ou ROLLBACK). É possível usar bloqueios nas tabelas ou nas linhas. Há dois tipos básicos de bloqueio:

- Bloqueios de compartilhamento (S locks) – adquiridos quando um aplicativo deseja ler e impedir que outros atualizem a mesma linha

- Bloqueios exclusivos (X locks) – adquiridos quando um aplicativo atualiza, insere ou exclui uma linha

Agora considere a *Figura 13.3*, similar à *Figura 13.2*, porém, neste caso, existe um bloqueio.

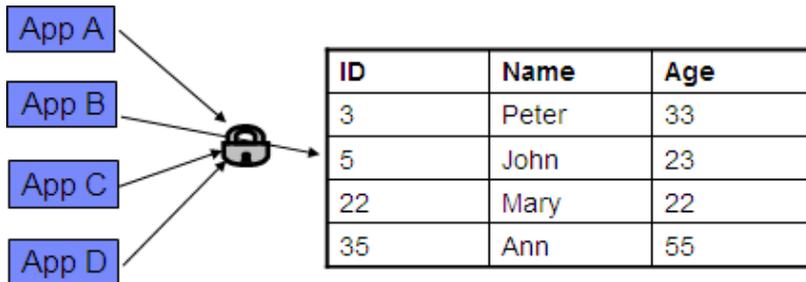


Figura 13.3 – Um exemplo de concorrência e da necessidade de bloqueios

Por exemplo, na *Figura 13.2*, se App B for o primeiro a acessar a linha 2 e executar um UPDATE, App B terá um X lock sobre a linha. Quando App A, App C e App D tentarem acessar a mesma linha, não serão capazes de executar um UPDATE devido ao X lock. Este controle permite manter a consistência e a integridade dos dados.

13.3 Problemas sem o controle de concorrência

Sem alguma forma de controle de concorrência, ocorreriam os seguintes problemas

- Perda de atualização
- Leitura sem alocação
- Leitura não reproduzível
- Leitura fantasma

13.3.1 Perda de atualização

A perda de atualização (UPDATE) é um problema semelhante ao explicado anteriormente nesta seção, no qual o aplicativo executando a última atualização (UPDATE) seria o “vencedor”.

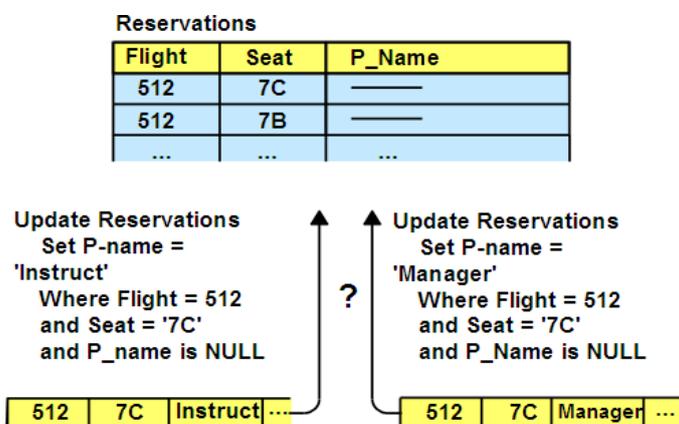


Figura 13.4 – Perda de atualização

Na *Figura 13.4* há dois aplicativos tentando atualizar a mesma linha. O aplicativo da esquerda é App1 e o aplicativo da direita é App2. Portanto, a sequência de eventos é:

1. App1 atualiza uma linha
2. App2 atualiza a mesma linha
3. App1 executa o COMMIT
4. App2 executa o COMMIT

A atualização de App1 'é perdida quando o App2 faz sua atualização, daí a expressão "Perda de atualização".

13.3.2 Leitura sem alocação

Uma leitura sem alocação ou "leitura suja" permite que um aplicativo leia informações que não foram "salvas" pelo comando COMMIT e, portanto, não são necessariamente precisas.

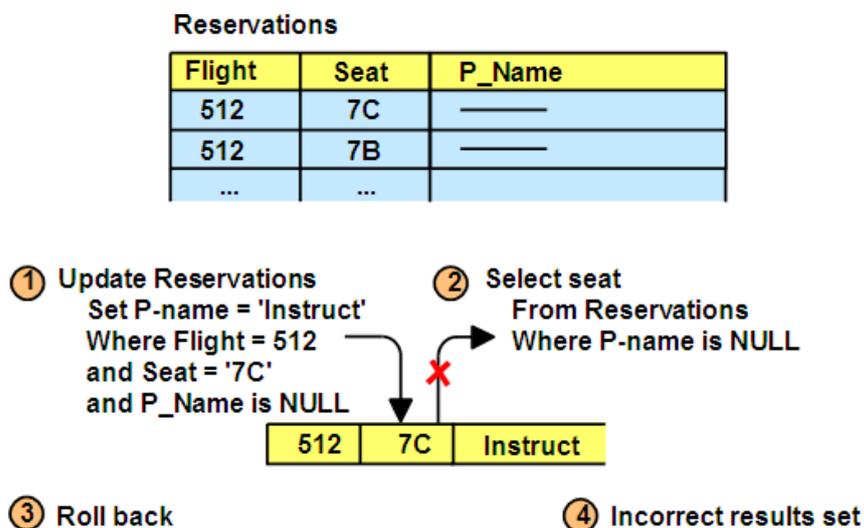


Figura 13.5 – Leitura sem alocação

A *Figura 13.5* segue esta sequência de eventos:

1. App1 atualiza uma linha
2. App2 lê o novo valor dessa linha
3. App1 reverte suas alterações na linha (ROLLBACK)

App2 está lendo dados não salvos e, portanto, inválidos, que é o motivo deste problema ser chamado de “leitura sem alocação”

13.3.3 Leitura não reproduzível

Uma leitura não reproduzível indica que você não pode obter o mesmo resultado após executar a mesma operação de leitura.

FLIGHT	SEAT	NAME	DESTINATION	ORIGIN
512	7B	—	DENVER	DALLAS
....				
....				
814	8A	—	SAN JOSE	DENVER
....				
134	1C	—	HONOLULU	SAN JOSE
....			

Figura 13.6 – Leitura não reproduzível

Na *Figura 13.6*, considere se você estiver tentando reservar passagens em um voo de Dallas para Honolulu. A sequência de eventos é:

1. App1 abre um cursor (também conhecido como conjunto de resultados) obtendo o que é visto na *Figura 13.6*
2. App2 exclui um linha que estava qualificada para o cursor (por exemplo, a linha com destino “San Jose”)
3. App2 aloca as alterações
4. App1 fecha e reabre o cursor

Neste caso, como App1 não receberia os mesmos dados em uma leitura repetida, ele não pode reproduzir o conjunto de dados; é por isto que o problema é chamado de “leitura não reproduzível”.

13.3.4 Leitura fantasma

O problema da leitura fantasma é semelhante ao da leitura não reproduzível, mas a diferença é que nos acessos subsequentes, você poderá obter linhas adicionais em vez de menos linhas. A *Figura 13.7* fornece um exemplo deste problema.

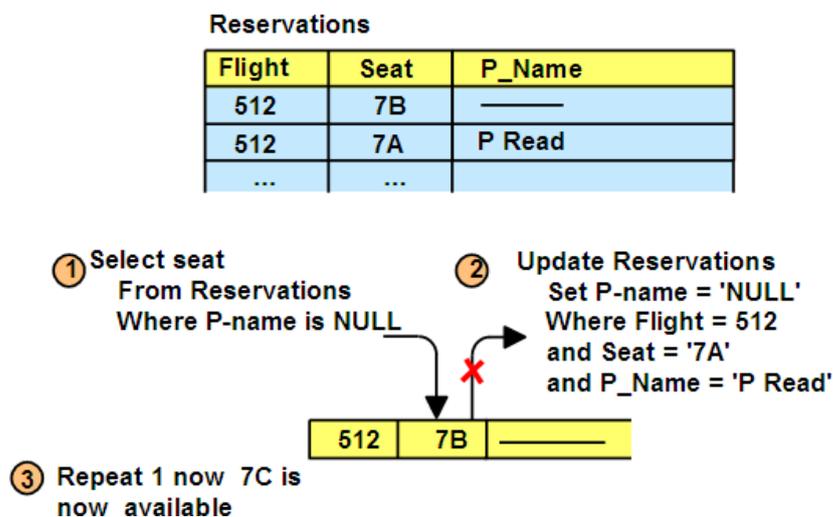


Figura 13.7 – Leitura fantasma

A *Figura 13.7* mostra a seguinte sequência de eventos:

1. App1 abre um cursor
2. App2 adiciona uma linha ao banco de dados, que se qualificaria para o cursor
3. App2 aloca as alterações
4. App1 fecha e reabre o cursor

Neste caso, App1 não receberia os mesmos dados em uma leitura repetida pois receberia mais linhas: é por isto que o problema é chamado de “leitura fantasma”.

13.4 Níveis de isolamento

Você pode pensar nos níveis de isolamento como políticas de bloqueio onde, dependendo do nível de isolamento escolhido, você pode obter comportamentos diferentes de bloqueio do banco de dados com um aplicativo.

O DB2 fornece diferentes níveis de proteção para isolar os dados:

- Uncommitted Read (UR, Leitura sem alocação)
- Cursor Stability (CS, Estabilidade do cursor)
- Read Stability (RS, Estabilidade da leitura)
- Repeatable Read (RR, Leitura reproduzível)

13.4.1 Uncommitted read (Leitura sem alocação)

A leitura sem alocação também é conhecida como leitura suja. Este é o nível de isolamento mais baixo e fornece o mais alto grau de concorrência. Não são obtidos bloqueios de linha nas operações de leitura a menos que outro aplicativo tente descartar ou alterar uma tabela; e as operações de atualização agem como se estivessem usando o nível de isolamento de estabilidade do cursor.

Os problemas que ainda são possíveis com este nível de isolamento:

- Leitura sem alocação
- Leitura não reproduzível
- Leitura fantasma

Os problemas evitados com este nível de isolamento:

- Perda de atualização

13.4.2 Cursor stability (Estabilidade do cursor)

A estabilidade do cursor é o nível de isolamento padrão. Ela fornece um grau de bloqueio mínimo. Basicamente com este nível de isolamento, a linha “atual” de um cursor é bloqueada. Se a linha for somente para leitura, o bloqueio é mantido até ser acessada uma nova linha ou a unidade de trabalho ser encerrada. Se a linha for atualizada, o bloqueio é mantido até a unidade de trabalho ser encerrada.

Os problemas que ainda são possíveis com este nível de isolamento:

- Leitura não reproduzível
- Leitura fantasma

Os problemas evitados com este nível de isolamento:

- Perda de atualização
- Leitura sem alocação



13.4.2.1 Currently committed (Atualmente comprometido)

Antes do DB2 9.7, ao usar o nível de isolamento cursor stability, um gravador (operação UPDATE) impediria que um leitor (operação SELECT) acessasse a mesma linha. A lógica era a de que, uma vez que o gravador está fazendo alterações na linha, o leitor deve aguardar até a atualização ser concluída, para ver o valor final alocado. No DB2 9.7, há um novo comportamento padrão para o nível de isolamento cursor stability nos novos bancos de dados. Este novo comportamento é implementado usando a semântica **currently committed (CC)** (Atualmente comprometido). Com o CC, um gravador não impede que um leitor acesse a mesma linha. Este comportamento era possível no passado caso se usasse o nível de isolamento uncommitted read (UR); entretanto, a diferença agora é que no UR, o leitor recupera o valor **sem alocação**, enquanto com o CC, o leitor recupera o valor **atualmente comprometido**. O valor atualmente comprometido é o valor salvo antes do início da operação de gravação.

Por exemplo, se você tiver a tabela T1 com o conteúdo abaixo:

FIRSTNAME	LASTNAME
Raul	Chong
Jin	Xie

Agora, seu aplicativo AppA executa a instrução abaixo sem alocar:

```
update T1 set lastname = 'Smith' where firstname = 'Raul'
```

Em seguida, o aplicativo AppB executa esta instrução:

```
select lastname from T1 where firstname = 'Raul' with CS
```

Antes do DB2 9.7, esta instrução ficaria em espera, aguardando que o bloqueio exclusivo obtido pela instrução de atualização de AppA (o gravador) fosse liberado.

No DB2 9.7 e com `currently committed` ativado (o padrão dos novos bancos de dados), a instrução retornaria o valor atualmente comprometido, que é *Chong*.

Observe que, mesmo que o padrão seja CS, estamos incluindo 'with CS' na instrução para manter a clareza. Discutiremos esta cláusula mais adiante no capítulo.

Se AppB tentar executar esta instrução:

```
select lastname from T1 where firstname = 'Raul' with UR
```

Como é usado o isolamento UR, o resultado seria *Smith*, que é o valor não comprometido.

Este exemplo mostra que existe mais concorrência para os aplicativos com o uso de CC, permitindo que os leitores acessem a linha que está sendo atualizada pelo gravador.

Outro cenário que poderia causar conflito antes do DB2 9.7 é um leitor impedindo o acesso de um gravador a uma linha. Este cenário foi um dos motivos pelo qual o comando COMMIT foi recomendado mesmo em operações de leitura, pois assegura que os bloqueios de compartilhamento (S) sejam liberados. Com o CC isto deixou de ser um problema, pois um leitor não bloqueia um gravador.

Quanto às operações de INSERT não comprometidas, a operação de leitura irá saltá-las como padrão, ou seja, o conjunto de resultados não exibe estas linhas. No caso dos comandos DELETE, a operação de leitura também deve saltar (ignorar) as linhas afetadas, mas o comportamento depende do valor da variável de registro do DB2 DB2_SKIPDELETED. Outras variáveis de registro e propriedades dos comandos BIND e PREPARE podem alterar o comportamento padrão do CC.

Lembre-se: `Currently committed` significa que serão exibidas somente as informações atualmente comprometidas; portanto, as operações de INSERT ou DELETE sem comprometimento serão ignoradas.

Como mencionado anteriormente, como padrão o CC está ativado nos novos bancos de dados. Se desejar desativá-lo ou ativá-lo para um banco de dados criado antes do DB2 9.7 e atualizado para o DB2 9.7, você pode atualizar o valor de configuração do banco de dados para CUR_COMMIT. Por exemplo, para desativá-lo para o comando abaixo, executado no banco de dados **SAMPLE**:

```
db2 update db cfg for sample using CUR_COMMIT off
db2stop
db2start
```

13.4.3 Estabilidade de leitura

Com a estabilidade de leitura, todas as linhas recuperadas por um aplicativo dentro de uma unidade de trabalho são bloqueadas. Em um dado cursor, são bloqueadas todas as linhas qualificadas para o conjunto de resultados. Por exemplo, se você possui uma tabela contendo 10.000 linhas e a consulta retorna 10 linhas, somente estas 10 linhas estão bloqueadas. A estabilidade de leitura usa um grau de bloqueio moderado.

Os problemas que ainda são possíveis com este nível de isolamento:

- Leitura fantasma

Os problemas evitados com este nível de isolamento:

- Perda de atualização
- Leitura sem alocação
- Leitura não reproduzível

13.4.4 Leitura reproduzível

A leitura reproduzível é o nível de isolamento mais alto. Ela fornece o grau mais elevado de bloqueio e menor concorrência. Os bloqueios são mantidos em todas as linhas processadas para construir o conjunto de resultados, ou seja, podem ser bloqueadas linhas que não estão necessariamente no resultado final. Nenhum outro aplicativo poderá atualizar, excluir ou inserir uma linha que afete o conjunto de resultados, até a unidade de trabalho ser concluída. A leitura reproduzível assegura que a mesma consulta executada por um aplicativo mais de uma vez em uma unidade de trabalho produza o mesmo resultado todas as vezes.

Os problemas que ainda são possíveis com este nível de isolamento:

- nenhum

Os problemas evitados com este nível de isolamento:

- Perda de atualização
- Leitura sem alocação
- Leitura não reproduzível
- Leitura fantasma

13.4.5 Comparação de níveis de isolamento

A *Figura 13.8* compara os diferentes níveis de isolamento para um acesso. Na figura, vemos que o nível de isolamento uncommitted read (UR) não faz bloqueios. O nível de isolamento cursor stability (CS) bloqueia a linha 1 ao acessá-la, mas libera logo que acessa a linha 2 e assim por diante. Nos níveis de isolamento read stability (RS) ou repeatable read (RR), qualquer linha acessada será bloqueada, e o bloqueio não é liberado até o final da transação (em um ponto de comprometimento).

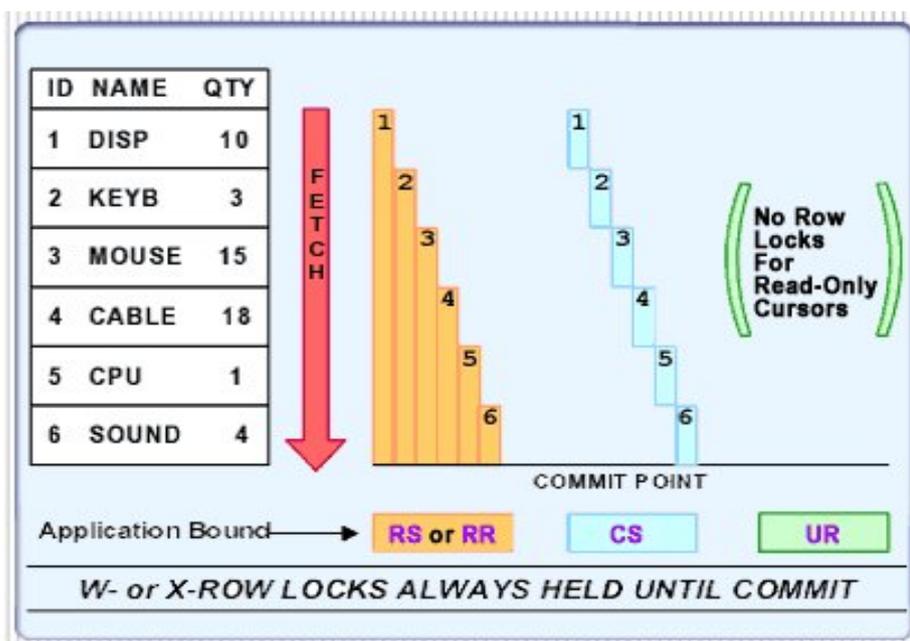


Figura 13.8 – Comparação dos níveis de isolamento de um acesso

13.4.6 Configuração do nível de isolamento

Os níveis de isolamento podem ser especificados em diversos níveis:

- Sessão (aplicativo)
- Conexão
- Instrução

O nível de isolamento normalmente é definido no nível de sessão ou de *Aplicativo*. Se não for especificado nenhum nível de isolamento no aplicativo, ele usa cursor stability como padrão. Por exemplo, a *Tabela 13.1* mostra os níveis de isolamento possíveis para um programa .NET ou JDBC e como essas propriedades, quando configuradas, correspondem a um nível de isolamento do DB2.

DB2

.NET

JDBC

Uncommitted Read (UR, Leitura sem alocação)	ReadUncommitted	TRANSACTION_READ_UNCOMMITTED
Cursor Stability (CS, Estabilidade do cursor)	ReadCommitted	TRANSACTION_READ_COMMITTED
Read Stability (RS, Estabilidade da leitura)	RepeatableRead	TRANSACTION_REPEATABLE_READ
Repeatable Read (RR, Leitura reproduzível)	Serializable	TRANSACTION_SERIALIZABLE

Tabela 13.1 – Comparação da terminologia de nível de isolamento

O nível de isolamento de instrução pode ser definido usando a cláusula WITH {nível de isolamento}. Por exemplo:

```
SELECT ... WITH {UR | CS | RS | RR}
```

Cenário de exemplo:

Um aplicativo precisa realizar uma contagem “aproximada” de quantas linhas há em uma tabela. O desempenho é fundamental. O nível de isolamento cursor stability é necessário, com exceção de uma instrução SQL:

```
SELECT COUNT(*) FROM tabl WITH UR
```

Para o SQL incorporado, o nível é definido no momento da associação; no SQL dinâmico, o nível é definido na execução.

A seleção do nível de isolamento a ser usado depende do aplicativo. Se o aplicativo não necessita de contagens exatas como no exemplo acima, selecione isolamento UR. Se o aplicativo necessita de um controle muito rígido sobre os dados com os quais trabalha, selecione o isolamento RR.



Para usar a semântica currently committed no momento da associação ou preparação, use esta sintaxe:

BIND:

```
>--+-----+----->
  '--CONCURRENTACCESSRESOLUTION-----USE CURRENTLY COMMITTED-----'
                                     '--WAIT FOR OUTCOME-----'
```

PREPARE:

```
concurrent-access-resolution:
|--+USE CURRENTLY COMMITTED+-----|
  '-WAIT FOR OUTCOME-----'
```

Em um aplicativo JDBC usando o IBM Data Server Driver para JDBC e SQLJ, você pode usar a propriedade concurrentAccessResolution para ativar o nível currently committed.

13.5 Ampliação de bloqueio

Cada bloqueio realizado pelo DB2 consome alguma memória. A ampliação de bloqueio ocorre quando o Optimizer acha que é melhor ter um bloqueio na tabela inteira em vez de diversos bloqueios nas linhas. A *Figura 13.9* ilustra isso.

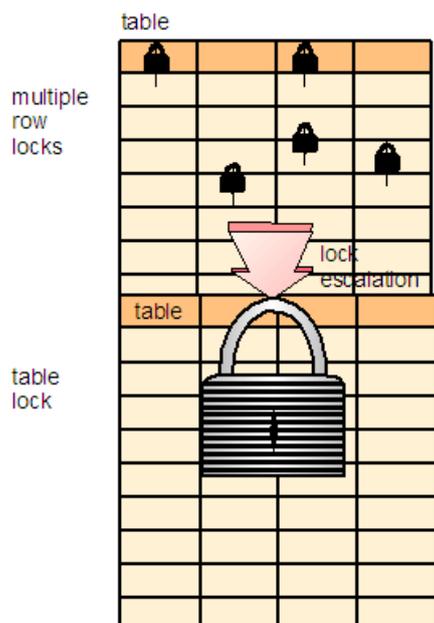


Figura 13.9 – Ampliação de bloqueio

Há dois parâmetros principais de configuração de banco de dados relacionados à ampliação de bloqueio:

- **LOCKLIST** – A quantidade de memória (em páginas de 4k) reservada para gerenciar os bloqueios em todos os aplicativos conectados.
- **MAXLOCKS** – O percentual máximo da lista de bloqueio inteira que um único aplicativo pode consumir. .

O padrão de ambos os parâmetros é AUTOMATIC, o que significa que o gerenciador de memória auto-ajustável (STMM) modificará o tamanho. Se você não ativar o STMM e definir o valor, estes valores afetarão o momento no qual ocorrerá a ampliação de bloqueio. Por exemplo, se configurar LOCKLIST para 200K e MAXLOCKS para 22%, a ampliação de bloqueio ocorrerá quando um único aplicativo exigir mais de 44K de memória de bloqueio ($200\text{ K} * 22\% = 44\text{K}$). Se a ampliação de bloqueio ocorrer frequentemente com esta configuração, aumente o valor de LOCKLIST e MAXLOCKS. A ampliação de bloqueio não é boa para o desempenho, pois reduz a concorrência. O arquivo de log de diagnóstico do DB2 (`db2diag.log`) pode ser usado para determinar se está havendo ampliação de bloqueio. Consulte o Apêndice A para saber mais sobre este arquivo.

13.6 Monitoramento de bloqueio

Você pode monitorar a ocorrência de bloqueios, utilizando a aplicação DB2 lock snapshots. Para ativar os instantâneos dos bloqueios, execute este comando:

```
UPDATE MONITOR SWITCHES USING LOCK ON
```

Através deste comando, a monitoração estará ativa e os dados serão coletados. Para obter um relatório de bloqueios ocorridos em um determinado período, utilize o comando:

```
GET SNAPSHOT FOR LOCKS FOR APPLICATION AGENTID <handle>
```

A *Figura 13.9* mostra um exemplo da saída da aplicação lock snapshot.

```

Application Lock Snapshot

Snapshot timestamp                = 11-05-2002
00:09:08.672586

Application handle                 = 9
Application ID                     = *LOCAL.DB2.00B9C5050843
Sequence number                   = 0001
Application name                   = db2bp.exe
Authorization ID                   = ADMINISTRATOR
Application status                 = UOW Waiting
Status change time                = Not Collected
Application code page              = 1252
Locks held                         = 4
Total wait time (ms)              = 0

List Of Locks
Lock Name                         = 0x0500070004800100000000000052
Lock Attributes                    = 0x00000000
Release Flags                      = 0x40000000
Lock Count                         = 255
Hold Count                         = 0
Lock Object Name                   = 98308
Object Type                        = Row
Tablespace Name                    = TEST4K
Table Schema                       = ADMINISTRATOR
    
```

Figura 13.9 – Instantâneo de bloqueio de aplicativo

new in
V9.7

Observação:

No DB2 9.7, um esforço está sendo feito para mover o monitoramento do banco de dados para fora do monitor do sistema e da tecnologia de snapshots para o uso de queries acessando diretamente a memória interna pelo uso de "function tables", para gerenciamento de carga de trabalho, bem como as ferramentas do IBM Data Studio. Consulte a documentação oficial do DB2 para obter mais informações.

13.7 Espera de bloqueio

Quando dois ou mais aplicativos precisam executar uma operação no mesmo objeto, um deles pode ter que aguardar para obter o bloqueio necessário. Como padrão, um aplicativo aguarda indefinidamente. O tempo que um aplicativo aguarda um bloqueio é controlado pelo parâmetro LOCKTIMEOUT de configuração do banco de dados. O valor padrão deste parâmetro é -1 (espera infinita).

O registro CURRENT LOCK TIMEOUT pode ser usado para definir a espera de bloqueio para uma dada conexão. Como padrão, este registro é configurado com o valor de LOCKTIMEOUT. Use a instrução SET LOCK TIMEOUT para alterar seu valor. Após definir o valor desse registro para uma conexão, ele persistirá em todas transações.

Exemplo:

```
SET LOCK TIMEOUT=WAIT n
```

13.8 Causas e detecção de impasse de bloqueios

Um impasse de bloqueios (deadlock) ocorre quando dois ou mais aplicativos conectados ao mesmo banco de dados aguardam um recurso indefinidamente. A espera nunca é resolvida, porque cada aplicativo está retendo um recurso necessário ao outro. Na maioria das vezes, os impasses de bloqueios são um problema de projeto de aplicativo. A *Figura 13.10* ilustra um impasse de bloqueios.

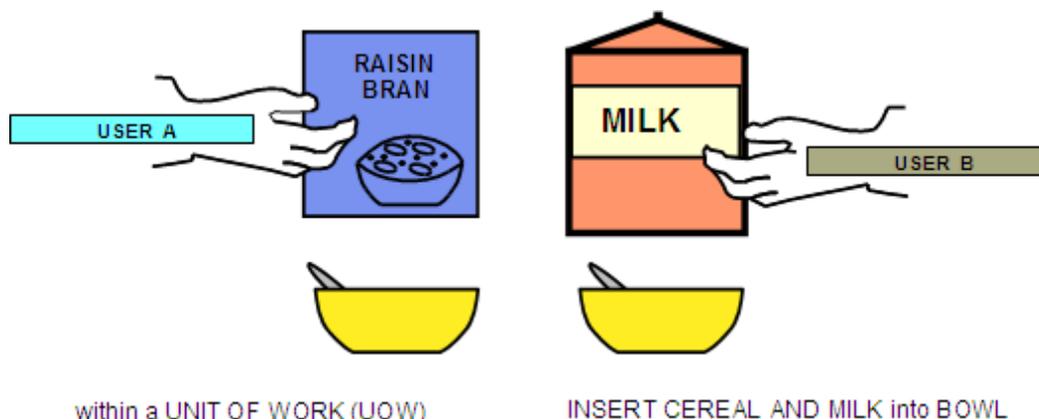


Figura 13.10 – Cenário de impasse de bloqueios

Na *Figura 13.10*, o usuário A está segurando o cereal e não o soltará até conseguir o leite. Por outro lado, o usuário B está segurando o leite e não o soltará até conseguir o cereal. Assim, temos uma situação de impasse de bloqueios.

new in
V9.7

Com o DB2 9.7, o uso do isolamento `currently committed` reduziu consideravelmente a ocorrência de impasses de bloqueios, pois um aplicativo não precisa aguardar o outro aplicativo liberar seu bloqueio; em vez disso, ele acessa o valor `currently committed`.

Para simular uma situação de impasse de bloqueio no DB2, siga estas etapas:

1. Desative o `currently committed`:

```
db2 update db cfg for sample using cur_commit off
db2stop force
db2start
```

2. Abra duas Command Windows do DB2 (que chamaremos respectivamente “CLP1” e “CLP2”), representando dois aplicativos diferentes conectando-se ao banco de dados

3. Em CLP1, execute os comandos:

```
db2 connect to sample
db2 +c update employee set firstnme = 'Mary' where empno = '000050'
```

Primeiro, estamos nos conectando ao banco de dados `SAMPLE` e, em seguida, executando uma instrução de atualização (`UPDATE`) na linha com “`empno = 000050`” na tabela `employee`. A opção “`+c`” na instrução indica que não desejamos que a Command Window do DB2 não execute o `COMMIT` automaticamente. Estamos fazendo isso propositalmente para reter os bloqueios.

4. Em CLP2, execute os comandos:

```
db2 connect to sample
db2 +c update employee set firstnme = 'Tom' where empno = '000030'
```

Na janela CLP2, que representa o segundo aplicativo, também estamos nos conectando ao banco de dados `SAMPLE`, mas atualizando outra linha na tabela `employee`.

5. Em CLP1, execute:

```
db2 +c select firstnme from employee where empno = '000030'
```

Após pressionar `Enter` para executar a instrução `SELECT` acima, pode parecer que o `SELECT` ficou parado. Na verdade não está parado, mas aguardando a liberação do bloqueio exclusivo que foi colocado por CLP2 neste linha, na etapa 4. Neste ponto, se `LOCKTIMEOUT` ainda tiver o valor padrão `-1`, o aplicativo CLP1 ficará aguardando eternamente.

6. Em CLP2, execute:

```
db2 +c select firstnme from employee where empno = '000050'
```

Ao executar a instrução SELECT acima, agora estamos criando um impasse de bloqueios. Esta instrução SELECT também parecerá parada, pois está aguardando a liberação do bloqueio exclusivo colocado por CLP1 neste linha, na etapa 3.

No cenário de impasse de bloqueio acima, o DB2 irá verificar o parâmetro de configuração de banco de dados DLCHKTIME. Este parâmetro define o intervalo de tempo da verificação de impasses de bloqueios. Por exemplo, se este parâmetro estiver definido para 10 segundos, o DB2 irá verificar se ocorreu um impasse de bloqueios a cada 10 segundos. Se realmente ocorreu o impasse, o DB2 usará um algoritmo interno para determinar qual das duas transações deve ser revertida e qual deverá continuar.

Se você estiver presenciando diversos impasses de bloqueios, reexamine as transações existentes e verifique se é possível reestruturá-las.

13.9 Melhores práticas de concorrência e bloqueio

Você encontrará abaixo algumas dicas a serem seguidas para permitir a melhor concorrência possível:

1. Procure ativar o nível currently committed (CC), caso a lógica do seu aplicativo permita.
2. Mantenha as transações o mais curtas possível. Isto pode ser obtido através da execução de instruções COMMIT frequentes (mesmo em transações somente de leitura), quando a lógica do seu aplicativo permitir.
3. Gere um log das informações da transação somente quando necessário.
4. Limpe os dados rapidamente. Você pode executar o comando

```
ALTER TABLE ACTIVATE NOT LOGGED INITIALLY WITH EMPTY TABLE
```

ou agora com o DB2 9.7, use o comando TRUNCATE:

```
TRUNCATE <table name>
```

5. Execute as modificações de dados em lotes/grupos. Por exemplo:

```
DELETE FROM (
  SELECT * FROM tedwas.t1 WHERE c1 = ... FETCH FIRST 3000 ROWS ONLY)
```

6. Use os recursos de concorrência presentes nas ferramentas de movimentação de dados do DB2.
7. Configure o parâmetro de nível de banco de dados LOCKTIMEOUT (os tempos sugeridos estão entre 30 e 120 segundos). Não o deixe no padrão, que é -1. Você também pode usar o tempo de espera de bloqueio baseado em sessão.



8. Não recupere mais dados do que é necessário. Por exemplo, use a cláusula `FETCH FIRST n ROWS ONLY` nas instruções `SELECT`.

Observação:

Para obter mais informações sobre as melhores práticas de concorrência e bloqueio, veja os documentos sobre Melhores práticas disponíveis em <http://www.ibm.com/developerworks/data/bestpractices/>

13.10 Resumo

Neste capítulo examinamos a manutenção da integridade dos dados através do controle de transações, acesso simultâneo de usuários e níveis de bloqueio. Todos os níveis diferentes de concorrência têm questões que podem afetar como você acessa e gerencia seus dados.

Também analisamos em detalhes a função de configurar os níveis de isolamento para lidar com essas questões, e como o nível de isolamento pode ser manipulado para fornecer a flexibilidade máxima necessária ao seu aplicativo e necessidades de dados.

Também vimos a ampliação de bloqueio, as esperas de bloqueio e o monitoramento de bloqueio, juntamente com as causas, detecção e solução dos impasses de bloqueio do banco de dados.

Finalmente, examinamos algumas ideias de melhores práticas para obter o melhor resultado possível para suas necessidades de concorrência.

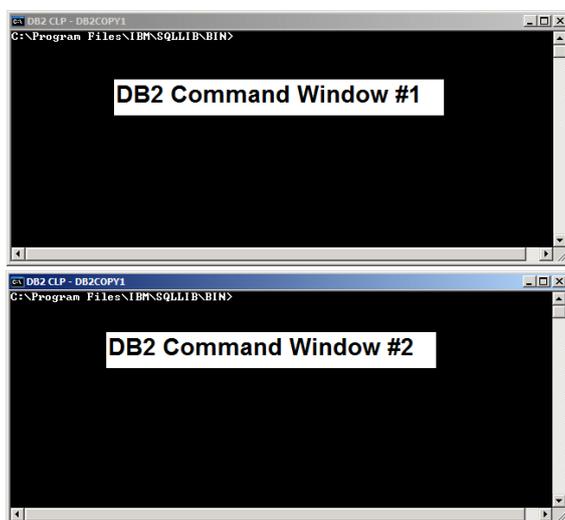
13.11 Exercícios

Neste exercício, você irá praticar os conceitos de concorrência e bloqueio discutidos neste capítulo, usando a Janela de comando do DB2. Esta ferramenta usa o nível de isolamento de bloqueio CS, como padrão. Após executar uma instrução SQL, a Janela de comando executará automaticamente o comando `commit` (este também é conhecido como `autocommit`). Para fins de ilustração neste exercício, usaremos o sinalizador `+c` para desativar o `autocommit`, e a cláusula `WITH <isolation level>` após algumas instruções SQL para sobrescrever o isolamento CS padrão.

Parte 1: Teste do problema de leitura fantasma e isolamento RR

Procedimento:

1. Abra duas Janelas de comando do DB2 como mostrado na figura abaixo. Chamaremos a janela do alto de “DB2 Command Window #1” e a janela inferior de “DB2 Command Window #2”



2. Na Command Window #1 do DB2, execute:

```
db2 connect to sample  
db2 +c select * from staff
```

Isto retornará 35 registros
3. Na Command Window #2 do DB2, execute:

```
db2 connect to sample  
db2 +c insert into staff (id,name) values (400, 'test')
```
4. Na Command Window #1 do DB2, execute:

```
db2 +c select * from staff
```

Isto deve retornar ainda 35 registros
5. Na Command Window #2 do DB2, execute:

```
db2 commit
```
6. Na Command Window #1 do DB2, execute:

```
db2 +c select * from staff
```

Agora, este retorna 36 registros!

A DB2 Command Window #1 representa um aplicativo que abre um cursor ou conjunto de resultados (`select * from staff`) obtendo 35 registros. Dentro da mesma transação (porque não estamos executando nenhuma instrução `commit` nesta janela), o aplicativo abre o mesmo cursor e ainda vê 35 registros, mesmo pós o aplicativo da DB2 Command Window #2 inserir (sem executar `commit`) um novo registro.

Em seguida, o aplicativo do DB2 na Command Window #2 executa um commit em relação ao inserido, e portanto na terceira vez em que o aplicativo do DB2 na Command Window #1 abre o cursor, o conjunto de resultados retorna mais uma linha (uma leitura fantasma), obtendo 36 registros. Este exemplo ilustra o problema da leitura fantasma: Dentro da mesma transação, a abertura do mesmo cursor retorna mais linhas. Estamos usando um nível de isolamento CS e, como mencionado anteriormente neste capítulo, o CS não impede as leituras fantasma.

7. Limpe o registro inserido antes de prosseguir para as próximas etapas:

Na Command Window #1 do DB2:

```
db2 rollback
```

Na Command Window #2 do DB2:

```
db2 delete from staff where id = 400
```

```
db2 select * from staff
```

Isto retorna 35 registros novamente.

8. Agora, vamos ver se um nível de isolamento RR pode evitar este problema de leitura fantasma.

Na Command Window #1 do DB2, execute:

```
db2 connect to sample
```

```
db2 +c select * from staff with RR
```

Isto retorna 35 registros

Na Command Window #2 do DB2:

```
db2 connect to sample
```

```
db2 +c insert into staff (id,name) values (400, 'test')
```

Esta instrução ficará parada, o que é esperado.

Como a cláusula WITH RR foi adicionada à instrução SELECT da Command Window #1 do DB2, este nível de isolamento impede um INSERT em uma linha que alteraria o conjunto de resultados. Este exemplo ilustra que o nível de isolamento RR realmente evita problemas de leitura fantasma.

9. Limpe, antes de seguir para a parte 2 deste exercício:

Na Command Window #2 do DB2:

```
Ctrl-C (to interrupt)
```

Feche a janela

Na Command Window #1 do DB2:

```
db2 rollback
```

Feche a janela

Parte 2 – Teste dos níveis currently committed (CC) e uncommitted read (UR)**Procedimento 1: Analisar o comportamento do isolamento CS sem o currently committed**

1. Abra uma Command Window do DB2 e execute as instruções abaixo:

```
db2 connect to sample
db2 select * from staff
```

Inspecione o conteúdo da tabela *STAFF*, procurando especificamente o *ID* com valor '10'. A coluna *NAME* correspondente tem valor *Sanders*. Feche a janela.

2. O isolamento currently committed é padrão nos novos bancos de dados. Verifique se ele foi realmente ativado na amostra do banco de dados:

```
db2 get db cfg for sample
```

Procure uma linha próxima ao final da saída, dizendo:

```
Currently Committed (CUR_COMMIT) = ON
```

Se o valor for ON, desative-o (OFF) primeiro para podermos analisar o comportamento do isolamento CS, como era antes do DB2 9.7:

```
db2 update db cfg for sample using CUR_COMMIT off
db2 force applications all
```

Adicionar a opção *force* assegura que não haja conexões, fazendo a atualização para CUR_COMMIT entrar em vigor na próxima conexão)

Confirme se CUR_COMMIT está desativado. A instrução deveria ser:

```
Currently Committed (CUR_COMMIT) = DISABLED
```

3. Abra duas Janelas de comando do DB2 como na parte 1, colocando uma no alto da outra. Vamos ver como funciona o isolamento CS sem a opção currently committed, quando uma operação de atualização (gravador) e de seleção (leitor) têm interesse na mesma linha. Observe que não é necessário colocar "WITH CS" após as instruções (já que este é o padrão).

Na Command window #1 do DB2 (este é o gravador):

```
db2 connect to sample
db2 +c update staff set name = 'Chong' where id = 10
```

Na Command window #2 do DB2 (este é o leitor):

```
db2 connect to sample
db2 +c select * from staff
```

Esta operação SELECT fica parada, aguardando o bloqueio exclusivo (X) ser liberado pelo aplicativo da Command Window #1 do DB2. Como você pode ver, o

comportamento padrão do CS antes do DB2 9.7 permite menos concorrência.

Na Command Window #2 do DB2:

CTRL-C (press these keys to interrupt)

Feche a janela

Na Command Window #1 do DB2:

db2 rollback

Feche a janela

Procedimento 2: Analisar o comportamento do isolamento CS com o currently committed

1. Ative a opção currently committed (ON):

Abra uma Janela de comando do DB2 e execute:

```
db2 update db cfg for sample using CUR_COMMIT on
db2 force applications all
```

Feche a janela.

2. Abra duas Janelas de comando do DB2 como na parte 1, colocando uma no alto da outra. Em seguida, execute o seguinte:

Na Command Window #1 do DB2:

```
db2 connect to sample
db2 +c update staff set name = 'Chong' where id = 10
```

Na Command Window #2 do DB2:

```
db2 connect to sample
db2 +c select * from staff
```

Agora esta instrução SELECT funciona! Ela não fica parada e o valor exibido é **Sanders**, que é o valor atualmente comprometido.

Na Command Window #1 do DB2:

```
db2 rollback
Feche a janela.
```

Na Command Window #2 do DB2:

```
db2 rollback
Feche a janela.
```

Procedimento 3: Analisar o comportamento do isolamento UR

1. Abra duas Janelas de comando do DB2 como na parte 1, colocando uma no alto da outra. Em seguida, execute o seguinte:

Na Command Window #1 do DB2:

```
db2 connect to sample
db2 +c update staff set name = 'Chong' where id = 10
```

Na Command Window #2 do DB2:

```
db2 connect to sample
db2 +c select * from staff with UR
```

Esta operação SELECT funciona, mas observe que o valor exibido é *Chong*, que é o valor não comprometido.

Na Command Window #1 do DB2:

```
db2 rollback
```

Feche a janela.

Na Command Window #2 do DB2:

```
db2 rollback
```

Feche a janela.

**PARTE III – APRENDENDO O DB2:
DESENVOLVIMENTO DE
APLICATIVOS**

14

Capítulo 14 – Introdução ao desenvolvimento de aplicativos DB2

O IBM DB2 é um poderoso software para o gerenciamento de dados relacionais e XML. Ele oferece flexibilidade não apenas aos administradores de banco de dados, mas também aos desenvolvedores de bancos de dados. Independente da linguagem que você usa para desenvolver seus programas, o DB2 fornece os drivers, adaptadores e extensões necessárias para trabalhar com bancos de dados como parte do seu aplicativo. Além disso, com o DB2 Express-C, você pode desenvolver seus aplicativos sem custo, sem limites de tamanho de banco de dados e com o mesmo nível de suporte a linguagem de programação das outras versões do DB2. Desenvolva apenas uma vez usando o DB2 Express-C, e você poderá executar em qualquer edição do DB2 sem qualquer necessidade de modificação do seu aplicativo.

Observação:

Esta seção é apenas uma introdução ao desenvolvimento de aplicativos no DB2. Uma série de mais de 25 livros gratuitos on-line está sendo desenvolvida com o parte da **série DB2 on Campus** incluindo um livro específico para desenvolvimento de aplicativos para o DB2. Outros livros da série incluem tópicos que não são relativos ao DB2, como Java, PHP, Ruby on Rails, Python, Perl, Web 2.0, SOA, Eclipse, desenvolvimento em código-fonte aberto, computação em nuvem e muito mais! Outros livros abrangem as tecnologias da IBM, como pureQuery, Data Studio, InfoSphere Data Architect em maiores detalhes. Estes livros serão lançados a partir de outubro de 2009.

14.1 Desenvolvimento de aplicativos no DB2: O quadro geral

O DB2 oferece aos desenvolvedores de bancos de dados a flexibilidade de tirar proveito dos recursos de desenvolvimento no servidor, como procedimentos armazenados e funções definidas pelo usuário, enquanto os desenvolvedores de aplicativos podem desenvolver aplicativos cliente usando sua linguagem de programação preferida. Esta flexibilidade é mostrada na *Figura 14.1*.

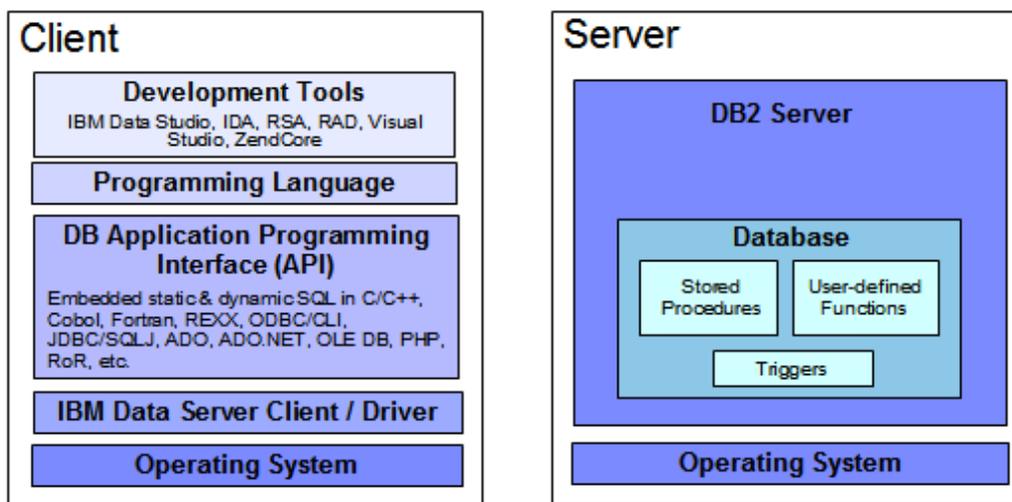


Figura 14.1 – O DB2 é para todos: Desenvolvedores de aplicativos e bancos de dados

Na Figura 14.1, o lado esquerdo representa uma máquina cliente onde um programador de aplicativos desenvolve e executa seu programa. Nesta máquina cliente, pode-se instalar um IBM Data Server Client além do sistema operacional, dependendo do tipo de aplicativo sendo desenvolvido. O IBM Data Server Client inclui os drivers de conexão necessários, como os drivers JDBC e ODBC/CLI. Também é possível fazer o download desses drivers de maneira independente, visitando o site do IBM DB2 Express-C em www.ibm.com/db2/express.

Usando ferramentas de programação como o IBM Data Studio, o InfoSphere Data Architect (IDA), o Rational Software Architect (RSA), o Rational Application Developer (RAD) e assim por diante, você pode desenvolver seu aplicativo com a linguagem de programação de sua preferência. As bibliotecas de API que suportam tais linguagens também foram incluídas no IBM Data Server Client, para que, ao se conectar a um DB2 Server, todas as instruções do programa sejam traduzidas adequadamente, usando estas APIs, nas instruções SQL ou XQuery compreendidas pelo DB2. A Tabela 1.1 fornece uma breve descrição das ferramentas mencionadas anteriormente.

Nome da ferramenta	Descrição
IBM Data Studio	O IBM Data Studio é uma ferramenta baseada em Eclipse que permite aos usuários gerenciar seus servidores de dados e desenvolver procedimentos armazenados (<i>stored procedures</i>), funções e serviços de dados de Web. O IBM Data Studio foi discutido anteriormente neste livro.
InfoSphere Data Architect (IDA)	O IDA é uma ferramenta de modelagem de dados para ajudá-lo a criar o projeto lógico

	e físico do seu banco de dados.
Rational Software Architect (RSA)	O RSA é uma ferramenta baseada em Eclipse para a engenharia de software, para ajudá-lo a desenvolver diagramas de UML.
Rational Application Developer (RAD)	O RAD é uma ferramenta de desenvolvimento rápido de aplicativos baseada em Eclipse, para desenvolvedores de software.
Visual Studio	O Microsoft Visual Studio é um IDE que permite que você desenvolva aplicativos na plataforma Windows, usando tecnologia da Microsoft.
ZendCore	Chamado anteriormente de <i>ZendCore for IBM</i> , este é um IDE gratuito para o desenvolvimento de aplicativos em PHP.

Tabela 14.1 – Ferramentas que podem ajudá-lo a desenvolver aplicativos com o DB2

No lado direito da *Figura 14.1* um servidor DB2 é ilustrado, contendo um banco de dados. Dentro desse banco de dados há imagens armazenadas, funções e disparadores (*triggers*) definidos pelo usuário. Descreveremos todos esses objetos em mais detalhes nas próximas seções.

14.2 Desenvolvimento no lado do servidor

O desenvolvimento no DB2 no lado do servidor implica que os objetos do aplicativo sejam desenvolvidos e armazenados no banco de dados DB2. Os objetos de aplicativo abaixo serão discutidos rapidamente nesta seção:

- Procedimentos armazenados (Stored Procedures)
- Funções definidas pelo usuário (UDFs)
- Disparadores (Triggers)

14.2.1 Procedimentos armazenados

Um **procedimento armazenado (stored procedure)** é um objeto de aplicativo de banco de dados que pode encapsular instruções SQL e lógica de negócios. Manter parte da lógica do aplicativo no banco de dados aumenta o desempenho, pois a quantidade de tráfego de rede entre o aplicativo e o banco de dados é reduzida. Além disso, os procedimentos armazenados fornecem uma localização centralizada para armazenar seu código, para que outros aplicativos possam reutilizar os mesmos procedimentos armazenados. Para chamar um procedimento armazenado, use a instrução **CALL**. No DB2, você pode desenvolver procedimentos armazenados em diversas linguagens, incluindo SQL PL, Java, C/C++, CLR, OLE e COBOL. Veja abaixo um exemplo simples de como criar e chamar um procedimento armazenado em SQL PL na Janela de comando do DB2 ou no shell do Linux:

```
db2 create procedure P1 begin end
db2 call P1
```

No exemplo, o procedimento P1 é um procedimento armazenado vazio, que não faz nada. O exemplo ilustra como é fácil criar um procedimento armazenado. Para desenvolver procedimentos armazenados com lógica mais complexa, recomendamos usar o IBM Data Studio, que inclui um depurador.

14.2.2 Funções definidas pelo usuário

Uma **função definida pelo usuário (UDF)** é um objeto de aplicativo de banco de dados que permite aos usuários ampliar a linguagem SQL com sua própria lógica. Uma função sempre retorna um valor ou vários valores, normalmente como resultado da lógica de negócios incluída na função. Para chamar uma função, use-a dentro de uma instrução SQL ou com a função `values`. No DB2, você pode desenvolver UDFs em diversas linguagens, incluindo SQL PL, Java, C/C++, OLE e CLR.

O exemplo simples abaixo mostra como criar e chamar uma UDF em SQL PL na Janela de comando do DB2 ou no shell do Linux:

```
db2 create function F1() returns integer begin return 1000; end
db2 values F1
```

No exemplo, a função F1 retorna o valor inteiro 1000. A instrução `VALUES` pode ser usada para chamar a função. Como no caso dos procedimentos armazenados, recomendamos que você crie as funções usando o IBM Data Studio.

14.2.3 Disparadores

Um **disparador (trigger)** é um objeto que executa automaticamente uma operação em uma tabela ou visualização. Uma ação de disparo no objeto onde o disparador está definido faz o disparador ser disparado. Um disparador normalmente não é considerado um objeto de aplicativo; portanto, os desenvolvedores de banco de dados normalmente não programam disparadores, o que é feito pelos administradores de banco de dados. Como é necessário alguma programação, incluímos os disparadores nesta seção. Veja abaixo um exemplo de um disparador:

```
create trigger myvalidate no cascade before insert on T1
referencing NEW as N
for each row
begin atomic
set (N.myxmlcol) = XMLVALIDATE(N.myxmlcol
according to xmlschema id myxmlschema);
end
```

Neste exemplo, o disparador é disparado antes de uma operação INSERT na tabela T1. O disparador irá inserir o valor (que é um documento XML), mas chamará a função

XMLVALIDATE para validar este documento XML com um dado esquema. O *Capítulo 15, DB2 pureXML*, entra em mais detalhes sobre XML e esquemas XML.

14.3 Desenvolvimento no lado do cliente

Como o nome sugere, no desenvolvimento no lado do cliente, os desenvolvedores de aplicativos fazem seus programas em um cliente e, em seguida, conectam-se ao banco de dados do DB2 usando as interfaces de programa de aplicativo (APIs) que são fornecidas com o DB2. Nesta seção, discutiremos:

- SQL incorporado
- SQL estático x SQL dinâmico
- CLI e ODBC
- JDBC, SQLJ e pureQuery
- OLE DB
- ADO.NET
- PHP
- Ruby on Rails
- Perl
- Python

14.3.1 SQL incorporado

Os **aplicativos com SQL incorporado** são aplicativos onde o SQL é incorporado em uma linguagem host como C, C++ ou COBOL. O aplicativo com SQL incorporado pode incluir SQL estático ou dinâmico, como será descrito na próxima seção. A *Figura 14.2* mostra como é construído um aplicativo com SQL incorporado.

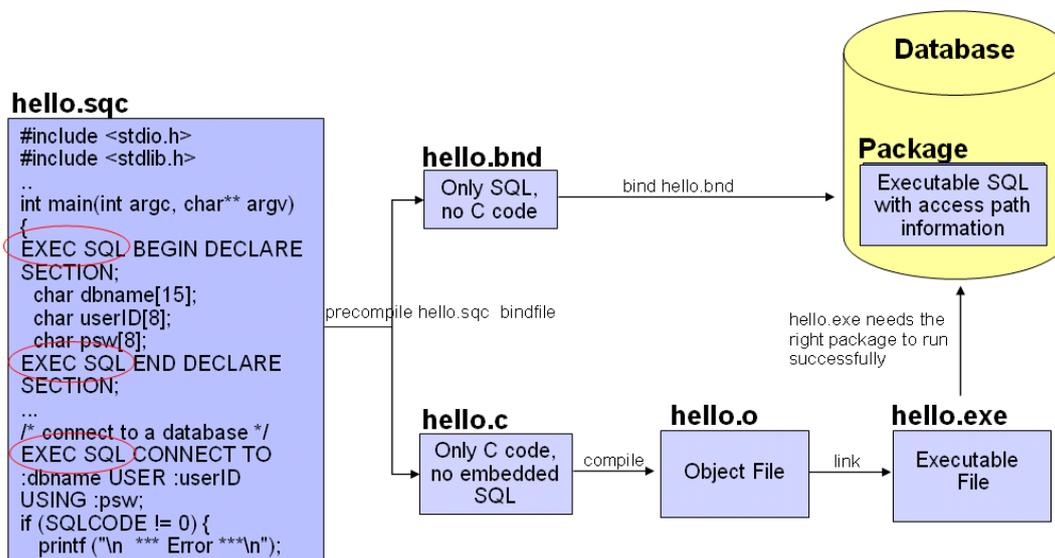


Figura 14.2 – Construção de aplicativos com SQL incorporado

Na figura, o programa em C `hello.sqc` contém SQL incorporado. A API de SQL incorporado para a linguagem C usa `EXEC SQL` (em destaque na *Figura 14.2*) para permitir um processo de pré-compilação para distinguir entre as instruções SQL incorporadas e o código C em si. Você também pode observar, na listagem `hello.sqc`, que algumas variáveis são pré-fixadas com dois pontos, como `:dbname`, `:userID` e `:psw`. Elas são denominadas variáveis host. As variáveis host são variáveis da linguagem host mencionadas nas instruções SQL incorporadas.

Ao executar o comando `precompile` (também conhecido como comando `prep`) com a opção `bindfile`, são gerados dois arquivos, o arquivo de associação `hello.bnd` contendo apenas instruções em SQL e o arquivo `hello.c` contendo somente código C. O arquivo de associação será compilado usando o comando `bind` para obter um **pacote** que será armazenado no banco de dados. Um pacote inclui o SQL compilado/executável e o caminho de acesso que o DB2 seguirá para recuperar os dados. Para executar o comando `bind`, uma conexão ao banco de dados deve existir. Na parte inferior da figura, o arquivo `hello.c` é compilado e vinculado como qualquer programa comum em C. O arquivo executável `hello.exe` resultante deve corresponder ao pacote armazenado no banco de dados, para ser executado com sucesso.

14.3.2 SQL estático x SQL dinâmico

As instruções **SQL estático** são aquelas nas quais a estrutura SQL é completamente conhecida no momento da pré-compilação. Por exemplo:

```
SELECT lastname, salary FROM employee
```

Neste exemplo, os nomes das colunas (`lastname`, `salary`) e da tabela (`employee`) mencionados em uma instrução são totalmente conhecidos no momento da pré-compilação. O exemplo abaixo também é uma instrução em SQL estático:

```
SELECT lastname, salary FROM employee WHERE firstnme = :fname
```

Neste segundo exemplo, a variável host `:fname` é usada como parte de uma instrução SQL incorporada. Embora o valor da variável host seja desconhecido até a execução, seu tipo de dados é conhecido a partir do programa, e todos os outros objetos (nomes de coluna, nomes de tabela) são totalmente conhecidos antecipadamente. O DB2 usa estimativas para estas variáveis host, para calcular antecipadamente o plano de acesso; portanto, este caso ainda é considerado SQL estático.

Você pré-compila, associa e compila as instruções SQL executadas estaticamente antes de executar o aplicativo. O SQL estático é melhor usado em bancos de dados nos quais as estatísticas não mudam muito. Agora, vejamos mais um exemplo:

```
SELECT ?, ? FROM ?
```

Neste exemplo, os nomes das colunas e da tabela mencionados pela instrução não são conhecidos até a execução. Portanto, o plano de acesso é calculado somente na execução, e usando as estatísticas disponíveis nesse momento. Esse tipo de instrução é considerado **SQL dinâmico**.

Algumas APIs de programação, como JDBC e ODBC, sempre usam SQL dinâmico independentemente das instruções SQL incluírem objetos conhecidos ou não. Por exemplo, a instrução `SELECT lastname, salary FROM employee` possui todos os nomes de coluna e tabela conhecidos antecipadamente; entretanto, as instruções não são pré-compiladas quando feitas através de JDBC ou ODBC. Todos os planos de acesso das instruções são calculados no momento da execução.

Em geral, duas instruções são usadas para tratar uma instrução SQL como dinâmica:

- **PREPARE:** Esta instrução prepara ou compila a instrução SQL, calculando o plano de acesso a ser usado para recuperar os dados
- **EXECUTE:** Esta instrução executa o SQL

Como alternativa, você pode executar **PREPARE** e **EXECUTE** em uma única instrução: **EXECUTE IMMEDIATELY**

A Listagem 14.1 mostra um exemplo de instrução SQL dinâmico incorporado em C que é preparada e executada.

```
strcpy(hVStmtDyn, "SELECT name FROM emp WHERE dept = ?");
PREPARE StmtDyn FROM :hVStmtDyn;
EXECUTE StmtDyn USING 1;
EXECUTE StmtDyn USING 2;
```

Listagem 14.1 – Uma instrução SQL dinâmico incorporado em C usando PREPARE e EXECUTE

A Listagem 14.2 mostra o mesmo exemplo da Listagem 14.1, mas usando a instrução EXECUTE IMMEDIATELY

```
EXECUTE IMMEDIATELY SELECT name from EMP where dept = 1
```

```
EXECUTE IMMEDIATELY SELECT name from EMP where dept = 2
```

Listagem 14.2 – Uma instrução SQL dinâmico incorporado em C usando EXECUTE IMMEDIATELY

Em diversas linguagens de programação dinâmicas, como PHP ou Ruby on Rails, onde o SQL é executado dinamicamente, os programadores tendem a escrever as mesmas instruções SQL com valores de campo diferentes, como abaixo:

```
SELECT lastname, salary FROM employee where firstnme = 'Raul'
```

```
SELECT lastname, salary FROM employee where firstnme = 'Jin'
```

...

Neste exemplo, as instruções são idênticas, exceto quanto ao valor da coluna `firstnme`. O DB2 considera duas instruções SQL dinâmico como diferentes; portanto, durante a execução, ele as prepara e executa de maneira independente. A sobrecarga de preparar a mesma instrução diversas vezes pode causar degradação de desempenho; sendo assim, antes do DB2 9.7, era recomendado codificar as instruções como mostrado abaixo:

```
SELECT lastname, salary FROM employee where firstnme = ?
```

O sinal de interrogação (?) na instrução é conhecido como um **marcador de parâmetro**. Ao usar marcadores de parâmetro, o programa pode preparar a instrução somente uma vez e executar as instruções EXECUTE fornecendo os valores diferentes para o marcador de parâmetro.

O DB2 9.7 introduziu uma tecnologia denominada **concentrador de instrução**, no qual todas as instruções que sejam iguais exceto pelos valores de campo são automaticamente reunidas em uma única instrução com marcadores de parâmetro e, em seguida, são realizadas instruções EXECUTE com os valores diferentes. O concentrador de instrução tem inteligência para determinar quando não deve agrupar algumas instruções; por exemplo, quando você adiciona propositalmente algumas cláusulas para influenciar o otimizador (Optimizer) do DB2.

Quanto ao desempenho, o SQL estático normalmente tem melhor desempenho que o SQL dinâmico, pois o plano de acesso do SQL estático é realizado no momento da pré-compilação, e não durante a execução. Entretanto, para ambientes onde há muita atividade do tipo INSERT e DELETE, as estatísticas calculadas durante a pré-compilação podem não estar atualizadas, e assim o plano de acesso do SQL estático pode não ser o ideal. Neste caso, o SQL dinâmico pode ser uma escolha melhor, levando em consideração que o comando RUNSTATS seja executado frequentemente para coletar estatísticas atuais.

Observação:

Muitos usuários acham que o SQL incorporado é apenas estático. Na realidade, ele pode ser estático ou dinâmico.

14.3.3 CLI e ODBC

A **Interface de Nível de Chamada (CLI)** foi desenvolvida originalmente pela X/Open Company e pelo SQL Access Group. Ela era uma especificação para uma interface SQL que podia ser chamada, com a finalidade de desenvolver aplicativos portáteis em C/C++, independentemente do fornecedor de RDBMS. Com base em um esboço preliminar da Interface de Nível de Chamada da X/Open, a Microsoft desenvolveu a **Conectividade Aberta de Banco de Dados (ODBC)** e posteriormente, a ISO CLI International Standard aceitou a maior parte da especificação da Interface de Nível de Chamada da X/Open. A CLI do DB2 baseia-se em ambas: ODBC e a norma internacional para SQL/CLI, como mostrado na *Figura 14.3*.

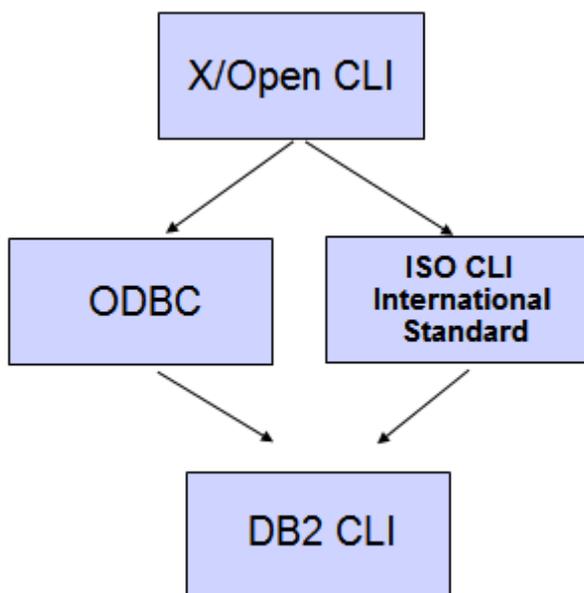


Figura 14.3 – A CLI do DB2 é baseada em ODBC e na norma internacional ISO CLI

A CLI do DB2 atende à ODBC 3.51 e pode ser usada como Driver ODBC quando carregada por um ODBC Driver Manager. A Figura 14.4 pode ajudá-lo a visualizar o suporte da CLI do DB2 para ODBC.

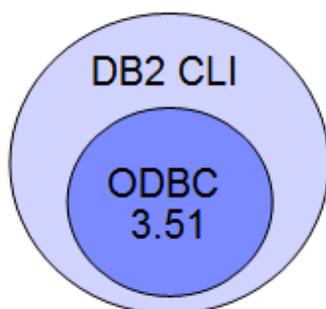


Figura 14.4 – A CLI do DB2 em conformidade com ODBC 3.51

A CLI/ODBC apresenta as seguintes características:

- O código é facilmente portátil entre diversos fornecedores de RDBMS
- Diferentemente do SQL incorporado, não há necessidade de um pré-compilador ou de variáveis host
- Ela executa SQL dinâmico
- É muito popular

Para **executar** um aplicativo CLI/ODBC, basta usar o driver de CLI do DB2. este driver é instalado a partir de um dos clientes e drivers abaixo, que podem ser obtidos para serem usados gratuitamente em www.ibm.com/db2/express:

- IBM Data Server Client
- IBM Data Server Runtime Client
- IBM Data Server Driver for ODBC and CLI

Para **desenvolver** um aplicativo CLI/ODBC, é necessário o driver de CLI do DB2 e também as bibliotecas apropriadas. Estas podem ser encontradas somente no IBM Data Server Client.

Vejamos agora o exemplo abaixo, para que você entenda melhor como configurar o driver de CLI do DB2 para seus aplicativos. A *Figura 14.5* ilustra três máquinas diferentes, uma na Indonésia, uma no Brasil e a outra no Canadá.

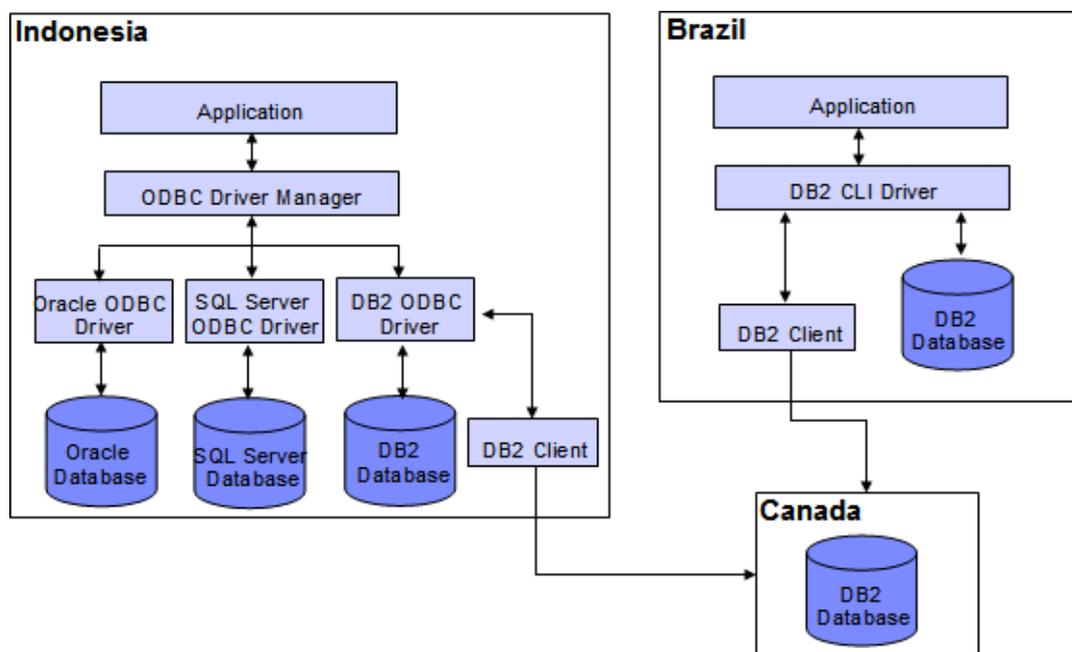


Figura 14.5 – Amostra de cenário CLI/ODBC no DB2

A figura mostra dois casos:

À esquerda, digamos que a máquina da Indonésia está executando um *aplicativo ODBC* que poderia funcionar com qualquer RDBMS, como Oracle, Microsoft SQL Server ou DB2. Um ODBC Driver Manager carregará o driver ODBC apropriado, dependendo do banco de dados a ser acessado. No caso em que o aplicativo acessa um banco de dados DB2 no Canadá, a conexão necessita passar através de um cliente DB2 que possua os componentes para a conexão remota.

No lado direito, digamos que um *aplicativo CLI* esteja sendo executado em uma máquina no Brasil. É um aplicativo CLI porque pode estar usando algumas funções específicas não disponíveis em ODBC, e também porque o aplicativo funciona somente com um banco de dados DB2. O aplicativo CLI passará pelo driver de CLI do DB2. O aplicativo pode se conectar ao banco de dados DB2 local no Brasil. Quando precisar se conectar ao banco de dados remoto no Canadá, ele passará por um cliente DB2.

Uma última observação a ser feita é a comparação entre um aplicativo CLI/ODBC e um aplicativo com SQL dinâmico incorporado em C. A *Figura 14.6* ilustra esta comparação.

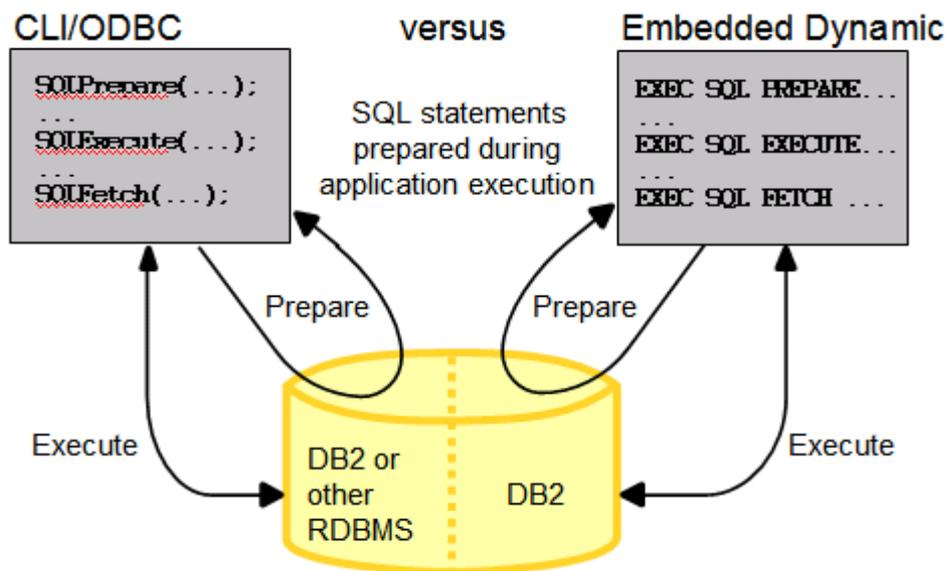


Figura 14.6 – Aplicativo CLI/ODBC x aplicativo com SQL dinâmico incorporado em C

Como mostrado na *Figura 14.6*, a única diferença entre CLI/ODBC x SQL dinâmico incorporado em C é que, no CLI/ODBC, o código é portátil e pode acessar outros RDBMS simplesmente alterando a cadeia de texto de conexão, enquanto que, na versão do SQL dinâmico incorporado em C, pode ser necessário programar elementos específicos para DB2. Obviamente, a outra diferença é a maneira pela qual são chamadas as diferentes funções para `PREPARE` e `EXECUTE`.

14.3.4 JDBC, SQLJ e pureQuery

A **Conectividade de Banco de Dados Java (JDBC)** é uma API de programação em Java que padroniza o modo de usar e acessar bancos de dados. Em JDBC, o código é facilmente portátil entre diversos fornecedores de RDBMS. Normalmente, as únicas

alterações necessárias no código são o driver JDBC que será carregado e a cadeia de texto da conexão. A JDBC usa apenas SQL dinâmico e é muito popular.

O **SQLJ** é o padrão para incorporar SQL em programas Java. Ele é usado principalmente com SQL estático, embora possa interoperar com JDBC, como mostrado na *Figura 14.7*. Embora normalmente seja mais compacto que os programas JDBC e forneça melhor desempenho, ele ainda não foi amplamente aceito. Os programas em SQLJ devem passar através de um pré-processador (o tradutor SQLJ) antes de serem compilados.

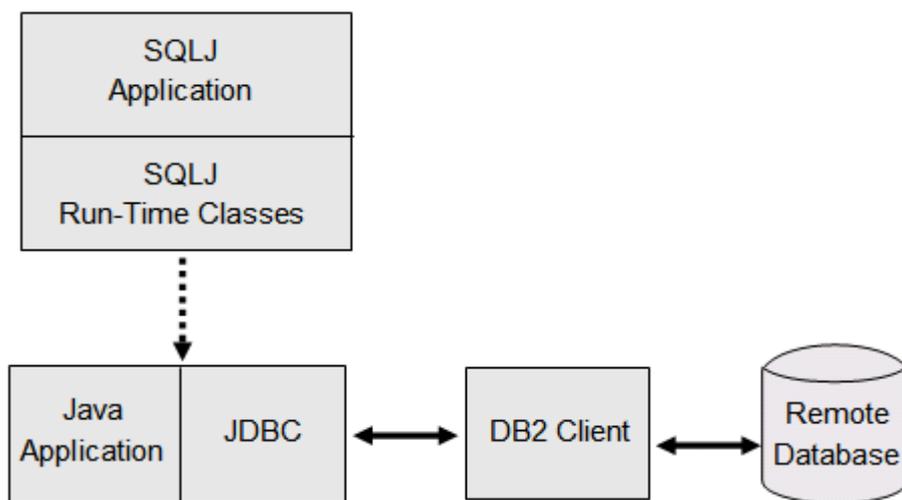


Figura 14.7 – Relacionamento entre aplicativos SQLJ e JDBC

Na *Figura 14.7*, um cliente DB2 pode ou não ser necessário, dependendo do tipo de driver JDBC usado, conforme discutiremos mais adiante nesta seção.

O **pureQuery** é um plugin da IBM baseado em Eclipse para gerenciar dados relacionais como objetos. Disponível desde 2007, o pureQuery pode gerar automaticamente o código para estabelecer um mapeamento objeto-relacional (ORM) entre seu código orientado a objeto e os objetos do banco de dados relacional. Comece criando um projeto Java com o Optim Development Studio (ODS), conecte-se a um banco de dados DB2 e, em seguida, use o ODS para descobrir todos os objetos do banco de dados. Através da GUI do ODS, você pode selecionar uma tabela e, em seguida, gerar o código pureQuery, que transformará qualquer uma das entidades da tabela relacional básica em um objeto Java. É gerado o código para criar as instruções SQL relevantes e os objetos-pais em Java que encapsulam essas instruções. Os objetos Java gerados e as instruções SQL contidas podem ser ainda mais personalizados. Com o pureQuery, você pode decidir no momento da execução se deseja executar seu SQL em modo estático ou dinâmico. O pureQuery suporta Java e .NET.

14.3.4.1 Drivers JDBC e SQLJ

Embora existam diversos tipos de drivers JDBC, como os tipos 1, 2, 3 e 4; os tipos 1 e 3 não são usados comumente e o suporte do DB2 a esses tipos foi descontinuado. Para o

tipo 2, há dois drivers que descreveremos em breve, mas um deles também foi descontinuado.

Os tipos 2 e 4 são suportados no DB2, como mostrado na *Tabela 14.2*. Os drivers do tipo 2 precisam de um cliente DB2 instalado, pois o driver irá usá-lo para estabelecer a comunicação com o banco de dados. O tipo 4 é um cliente Java puro, portanto não há necessidade de um cliente DB2, mas o driver deve ser instalado na máquina onde o aplicativo JDBC está sendo executado.

Tipo de driver	Nome do driver	Fornecido como	Suporta	Nível mínimo necessário de SDK para Java
Tipo 2	DB2 JDBC Type 2 Driver for Linux, UNIX and Windows (obsoleto*)	db2java.zip	JDBC 1.2 e JDBC 2.0	1.4.2
Tipo 2 e Tipo 4	IBM Data Server Driver for JDBC and SQLJ	db2jcc.jar e sqlj.zip	Compatível com JDBC 3.0	1.4.2
		db2jcc4.jar e sqlj4.zip	JDBC 4.0 e anterior	6

Tabela 14.2 – Drivers JDBC e SQLJ do DB2

* Obsoleto significa que ainda é suportado, mas não recebe mais aperfeiçoamentos

Como mencionado anteriormente e mostrado na *Tabela 14.2*, o tipo 2 é fornecido com dois drivers diferentes; entretanto, o DB2 JDBC Type 2 Driver for Linux, UNIX e Windows, com nome de arquivo db2java.zip, está obsoleto.

Ao instalar um servidor DB2, um cliente DB2 ou o IBM Data Server Driver para JDBC e SQLJ, os arquivos db2jcc.jar e sqlj.zip compatíveis com JDBC 3.0 são adicionados automaticamente ao seu classpath.

14.3.5 OLE DB

A **Vinculação e Incorporação de Objeto, O Banco de dados (OLE DB)** é um conjunto de interfaces que fornece acesso a dados armazenados em diversas fontes. Ele foi projetado como substituto para o ODBC, mas estendido para suportar uma variedade de fontes mais ampla, incluindo bancos de dados não relacionais, como bancos de dados orientados a objeto e planilhas. A OLE DB é implementada usando a tecnologia Component Object Model (COM).

Os usuários de OLE DB podem acessar um banco de dados DB2 com o IBM OLE DB Provider for DB2. Este provedor tem as seguintes características:

- Nome do provedor: IBMDADB2

- Suporta o nível 0 da especificação de provedor OLE DB, incluindo algumas interfaces adicionais do nível 1
- Compatível com a versão 2.7 ou posterior da especificação OLE DB da Microsoft
- É necessário instalar um IBM Data Server Client com os Microsoft Data Access Components (MDAC)
- Se o IBM DADB2 não for especificado explicitamente, o driver OLE DB da Microsoft (MSDASQL) será utilizado como padrão. O MSDASQL permite que clientes utilizando OLE DB acessem fontes de dados de servidor SQL, que não são Microsoft, usando o driver ODBC, mas não assegura uma funcionalidade completa do driver OLE DB.

14.3.6 ADO.NET

O **.NET Framework** é o substituto da Microsoft para a tecnologia Component Object Model (COM). Usando o .NET Framework, você pode programar aplicativos .NET em mais de quarenta linguagens de programação diferentes, sendo as mais populares C# e Visual Basic .NET.

As bibliotecas da classe .NET Framework fornecem os elementos fundamentais com os quais você pode construir aplicativos .NET. Esta biblioteca de classe é independente de linguagem e fornece interfaces para serviços de sistema operacional e aplicativos. Seu aplicativo .NET (independente da linguagem) é compilado em Intermediate Language (IL), um tipo de código de bytes.

O Common Language Runtime (CLR) é o cerne do .NET Framework, compilando o código IL durante o processo e executando-o em seguida. Ao executar o código IL compilado, o CLR ativa os objetos, verifica seu nível de segurança, aloca a memória, executa-os e limpa a memória após concluir a execução.

Fazendo uma analogia ao funcionamento do Java; em Java, um programa pode ser executado em diversas plataformas com modificações mínimas ou nenhuma modificação: uma linguagem, mas diversas plataformas. Em .NET, um programa escrito em qualquer uma das quarenta linguagens suportadas pode ser executado em uma plataforma, Windows, com modificações mínimas ou nenhuma modificação: diversas linguagens, mas uma única plataforma.

O **ADO.NET** é como é fornecido suporte de acesso aos dados no .NET Framework. O ADO.NET suporta acesso conectado e desconectado. O componente essencial do acesso desconectado a dados no ADO.NET é a classe `DataSet`, cujas instâncias atuam como cache de banco de dados residente na memória do seu aplicativo.

Para ambos os acessos, conectado e desconectado, seus aplicativos usam os bancos de dados através do que é conhecido por **provedor de dados**. Diversos produtos de banco de dados incluem seus próprios provedores de dados .NET, incluindo o DB2 for Windows.

Um provedor de dados .NET apresenta implementações das seguintes classes básicas:

- Connection: estabelece e gerencia uma conexão de banco de dados.

- Command: executa uma instrução SQL em um banco de dados.
- DataReader: lê e retorna os dados do conjunto de resultados de um banco de dados.
- DataAdapter: vincula uma instância de DataSet a um banco de dados. Através de uma instância DataAdapter, o DataSet pode ler e gravar dados de tabela do banco de dados.

Os três provedores de dados que podem funcionar com o DB2 são mostrados na *Tabela 14.3*

Provedor de dados	Características
ODBC .NET Data provider (não recomendado)	<ul style="list-style-type: none"> ▪ Faz chamadas ODBC a uma fonte de dados do DB2 usando o driver de CLI do DB2 ▪ Ele apresenta o mesmo suporte e restrições a palavras-chave que o driver de CLI do DB2 ▪ Pode ser usado com .NET Framework versão 1.1, 2.0 ou 3.0
OLE DB .NET Data provider (não recomendado)	<ul style="list-style-type: none"> ▪ Usa o IBM DB2 OLE DB Driver (IBMDADB2) ▪ Ele apresenta o mesmo suporte e restrições a palavras-chave do driver OLE DB do DB2 ▪ Pode ser usado somente com .NET Framework versão 1.1, 2.0 ou 3.0.
DB2 .NET Data provider (recomendado)	<ul style="list-style-type: none"> ▪ Amplia o suporte do DB2 à interface ADO.NET. ▪ O provedor gerenciado pelo DB2 implementa o mesmo conjunto padrão de classes e métodos ADO.NET ▪ Ele é definido sob o namespace IBM.DATA.DB2 ▪ Pode ser obtido através do download de: <ul style="list-style-type: none"> - Data Server Driver for ODBC, CLI, and .NET - IBM Data Server Runtime Client - DB2 Data Server

Tabela 14.3 – Provedores de dados ADO.NET

14.3.7 PHP

O **PHP Hypertext Preprocessor (PHP)** é uma linguagem de scripts de código-fonte aberto, independente de plataforma, criada para o desenvolvimento de aplicativos de Web. Ela pode ser incorporada a HTML e geralmente é executada em um servidor de Web, que usa o código PHP para gerar, como saída, páginas da Web.

O PHP é uma linguagem modular. Você pode usar extensões para personalizar a funcionalidade disponível. Algumas das extensões PHP mais populares são as usadas para acessar bancos de dados. A IBM suporta o acesso aos bancos de dados DB2 através de duas extensões:

- **ibm_db2:** A extensão `ibm_db2` oferece uma interface de programação de aplicativos processuais para criar, ler, atualizar e gravar operações de bancos de dados, além de um amplo acesso aos metadados de banco de dados. Ela pode ser compilada com PHP 4 ou PHP 5.
- **pdo_ibm:** O `pdo_ibm` é um driver para a extensão PHP Data Objects (PDO) que oferece acesso ao banco de dados DB2 através da interface padrão de banco de dados orientado a objeto introduzida com o PHP 5.1. Ele pode ser compilado diretamente com as bibliotecas do DB2.

As extensões e drivers PHP estão disponíveis gratuitamente no repositório PECL em <http://pecl.php.net/>, ou com o IBM Data Server Client. Ambos, `ibm_db2` e `pdo_ibm`, são baseados na IBM DB2 CLI Layer

A IBM também formou uma parceria com a Zend Technologies Inc. para suportar o ZendCore, um kit de ferramentas de ambiente gratuito, pronto para o desenvolvimento com PHP e DB2 Express-C. O pacote ZendCore inclui bibliotecas PHP, o servidor de web Apache e o DB2 Express-C. Para fazer o download do ZendCore, visite <http://www.ibm.com/software/data/info/zendcore>

14.3.8 Ruby on Rails

Ruby é uma linguagem de código-fonte aberto orientada a objeto. Rails é um framework de Web criado com Ruby. Ruby on Rails (RoR) é um método ideal para desenvolver aplicativos baseados em web com auxílio de banco de dados. Esta novíssima tecnologia é baseada na arquitetura Model, View, Controller (MVC) e segue os princípios do desenvolvimento ágil de software.

O Rails não requer formatos de arquivo especiais ou ambientes de desenvolvimento integrado (IDEs); você pode começar usando um editor de texto. Entretanto, há diversos IDEs disponíveis com suporte a Rails, como RadRails, que é um ambiente Rails para Eclipse. Para obter mais informações sobre o RadRails, visite <http://www.radrails.org/>.

O DB2 suporta Ruby 1.8.5 e posterior e Ruby on Rails 1.2.1 e posterior. O `IBM_DB` gem inclui o driver `IBM_DB` Ruby e o adaptador Rails, permitindo que você trabalhe com DB2, e são baseados na camada CLI. Este gem deve ser instalado juntamente com um IBM Data Server Client. Para instalar o driver `IBM_DB` e o adaptador, você pode usar o Ruby gem ou instalá-los como um plugin do Rails.

14.3.9 Perl

Perl é uma linguagem de programação interpretada popular, que está disponível gratuitamente para diversos sistemas operacionais. Ela usa SQL dinâmico e é ideal para a prototipagem de aplicativos.

A linguagem Perl fornece um módulo padrão denominado Interface de Banco de Dados (DBI), para acessar bancos de dados diferentes. Ele está disponível em <http://www.perl.com>. Este módulo “conversa” com os drivers de diferentes fornecedores de bancos de dados. No caso do DB2, é o driver `DBD::DB2`, que está disponível em <http://www.ibm.com/software/data/db2/perl>.

14.3.10 Python

Python é uma linguagem dinâmica usada frequentemente para criar scripts. Ela enfatiza a legibilidade de código e suporta diversos paradigmas de programação, incluindo a programação processual, orientada a objeto, orientada a aspecto, metaprogramação e funcional. Python é uma linguagem ideal para o desenvolvimento rápido de aplicativos.

A Tabela 14.4 mostra as extensões disponíveis para acessar os bancos de dados DB2 a partir de um aplicativo Python.

Extensão	Descrição
ibm_db	<p>Definido pela IBM.</p> <p>Fornece o melhor suporte aos recursos avançados.</p> <p>Permite executar consultas SQL, chamar procedimentos armazenados (stored procedures), usar pureXML e acessar as informações de metadados.</p>
ibm_db_dbi	<p>Implementa a Especificação v2.0 de API de Banco de Dados Python.</p> <p>Não oferece alguns dos recursos avançados suportados pela API <code>ibm_db</code>.</p> <p>Se você tiver um aplicativo com um driver que suporte a Especificação v2.0 de API de Banco de Dados Python, é possível substituí-la facilmente pela <code>ibm_db</code>. As APIs <code>ibm_db</code> e <code>ibm_db_dbi</code> são fornecidas juntas.</p>
ibm_db_sa	<p>Suporta SQLAlchemy, um popular kit de ferramentas SQL Python de código-fonte aberto e mapeador objeto-relacional (ORM).</p>

Tabela 14.4 – IBM Data Server - extensões Python

14.4 XML e DB2 pureXML

A **Extensible Markup Language (XML)** é a tecnologia por trás das ferramentas e técnicas da Web 2.0, além da **arquitetura orientada a serviços (SOA)**. A IBM reconheceu cedo a importância da XML, fazendo grandes investimentos para oferecer a tecnologia pureXML - uma tecnologia que proporciona documentos XML com melhor suporte de armazenamento no DB2.

Introduzido em 2006, o DB2 9 é um servidor de dados híbrido: ele permite armazenamento nativo de dados relacionais, bem como dados hierárquicos. Embora as versões anteriores do DB2 e de outros servidores de dados do mercado possam armazenar documentos XML, o método de armazenamento usado no DB2 9 aumentou o desempenho e a flexibilidade. Com a tecnologia pureXML do DB2 9, os documentos XML são armazenados internamente seguindo uma análise hierárquica, como uma árvore; portanto, o trabalho com documentos XML fica imensamente aperfeiçoado. As versões mais recentes do DB2, como o DB2 9.5 e o DB2 9.7, melhoraram ainda mais o suporte a pureXML. O *Capítulo 15, DB2 pureXML* é dedicado a este assunto em detalhes.

14.5 Serviços de web

Como definição básica, pense em um serviço de Web como uma função que pode ser chamada através da rede, onde não é necessário conhecer a linguagem de programação usada em seu desenvolvimento, não é necessário conhecer o sistema operacional no qual a função será executada, nem é necessário conhecer a localização onde será executada. Os serviços de Web permitem que um aplicativo troque dados com outro aplicativo usando protocolos extensíveis do padrão da indústria, baseados em XML. Isto é ilustrado na *Figura 14.8*.

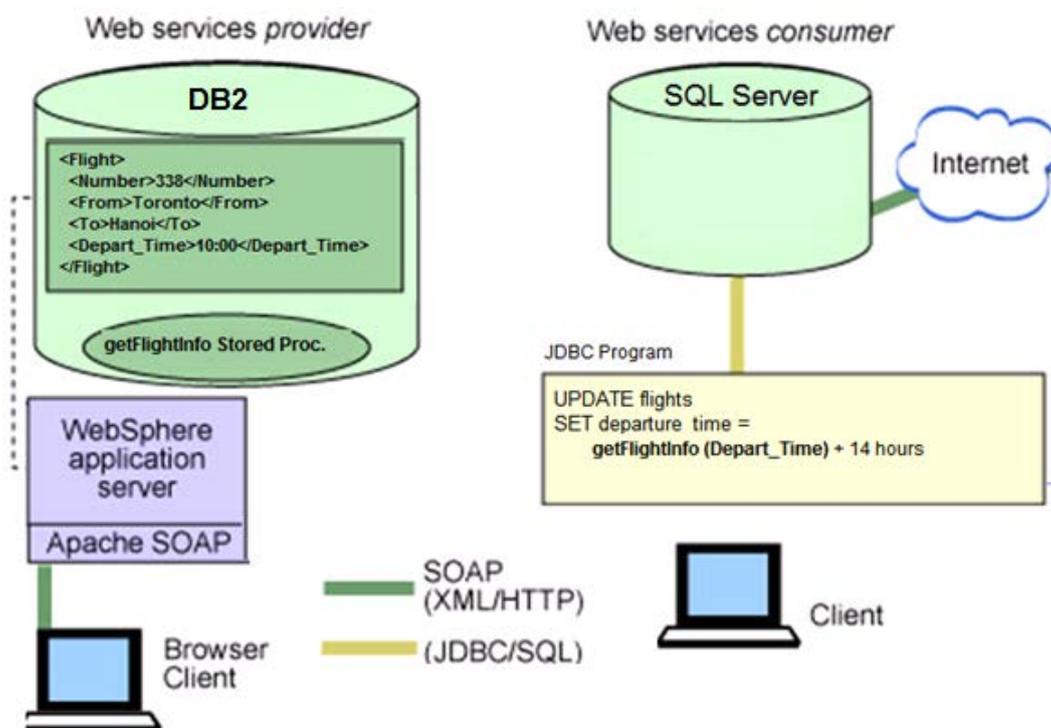


Figura 14.8 – Como funciona um exemplo de serviço de Web

Na figura, digamos que o lado esquerdo representa o sistema de uma companhia aérea fictícia, a Air Atlantis, que usa DB2 sobre Linux e armazena suas informações de voo em formato XML no banco de dados DB2. No lado direito, temos um sistema de outra companhia aérea fictícia, a Air Discovery, que usa SQL Server sendo executado em Windows. Agora digamos que a Air Atlantis e a Air Discovery assinem um contrato de parceria, onde ambas as empresas desejam compartilhar informações de programação e preços para coordenar seus voos. O compartilhamento de informações entre ambas pode ser um desafio, dado que as empresas estão usando sistemas operacionais diferentes (Linux, Windows) e servidores de dados diferentes (DB2, SQL Server). Quando a Air Atlantis altera sua programação de voos de uma viagem de Toronto a Pequim, como a Air Discovery pode ajustar sua própria programação de voos automaticamente para uma conexão de Pequim a Xangai? A resposta está nos serviços de Web. A Air Atlantis pode expor parte de suas informações de voo, criando um **Serviço de dados de Web** que

retorna a saída de um procedimento armazenado (**stored procedures**) (o procedimento armazenado **getFlightInfo**) com informações de voo do banco de dados do DB2. Um serviço de **dados** de web é um serviço de web baseado em informações de banco de dados. Quando este serviço de dados de web é implantado em um servidor de aplicativos como o WebSphere Application Server, um cliente ou parceiro como o Air Discovery pode usar um navegador para acessar facilmente as informações de voo da Air Atlantis. Neste exemplo, a Air Atlantis comporta-se como o fornecedor de serviços de Web, pois desenvolveu e disponibilizou o serviço de Web, enquanto a Air Discovery comporta-se como o consumidor de serviços de Web, pois está consumindo ou usando o serviço de Web.

A Air Discovery também pode chamar o serviço de Web de seu próprio aplicativo JDBC, para poder realizar cálculos usando dados de seu banco de dados do SQL Server. Por exemplo, se um voo de Toronto para Pequim leva em média 12 horas, a Air Discovery pode calcular o voo de conexão de Pequim para Xangai, somando o horário de partida do voo da Air Atlantis de Toronto, e a duração do voo somada a algumas horas de folga. A quantidade de horas a serem usadas de folga pode estar armazenada no banco de dados do SQL Server no sistema da Air Discovery, e a equação simples a ser usada no aplicativo JDBC teria a aparência abaixo:

Air Discovery Flight Departure time (Beijing to Shanghai)	=	Air Atlantis Flight Departure time (Toronto to Beijing)	+	Air Atlantis Flight Duration (Toronto to Beijing) 12 hours average	+	Air Atlantis Flight Buffer time (Toronto to Beijing) 2 hours
---	---	---	---	---	---	---

Se a Air Atlantis alterar seu horário de partida do voo, estas informações são comunicadas automaticamente ao sistema da Air Discovery, quando esta chamar o serviço de Web.

14.6 APIs administrativas

O DB2 fornece uma grande quantidade de APIs administrativas que podem ser usadas pelos desenvolvedores para construir seus próprios utilitários ou ferramentas. Por exemplo, para criar um banco de dados, você pode chamar a API **sqlcrea**; para iniciar uma instância, use a API **db2InstanceStart**; ou ainda, para importar dados para uma tabela, use a API **db2Import**. A lista completa está disponível no Information Center do DB2. Consulte a seção *Recursos* para obter o URL do Information Center do DB2.

14.7 Outros tipos de desenvolvimento

Alguns usuários do DB2 Express-C também interagem com produtos de terceiros, como o MS Excel e o MS Access, para criar formulários simples que se conectam ao DB2. Nesta seção, descreveremos como trabalhar com estes produtos e o DB2 Express-C.

O DB2 Express-C também está disponível em Mac OS X, de forma que você pode usar o DB2 nativamente para desenvolver aplicativos de banco de dados em um Mac. Isto pode ser especialmente atraente para a comunidade do RoR, que abraçou a plataforma Mac.

14.7.1 Como trabalhar com Microsoft Access e Microsoft Excel

O Microsoft Excel e o Microsoft Access são ferramentas populares usadas para gerar relatórios, criar formulários e desenvolver aplicativos simples que fornecem alguma inteligência de negócios aos seus dados. O DB2 interage com essas ferramentas com muita facilidade. Um DBA pode armazenar os dados da empresa em um servidor seguro do DB2 e os usuários regulares com Access ou Excel podem acessar esses dados e gerar relatórios. Isto está ilustrado na Figura 14.9.

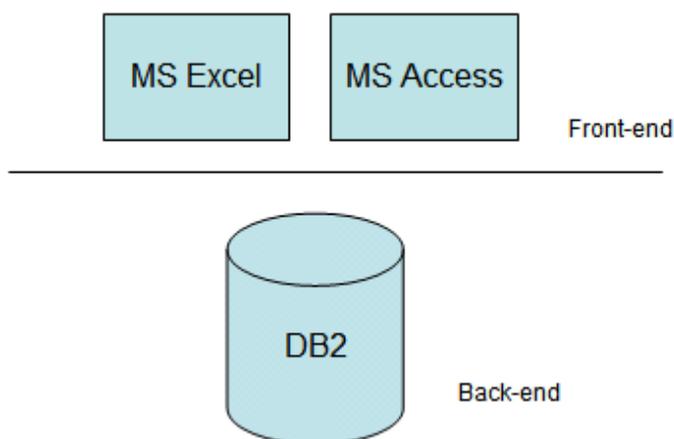


Figura 14.9 – Como trabalhar com Excel, Access e DB2

Na figura 14.9 acima, o Excel e o Access podem ser usados para desenvolver um aplicativo de front-end, enquanto o DB2 cuida da segurança dos dados, confiabilidade e desempenho, como back-end do aplicativo. A centralização de todos os dados no DB2 cria um modelo simplificado de armazenamento de dados.

No caso do Excel, a maneira mais fácil de obter acesso aos dados do DB2 é usar um driver OLE DB como o IBM OLE DB Provider for DB2. Este está incluído ao instalar o IBM Data Server Client gratuito, que pode ser obtido por download do site do DB2 Express-C em www.ibm.com/db2/express. Após a instalação, é necessário selecionar a fonte de dados com o provedor OLE DB apropriado para usar no menu do MS Excel. Selecione *Dados* → *Importar dados externos* → *Importar dados*. As próximas etapas são documentadas no artigo *IBM® DB2® Universal Database™ and the Microsoft® Excel Application Developer... for Beginners* (Banco de dados universal IBM DB2 e o desenvolvedor de aplicativos em Microsoft Excel... para Iniciantes) [1]. Consulte a seção *Referências* para obter mais detalhes.

No caso do Microsoft Access, você deve ter o seguinte instalado:

- IBM Data Server client ou
- IBM Data Server Driver for ODBC, CLI and .Net ou
- IBM Data Server Driver for ODBC and CLI

O IBM Data Server Driver for ODBC, CLI and .Net e o IBM Data Server Driver for ODBC and CLI também são conhecidos como o IBM DB2 ODBC Driver, que é o mesmo driver CLI do DB2. Este é o driver que deve ser usado para se conectar ao DB2 a partir do Access. Após a instalação do driver, crie um projeto no Access 2007 e selecione a opção *Banco de dados ODBC* disponível dentro da guia *Dados externos* na guia *Ferramentas de tabela*. As próximas etapas são documentadas no artigo *DB2 9 and Microsoft Access 2007 Part 1: Getting the Data...* (DB2 9 e Microsoft Access 2007 parte 1: obtenção dos dados...) [2]. Ao usar **tabelas vinculadas** no Microsoft Access, os dados ficam disponíveis para os usuários do Access 2007, mas os dados residem no servidor de dados do DB2.

Nas versões do Access anteriores a 2007, a configuração é um pouco diferente, mas você pode ler o artigo *Use Microsoft Access to interact with your DB2 data* (Uso do Microsoft Access para interagir com seus dados do DB2) [3]. Consulte a seção *Referências* para obter mais detalhes.

14.8 Ferramentas de desenvolvimento

O Microsoft Visual Studio e o Eclipse são dois dos mais populares ambientes de desenvolvimento integrado (IDEs) usado pelos desenvolvedores atualmente. Ambos os IDEs funcionam bem com o DB2.

Para o Microsoft Visual Studio, o DB2 fornece um add-in do Visual Studio para que, após a instalação, as ferramentas da IBM sejam adicionadas aos menus do Visual Studio. Dessa maneira, um desenvolvedor não precisa mudar para outras ferramentas para trabalhar com os bancos de dados do DB2. Você pode fazer o download do add-in do Visual Studio no site do DB2 Express-C em www.ibm.com/db2/express.

No que diz respeito ao Eclipse, a IBM lançou o IBM Data Studio, uma ferramenta gratuita baseada em Eclipse que permite desenvolver scripts SQL e XQuery, procedimentos armazenados (**Stored Procedures**), UDFs e serviços de Web. Como ele é baseado na plataforma Eclipse, muitos desenvolvedores podem aproveitar seus conhecimentos existentes para trabalhar com esta ferramenta.

14.9 Amostras de programas

Para ajudá-lo a aprender a programar em diferentes linguagens usando o DB2 como servidor de dados, você pode rever os aplicativos de amostra que são fornecidos com a instalação do servidor DB2 no diretório `SQLLIB\samples`. A *Figura 14.10* mostra alguns programas de amostra fornecidos com o DB2 em uma plataforma Windows.

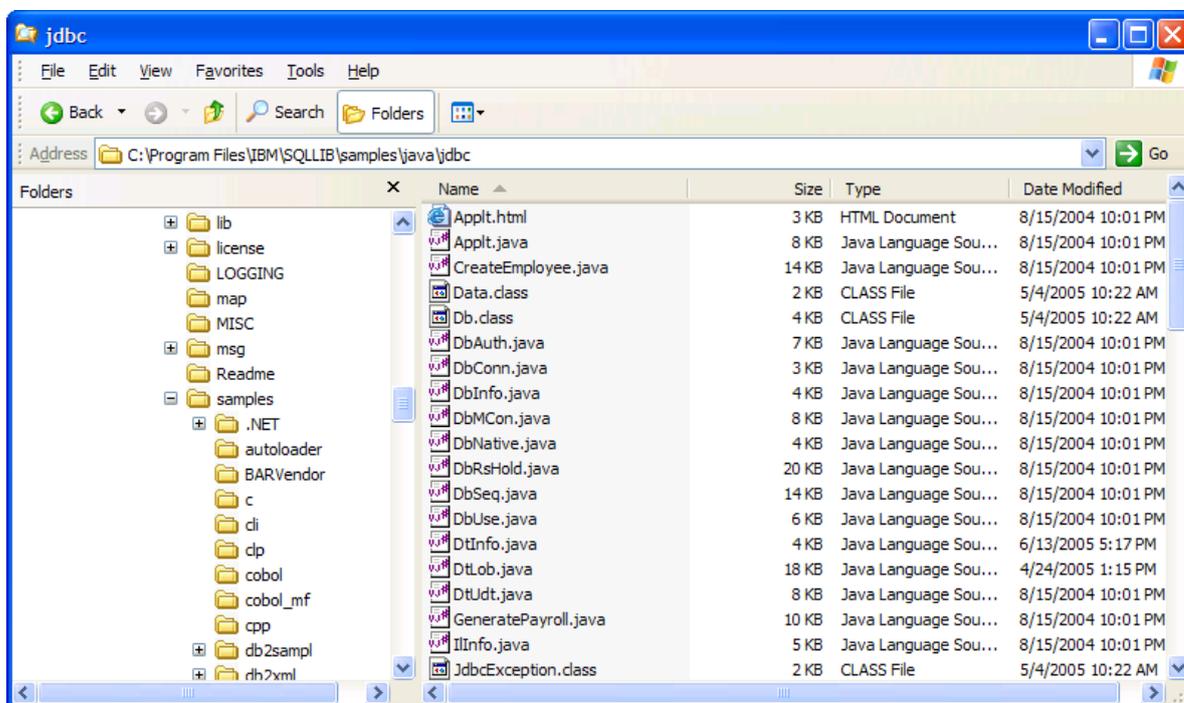


Figura 14.10 – Programas de amostra fornecidos com o DB2

14.10 Resumo

Neste capítulo, vimos como o DB2 fornece a flexibilidade de programar aplicativos de banco de dados dentro do banco de dados no servidor ou através de aplicativos no lado do cliente com conexões ao servidor de dados DB2.

A cobertura do aplicativo no lado do servidor inclui procedimentos armazenados (**Stored Procedures**), funções definidas pelo usuário (**UDF**) e disparadores (**Triggers**).

No lado do cliente, discutimos a enorme variedade de interfaces de programação e métodos permitidos pelo desenvolvimento de aplicativos no DB2, novamente exibindo a flexibilidade notável e a capacidade do DB2 como servidor de banco de dados.

15

Capítulo 15 – DB2 pureXML

Neste capítulo, discutiremos pureXML, a nova tecnologia introduzida no DB2 9 para oferecer suporte a armazenamento XML nativo. Muitos dos exemplos e conceitos discutidos neste capítulo foram tirados do IBM Redbook: *DB2 9: pureXML overview and fast start* (Visão geral e início rápido com pureXML). Consulte a seção Recursos para obter mais informações sobre este título. A *Figura 15.1* descreve que seção do “quadro geral” do DB2 discutiremos neste capítulo.

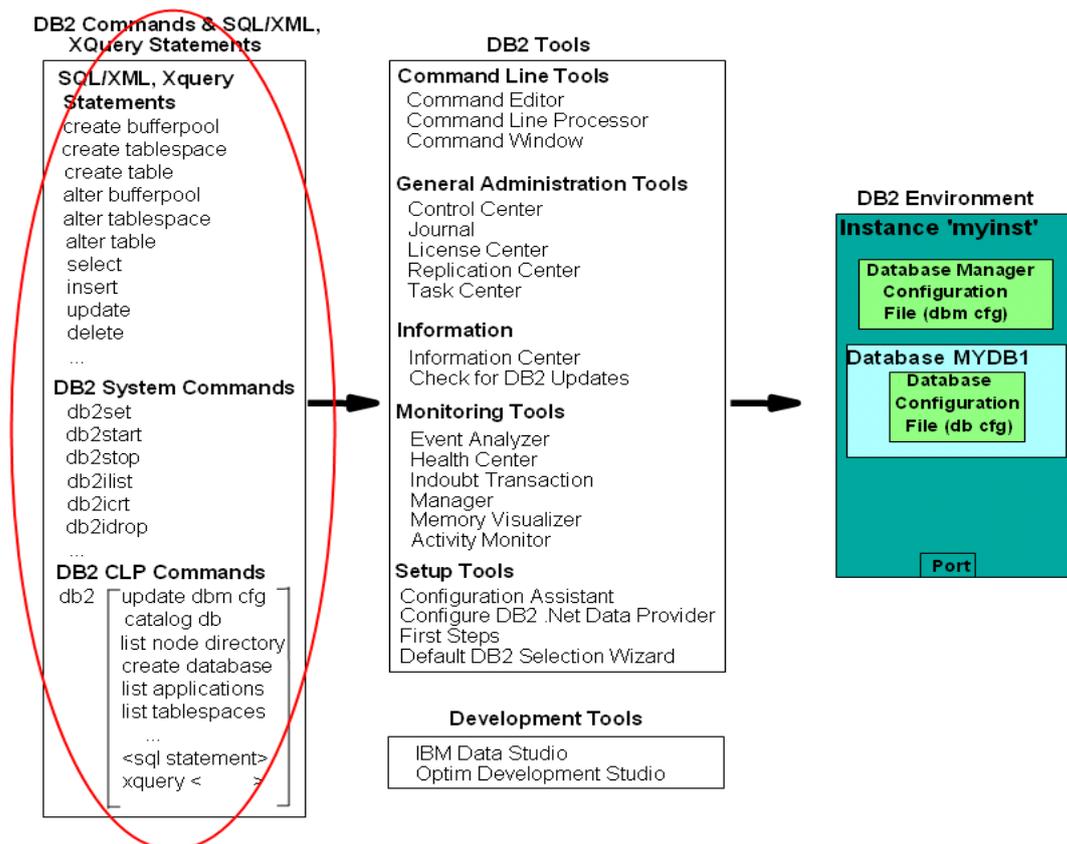


Figura 15.1 – O quadro geral do DB2: comandos do DB2, SQL/XML e XQuery

Observação:

Para obter mais informações sobre pureXML, assista ao vídeo:
<http://www.channeldb2.com/video/video/show?id=807741:Video:4382>

15.1 Como usar XML com bancos de dados

Os documentos XML podem ser armazenados em arquivos de texto, repositórios XML ou bancos de dados. Há dois motivos principais para que muitas empresas optem por armazená-los em bancos de dados:

- Gerenciar grandes volumes de dados XML é um problema de banco de dados. Os dados XML são como qualquer outro tipo de dados, exceto por apresentar um formato geral diferente. Os mesmos motivos que levam ao armazenamento de dados relacionais em bancos de dados se aplicam aos dados XML: Os bancos de dados fornecem busca e recuperação eficientes, suporte robusto à persistência de dados, backup e recuperação, suporte a transações, desempenho e escalabilidade.
- Integração: Ao armazenar dados relacionais e documentos XML juntos, você pode integrar novos dados XML aos dados relacionais existentes e combinar SQL a XPath ou XQuery em uma única consulta. Além disso, os dados relacionais podem ser publicados como XML e vice-versa. Através da integração, os bancos de dados podem oferecer um suporte melhor a aplicativos de Web, SOA e serviços de Web.

15.2 Bancos de dados XML

Há dois tipos de bancos de dados para armazenar dados XML:

- Bancos de dados com capacidade para XML
- Bancos de dados XML nativos

15.2.1 Bancos de dados com capacidade para XML

Um banco de dados com capacidade para XML utiliza o modelo relacional como modelo principal de armazenamento de dados para armazenar XML. Isto exige um mapeamento entre o modelo de dados XML (hierárquico) e o modelo de dados relacionais, ou então armazenar os dados XML como um objeto grande de caracteres. Embora isto possa ser considerado tecnologia “antiga”, ainda é usado por muitos fornecedores de bancos de dados. A *Figura 15.2* explica as duas opções de bancos de dados com capacidade para XML em mais detalhes.

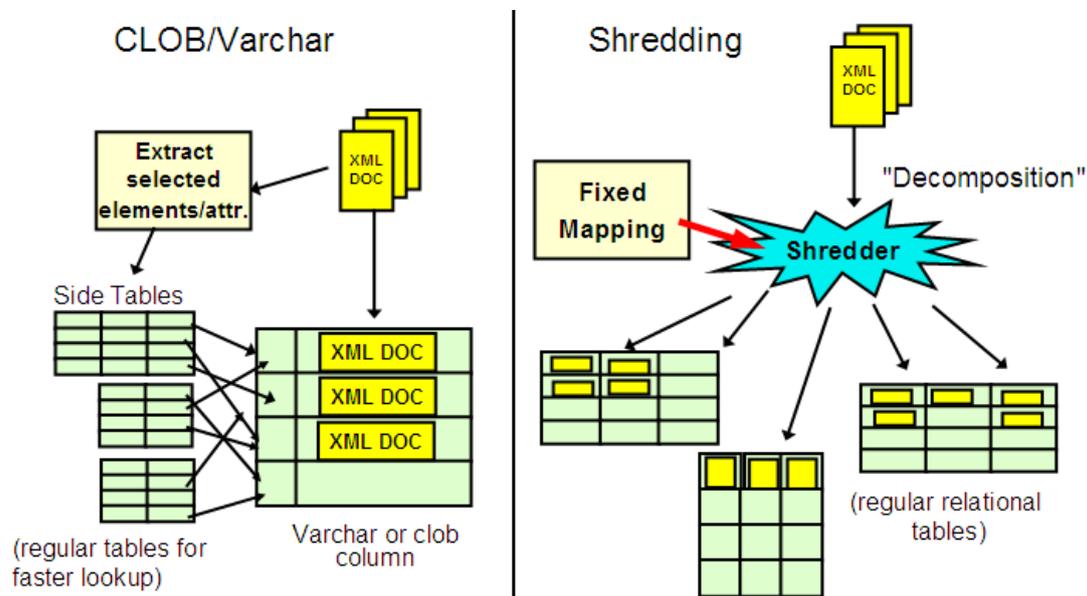


Figura 15.2 – Duas opções de armazenamento de XML em bancos de dados com capacidade para XML

O lado esquerdo da *Figura 15.2* mostra o **método CLOB e varchar** de armazenar documentos XML em um banco de dados. Usando este método, um documento XML é armazenado em uma cadeia de texto não analisada em um CLOB ou coluna varchar no banco de dados. Se o documento XML for armazenado como cadeia de texto, quando você quiser recuperar parte do documento XML, o programa terá que recuperar a cadeia inteira e analisá-la para encontrar o que você deseja. Pense em analisar como construir a árvore do documento XML na memória para que você possa navegar na árvore. Este método usa intensamente a memória e não é muito flexível.

A outra opção de bancos de dados com capacidade para XML é denominada **fragmentação ou decomposição** e é ilustrada no lado direito da *Figura 15.2*. Usando este método, um documento XML inteiro é fragmentado em partes menores, que são armazenadas em tabelas. Com este método, você literalmente força um documento XML, baseado no modelo hierárquico, a entrar no modelo relacional. Este método não é flexível, porque se o documento XML for alterado, a alteração não será propagada com facilidade para as tabelas correspondentes, e pode ser necessário criar muitas outras tabelas. Este método também não é bom para o desempenho: Se for necessário obter o documento XML original de volta, você terá que executar uma dispendiosa operação join (agrupar) de SQL, que pode se tornar ainda mais cara quando houver muitas tabelas envolvidas.

15.2.2 Bancos de dados XML nativos

Os bancos de dados XML nativos usam o modelo de dados hierárquico do XML para armazenar e processar o XML internamente. O formato de armazenamento é o mesmo do processamento: não há mapeamento para o modelo relacional e os documentos XML não são armazenados como cadeias de texto sem análise (CLOBs ou varchars). Quando são

usadas instruções XPath ou XQuery, elas são processadas nativamente pelo mecanismo e não convertidas para SQL. É por isso que estes bancos de dados são conhecidos como XML “nativo”. O DB2 é atualmente o único servidor de dados comercial que fornece este suporte.

15.3 XML no DB2

A *Figura 15.3* abaixo descreve como os dados relacionais e hierárquicos (documentos XML) são armazenados em um banco de dados híbrido do DB2. A figura também mostra a instrução `CREATE TABLE` que foi usada para criar a tabela `dept`.

```
CREATE TABLE dept (deptID CHAR(8),..., deptdoc XML);
```

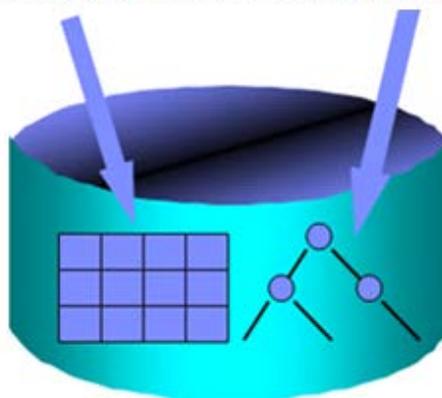


Figura 15.3 – XML no DB2

Observe que a definição da tabela usa um novo tipo de dados, XML, para a coluna `deptdoc`. Na figura, a seta para a esquerda indica a coluna relacional `deptID` armazenada em formato relacional (tabelas), enquanto a coluna XML `deptdoc` é armazenada em formato analisado hierárquico.

A *Figura 15.4* ilustra isso no DB2 9, onde agora há quatro maneiras de acessar dados:

- Usar SQL para acessar dados relacionais
- Usar SQL com extensões XML (SQL/XML) para acessar dados XML
- Usar XQuery para acessar dados XML
- Usar XQuery para acessar dados relacionais

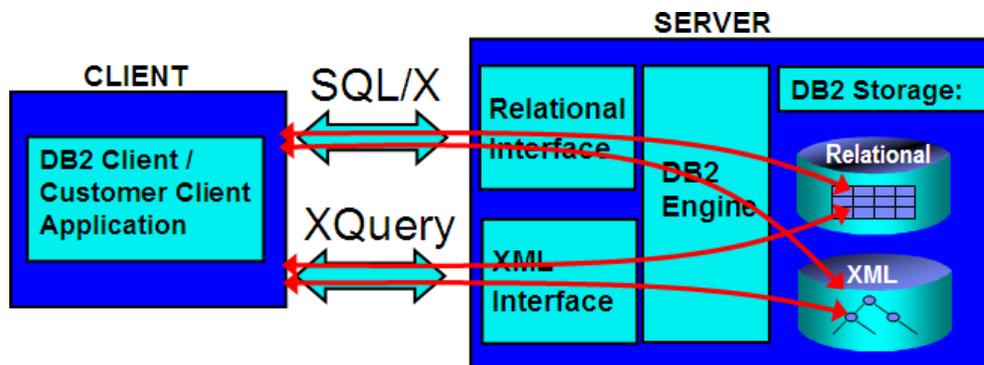


Figura 15.4 – Quatro maneiras de acessar os dados no DB2

Assim, dependendo do seu histórico, se você for uma pessoa que usa SQL, verá o DB2 como um RDBMS de primeira classe que também suporta XML. Se você for uma pessoa que usa XML, verá o DB2 como um repositório XML de primeira classe que também suporta SQL.

Observe que a IBM usa o termo *pureXML* em vez de *XML nativo* para descrever esta tecnologia. Embora outros fornecedores ainda usem as antigas tecnologias CLOB/varchar ou fragmentação para armazenar documentos XML, elas chamam essas tecnologias antigas de “XML nativo”. Para evitar confusão, a IBM decidiu usar o novo termo *pureXML* e registrar comercialmente este nome, para que nenhum outro banco de dados ou fornecedor de XML possa usar o mesmo termo para descrever uma tecnologia diferente. O suporte a *pureXML* é fornecido para bancos de dados criados como Unicode ou não Unicode.

15.3.1 As vantagens da tecnologia pureXML

A tecnologia *pureXML* oferece muitas vantagens.

1. Você pode alavancar seus investimentos em dados relacionais de maneira simples e ágil, uma vez que os documentos XML são armazenados em colunas de tabelas usando o novo tipo de dados XML.
2. Você pode reduzir a complexidade do código. Por exemplo, a *Figura 15.5* ilustra um script PHP escrito com e sem *pureXML*. Com *pureXML* (a caixa menor à esquerda) reduz a quantidade de linhas do código. Isto não apenas significa que o código é menos complexo, mas o desempenho geral melhora, pois há menos linhas a serem analisadas e mantidas no código.

```

<?php
$conn = db2_connect($dbname, $dbuser, $dbpass);

/* Insert Customer Documents */

$stmt = db2_prepare($conn, "VALUES (NEXT VALUE FOR
Cid)");
db2_execute($stmt);
list($Cid) = db2_fetch_array($stmt);

$fileContents = file_get_contents
("customers/c1.xml");

$stmt = db2_prepare($conn, "INSERT INTO xmlicustomer
(Cid, Info) VALUES (?, ?)");
if(!db2_execute($stmt, array($Cid, $fileContents)))
    echo db2_stmt_errormsg($stmt);
}

/* Insert Product Documents */

$fileContents = file_get_contents
("products/p1.xml");
$dom = simplexml_load_string($fileContents);

$prodID = (string) $dom["pid"];

$stmt = db2_prepare($conn, "INSERT INTO xmlproduct
(Pid, Description) VALUES (?, ?)");
if(!db2_execute($stmt, array($prodID,

```

```

...
}

db2_execute($stmt);
list($Cid) = db2_fetch_array($stmt);

$fileContents = file_get_contents
("customers/c1.xml");
$dom = simplexml_load_string($fileContents);

$custName = (string) $dom->name;
$custCountry = (string) $dom->addr->country;
$custStreet = (string) $dom->addr->street;
$custCity = (string) $dom->addr->city;
$custProvince = (string) $dom->addr->{"prov-state"};
$custZip = (string) $dom->addr->{"pcode-zip"};
$custPhone = (string) $dom->phone;

$stmt = db2_prepare($conn, "INSERT INTO sqlcustomer
(Cid, Name, Country, Street, City, Province, Zip,
Phone, Info) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)");
if(!db2_execute($stmt, array($Cid, $custName,
$custCountry, $custStreet, $custCity, $custProvince,
$custZip, $custPhone, $fileContents) )) {
    echo db2_stmt_errormsg($stmt);
}

/* Insert Product Documents */

$fileContents = file_get_contents
("products/p1.xml");
$dom = simplexml_load_string($fileContents);

$prodID = (string) $dom["pid"];

```

Figura 15.5 – Complexidade do código com e sem pureXML

3. É mais fácil alterar o esquema usando XML e tecnologia pureXML . A Figura 15.6 ilustra um exemplo deste aumento de flexibilidade. Na figura, suponha que você tivesse um banco de dados composto pelas tabelas *Employee* e *Department*. Normalmente, com um banco de dados não XML, se o gerente solicitasse que você armazenasse não apenas um telefone por funcionário (o telefone residencial), mas também um segundo telefone (um número de celular), você poderia adicionar uma coluna extra à tabela *Employee* e armazenar o número do celular na nova coluna. Entretanto, este método seria contra as regras de normalização dos bancos de dados relacionais. Se você deseja preservar estas regras, em vez disso seria necessário criar uma nova tabela *Phone* ao lado e mover todas as informações de telefone para esta tabela. Você poderia, então, adicionar também os números de celular. Criar uma nova tabela *Phone* é caro, não apenas porque é necessário mover grandes quantidades de dados pré-existentes, mas também porque seria necessário alterar o SQL de seus aplicativos para apontar para a nova tabela.

Em vez disso, no lado esquerdo da figura, mostramos como isso poderia ser feito usando XML. Se a funcionária *Christine* também tem um número de celular, é possível adicionar um novo indexador para acrescentar esta informação. Se o funcionário *Michael* não possui celular, deixamos do jeito que está.

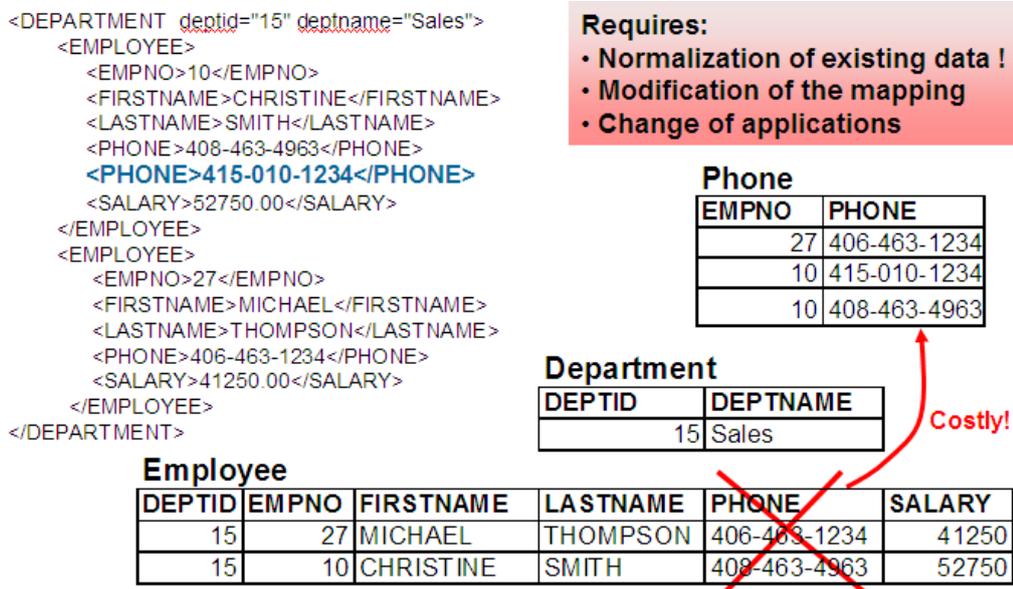


Figura 15.6 – Aumento da flexibilidade de dados usando XML

4. Você pode aumentar o desempenho do seu aplicativo XML. Os testes realizados usando a tecnologia pureXML demonstraram um aumento imenso de desempenho nos aplicativos XML. A Tabela 15.1 mostra os resultados dos testes para uma empresa que mudou das tecnologias mais antigas para pureXML. A segunda coluna mostra os resultados usando-se o método antigo de trabalho com XML usando outro banco de dados relacional e a terceira coluna mostra os resultados usando-se o DB2 com pureXML.

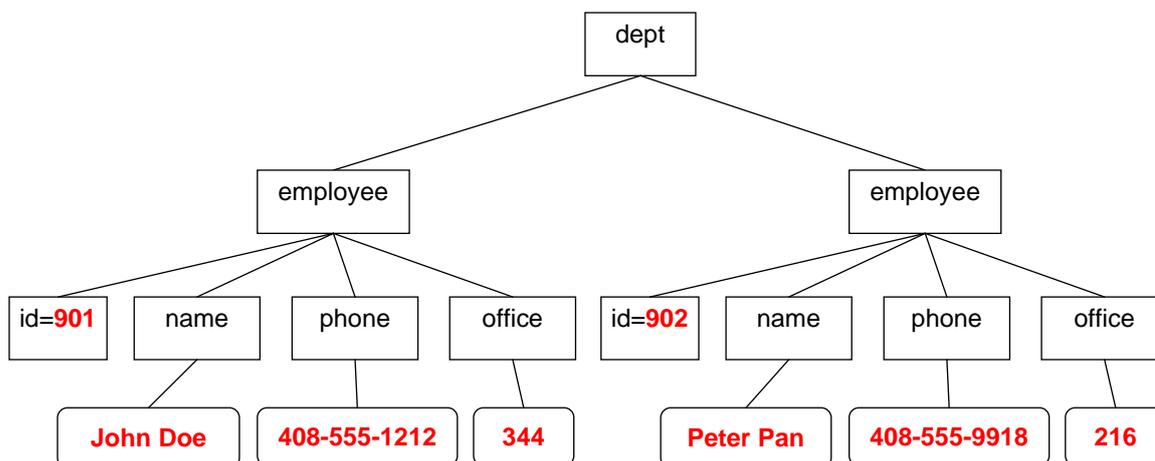
Tarefa	Outro banco de dados relacional	DB2 com pureXML
Desenvolvimento de processos de pesquisa e recuperação de negócios	CLOB: 8 h	30 min.
	Fragmentação: 2 h	
Linhas relativas de código de E/S	100	35 (65% de redução)
Adicionar campo ao esquema	1 semana	5 min.
Consultas	24 a 36 h	20 s a 10 min

Tabela 15.1 – Aumento do desempenho usando a tecnologia pureXML

15.3.2 Fundamentos de XPath

XPath é uma linguagem que pode ser usada para consultar documentos XML. A Listagem 15.1 mostra um documento XML e a Figura 15.7 ilustra o mesmo documento representado em formato hierárquico analisado (também chamado de “nó” ou “folha”). Usaremos o formato hierárquico analisado para explicar o funcionamento de XPath.

```
<dept bldg="101">
  <employee id="901">
    <name>John Doe</name>
  </employee>
  <employee id="902">
    <name>Peter Pan</name>
  </employee>
</dept>
```

Listagem 15.1 – Um documento XML**Figura 15.7 – Representação hierárquica analisada do documento XML da Listagem 15.1**

Uma maneira rápida de aprender XPath é compará-lo ao comando `change directory (cd)` do MS-DOS ou Linux/UNIX. Usando o comando `cd`, você percorre uma árvore de diretório como abaixo:

```
cd /directory1/directory2/...
```

De um modo semelhante, em XPath você usa as barras ("`/`") para ir de um elemento para outro dentro do documento XML. Por exemplo, usando o documento da *Listagem 15.1* em XPath, você poderia recuperar os nomes de todos os funcionários usando a consulta abaixo:

```
/dept/employee/name
```

15.3.2.1 Expressões XPath

As expressões XPath usam caminhos totalmente qualificados para especificar elementos e atributos. Um símbolo “@” é usado para especificar um atributo. Para recuperar somente o valor (nó de texto) de um elemento, use a função `text()`. A *Tabela 15.2* mostra consultas XPath e os resultados correspondentes usando o documento XML da *Listagem 15.1*.

XPath	Resultado
<code>/dept/@bldg</code>	101
<code>/dept/employee/@id</code>	901 902
<code>/dept/employee/name</code>	<name>Peter Pan</name> <name>John Doe</name>
<code>/dept/employee/name/text()</code>	Peter Pan John Doe

Tabela 15.2 – Exemplos de expressão XPath

15.3.2.2 Curingas XPath

Há dois curingas principais em XPath:

- “*” corresponde a qualquer nome de indexador
- “/” é o curinga “descendente-ou-próprio”

A *Tabela 15.3* fornece mais exemplos usando o documento XML da *Listagem 15.1*

XPath	Resultado
<code>/dept/employee/*/text()</code>	John Doe 408 555 1212 344 Peter Pan 408 555 9918 216
<code>/dept/*/@id</code>	901 902

//name/text()	Peter Pan John Doe
/dept//phone	<phone>408 555 1212</phone> <phone>408 555 9918</phone>

Tabela 15.3 – Exemplos de curingas XPath

15.3.2.3 Predicados XPath

Os predicados são limitados por colchetes []. Como analogia, pense neles como o equivalente à cláusula WHERE em SQL. Por exemplo, [*@id="902"*] pode ser lido como: "WHERE atributo id é igual a 902". Pode haver diversos predicados em uma única expressão XPath. Para especificar um predicado de posição use [n], que significa que o n^o filho seria selecionado. Por exemplo, *employee[2]* significa que o segundo funcionário deve ser o selecionado. A *Tabela 15.4* fornece mais exemplos.

XPath	Resultado
/dept/employee[@id="902"]/name	<name>Peter Pan</name>
/dept[@bldg="101"]/employee[office > "300"]/name	<name>John Doe</name>
//employee[office="344" OR office="216"]/@id	901 902
/dept/employee[2]/@id	902

Tabela 15.4 – Exemplos de predicado XPath

15.3.2.4 O eixo pai

Semelhante ao MS-DOS ou Linux/UNIX, você pode usar "." (ponto) para indicar na expressão que está se referindo ao contexto atual, e ".." (ponto ponto) para se referir ao contexto pai. A *Tabela 15.5* fornece mais exemplos.

XPath	Resultado
/dept/employee/name[../@id="902"]	<name>Peter Pan</name>
/dept/employee/office[.>"300"]	<office>344</office>
/dept/employee[office > "300"]/office	<office>344</office>
/dept/employee[name="John Doe"]/../@bldg	101
/dept/employee/name[.="John Doe"]/../../@bldg	101

Tabela 15.5 – O eixo pai em XPath

15.3.3 Fundamentos do XQuery

XQuery é uma linguagem de consultas criada para XML. XQuery suporta expressões de caminho para navegar na estrutura hierárquica de XML. De fato, XPath é um subconjunto de XQuery; assim, tudo o que aprendemos antes sobre XPath também se aplica a XQuery. XQuery suporta dados com e sem tipo. XQuery não trabalha com valores nulos, porque os documentos XML omitem dados ausentes ou desconhecidos. As expressões XQuery e XPath fazem distinção entre maiúsculas e minúsculas, e XQuery retorna sequências de dados XML.

XQuery suporta a expressão FLWOR. Usando SQL como analogia, ela equivale à expressão SELECT-FROM-WHERE. A próxima seção descreve FLWOR em mais detalhes.

15.3.3.1 XQuery: expressão FLWOR

FLWOR significa:

- FOR: repetição através de uma sequência, associa uma variável a itens
- LET: associa uma variável a uma sequência
- WHERE: elimina os itens da repetição
- ORDER: reordena os itens da repetição
- RETURN: constrói os resultados da consulta

É uma expressão que permite manipular os documentos XML, possibilitando o retorno de outra expressão. Por exemplo, suponha uma tabela com esta definição:

```
CREATE TABLE dept(deptID CHAR(8),deptdoc XML);
```

E o documento XML da *Listagem 15.2* é inserido na coluna *deptdoc*

```
<dept bldg="101">
  <employee id="901">
    <name>John Doe</name>
    <phone>408 555 1212</phone>
    <office>344</office>
  </employee>
  <employee id="902">
    <name>Peter Pan</name>
    <phone>408 555 9918</phone>
    <office>216</office>
  </employee>
</dept>
```

Listagem 15.2 – Uma amostra de documento XML

Assim, a instrução XQuery da *Listagem 15.3* usando a expressão FLWOR poderia ser executada:

```
xquery
for $d in db2-fn:xmlcolumn('dept.deptdoc')/dept
let $emp := $d//employee/name
where $d/@bldg > 95
order by $d/@bldg
return
  <EmpList>
    {$d/@bldg, $emp}
  </EmpList>
```

Listagem 15.3 – Uma amostra de instrução XQuery com a expressão FLWOR

Isto retornaria a saída mostrada na *Listagem 15.4*

```
<EmpList bldg="101">
  <name>
    John Doe
  </name>
  <name>
    Peter Pan
  </name>
</EmpList>
```

Listagem 15.4 – Saída após a execução da instrução XQuery da Listagem 15.3

15.3.4 Inserção de documentos XML

Para inserir documentos XML em um banco de dados DB2, use a instrução SQL INSERT ou o utilitário IMPORT. Não é possível usar XQuery com esta finalidade, pois isto ainda não foi definido no padrão.

Vamos examinar o script `table_creation.txt` mostrado na *Listagem 15.5* abaixo, que pode ser executado na Command Window do DB2 ou no shell do Linux usando a seguinte instrução:

```
db2 -tvf table_creation.txt

-- (1)
drop database mydb
;

-- (2)
create database mydb using codeset UTF-8 territory US
;

-- (3)
connect to mydb
```

```

;

-- (4)
create table items (
  id          int primary key not null,
  brandname   varchar(30),
  itemname    varchar(30),
  sku         int,
  srp        decimal(7,2),
  comments    xml
);

-- (5)
create table clients(
  id          int primary key not null,
  name        varchar(50),
  status      varchar(10),
  contact     xml
);

-- (6)
insert into clients values (77, 'John Smith', 'Gold',
  '<addr>111 Main St., Dallas, TX, 00112</addr>')
;

-- (7)
IMPORT FROM "D:\Raul\clients.del" of del xml from "D:\Raul" INSERT INTO
CLIENTS (ID, NAME, STATUS, CONTACT)
;

-- (8)
IMPORT FROM "D:\Raul\items.del" of del xml from "D:\Raul" INSERT INTO
ITEMS (ID, BRANDNAME, ITEMNAME, SKU, SRP, COMMENTS)
;

```

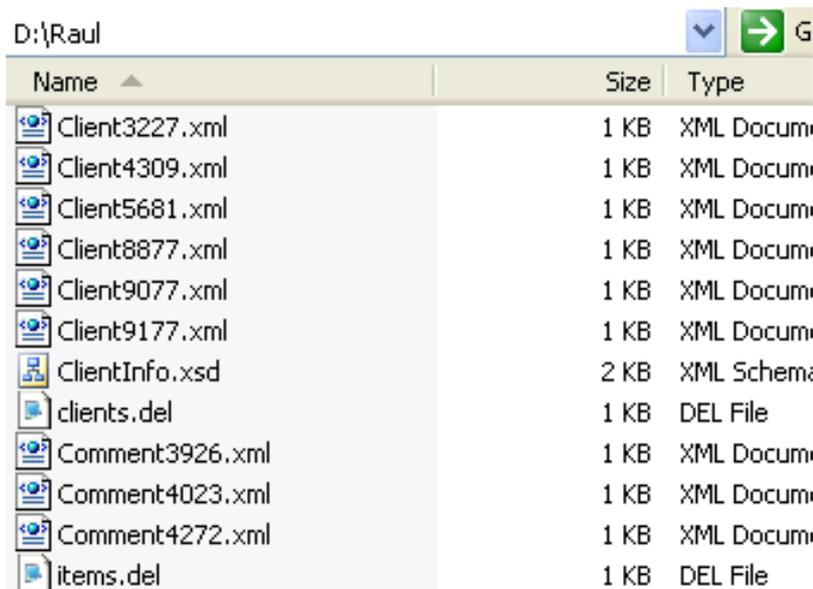
Listagem 15.5 – Conteúdo do arquivo table_creation.txt

Observe que este arquivo de script e os arquivos relacionados são fornecidos no arquivo compactado **Expressc_book_exercises_9.7.zip** que acompanha este livro. Acompanhe enquanto descrevemos cada linha do script da *Listagem 15.5*.

1. Descarte o banco de dados *mydb*. Nos arquivos de script, isto normalmente é feito para realizar uma limpeza. Se *mydb* não existia antes, você receberá uma mensagem de erro, mas isto está certo.
2. Crie o banco de dados *mydb* usando o código UTF-8. Isto cria um banco de dados Unicode. O pureXML é suportado em ambos os bancos de dados: Unicode ou não.

3. Conecte-se ao banco de dados *mydb* que acabou de ser criado. Isto é necessário para criar os objetos dentro do banco de dados.
4. Crie a tabela *items*. Observe que a última coluna da tabela (coluna *comments*) é definida como uma coluna XML usando o novo tipo de dados XML.
5. Criamos a tabela *clients*. Observe que a última coluna da tabela (coluna *contact*) também é definida com o novo tipo de dados XML.
6. Usando esta instrução SQL INSERT, insira um documento XML em uma coluna XML. Na instrução INSERT, passe o documento XML como uma cadeia de texto delimitada por aspas simples.
7. Usando o comando IMPORT, você pode inserir ou importar diversos documentos XML junto com dados relacionais para o banco de dados. Na etapa (7), estamos importando os dados do arquivo *clients.del* (um arquivo ascii delimitado), e também indicamos onde estão localizados os dados XML referenciados no arquivo *clients.del* (neste exemplo, em *D:\Raul*).

Vamos observar o arquivo *clients.del* com mais atenção mas primeiramente, vejamos o conteúdo do diretório *D:\Raul*. A *Figura 15.8* fornece estas informações.



Name	Size	Type
Client3227.xml	1 KB	XML Document
Client4309.xml	1 KB	XML Document
Client5681.xml	1 KB	XML Document
Client8877.xml	1 KB	XML Document
Client9077.xml	1 KB	XML Document
Client9177.xml	1 KB	XML Document
ClientInfo.xsd	2 KB	XML Schema
clients.del	1 KB	DEL File
Comment3926.xml	1 KB	XML Document
Comment4023.xml	1 KB	XML Document
Comment4272.xml	1 KB	XML Document
items.del	1 KB	DEL File

Figura 15.8 – Conteúdo do diretório *D:\Raul* com documentos XML

A *Listagem 15.6* mostra o conteúdo do arquivo de texto *clients.del*.

```
3227,Ella Kimpton,Gold,<XDS FIL='Client3227.xml' />,
8877,Chris Bontempo,Gold,<XDS FIL='Client8877.xml' />,
9077,Lisa Hansen,Silver,<XDS FIL='Client9077.xml' />
9177,Rita Gomez,Standard,<XDS FIL='Client9177.xml' />,
```

```
5681,Paula Lipenski,Standard,<XDS FIL='Client5681.xml' />,
4309,Tina Wang,Standard,<XDS FIL='Client4309.xml' />
```

Listagem 15.6 – Conteúdo do arquivo clients.del

No arquivo `clients.del` “XDS FIL=” é usado para apontar para um documento XML específico.

A Figura 15.9 mostra o Control Center após a execução do script acima.

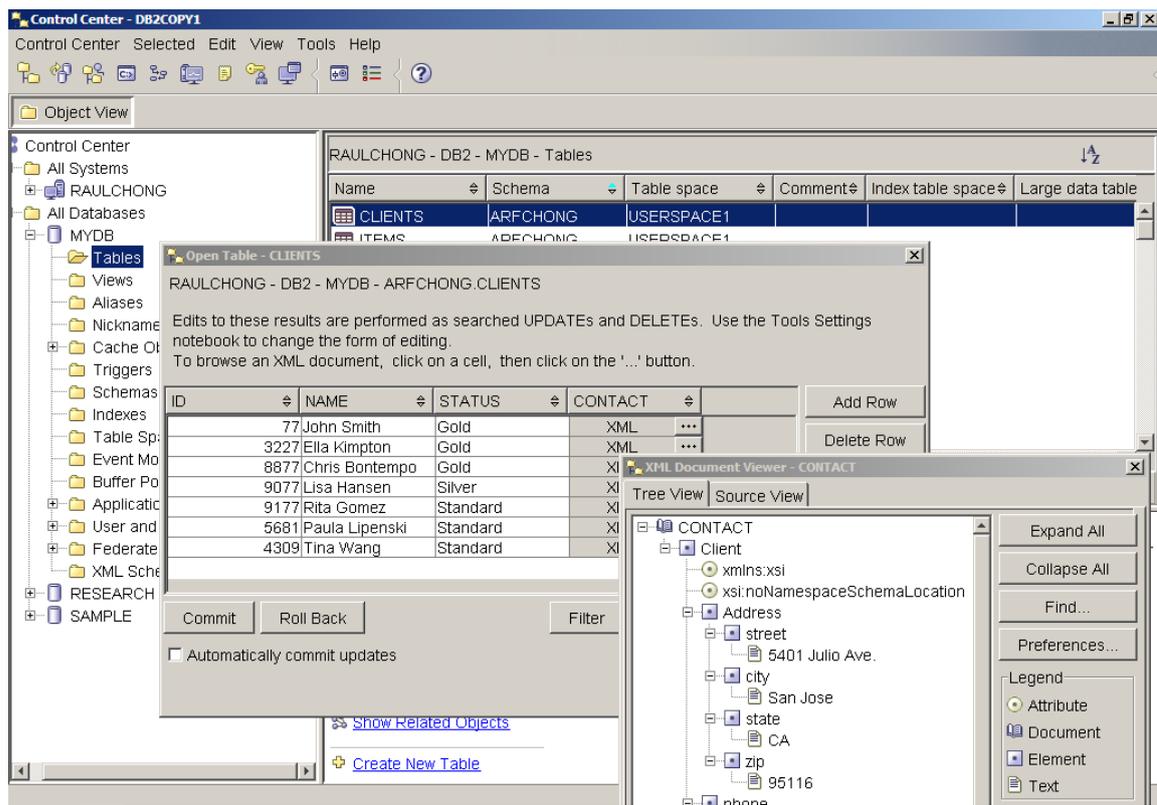


Figure 15.9 – O Control Center após a execução do script table_creation.txt

Observe que mostramos na figura o conteúdo da tabela `CLIENTS`. A última coluna `contact` é uma coluna XML. Ao clicar no botão com três pontos, abre-se outra janela exibindo o conteúdo do documento XML. Isto é mostrado no canto inferior direito da *Figura 15.9*.

15.3.5 Como consultar dados XML

Há duas maneiras de consultar dados XML no DB2:

- Usando SQL com extensões XML (SQL/XML)
- Usando XQuery

Em ambos os casos, o DB2 segue os padrões XML internacionais.

15.3.5.1 Como consultar dados XML com SQL/XML

O uso de instruções comuns em SQL permite trabalhar com linhas e colunas. Uma instrução SQL pode ser usada para trabalhar com documentos XML completos; entretanto, isso não é útil ao tentar recuperar somente parte do documento. Nesse caso, é necessário usar SQL com extensões XML (SQL/XML).

A *Tabela 15.6* descreve algumas das funções de SQL/XML disponíveis no padrão SQL 2006

Nome da função	Descrição
XMLPARSE	Analisa os caracteres ou grande objeto de dados binários, produz valores em XML
XMLSERIALIZE	Converte um valor XML em caracteres ou grande objeto de dados binários
XMLVALIDATE	Valida o valor XML em relação a um esquema XML e anota o tipo do valor XML
XMLEXISTS	Determina se uma consulta XQuery retorna um resultado (ou seja, uma sequência de um ou mais itens)
XMLQUERY	Executa uma consulta XQuery e retorna a sequência de resultados
XMLTABLE	Executa uma consulta XQuery, retorna a sequência de resultados como uma tabela relacional (se possível)
XMLCAST	Converte de ou para um tipo XML

Tabela 15.6 – Funções de SQL/XML

Os exemplos abaixo podem ser testados com o banco de dados *mydb* criado anteriormente.

Exemplo 1

Imagine que você precisa localizar os nomes de todos os clientes que residem em um CEP específico. A tabela `clients` armazena os endereços dos clientes, incluindo CEP, em uma coluna XML. Usando `XMLEXISTS`, você pode pesquisar o CEP desejado na coluna XML e restringir o conjunto de resultados retornado de acordo. A *Listagem 15.7* abaixo ilustra a consulta necessária.

```
SELECT name FROM clients
WHERE xmlexists(
    '$c/Client/Address[zip="95116"]'
    passing clients.contact as "c"
)
```

Listagem 15.7 – Um exemplo usando XMLEXISTS

Na *Listagem 15.7*, a primeira linha é uma cláusula SQL especificando que você deseja recuperar informações na coluna `name` da tabela `clients`.

A cláusula WHERE chama a função XMLEXISTS, especificando a expressão XPath que orienta o DB2 a navegar até o elemento `zip` e verificar o valor 95116

A cláusula `$c/Client/Address` indica o caminho dentro da hierarquia do documento XML onde o DB2 pode localizar o elemento `zip`. Um cifrão (\$) é usado para especificar uma variável; portanto, “c” é uma variável. Em seguida, esta variável é definida pela linha: `passing clients.contact as "c"`. Aqui, `clients` é o nome da tabela e `contact` é o nome da coluna com tipo de dados XML. Em outras palavras, estamos passando o documento XML para a variável “c”.

O DB2 inspeciona os dados XML contidos na coluna `contact`, navega desde o nó raiz `Client` até o nó `Address` e, em seguida, até o nó `zip`, e finalmente determina se o cliente reside no CEP desejado. A função XMLEXISTS avalia se é “true” (verdadeiro) e o DB2 retorna o nome do cliente associado a essa linha.

A partir do DB2 9.5, a consulta acima pode ser simplificada como mostrado na Listagem 15.8 abaixo.

```
SELECT name FROM clients
WHERE xmlexists(
    '$CONTACT/Client/Address[zip="95116"]'
)
```

Listagem 15.8 – Versão simplificada da consulta mostrada na Listagem 15.7

Uma variável com o mesmo nome de uma coluna XML é criada automaticamente pelo DB2. No exemplo acima, a variável `CONTACT` é criada automaticamente pelo DB2. Seu nome corresponde ao nome da coluna XML `CONTACT`.

Exemplo 2

Vamos pensar agora em como resolver o problema de criar uma listagem de relatório com os endereços de email dos clientes com status “Gold”. A consulta na *Listagem 15.9* abaixo pode ser executada com esta finalidade.

```
SELECT xmlquery('$c/Client/email' passing contact as "c")
FROM clients
WHERE status = 'Gold'
```

Listagem 15.9 – Um exemplo usando XMLQUERY

A primeira linha indica que desejamos retornar o endereço de email, que é um elemento do documento XML (não é uma coluna relacional). Como no exemplo anterior, “c” é uma variável contendo o documento XML. Neste exemplo usamos a função XMLQUERY, que pode ser usada após SELECT, enquanto a função XMLEXISTS pode ser usada após a cláusula WHERE.

Exemplo 3

Em algumas situações, talvez você deseje apresentar dados XML na forma de tabelas. Isto é possível com a função XMLTABLE como mostrado na *Listagem 15.10* abaixo.

```
SELECT t.comment#, i.itemname, t.customerID, Message
FROM items i,
xmltable('$c/Comments/Comment' passing i.comments as "c"
columns Comment# integer path 'CommentID',
CustomerID integer path 'CustomerID',
Message varchar(100) path 'Message') AS t
```

Listagem 15.10 – Um exemplo usando XMLTABLE

A primeira linha especifica as colunas que serão incluídas no conjunto de resultados. As colunas prefixadas com a variável “t” são baseadas em valores de elementos XML.

A terceira linha chama a função XMLTABLE para especificar a coluna XML do DB2 que contém os dados desejados (*i.comments*) e o caminho, dentro dos documentos XML, da coluna onde estão localizados os elementos que interessam.

A cláusula *columns*, que abrange as linhas 4 a 6, identifica os elementos XML específicos que serão mapeados nas colunas de saída do conjunto de resultados SQL especificado na linha 1. Parte deste mapeamento envolve especificar os tipos de dados para os quais serão convertidos os valores dos elementos XML. Neste exemplo, todos os dados XML são convertidos para tipos de dados SQL tradicionais.

Exemplo 4

Agora, vamos explorar um exemplo simples, no qual será incluída uma expressão FLWOR de XQuery dentro de uma função XMLQUERY de SQL/XML. Isto está ilustrado na *Listagem 15.11*.

```
SELECT name, xmlquery(
  'for $e in $c/Client/email[1] return $e'
  passing contact as "c"
)
FROM clients
WHERE status = 'Gold'
```

Listagem 15.11 – Um exemplo usando XMLQUERY e FLWOR

A primeira linha especifica que os nomes de clientes e a saída da função XMLQUERY serão incluídos no conjunto de resultados. A segunda linha indica que o primeiro subelemento *email* do elemento *Client* deverá ser retornado. A terceira linha identifica a fonte dos dados XML (a coluna *contact*). A quarta linha informa que esta coluna é originária da tabela *clients*, e a quinta linha indica que somente nos interessam os clientes *Gold*.

Exemplo 5

O exemplo ilustrado na *Listagem 15.12* demonstra novamente a função XMLQUERY, que assume uma expressão FLWOR de XQuery; entretanto, observe que dessa vez estamos retornando não apenas XML, mas também HTML.

```
SELECT xmlquery('for $e in $c/Client/email[1]/text()
  return <p>{$e}</p>'
  passing contact as "c")
FROM clients
WHERE status = 'Gold'
```

Listagem 15.12 – Um exemplo retornando XML e HTML

A cláusula de retorno de XQuery possibilita transformar uma saída XML conforme necessário. O uso da função `text()` na primeira linha indica que nosso interesse está somente na representação de texto do primeiro endereço de email dos clientes qualificados. A segunda linha especifica que estas informações devem ser envolvidas por tags de parágrafo HTML.

Exemplo 6

O exemplo a seguir usa a função `XMLEMENT` para criar uma série de elementos de item, cada um contendo subelementos para os valores de ID, nome da marca e unidade de manutenção de estoque (SKU) obtidos nas colunas correspondentes da tabela `items`. Basicamente, podemos usar a função `XMLEMENT` quando desejamos converter dados relacionais em XML. Isto é ilustrado na Listagem 15.13.

```
SELECT
  xmlelement (name "item", itemname),
  xmlelement (name "id", id),
  xmlelement (name "brand", brandname),
  xmlelement (name "sku", sku)
FROM items
WHERE srp < 100
```

Listagem 15.13 – Um exemplo usando XMLEMENT

A consulta na *Listagem 15.13* retornaria a saída mostrada na *Listagem 15.14*

```
<item>
  <id>4272</id>
  <brand>Classy</brand>
  <sku>981140</sku>
</item>
...
<item>
  <id>1193</id>
  <brand>Natural</brand>
  <sku>557813</sku>
</item>
```

Listagem 15.14 – Saída da consulta na Listagem 15.13

15.3.5.2 Como consultar dados XML com XQuery

Na seção anterior, examinamos como consultar dados XML usando SQL com extensões XML. O SQL sempre foi o método de consulta principal, com XPath ou XQuery incorporado ao SQL. Nesta seção, discutiremos como consultar dados XML usando XQuery. Desta vez, o método de consulta principal será XQuery e em alguns casos, usaremos SQL incorporado em XQuery (usando a função `db2-fn:sqlquery`). Ao usar XQuery, chamaremos algumas funções e também usaremos a expressão FLWOR.

Exemplo 1

Esta é uma consulta XQuery simples para retornar os dados de contato do cliente. Neste exemplo, `CONTACT` é o nome da coluna XML e `CLIENTS` é o nome da tabela.

```
xquery db2-fn:xmlcolumn('CLIENTS.CONTACT')
```

Sempre coloque prefixo em qualquer expressão XQuery com o comando `xquery` para que o DB2 saiba que deve usar o analisador XQuery; caso contrário, o DB2 assumirá que você está tentando executar uma expressão SQL. A função `db2-fn:xmlcolumn` recupera os documentos XML da coluna especificada como parâmetro. Ela equivale à instrução SQL abaixo, pois está recuperando o conteúdo da coluna inteira:

```
SELECT contact FROM clients
```

Exemplo 2

Neste exemplo mostrado na *Listagem 15.15*, usamos a expressão FLWOR para recuperar dados de fax dos clientes

```
xquery
  for $y in db2-fn:xmlcolumn('CLIENTS.CONTACT')/Client/fax
  return $y
```

Listagem 15.15 – Xquery e a expressão FLWOR

A primeira linha chama o analisador XQuery. A segunda linha orienta o DB2 a iterar através dos subelementos de fax contidos na coluna `CLIENTS.CONTACT`. Cada elemento de fax está vinculado à variável `$y`. A terceira linha indica que, para cada repetição, é retornado o valor “`$y`”.

A saída desta consulta é ilustrada na *Listagem 15.16* (Omitimos o namespace na saída para simplificar a leitura; caso contrário ela ocuparia muitas linhas):

```
<fax>4081112222</fax>
<fax>5559998888</fax>
```

Listagem 15.16 – Saída da consulta mostrada na Listagem 15.15

Exemplo 3

O exemplo da Listagem 15.17 consulta dados XML e retorna os resultados como HTML.

```
xquery
  <ul> {
    for $y in db2-fn:xmlcolumn('CLIENTS.CONTACT')/Client/Address
    order by $y/zip
    return <li>{$y}</li>
  }
</ul>
```

Listagem 15.17 – Instrução XQuery com a expressão FLWOR retornando HTML

A amostra de HTML retornada teria a aparência mostrada na *Listagem 15.18*.

```
<ul>
<li>
<address>
  <street>9407 Los Gatos Blvd.</street>
  <city>Los Gatos</city>
  <state>ca</state>
  <zip>95302</zip>
</address>
</li>
<address>
<street>4209 El Camino Real</street>
  <city>Mountain View</city>
  <state>CA</state>
  <zip>95302</zip>
</address>
</li>
...
</ul>
```

Listagem 15.18 – Saída da consulta executada na Listagem 15.17

Exemplo 4

O exemplo abaixo mostra como incorporar SQL dentro de XQuery, usando a função `db2-fn:sqlquery`. A função `db2-fn:sqlquery` executa uma consulta SQL e retorna somente os dados XML selecionados. A consulta SQL passada para `db2-fn:sqlquery` deve retornar somente dados XML. Esses dados XML poderão ser processados em seguida pela XQuery. Isto é ilustrado na Listagem 15.19.

```
xquery
  for $y in
  db2-fn:sqlquery(
    'select comments from items where srp > 100'
  )/Comments/Comment
  where $y/ResponseRequested='Yes'
```

```

return (
  <action>
    {$y/ProductID
     $y/CustomerID
     $y/Message}
  </action>
)

```

Listagem 15.19 – Um exemplo da função `db2-fn:sqlquery` incorporando SQL em XQuery

No exemplo, a consulta SQL filtra as linhas com base na condição de que a coluna `srp` tenha valor superior a 100. Das linhas filtradas, ela seleciona a coluna `comments`, que é a coluna XML. Em seguida, a consulta XQuery (ou XPath) é aplicada para ir para os subelementos.

Observação:

SQL não faz distinção entre maiúsculas e minúsculas, e o DB2 armazena todos os nomes de tabela e coluna em maiúsculas, como padrão. Por outro lado, XQuery faz distinção entre maiúsculas e minúsculas. As funções acima são funções de interface XQuery, portanto todos os nomes de tabela e de coluna devem ser passados a estas funções em maiúsculas. Se os nomes de objetos forem passados em minúsculas, poderá ocorrer um erro de nome de objeto indefinido.

15.3.6 Operações Join com SQL/XML

Esta seção descreve como executar operações JOIN entre duas colunas XML de tabelas diferentes ou entre uma coluna XML e uma coluna relacional. Suponha que tenha criado duas tabelas com as instruções mostradas na *Listagem 15.20*

```

CREATE TABLE dept (unitID CHAR(8), deptdoc XML)
CREATE TABLE unit (unitID CHAR(8) primary key not null,
                   name CHAR(20),
                   manager VARCHAR(20),
                   ...
)

```

Listagem 15.20 – DDL de tabelas a serem usadas nos exemplos de JOIN

A operação JOIN pode ser realizada de duas maneiras. O primeiro método é mostrado na *Listagem 15.21*.

```

SELECT u.unitID
FROM dept d, unit u
WHERE XMLEXISTS (
  '$e//employee[name = $m]'
  passing d.deptdoc as "e", u.manager as "m")

```

Listagem 15.21 – Primeiro método de execução de JOIN com SQL/XML

A linha 4 da instrução na listagem acima mostra que a operação JOIN ocorre entre o elemento **name**, que é um subelemento da coluna XML **deptdoc** na tabela **dept**, e a coluna relacional **manager** na tabela **unit**.

A *Listagem 15.22* mostra o segundo método de execução da operação JOIN.

```
SELECT u.unitID
FROM dept d, unit u
WHERE u.manager = XMLCAST(
    XMLQUERY('$e//employee/name '
    passing d.deptdoc as "e")
AS char(20))
```

Listagem 15.22 – Segundo método de execução de JOIN com SQL/XML

Neste segundo método, a coluna relacional está no lado esquerdo de JOIN. Se a coluna relacional estiver no lado esquerdo da equação, pode-se usar um índice relacional em vez de índice XML.

15.3.7 Joins com XQuery

Suponhamos que as tabelas abaixo foram criadas:

```
CREATE TABLE dept (unitID CHAR(8), deptdoc XML)
CREATE TABLE project(projectDoc XML)
```

Se usarmos SQL/XML, uma operação JOIN teria a aparência mostrada na *Listagem 15.23*.

```
SELECT XMLQUERY (
    '$d/dept/employee' passing d.deptdoc as "d")
FROM dept d, project p
WHERE XMLEXISTS (
    '$e/dept[@deptID=$p/project/deptID]'
    passing d.deptdoc as "e", p.projectDoc as "p")
```

Listagem 15.23 – JOIN com SQL/XML

A operação equivalente a JOIN usando XQuery é mostrada na *Listagem 15.24*.

```
xquery
for $dept in db2-fn:xmlcolumn("DEPT.DEPTDOC")/dept
for $proj in db2-fn:xmlcolumn("PROJECT.PROJECTDOC")/project
where $dept/@deptID = $proj/deptID
return $dept/employee
```

Listagem 15.24 – Uma operação JOIN com XQuery

Este segundo método é mais fácil de interpretar -- a variável **\$dept** contém o documento XML da coluna XML **deptdoc** da tabela **dept**. A variável **\$proj** contém o documento XML da coluna XML **projectdoc** da tabela **project**. Em seguida, a linha 4 realiza a operação JOIN entre um atributo do primeiro documento XML e um elemento do segundo documento XML.

15.3.8 Operações de atualização e exclusão

As operações de atualização e exclusão em dados XML podem ser realizadas de duas maneiras:

- Usando as instruções SQL UPDATE e DELETE
- Usando a expressão TRANSFORM

Para a primeira maneira, usando as instruções SQL UPDATE e DELETE, a atualização ou exclusão ocorre no nível do documento, ou seja, o documento XML inteiro é substituído pelo atualizado. Por exemplo, na instrução UPDATE na *Listagem 15.25* abaixo, mesmo que você deseje alterar somente o elemento <state>, na realidade o documento XML inteiro é substituído.

```
UPDATE clients SET contact=(
  xmlparse(document
    '<Client>
      <address>
        <street>5401 Julio ave.</street>
        <city>San Jose</city>
        <state>CA</state>
        <zip>95116</zip>
      </address>
      <phone>
        <work>4084633000</work>
        <home>4081111111</home>
        <cell>4082222222</cell>
      </phone>
      <fax>4087776666</fax>
      <email>newemail@someplace.com</email>
    </Client>')
  )
WHERE id = 3227
```

Listagem 15.25 – Um exemplo de instrução SQL UPDATE

No segundo método, você pode realizar atualizações em subdocumentos usando a expressão TRANSFORM, que é muito mais eficiente. Isto permite substituir, inserir, excluir ou renomear os nós de um documento XML. Você também pode alterar o valor de um nó sem substituir o nó em si, normalmente para alterar o valor de um elemento ou atributo—que é um tipo de atualização muito comum. Este suporte foi adicionado no DB2 9.5.

A expressão TRANSFORM é parte da linguagem XQuery; você pode usá-la em qualquer lugar onde normalmente usaria XQuery, por exemplo, em uma expressão FLWOR ou na função XMLQUERY em uma instrução SQL/XML. O uso mais típico é em uma instrução SQL UPDATE para modificar um documento XML em uma coluna XML.

A Listagem 15.26 mostra a sintaxe da expressão TRANSFORM.

```
>>--transform--| copy clause |--| modify clause |--| return clause |--><
```

copy clause

```

      .-,-----
      v                                     |
|--copy---$VariableName--:=---CopySourceExpression+-----|

```

modify clause

```
|--modify--ModifyExpression-----|
```

return clause

```
|--return--ReturnExpression-----|
```

Listagem 15.26 – A sintaxe da expressão TRANSFORM

A cláusula `copy` é usada para atribuir os documentos XML que você deseja processar a uma variável.

Na cláusula `modify`, você pode chamar uma expressão `insert`, `delete`, `rename` ou `replace`. Estas expressões permitem realizar atualizações no documento XML.

Por exemplo:

- Se você deseja adicionar novos nós ao documento, use a expressão `insert`
- Para excluir nós de um documento XML, use a expressão `delete`
- Para renomear um elemento ou atributo no documento XML, use a expressão `rename`
- Para substituir um nó existente por um novo nó ou sequência de nós, use a expressão `replace`. O valor de substituição da expressão só pode ser usado para alterar o valor de um elemento ou atributo.

A cláusula `return` retorna o resultado da expressão `transform`.

A *Listagem 15.27* mostra um exemplo de instrução `UPDATE` usando a expressão `TRANSFORM`.

```

(1)-- UPDATE customers
(2)-- SET contactinfo = xmlquery( 'declare default element namespace
(3)--                               "http://posample.org";
(4)--   transform
(5)--   copy $newinfo := $c
(6)--       modify do insert <email2>my2email.gm.com</email2>
(7)--       as last into $newinfo/customerinfo
(8)--   return $newinfo' passing contactinfo as "c")
(9)-- WHERE id = 100

```

Listagem 15.27 – Uma operação UPDATE usando a expressão TRANSFORM

No exemplo acima, as linhas (1), (2) e (9) fazem parte da sintaxe UPDATE de SQL. A linha (2) é usada para chamar a função XMLQUERY, que chama a expressão transform da linha (4). O bloco da expressão transform vai da linha (4) à linha (8), e é usado para inserir um novo nó no documento XML que contém o elemento *email2*. Observe que a atualização de elementos de um documento XML através de uma visualização não é suportada.

A exclusão de documentos XML inteiros de tabelas é tão simples quanto usar a instrução SELECT em SQL/XML. Use a instrução SQL DELETE e especifique qualquer predicado WHERE necessário.

15.3.9 Indexação XML

Em um documento XML, é possível criar índices para os elementos, atributos ou valores (nós de texto). A seguir, alguns exemplos. Suponha que foi criada a tabela abaixo:

```
CREATE TABLE customer(info XML)
```

E suponha que o documento XML da Listagem 15.28 é um dos documentos armazenados na tabela.

```
<customerinfo Cid="1004">
  <name>Matt Foreman</name>
  <addr country="Canada">
    <street>1596 Baseline</street>
    <city>Toronto</city>
    <state>Ontario</state>
    <pcode>M3Z-5H9</pcode>
  </addr>
  <phone type="work">905-555-4789</phone>
  <phone type="home">416-555-3376</phone>
  <assistant>
    <name>Peter Smith</name>
    <phone type="home">416-555-3426</phone>
  </assistant>
</customerinfo>
```

Listagem 15.28 – O documento XML a ser usado nos exemplos relacionados aos índices XML

A instrução mostrada na *Listagem 15.29* cria um índice no atributo *cid*

```
CREATE UNIQUE INDEX idx1 ON customer(info)
  GENERATE KEY USING
  xmlpattern '/customerinfo/@Cid'
  AS sql DOUBLE
```

Listagem 15.29 – Um índice no atributo Cid

A instrução mostrada na *Listagem 15.30* cria um índice no elemento *name*

```
CREATE INDEX idx2 ON customer(info)
  GENERATE KEY USING
  xmlpattern '/customerinfo/name'
  AS sql VARCHAR(40)
```

Listagem 15.30 – Um índice no elemento name

A instrução mostrada na *Listagem 15.31* cria um índice em todos os elementos *name*

```
CREATE INDEX idx3 ON customer(info)
  GENERATE KEY USING
  xmlpattern '//name'
  AS sql VARCHAR(40);
```

Listagem 15.31 – Um índice em todos os elementos name

A instrução mostrada na *Listagem 15.32* cria um índice em todos os nós de texto (todos os valores). Isto não é recomendado, pois seria muito caro manter o índice para as operações de atualização, exclusão e inserção, e o índice ficaria grande demais.

```
CREATE INDEX idx4 ON customer(info)
  GENERATE KEY USING
  xmlpattern '//text()'
  AS sql VARCHAR(40);
```

Listagem 15.32 – Um índice em todos os nós de texto (não recomendado)

15.4 Como trabalhar com esquemas XML

O DB2 permite inserir um documento XML no banco de dados caso esteja bem estruturado. Caso contrário, você receberá uma mensagem de erro no momento da inserção. Por outro lado, o DB2 não força você a validar um documento XML. Se desejar validar um documento XML, há diversas alternativas que discutiremos nesta seção.

15.4.1 Registro dos esquemas XML

Os esquemas XML são armazenados nos bancos de dados DB2 dentro do que é chamado de repositório do esquema XML. Para adicionar um esquema XML a um repositório, use o comando REGISTER XMLSCHEMA.

Por exemplo, digamos que há um documento XML armazenado no arquivo `order.xml` como mostrado na *Figura 15.10*

```
<?xml version="1.0" encoding="UTF-8"?>
<po:PurchaseOrder xmlns:po="http://www.test.com/po">
  <Header>
    <Id>1</Id>
    <date>2004-01-29</date>
    <description>purchase order</description>
    <value>20</value>
    <status>shipped</status>
  </Header>
  <Items>
    <Item>
      <ItemDescription color="red" weight="5">
        <Name>Widget C</Name>
        <SKU>1</SKU>
        <Price>30</Price>
        <Comment>no comment</Comment>
      </ItemDescription>
      <NumberOrdered>1</NumberOrdered>
    </Item>
  </Items>
  <Customer type="regular">
    <Name>Manoj K Sardana</Name>
    <Address>ring road, bangalore</Address>
    <Phone>918051055109</Phone>
    <email>msardana@in.ibm.com</email>
  </Customer>
</po:PurchaseOrder>
```

Figura 15.10 – O arquivo order.xml contendo um documento XML

Agora digamos que há um esquema XML armazenado no arquivo order.xsd como mostrado na *Figura 15.11*

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.test.com/po"
  xmlns:po="http://www.test.com/po"
  xmlns:head="http://www.test.com/header"
  xmlns:prod = "http://www.test.com/product"
  xmlns:cust = "http://www.test.com/customer">

  <xsd:import namespace="http://www.test.com/product" schemaLocation="product.xsd" />
  <xsd:import namespace="http://www.test.com/customer" schemaLocation="customer.xsd" />
  <xsd:import namespace="http://www.test.com/header" schemaLocation="header.xsd" />
  <xsd:complexType name="itemType">
    <xsd:sequence>
      <xsd:element name="Item" minOccurs="1" maxOccurs="unbounded" >
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="ItemDescription" type="prod:prodType" />
            <xsd:element name="NumberOrdered" type="xsd:integer" />
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name="potype">
    <xsd:sequence>
      <xsd:element name="Header" type="head:headerType" />
      <xsd:element name="Items" type="po:itemType" />
      <xsd:element name="Customer" type="cust:customerType" />
    </xsd:sequence>
  </xsd:complexType>

  <xsd:element name="PurchaseOrder" type="po:potype" />

</xsd:schema>

```

Figura 15.11 – O arquivo order.xsd contendo um esquema XML

Neste documento de esquema XML, usamos uma elipse para destacar o seguinte:

- `<xsd:schema>`: Indica que é um documento de esquema XML
- `<xsd:import ...>`: Importamos outros arquivos xsd (outros esquemas XML) que seriam parte deste esquema XML maior.
- `minOccurs="1"`: Um exemplo de “regra” de esquema XML, onde para o elemento `Item` dizemos que deve ocorrer pelo menos uma vez, ou em outras palavras, deve haver pelo menos um elemento `Item`.

Em seguida, para registrar o esquema XML no banco de dados, podemos usar um script semelhante ao mostrado na *Listagem 15.33* abaixo. O script inclui comentários, explicando-se por si mesmo.

```

-- CONECTAR-SE AO BANCO DE DADOS
CONNECT TO SAMPLE;

-- REGISTRAR O ESQUEMA XML PRINCIPAL
REGISTER XMLSCHEMA http://www.test.com/order FROM D:\example3\order.xsd AS
order;

```

```
-- ADICIONAR O DOCUMENTO DO ESQUEMA XML AO ESQUEMA PRINCIPAL
ADD XMLSCHEMA DOCUMENT TO order ADD http://www.test.com/header FROM
D:\example3\header.xsd;
```

```
-- ADICIONAR O DOCUMENTO DO ESQUEMA XML AO ESQUEMA PRINCIPAL
ADD XMLSCHEMA DOCUMENT TO order ADD http://www.test.com/product FROM
D:\example3\product.xsd;
```

```
-- ADICIONAR O DOCUMENTO DO ESQUEMA XML AO ESQUEMA PRINCIPAL
ADD XMLSCHEMA DOCUMENT TO order ADD http://www.test.com/customer FROM
D:\example3\customer.xsd;
```

```
-- CONCLUIR O REGISTRO DO ESQUEMA
COMPLETE XMLSCHEMA order;
```

Listagem 15.33 – Uma amostra de script mostrando as etapas para registrar um esquema XML

Para verificar estas informações posteriormente, execute SELECT nas informações das tabelas de catálogo, como mostrado na *Listagem 15.34* abaixo.

```
SELECT CAST(OBJECTSCHEMA AS VARCHAR(15)), CAST(OBJECTNAME AS VARCHAR(15))
FROM syscat.xsobjects
WHERE OBJECTNAME='ORDER';
```

Listagem 15.34 – Recuperação das informações do esquema XML das tabelas de catálogo do DB2

15.4.2 Validação de esquema XML

Após registrar os esquemas XML no DB2, você pode validar os documentos XML de duas maneiras:

- Usar a função XMLVALIDATE durante uma instrução INSERT
- Usar um disparador BEFORE

A *Figura 15.12* mostra um exemplo no qual o documento XML mostrado na *Figura 15.10* é validado de acordo com o esquema XML mostrado na *Figura 15.11*.

```

DROP TABLE t1;
CREATE TABLE t1 (po xml);

INSERT INTO t1 VALUES(xmlvalidate(xmlparse(document('<?xml version="1.0" encoding="UTF-8"?>
<po:PurchaseOrder xmlns:po="http://www.test.com/po">
  <Header>
    <Id>1</Id>
    <date>2004-01-29</date>
    <description>purchase order</description>
    <value>20</value>
    <status>shipped</status>
  </Header>
  <Items>
    <Item>
      <ItemDescription color="red" weight="5">
        <Name>Widget C</Name>
        <SKU>1</SKU>
        <Price>30</Price>
        <Comment>no comment</Comment>
      </ItemDescription>
      <NumberOrdered>1</NumberOrdered>
    </Item>
  </Items>
  <Customer type="regular">
    <Name>Manoj K Sardana</Name>
    <Address>ring road, bangalore</Address>
    <Phone>918051055109</Phone>
    <email>msardana@in.ibm.com</email>
  </Customer>
</po:PurchaseOrder>')) ACCORDING TO XMLSCHEMA ID order));

```

Figura 15.12 – Validação do esquema XML usando XMLVALIDATE

Para testar se um documento XML foi validado, você pode usar o predicado “IS VALIDATED” em uma restrição CHECK.

Você pode validar documentos XML em uma coluna, usando esquemas XML diferentes. Isto é importante para uma migração fácil da versão 1 para a versão 2 de um esquema XML. Na mesma coluna XML, também pode haver documentos XML sem qualquer validação. Isto é útil se os documentos forem recebidos de fontes confiáveis ou não, onde somente estas últimas exigem validação do esquema.

15.4.3 Outro suporte a XML

Agora, pequenos documentos XML podem ser alinhados na tabela base. Isto significa que os dados XML são armazenados no mesmo lugar dos dados relacionais e podem tirar proveito dos mesmos mecanismos de compressão que os dados relacionais comuns. Os documentos XML maiores são armazenados em um objeto interno separado, que também pode ser compactado.

O DB2 também suporta a evolução de esquema XML. Isto significa que se o esquema XML mudar, você pode atualizá-lo facilmente com o comando UPDATE XMLSCHEMA. Se as alterações no esquema XML forem muito drásticas, é possível que você receba alguns erros.

O DB2 também suporta a decomposição ou “fragmentação” de XML. Este é o método “antigo” de armazenar o XML nos bancos de dados, e é o que os outros fornecedores usam para armazenar XML. O DB2 ainda suporta este método, se você quiser usá-lo; no entanto, recomendamos usar pureXML. O DB2 também suporta o XML Extender usando o antigo método para armazenar XML, mas este não será mais aperfeiçoado.

Com o DB2 9.7, todos os benefícios do pureXML foram ampliados para as partições de banco de dados usadas normalmente em depósitos de dados. O Database Partitioning Feature (DPF) é oferecido com o DB2 Enterprise Edition.

new in
V9.7

15.6 Resumo

Este capítulo foi uma introdução ao XML e à tecnologia pureXML. O uso de documentos XML cresce exponencialmente, devido às ferramentas e técnicas da Web 2.0, além da SOA. Ao armazenar documentos XML em um banco de dados do DB2, você pode tirar proveito da segurança, desempenho e flexibilidade de programação usando pureXML. A tecnologia pureXML permite armazenar documentos XML em formato hierárquico analisado, como uma árvore, e isto é realizado no momento da inserção no banco de dados. No momento da consulta, não há necessidade de analisar o documento XML para construir uma árvore antes do processamento. A árvore do documento XML já foi construída e armazenada no banco de dados. Além disso, a tecnologia pureXML usa um mecanismo XML nativo que compreende XQuery. Portanto, não há necessidade de mapear XQuery para SQL, que é o que fazem os outros produtos de RDBMS.

O capítulo também explicou como inserir, excluir, atualizar e consultar documentos XML usando SQL/XML e XQuery. Também discutiu os índices XML, o esquema XML e outros recursos como a compressão e a evolução de esquema XML.

15.7 Exercícios

Ao longo deste capítulo, você viu diversos exemplos de sintaxe SQL/XML e XQuery, e foi apresentado ao DB2 Command Editor e ao IBM Data Studio. Neste exercício, você irá testar seu conhecimento em SQL/XML e XQuery e ganhar experiência com estas ferramentas. Usaremos o banco de dados **mydb** criado usando o arquivo de script **table_creation.txt** que foi explicado anteriormente neste capítulo (*Listagem 15.5*).

Procedimento

1. Crie o banco de dados **mydb** e carregue os dados XML, como discutido antes neste capítulo. O arquivo **table_creation.txt** foi incluído no arquivo anexo **Expressc_book_exercises_9.7.zip** na pasta Chapter 2 (Capítulo 2). Execute o arquivo de script **table_creation.txt** em uma Command Window do DB2 ou no shell do Linux, na forma a seguir:

```
db2 -tvf table_creation.txt
```

2. Se o script falhar em qualquer uma das etapas, procure descobrir qual é o problema, lendo as mensagens de erro. Um problema típico ao executar o script é que você pode precisar alterar os caminhos dos arquivos, pois eles podem estar localizados em diretórios diferentes. Você sempre pode descartar o banco de dados e começar novamente, executando este comando na Command Window do DB2 ou no shell do Linux:

```
db2 drop database mydb
```

3. Se receber uma mensagem de erro devido às conexões ativas ao tentar descartar o banco de dados, execute este comando primeiro:

```
db2 force applications all
```

4. Após executar o script com sucesso, use o Control Center do DB2 ou o IBM Data Studio para confirmar se as tabelas **items** e **clients** foram criadas e que contêm 4 e 7 linhas, respectivamente.
5. Com o banco de dados **mydb** criado e as duas tabelas carregadas, você agora pode conectar-se a ele e executar as consultas mostradas nas *Listagens 15.7 a 15.19*

A

Apêndice A – Solução de problemas

Este apêndice discute como resolver problemas que podem ser encontrados ao se trabalhar com o DB2. A *Figura A.1* fornece uma rápida visão geral das ações que devem ser tomadas caso ocorra um problema.

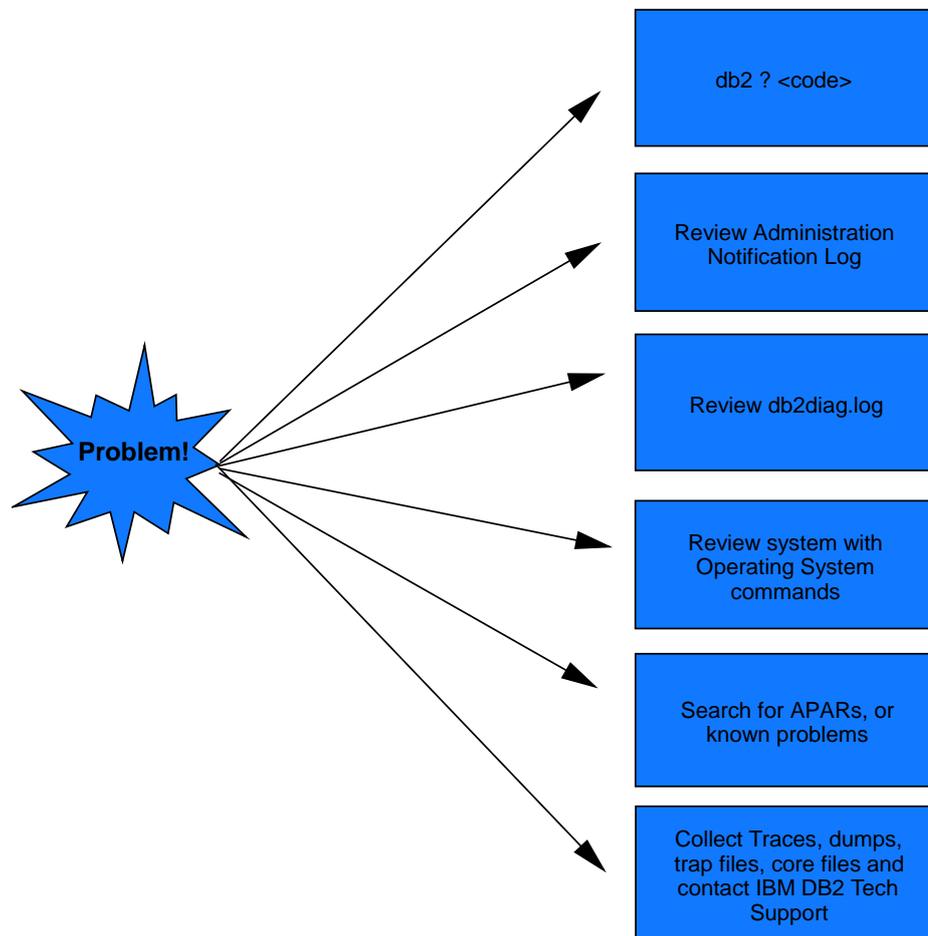


Figura A.1 – Visão geral da solução de problemas

Observação:

Para obter mais informações sobre solução de problemas, assista ao vídeo:

<http://www.channeldb2.com/video/video/show?id=807741:Video:4462>

A.1 Obter mais informações sobre os códigos de erro

Para obter mais informações sobre um código de erro recebido, digite o código usando uma interrogação como prefixo na área de entrada do Command Editor e clique no botão *Execute* (Executar). Isto é mostrado na *Figura A.2*.

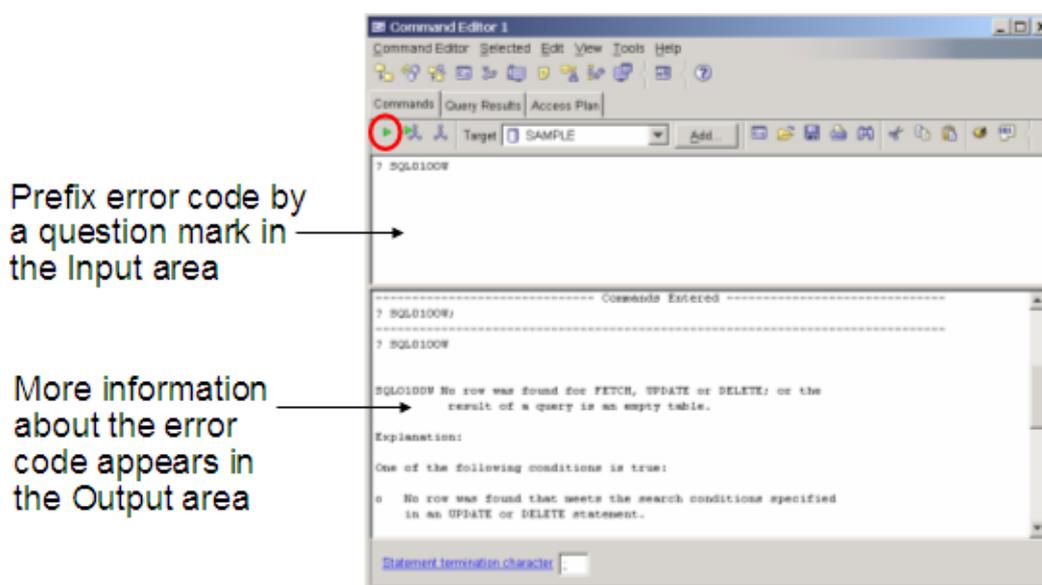


Figura A.2 – Como obter mais informações sobre os códigos de erro do DB2

A interrogação (?) chama o comando de ajuda do DB2. Veja abaixo diversos exemplos de como chamá-lo para obter ajuda caso receba, por exemplo, o código de erro de SQL “-104”. Todos os exemplos abaixo são equivalentes.

```

db2 ? SQL0104N
db2 ? SQL104N
db2 ? SQL-0104
db2 ? SQL-104
db2 ? SQL-104N

```

A.2 SQLCODE e SQLSTATE

SQLCODE é um código recebido após a execução de cada instrução SQL. Os significados dos valores são resumidos abaixo:

- SQLCODE = 0; o comando foi bem sucedido
- SQLCODE > 0; o comando foi bem-sucedido, mas retornou um aviso
- SQLCODE < 0; o comando falhou e retornou um erro

SQLSTATE é uma cadeia de texto de cinco caracteres que atende à norma ISO/ANSI SQL92. Os dois primeiros caracteres são conhecidos como o código de classe SQLSTATE:

- Um código de classe 00 significa que o comando foi bem-sucedido.
- Um código de classe 01 indica um aviso.
- Um código de classe 02 indica uma condição não encontrado.
- Todos os outros códigos de classe são considerados erros.

A.3 Log de notificação de administração do DB2

O log de notificação de administração do DB2 fornece informações de diagnóstico sobre erros no ponto de falha. Nas plataformas Linux e UNIX, o log de notificação de administração é um arquivo de texto denominado <nome da instância>.nfy (por ex., "db2inst.nfy"). No Windows, todas as mensagens de notificação de administração são gravadas no Log de Eventos do Windows.

O parâmetro `notifylevel` de configuração do DBM permite que os administradores especifiquem o nível de informações que devem ser registradas:

- 0 -- Sem captura de mensagens de notificação de administração (não recomendado)
- 1 -- Erros fatais ou sem recuperação
- 2 -- É necessário ação imediata
- 3 -- Informações importantes, sem necessidade de ação imediata (o padrão)
- 4 -- Mensagens informativas

A.4 db2diag.log

O db2diag.log fornece mais informações detalhadas que o log de notificação de administração do DB2. Normalmente, ele é usado apenas pelo suporte técnico do IBM DB2 ou DBAs experientes. As informações do db2diag.log incluem:

- A localização do código do DB2 informando um erro.
- Identificadores do aplicativo que permitem comparar as entradas relacionadas a um aplicativo nos db2diag.logs de servidores e clientes.
- Uma mensagem de diagnóstico (iniciando por "DIA") explicando o motivo do erro.
- Qualquer dado de suporte disponível, como as estruturas de dados SQLCA e os ponteiros para a localização de qualquer arquivo extra de despejo ou interceptação.

No Windows (versão diferente do Vista), o db2diag.log está localizado, como padrão, no diretório:

```
C:\Documents and Settings\All Users\Application  
Data\IBM\DB2\DB2COPY1\<<instance name>
```

No Windows Vista, o db2diag.log está localizado, como padrão, no diretório:

```
C:\ProgramData\IBM\DB2\DB2COPY1\<<instance name>
```

No Linux/UNIX, o db2diag.log está localizado, como padrão, no diretório:

```
/home/<instance_owner>/sqlllib/db2dump
```

O detalhamento do texto de diagnóstico é determinado pelo parâmetro de configuração DIAGLEVEL do dbm cfg. O intervalo é de 0 a 4, onde 0 é o menor detalhamento e 4 é o máximo. O nível padrão é 3.

A.5 Rastreamentos CLI

Para os aplicativos CLI, Java, PHP e Ruby on Rails, você pode ativar o utilitário de rastreamento CLI para resolver problemas em seu aplicativo. Para isso, faça alterações no arquivo db2cli.ini no servidor onde o aplicativo está sendo executado. As entradas típicas do arquivo db2cli.ini são mostradas abaixo, na *Listagem A.1*.

```
[common]  
trace=0  
tracerefreshinterval=300  
tracepathname=/path/to/writeable/directory  
traceflush=1
```

Listagem A.1 – Entradas do arquivo db2cli.ini para ativar o rastreamento CLI

O rastreamento de baixo nível (db2trc) também está disponível, mas normalmente só é útil para o suporte técnico do DB2.

A.6 Defeitos e correções do DB2

Algumas vezes, o problema encontrado pode ser causado por um defeito do DB2. A IBM lança regularmente os fix packs, que contêm correções de programação para os defeitos (APARs). A documentação do fix pack contém uma lista das correções incluídas. Ao desenvolver novos aplicativos, sempre recomendamos usar o fix pack mais recente para aproveitar as últimas correções. Para exibir o nível atual de versão e fix pack: no Control Center, selecione a opção **About** no menu **Help** ; ou na Command Window, digite **db2level**. Observe que os fix packs e suporte técnico oficial do IBM DB2 não são oferecidos com o DB2 Express-C. Neste caso, as correções são incorporadas na própria imagem, em vez de aplicadas em fix packs.

B

Apêndice B – Referências e recursos

B.1 Referências

[1] ZIKOPOULOS, P. *IBM® DB2® Universal Database™ and the Microsoft® Excel Application Developer... for Beginners* (IBM DB2 Universal Database e o desenvolvedor de aplicativos do Microsoft Excel... para iniciantes), artigo em dbazine.com, abril de 2005 <http://www.dbazine.com/db2/db2-disarticles/zikopoulos15>

[2] ZIKOPOULOS, P. *DB2 9 and Microsoft Access 2007 Part 1: Getting the Data...* (DB2 9 e Microsoft Access 2007 Parte1: Obtenção de dados), artigo no Database Journal, maio de 2008 <http://www.databasejournal.com/features/db2/article.php/3741221>

[3] BHOGAL, K. *Use Microsoft Access to interact with your DB2 data* (Use o Microsoft Access para interagir com seus dados do DB2), artigo na developerWorks, maio de 2006. <http://www.ibm.com/developerworks/db2/library/techarticle/dm-0605bhogal/>

[4] SARACCO, C. et all. IBM Redbook *DB2 9: pureXML overview and fast start* (Visão geral e início rápido com pureXML).

julho de 2006. <http://www.redbooks.ibm.com/abstracts/sg247298.html>

B.2 Sites:

1. Site do DB2 Express-C: www.ibm.com/db2/express

Use este site para fazer o download da imagem dos servidores DB2 Express-C, clientes DB2, drivers do DB2, manuais, acesso ao blog da equipe, assinar a lista para recebimento de mensagens, etc.

2. Fórum do DB2 Express:
www.ibm.com/developerworks/forums/dw_forum.jsp?forum=805&cat=19

Use o fórum para fazer perguntas técnicas quando não conseguir encontrar as respostas sozinho nos manuais.

3. Information Center do DB2

O Information Center fornece acesso aos manuais on-line. Ele é a fonte de informações mais atualizada. Para cada versão do DB2, há um Information Center correspondente:

- DB2 9.1: <http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp>
- DB2 9.5: <http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/index.jsp>
- DB2 9.7: <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/index.jsp>

4. developerWorks: <http://www.ibm.com/developerworks/db2>

Este site é um recurso excelente para os desenvolvedores e DBAs, fornecendo gratuitamente acesso aos atuais artigos, tutoriais, etc.

5. alphaWorks: <http://www.alphaworks.ibm.com/>

Este site fornece acesso direto à tecnologia emergente da IBM. É um local onde se pode encontrar as tecnologias mais recentes da IBM Research.

6. planetDB2: www.planetDB2.com

Este é um agregador de blogs de diversos contribuintes que mantêm blogs sobre o DB2.

7. Suporte técnico do DB2: http://www.ibm.com/software/data/db2/support/db2_9/

Aqui, você pode procurar relatórios sobre defeitos e problemas (APARs) e outras informações técnicas.

8. ChannelDB2: <http://www.ChannelDB2.com/>

O ChannelDB2 é uma rede social para a comunidade do DB2. Ele traz conteúdo como vídeos relacionados ao DB2, demonstrações, podcasts, blogs, discussões, recursos, etc. para Linux, UNIX, Windows, z/OS e i5/OS.

B.3 Livros

1. Redbook gratuito: DB2 Express-C: The Developer Handbook for XML, PHP, C/C++, Java, and .NET (Manual do desenvolvedor para XML, PHP, C/C++, Java e .NET)
Whei-Jen Chen, John Chun, Naomi Ngan, Rakesh Ranjan, Manoj K. Sardana,
agosto de 2006 - SG24-7301-00
<http://www.redbooks.ibm.com/abstracts/sg247301.html?Open>

2. Redbook gratuito: DB2 pureXML Guide (Guia sobre pureXML no DB2)
Whei-Jen Chen, Art Sammartino, Dobromir Goutev, Felicity Hendricks, Ippei Komi, Ming-Pang Wei, Rav Ahuja, Matthias Nicola. agosto de 2007
<http://www.redbooks.ibm.com/abstracts/sg247315.html?Open>
3. Redbook gratuito: Developing PHP Applications for IBM Data Servers (Desenvolvimento de aplicativos PHP para servidores de dados IBM).
Whei-Jen Chen, Holger Kirstein, Daniel Krook, Kiran H Nair, Piotr Pietrzak
maio de 2006 - SG24-7218-00
<http://www.redbooks.ibm.com/abstracts/sg247218.html?Open>
4. Understanding DB2 – Learning Visually with Examples V9.5 (Compreendendo o DB2 - aprendendo visualmente com exemplos V9.5)
Raul F. Chong, et all. Janeiro de 2008
ISBN-10: 0131580183
5. DB2® SQL PL: Essential Guide for DB2® UDB on Linux™, UNIX®, Windows™, i5/OS™, and z/OS®, 2nd Edition (Guia essencial para DB2® UDB em Linux™, UNIX®, Windows™, i5/OS™ e z/OS®, 2ª edição)
Zamil Janmohamed, Clara Liu, Drew Bradstock, Raul Chong, Michael Gao, Fraser McArthur, Paul Yip
ISBN: 0-13-100772-6
6. DB2 9: pureXML overview and fast start
(Visão geral e início rápido com pureXML) Cynthia M. Saracco, Don Chamberlin, Rav Ahuja June 2006 SG24-7298
<http://www.redbooks.ibm.com/abstracts/sg247298.html?Open>
7. Information on Demand - Introduction to DB2 9 New Features (Informação sob demanda - introdução aos novos recursos do DB2 9)
Paul Zikopoulos, George Baklarz, Chris Eaton, Leon Katsnelson
ISBN-10: 0071487832
ISBN-13: 978-0071487832

B.4 Emails para contato

Caixa postal geral do DB2 Express-C (Para perguntas do tipo administrativo):
db2x@ca.ibm.com

Caixa postal geral do programa DB2 on Campus: db2univ@ca.ibm.com

Começar a usar o DB2 9.7 não poderia ser mais fácil. Leia este livro para:

- **Descobrir o DB2 usando sua edição gratuita - Express-C**
- **Compreender a arquitetura, ferramentas, segurança do DB2**
- **Aprender como administrar bancos de dados do DB2**
- **Escrever SQL, XQuery, procedimentos armazenados**
- **Desenvolver aplicativos de bancos de dados para DB2**
- **Praticar com exercícios**

O DB2 Express-C da IBM é uma edição gratuita do servidor de dados DB2 para o gerenciamento fácil de dados relacionais e XML. Gratuito significa que o DB2 Express-C é gratuito para fazer o download, construir seus aplicativos, implementar na produção e até mesmo incorporar e distribuir com sua solução. E o DB2 não coloca qualquer limitação artificial sobre o tamanho dos bancos de dados, número de bancos de dados ou de usuários.

O DB2 Express-C é executado em sistemas Windows e Linux e fornece drivers de aplicativo para diversas linguagens de programação, incluindo C/C++, Java, .NET, PHP, Perl e Ruby. Uma assinatura de baixo custo e suporte com capacidades adicionais opcionais estão disponíveis. Se você necessitar de maior capacidade de escalabilidade ou funcionalidade mais avançada, pode implementar de forma simples os aplicativos criados com o DB2 Express-C em outras edições do DB2, como o Workgroup e o Enterprise.

Esta edição gratuita do DB2 é ideal para desenvolvedores, consultores, ISVs, DBAs, estudantes ou qualquer pessoa que pretenda desenvolver, testar, implementar ou distribuir aplicativos de banco de dados. Junte-se à comunidade crescente de usuários do DB2 Express-C hoje mesmo e faça um teste com o DB2 Express-C. Comece a descobrir como você pode criar a nova geração de aplicativos e fornecer soluções inovadoras.

Para saber mais ou fazer download do DB2 Express-C, visite:

ibm.com/db2/express

Para se relacionar com a comunidade do DB2, assistir a vídeos e ler blogs relacionados ao DB2, visite:

Channeldb2.com



9 780986 628351

Preço: US\$ 24,99