

CONHECENDO O

# DB2 Express-C

Um livro da comunidade para a comunidade

RAUL CHONG, IAN HAKES, RAV AHUJA

PREFÁCIO POR DR. ARVIND KRISHNA



SEGUNDA EDIÇÃO

**Segunda Edição (Abril 2008)**

**Esta edição foi atualizada para IBM® DB2® Express-C Version 9.5 for Linux®, UNIX® and Windows®.**

**© 2007, 2008 Copyright IBM Corporation. Todos os direitos reservados.**

# Sumário

<b>Sobre este livro</b> .....	<b>8</b>
Avisos e Marca Registrada.....	8
Quem deve ler este livro?.....	9
Como este livro é estruturado?.....	9
Um livro para a comunidade.....	9
Colaboradores.....	10
Agradecimentos .....	10
Prefácio .....	12
<b>PARTE I – VISÃO GERAL E SETUP</b> .....	<b>13</b>
<b>Capítulo 1 – O que é o DB2 Express-C?</b> .....	<b>15</b>
1.1 Grátis para desenvolver, distribuir e entregar... sem limites! .....	16
1.2 Assistência ao usuário e suporte técnico.....	16
1.3 Servidores DB2 .....	17
1.4 Clientes e drivers DB2.....	17
1.5 Liberdade no desenvolvimento de aplicações .....	19
1.6 Versões do DB2 versus edições do DB2.....	19
1.7 Migrando para uma nova edição do DB2 .....	20
1.8 A manutenção do DB2 Express-C .....	20
1.9 Softwares grátis relacionados ao DB2 Express-C .....	20
1.9.1 IBM Data Studio .....	21
1.9.2 DB2 Net Search Extender .....	21
1.9.3 Starter Toolkit for DB2 on Rails .....	21
1.9.4 Web 2.0 Starter Toolkit for DB2.....	21
1.9.5 WebSphere Application Server – Community Edition .....	22
<b>Capítulo 2 – Características e produtos relacionados</b> .....	<b>23</b>
2.1 Características incluídas com a assinatura do DB2 Express-C .....	27
2.1.1 Pacotes de correção.....	27
2.1.2 Alta Disponibilidade e Desastre e Recuperação (HADR - High Availability Disaster Recovery .....	27
2.1.3 Replicação de Dados .....	28
2.2 Características não disponíveis no DB2 Express-C .....	28
2.2.1 Particionamento de base de dados .....	29
2.2.2 Concentrador de Conexões.....	29
2.2.3 Extensor Geodésico .....	29
2.2.4 Gerenciamento do Balanço de Carga (Workload Management - WLM) .....	29
2.3 Produtos pagos que estão relacionados aos DB2 .....	30
2.3.1 DB2 Connect .....	30
2.3.2 WebSphere Federation Server .....	30
2.3.3 WebSphere Replication Server .....	31
<b>Capítulo 3 – A instalação do DB2</b> .....	<b>33</b>
3.1 Pré-requisitos de instalação .....	33
3.2 Regras de instalação relativas ao sistema operacional .....	33

## 4 Conhecendo o DB2 Express-C

---

3.3 O wizard de instalação .....	34
3.4 Instalação Silenciosa .....	39
QuickLab #1: Instalando o DB2 Express-C & criando a base de dados SAMPLE ...	41
<b>Capítulo 4 – Ambiente do DB2 .....</b>	<b>45</b>
Quicklab #2 – Criando uma nova base de dados .....	56
4.1 Configuração do DB2 .....	57
4.1.1 Variáveis de ambiente .....	58
4.1.2 Arquivo de configuração do gerenciador de base de dados (dbm cfg) .....	58
4.1.3 Arquivo de configuração da base de dados (db cfg).....	61
4.1.4 Registro de perfil do DB2 (DB2 profile registry) .....	62
4.2 O DB2 Administration Server.....	63
Quicklab #3 – Trabalhando com instâncias, bases de dados e configuração.....	65
<b>Capítulo 5 – Ferramentas do DB2 .....</b>	<b>67</b>
5.1 Control Center .....	69
5.2 Command Editor .....	72
5.3 SQL Assist Wizard.....	74
5.4 Botão Show SQL.....	76
Quicklab #4: Populando a base de dados EXPRESS usando scripts.....	78
5.5 Scripts .....	80
5.5.1 Scripts SQL .....	80
5.5.2 Scripts de sistema operacional (shell) .....	81
Quicklab #5: Criando um script de instalação para a base de dados EXPRESS.....	83
5.6 Task Center.....	86
5.6.1 Uma base de dados Tools Catalog.....	86
5.7 Journal .....	87
5.8 Health Monitor .....	89
5.8.1 Health Center .....	90
<b>PARTE II – CONHECENDO O DB2: ADMINISTRAÇÃO DE BANCOS DE DADOS ...</b>	<b>93</b>
<b>Capítulo 6 – Arquitetura do DB2 .....</b>	<b>95</b>
6.1 Modelo de Processo do DB2.....	95
6.2 O modelo de Memória DB2 .....	97
6.3 O Modelo de Armazenamento DB2.....	98
6.3.1 Páginas e Extents.....	98
6.3.2 Buffer pools .....	98
<b>Capítulo 7 – Conectividade do DB2 Cliente.....</b>	<b>105</b>
7.1 Diretórios do DB2 .....	105
7.2 Configuration Assistant.....	106
7.2.1 Configurações necessárias no servidor .....	106
7.2.2 Configurações necessárias no cliente .....	109
7.2.3 Criando perfis de cliente e de servidor .....	113
Quicklab #6: Usando o <i>Configuration Assistant</i> .....	117
<b>Capítulo 8 – Trabalhando com objetos da base de dados .....</b>	<b>121</b>
8.1 Esquema .....	121

---

8.2 Tabelas .....	122
8.2.1 Tipos de dados .....	122
8.2.2 Colunas Identity .....	125
8.2.3 Objetos SEQUENCE .....	125
8.2.4 Tabelas do catálogo do sistema .....	126
8.2.5 Tabelas temporárias .....	127
8.3 Visões .....	128
8.4 Índices .....	128
8.4.1 Design Advisor .....	129
8.5 Integridade Referencial .....	130
<b>Capítulo 9 – Utilitários de movimentação de dados .....</b>	<b>132</b>
9.1 Utilitário EXPORT .....	133
9.2 IMPORT utility .....	134
9.3 LOAD .....	135
9.4 O utilitário db2move .....	137
9.5 O utilitário db2look .....	137
<b>Capítulo 10 – Segurança da base de dados .....</b>	<b>145</b>
10.1 Autenticação .....	146
10.2 Autorização .....	147
10.3 Autorização DBADM .....	150
10.4 O grupo PUBLIC .....	150
10.5 As sentenças GRANT e REVOKE .....	151
10.6 Verificação de autorização e privilégio .....	152
10.7 Considerações sobre grupo de privilégio .....	153
Quicklab #9 – Concedendo e removendo permissões de usuário .....	154
<b>Capítulo 11 – Backup e Recuperação .....</b>	<b>157</b>
11.1 Log de base de dados .....	157
11.2 Tipos de logs .....	158
11.3 Tipos de logging .....	159
11.3.2 Log de arquivamento ou log de retenção .....	159
11.4 Log da base de dados no Control Center .....	160
11.5 Parâmetros de log .....	162
11.6 Backup da base de dados .....	162
Quicklab #10 – Agendando um backup .....	165
11.7 Recuperação de base de dados .....	168
11.7.1 Tipos de Recuperação .....	168
11.7.2 Restore da base de dados .....	169
11.8 Outras operações que utilizam BACKUP e RESTORE .....	169
<b>Capítulo 12 – Tarefas de Manutenção .....</b>	<b>171</b>
12.1 REORG, RUNSTATS, REBIND .....	171
12.1.1 O comando REORG .....	172
12.1.2 O comando RUNSTATS .....	172
12.1.3 BIND / REBIND .....	173
12.1.4 Tarefas de Manutenção no Control Center .....	174
12.2 Opções de Manutenção .....	175

Quicklab #11 – Configurando a manutenção automática .....	178
<b>Capítulo 13 – Concorrência e Bloqueios .....</b>	<b>181</b>
13.1 Transações.....	181
13.2 Concorrência .....	182
13.3 Problemas ocasionados pela falta de controle de concorrência .....	183
13.3.1 Operações de atualização (UPDATE) perdidas.....	183
13.3.2 Leitura de dados que não tenham sido finalizados (Uncommitted read)..	184
13.3.3 Leitura de dados não repetidos (Non-repeatable read) .....	185
13.3.4 Leituras Fantasmas (Phantom read) .....	186
13.4 Níveis de Isolamento .....	187
13.4.1 Leitura de Dados não finalizados (Uncommitted Read UR).....	187
13.4.2 Estabilidade de Cursor (Cursor Stability CS) .....	187
13.4.3 Estabilidade de Leitura (Read Stability RS) .....	188
13.4.4 Leitura Repetida (Repeatable Read RR) .....	188
13.4.5 Comparando os níveis de isolamento.....	188
13.4.6 Definindo o nível de isolamento.....	189
13.5 Escalada do Bloqueio (Lock Escalation).....	190
13.6 Monitoração de Bloqueio .....	191
13.7 Bloqueio em Espera (Lock wait) .....	192
13.8 Causas e como descobrir um deadlock.....	193
13.9 Boas práticas de concorrência e bloqueio .....	194
<b>PARTE III – CONHECENDO O DB2: DESENVOLVIMENTO DE APLICAÇÕES.....</b>	<b>196</b>
<b>Capítulo 14 – Procedimentos Armazenados SQL PL .....</b>	<b>198</b>
14.1 O IBM Data Studio.....	199
14.1.2 Criar um procedimento armazenado no studio de dados .....	200
14.2 Fundamentos dos procedimentos armazenados SQL PL.....	204
14.2.1 A estrutura de um procedimento armazenado .....	204
14.2.2 Atributos opcionais de procedimentos armazenados.....	205
14.2.3 Parâmetros.....	205
14.2.4 Comentários em um procedimento armazenado SQL PL.....	206
14.2.5 Instruções compostas.....	206
14.2.6 Declaração de variáveis .....	207
14.2.7 Instruções de atribuição .....	207
14.3 Cursores.....	208
14.4 Controle de fluxo .....	208
14.5 Executando procedimentos armazenados.....	209
14.6 Manipuladores de erros e condições .....	211
14.7 SQL Dinâmico .....	212
<b>Capítulo 15 – Inline SQL PL, Triggers, e UDFs.....</b>	<b>215</b>
15.1 Inline SQL PL .....	215
15.2 Triggers .....	216
15.2.1 Tipos de triggers.....	217
Quicklab #12 – Criando uma trigger no Control Center .....	220
15.3 Funções definidas pelo usuário (UDFs).....	223
15.3.1 Funções Escalares .....	223

---

15.3.2 Funções aplicadas em tabela .....	224
Quicklab #13 – Criando uma UDF usando o IBM Data Studio .....	226
<b>Capítulo 16 – DB2 pureXML.....</b>	<b>228</b>
16.1 Usando XML em bases de dados.....	229
16.2 Bases de dados XML .....	229
16.2.1 Bases de dados XML-enabled.....	229
16.2.2 Bases de dados de XML nativo .....	230
16.3 XML no DB2 .....	231
16.3.1 Vantagens da tecnologia pureXML.....	232
16.3.2 Básico sobre XPath .....	234
16.3.3 XQuery definida.....	237
16.3.4 Inserindo documentos XML.....	238
16.3.5 Consultando dados XML .....	241
16.3.6 Joins utilizando SQL/XML.....	247
16.3.7 Joins utilizando XQuery .....	248
16.3.8 Operações de atualização e exclusão .....	249
16.3.9 Índice em XML.....	250
QuickLab #14 - SQL/XML e XQuery.....	252
<b>Capítulo 17 – Desenvolvendo com Java, PHP, e Ruby.....</b>	<b>253</b>
17.1 Desenvolvimento de aplicações em Java .....	253
17.1.1 Driver JDBC tipo 2.....	253
17.1.2 Driver JDBC tipo 4.....	254
17.2 Desenvolvimento de aplicações em PHP .....	255
17.2.1 Opções de conexão do DB2 para PHP .....	255
17.2.2 Zend Core for IBM.....	257
17.3 Desenvolvimento de aplicações em Ruby on Rails .....	260
17.3.1 Startup Toolkit for DB2 on Rails.....	260
<b>Apêndice A – Resolvendo Problemas .....</b>	<b>261</b>
A.1 Encontrando mais informações sobre códigos de erro.....	262
A.2 SQLCODE e SQLSTATE .....	262
A.3 DB2 – Log de Administração e Notificação .....	263
A.4 db2diag.log .....	263
A.5 CLI traces.....	264
A.6 DB2 – Defeitos e Correções.....	264
<b>Recursos .....</b>	<b>265</b>
Web sites: .....	265
Livros.....	266

## **Sobre este livro**

### **Avisos e Marca Registrada**

© Copyright IBM Corporation 2007, 2008  
Todos os direitos reservados.  
IBM Canada  
8200 Warden Avenue  
Markham, ON  
L6G 1C7  
Canada

Este documento e/ou parte dele não podem ser copiados ou reproduzidos de qualquer forma ou por qualquer meio, ou traduzidos para idioma, sem o consentimento prévio de todos os titulares de copyright mencionados acima.

A IBM não fornece garantias ou declarações com relação ao conteúdo aqui apresentado e especificamente renuncia a qualquer garantia implícita de comercialização ou adequação a qualquer finalidade específica. A IBM não assume responsabilidade por qualquer erro que este documento possa conter, incluindo, sem limitação, erros na tradução. As informações contidas neste documento estão sujeitas a alterações sem aviso prévio. A IBM reserva-se o direito de fazer quaisquer alterações sem obrigação de notificar qualquer pessoa sobre tal revisão ou alteração. A IBM não se compromete a manter as informações aqui contidas atualizadas.

As informações deste documento referentes a produtos não-IBM foram obtidas de fornecedor(es) de tais produtos. A IBM não testou tais produtos e não pode confirmar a exatidão do desempenho, a compatibilidade ou qualquer outra reclamação relacionada a produtos não-IBM. Dúvidas sobre as capacidades de produtos não-IBM devem ser endereçadas ao(s) fornecedor(es) de tais produtos.

IBM, o logotipo da IBM, AIX, DB2, DB2 Connect, DB2 Universal Database, i5/OS, Informix, pureXML, Tivoli, WebSphere e z/OS são marcas registradas ou marcas de serviço da International Business Machines Corporation nos Estados Unidos e/ou em outros países.

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

Microsoft, Windows, Windows NT e o logotipo Windows são marcas registradas da Microsoft Corporation nos Estados Unidos e/ou em outros países.

Linux é uma marca registrada de Linus Torvalds nos Estados Unidos e/ou em outros países.

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

Outros nomes de empresas, produtos ou serviços podem ser marcas registradas ou marcas de serviço de terceiros.

Referências nesta publicação a produtos ou serviços IBM não significam que a IBM pretenda torná-los disponível em todos os países em que opera.

---

## Quem deve ler este livro?

Este livro é destinado a todos que trabalham ou pretendem trabalho com bancos de dados, como, administradores de banco de dados (DBA's), desenvolvedores de aplicações, consultores, arquitetos de software, gerentes de projetos, instrutores e estudantes.

## Como este livro é estruturado?

A Parte I, Visão Geral e Setup, explica como funciona a edição DB2 Express-C, faz uma introdução a família de produtos DB2 e suas principais características, mostrando como instalar e criar base de dados e explora as ferramentas disponíveis no DB2.

A Parte II, Aprendendo sobre o DB2: Administração do Banco de Dados, faz com que o leitor se familiarize com o ambiente do DB2, detalhando sua arquitetura, conexão remota, objetos de banco de dados, movimentação de dados (import/export/load), segurança, recuperação e backup, concorrência e bloqueio, e outras tarefas comuns de manutenção de base de dados.

A Parte III – Aprendendo sobre o DB2: Desenvolvendo Aplicações, trata sobre procedimentos armazenados (stored procedures), funções definidas pelo usuário, gatilhos (triggers), SQL/XML, XQuery, desenvolvimento em Java™, PHP e Ruby.

O Apêndice contém informações úteis sobre resolução de problemas.

Exercícios, chamados “Quicklabs” são fornecidos na maioria dos capítulos; todos os arquivos necessários para execução destes exercícios são fornecidos através do arquivo compactado `expressc_book_quicklabs_9.5.zip` que acompanha este livro e que também pode ser encontrado no site do IBM® DB2 Express-C:

[www.ibm.com/db2/express](http://www.ibm.com/db2/express).

O material deste livro também é utilizado em cursos oferecidos como parte do programa “DB2 on Campus”, o qual disponibiliza apresentações em vídeo através do site [www.channeldb2.com/oncampus](http://www.channeldb2.com/oncampus). Este livro também prepara para o exame DB2 on Campus. Este exame confirma a participação em um treinamento de 16h sobre o DB2. Você pode ler mais sobre este programa no site do DB2 Express-C [www.ibm.com/db2/express/students.html](http://www.ibm.com/db2/express/students.html)

**Nota:**

Para maiores informações sobre o programa DB2 on Campus, assista ao vídeo:

<http://www.channeldb2.com/video/video/show?id=807741:Video:3902>

## Um livro para a comunidade

Este livro foi criado pela equipe do DB2 Express-C e foi liberado para a comunidade sem custo nenhum. Vários membros da comunidade, ao redor do mundo, traduziram este livro para diferentes idiomas. Se você quiser fornecer um feedback, ou contribuir com novos materiais, incrementando o que já existe ou ajudando na tradução deste livro para outro

idioma, por favor, envie um e-mail para [db2x@ca.ibm.com](mailto:db2x@ca.ibm.com) com o seu plano de contribuição para com o seguinte texto no assunto "DB2 Express-C book changes."

## Colaboradores

Gostaríamos de agradecer ao incrível trabalho realizado pelos tradutores voluntários da comunidade DB2! Eles gastaram muitas horas e finais de semana para tornar a tradução deste livro uma realidade. A tabela abaixo lista os colaboradores voluntários:

Nome do Colaborador	Empresa / Universidade	Cargo	Contribuições
Andrea Rodacki	IBM Brasil	DBA	Líder do time de traduções para português do Brasil.  Traduziu os capítulos 1, 2, 3, 11, 12, 13 e 15, além de realizar a revisão final do trabalho realizado pelos demais voluntários. Verificou a precisão da tradução
Wellington Sampaio Martins	Pontifícia Universidade Católica (PUC/PR)	Estudante	Traduziu os capítulos 6, 7, 8, 9, 10 e 14.
Caroline Beatriz Perlin	Universidade Federal de São Carlos (UFSCar), IBM	Estudante / DBA	Traduziu os capítulos 4, 5, 16 e 17.

## Agradecimentos

Nós agradecemos muito as seguintes pessoas pelo seu apoio e desenvolvimento de materiais de referencia para este livro:

- Ted Wasserman, Clara Liu e Paul Yip do laboratório da IBM em Toronto os quais desenvolveram materiais que serviram como suporte a este livro.
- Don Chamberlin e Cindy Saracco por seus artigos sobre XQuery postados no site do IBM developerWorks, e Matthias Nicola por sua apresentação sobre pureXML™.
- Kevin Czap e Grant Hutchison pelo desenvolvimento de materiais de instrução técnica sobre o DB2.
- Katherine Boyachok por ter desenhado a capa deste livro.
- Susan Visser pela sua assistência na publicação deste livro.



## Prefácio

Inovação é a pedra fundamental do progresso na área de tecnologia. Na IBM, a inovação tem feito parte da evolução de nossos servidores de dados. A IBM foi pioneira nas técnicas de gerenciamento de dados entre os anos 1960 e 1970, disponibilizando novas tecnologias que refletiram em centenas de patentes credenciadas pela IBM na área de gerenciamento de banco de dados. Como resultado, algumas das maiores organizações do mundo utilizam e acreditam nos produtos IBM como o DB2, armazenando seus dados de missão crítica e desenvolvendo soluções que o utilizam.

Porém, o DB2 não é somente utilizado em empresas de grande porte. Através da edição of DB2 Express-C a tecnologia do DB2 está agora disponível para empresas de pequeno e médio porte – e tudo isto sem custo algum! Embora existam outros gerenciadores de banco de dados oferecidos gratuitamente, o DB2 Express-C oferece vantagens únicas.

Existem várias tecnologias inovadoras presentes no DB2 Express-C. Algumas destas inovações visam características avançadas, algumas reduzem a dificuldade de administração, outras trazem ganhos de performance, e muitas delas reduzem o custo de infra-estrutura. Não iremos discutir mais sobre estes pontos nesta seção, esperando que você seja estimulado a ler este livro – mas nós iremos resumir algumas características já provadas na prática.

O DB2 Express-C foi construído utilizando a tecnologia 'Viper', fazendo com que ele seja o primeiro servidor de banco de dados híbrido, armazenando os dados de ambas as maneiras, relacional e XML no seu formato nativo. Isto faz com que o DB2 seja o banco de dados ideal para a nova geração de aplicações SOA e Web 2.0 que utilizam a maioria de dados no formato XML. Diferente de outros fornecedores de banco de dados, o DB2 Express-C não limita a quantidade de dados que podem ser armazenados (tamanho do banco de dados) ou o número de base que podem ser criadas.

Este livro pode ser usado como um guia para começar a utilizar o DB2 Express-C. Ele o ajudará a entender os conceitos do DB2 e o habilitará para desempenhar funções administrativas e de desenvolvimento utilizando o DB2. As habilidades e conhecimentos adquiridos através deste livro são muito relevantes para outras edições avançadas do DB2 nas plataformas Linux, Unix e Windows.

Já que o DB2 Express-C não é um software de padrões abertos, a IBM acredita na prestação de suporte e incentivo as iniciativas da comunidade. Eu estou muito feliz por este livro ter sido desenvolvido por membros da comunidade de DB2 Express-C, sendo disponibilizado gratuitamente para todos da comunidade. Eu o encorajo a enriquecer e atualizar este livro com o seu próprio conhecimento e experiência, ajudando com as traduções para outros idiomas para que outras pessoas também possam se beneficiar de seu conhecimento.



Arvind Krishna  
Vice President, Data Servers  
Information Management, IBM Software Group

# **PARTE I – VISÃO GERAL E SETUP**



# 1

## Capítulo 1 – O que é o DB2 Express-C?

O DB2 Express-C faz parte da poderosa família de produtos IBM DB2 para gerenciamento de dados relacionais e no formato XML. O DB2 Express-C é um software grátis, sem limites e fácil de ser utilizado. A letra 'C' no nome DB2 Express-C significa Comunidade. A comunidade de usuários do DB2 Express-C está unida para auxiliar um ao outro, tanto de maneira online quanto off line. A comunidade de DB2 Express-C é composta de uma variedade de pessoas e empresas que modelam, desenvolvem, distribuem ou utilizam soluções de banco de dados. Os membros da comunidade incluem:

- Desenvolvedores de aplicação que precisam de padrões abertos de software de banco de dados para construir aplicações independentes (standalone), cliente-servidor, baseadas na web, e corporativas
- ISVs (Independent Software Vendor - Vendedores de Software Independentes), vendedores de hardware, vendedores de infraestrutura, e outros tipos de provedores de soluções que desejam empacotar ou embutir a característica de servidor de banco de dados em suas aplicações
- Consultores, administradores de banco de dados, e arquitetos de TI que precisam de um banco de dados robusto para treinamento, desenvolvimento, avaliação ou prototipação
- Empresas iniciantes, pequenas ou de porte médio que precisam da segurança de um servidor de banco de dados para suas aplicações e operações
- Fãs de banco de dados e entusiastas da tecnologia de última geração que querem utilizar um banco de dados fácil para construir aplicações para a próxima geração da Web 2.0
- Estudantes, professores, e outros usuários acadêmicos que querem um servidor de banco de dados versátil para lecionar, realizar projetos e pesquisa

O DB2 Express-C utiliza a mesma funcionalidade principal e o mesmo código fonte de outras edições do DB2 para Linux, UNIX, e Windows. O DB2 Express-C pode ser utilizado em sistemas 32-bit ou 64-bit em sistemas operacionais Linux ou Windows. Ele pode rodar em sistemas com qualquer quantidade de processadores e memória e não tem nenhum requisito especial de armazenamento ou setup do sistema. O DB2 Express-C também inclui o padrão pureXML sem nenhum custo. O padrão pureXML é uma tecnologia única do DB2 para armazenar e processar documentos XML nativo.

## 1.1 Grátis para desenvolver, distribuir e entregar... sem limites!

Esta frase sintetiza as idéias-chaves do DB2 Express-C:

- **Grátis para desenvolver:** Se você é um desenvolvedor de aplicação e precisa de um banco de dados para sua aplicação, você pode usar o DB2 Express-C.
- **Grátis para distribuir:** Se você está trabalhando em um ambiente de produção, e precisa de um gerenciador de banco de dados para os seus registros, você pode usar o DB2 Express-C.
- **Grátis para entregar:** Se você está desenvolvendo uma aplicação ou uma ferramenta que necessite de um servidor de banco de dados embutido, você pode utilizar o DB2 Express-C. Mesmo que o DB2 Express-C seja embutido na sua aplicação, e seja distribuído a cada vez que você realize uma venda, ele continua sendo grátis. Você precisa somente se registrar, gratuitamente, na IBM para distribuir o DB2 Express-C sem nenhum custo.
- **Sem limites:** Enquanto outros concorrentes de banco de dados oferecem um conjunto de limitações em relação ao tamanho da base de dados, o número de banco de dados, e o número de usuários, o DB2 Express-C não tem nenhum limite. Sua base de dados pode crescer e crescer sem nenhuma violação de acordo de licença. Não existe também nenhuma limitação em termos de número de conexões ou usuários por servidor.

### Nota:

Para aprender mais sobre o DB2 Express-C e suas funções no mundo da Web 2.0, assista a este vídeo:

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

## 1.2 Assistência ao usuário e suporte técnico

Se você tem dúvidas técnicas sobre o DB2 Express-C, você pode postá-las no fórum do DB2 Express-C. Este fórum é monitorado pela dedicada time do DB2 Express-C, e, de qualquer forma, a comunidade de voluntários que participam do fórum poderá responder as suas questões.

A IBM também fornece aos usuários a escolha de compra, por um custo baixo, de uma assinatura anual de suporte (também conhecida como Licença e Assinatura de 12 Meses ou Termo de Licença Fixa ou FTL).

Esta licença do DB2 Express-C engloba o suporte 24 x 7 da IBM e também as atualizações de software. Adicionalmente ao suporte e a manutenção de software, através de um pequeno custo anual de assinatura (US\$ 2.995 por servidor por ano nos Estados Unidos – este preço pode variar em outros países) você também pode utilizar duas características adicionais: HADR (cluster para Alta Disponibilidade e Desastre e Recuperação), e replicação de SQL (para replicar dados entre outros servidores DB2).

Maiores informações pode ser obtidas no site do DB2 Express-C  
[www.ibm.com/db2/express/support.html](http://www.ibm.com/db2/express/support.html)

### 1.3 Servidores DB2

Todas as edições do DB2 contém o mesmo componente principal; elas são empacotadas de uma maneira que os usuários podem escolher as funções necessárias de acordo com o preço. A Figura 1.1 ilustra a diferença entre as edições do DB2.

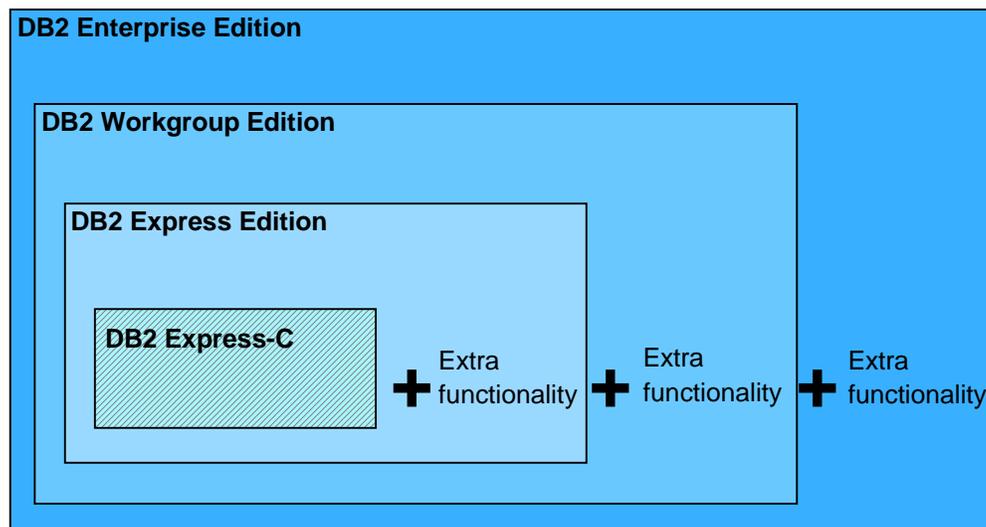


Figura 1.1 – Servidores DB2

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 grátis para a comunidade. Como mencionado anteriormente, a assistência técnica está disponível através do fórum on-line, ou você pode receber o suporte técnico oficial da IBM (24 x 7) se você comprar a licença de assinatura de 12 meses.

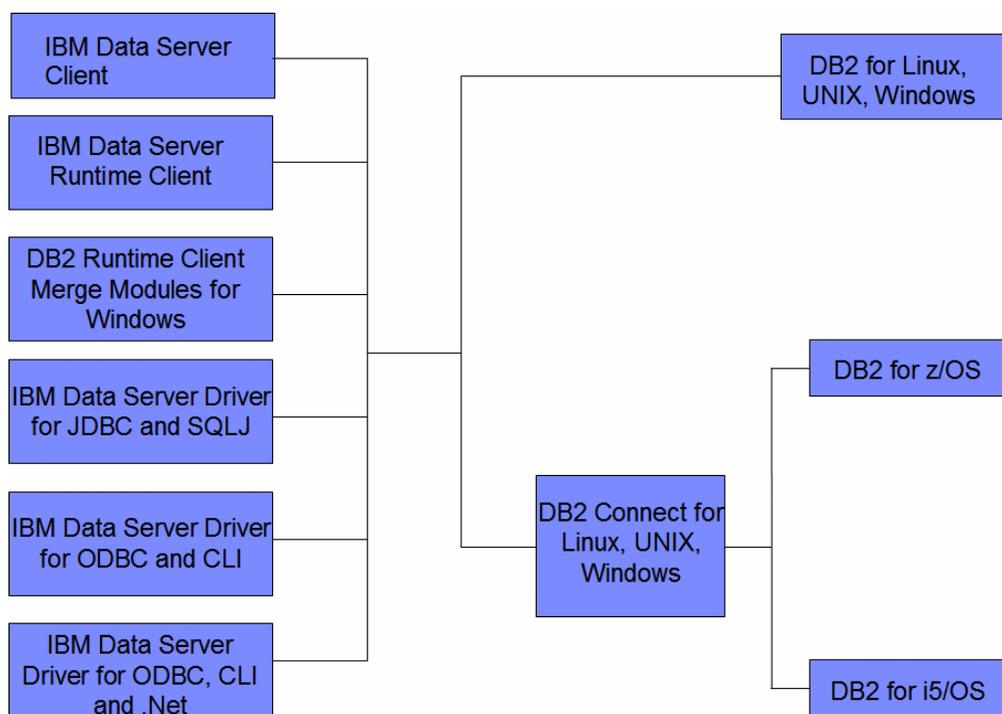
A Figura 1.1 também explica porque é tão fácil realizar o *upgrade* do DB2 Express-C. Se você desejar realizar o *upgrade* para qualquer outra edição do DB2 no futuro, todas as edições têm o mesmo componente principal (core). Isto também significa que qualquer aplicação desenvolvida em uma edição poderá ser executada, sem nenhuma modificação, em outras edições. E, todo conhecimento que você adquirir sobre uma edição poderá ser aplicado em outras edições.

### 1.4 Clientes e drivers DB2

Um cliente DB2 inclui a funcionalidade necessária para se conectar a um servidor DB2; porem, nem sempre é necessário instalar um cliente DB2. Por exemplo, uma aplicação JDBC tipo 4 pode se conectar diretamente a um servidor DB2. Clientes e drivers DB2 existem em diferentes maneiras:

- IBM Data Server Client: mais comum, inclui Ferramentas GUI e drivers.
- IBM Data Server Runtime Client: cliente mais leve que possui as funcionalidades básicas e inclui drivers
- DB2 Runtime Client Merge Modules for Windows: utilizado principalmente para empacotar o *runtime* do DB2 como parte da instalação de aplicações Windows
- IBM Data Server Driver for JDBC and SQLJ: permite que aplicações Java conectem-se a servidores DB2
- IBM Data Server Driver for ODBC and CLI: permite que aplicações ODBC e CLI conectem-se a um servidor DB2 sem que seja necessário instalar um cliente
- IBM Data Server Driver for ODBC, CLI and .NET: driver específico para Windows que suporte o ambiente .NET adicionalmente ao ODBC e ao CLI

Figura 1.2 mostra as diferenças entre os clientes e drivers DB2.



**Figura 1.2 – Clientes e drivers DB2**

No lado esquerdo da Figura 1.2 são mostrados todos os clientes e drivers do DB2. Embora todos os clientes DB2 já incluam seus drivers necessários, iniciando pelo DB2 9 nós também fornecemos os drivers individualmente. Todos os clientes e drivers DB2 estão disponíveis gratuitamente para download através do site do DB2 Express-C. Os clientes e drivers podem ser usados para conectar um servidor DB2 no Linux, UNIX ou Windows. Para conectar-se aos servidores DB2 for z/OS® ou DB2 for i5/OS®, você precisará se utilizar o servidor DB2 Connect™ (mostrado no meio da Figura 1.2). Nós discutiremos sobre o software DB2 Connect no Capítulo 2.

## 1.5 Liberdade no desenvolvimento de aplicações

O DB2 oferece um ambiente de desenvolvimento de aplicações que é padrão e transparente entre a família DB2. O padrão SQL no produto DB2 oferece um conjunto de interfaces de programação para acesso a banco de dados.

Adicionalmente, todo produto DB2 oferece pré-compiladores SQL que permitem aos desenvolvedores empacotar SQL estático e dinâmico em programas de aplicação portáteis. O DB2 também tem um gerenciador nativo .NET para se integrar com as ferramentas do Microsoft® Visual Studio.

Linguagens e padrões que você pode usar incluem:

- SQL, XQuery, XPath
- C/C++ (CLI, ODBC e SQL embutido)
- Java (JDBC and SQLJ)
- COBOL
- PHP
- Perl
- Python
- Ruby on Rails
- .NET languages
- OLE-DB
- ADO
- MS Office: Excel, Access, Word
- Web services

## 1.6 Versões do DB2 versus edições do DB2

Se você está começando a utilizar o DB2, você pode se confundir um pouco entre as diferenças de versões e edições do DB2.

De tempos em tempos, a IBM lança novas versões do DB2. Uma versão inclui nova características e melhorias significantes no produto. Atualmente, o DB2 versão 8 e o DB2 versão 9 são suportados pela IBM. Uma versão também pode ter novos lançamentos que podem incluir algumas novas funcionalidades porém não tão significantes como uma versão. Por exemplo, 8.1 e 8.2 são níveis de lançamento do DB2 versão 8. Tomando como base os últimos anos, a IBM tem realizado novos lançamentos do DB2 a cada ano, porém, novas versões ocorrem em um período de 2 a 3 anos. A versão mais nova é a V9.5 (anteriormente chamada de DB2 'Viper 2') a qual foi disponibilizada (Generally Available - GA) em Outubro de 2007. Cada lançamento também pode conter níveis de modificação, os quais normalmente contêm correções ou pacotes de correção, raramente novas funcionalidades. Atualmente o nível de Versão, Lançamento (Release) e Modificação (V, R, M) do DB2 Express-C é a 9.5.0 que corresponde ao nível de código 9.5 e pacote de correção (Fix pack) 0, o que significa que este é o nível GA.

Por outro lado, edições são oferecidas a cada nova versão. Como discutido anteriormente, uma edição é um pacote de diferentes funcionalidades de acordo com o preço e a licença. O DB2 versão 9.5 (também conhecido como DB2 9.5) tem muitas edições; por exemplo, o

DB2 Express-C 9.5, DB2 Express 9.5, DB2 Workgroup 9.5, e DB2 Enterprise 9.5 (veja a Figura 1.1).

## 1.7 Migrando para uma nova edição do DB2

Assim como a sua base de dados precisa crescer, você também poderá ter a necessidade de migrar para uma outra edição do DB2, devido a, por exemplo, um upgrade de configuração de hardware. Se acontecer esta situação, é muito fácil migrar para qualquer outra edição do DB2:

- Se você estiver migrando para outra edição do DB2 no mesmo computador, instale a nova edição em cima do DB2 Express-C e utilize a nova licença. Suas bases de dados não serão removidas (porém é recomendado realizar um backup de suas base de dados).
- Se você estiver migrando para uma nova edição do DB2 que será instalada em outro computador e sistema operacional, instale a nova edição do DB2, realize o backup de suas bases de dados antiga, mova esta imagem de backup para o novo computador, e restaure suas bases de dados. Você também precisará salvar os parâmetros de configuração de instância (dbm cfg) e aplicá-las no novo computador. Os comandos de backup e restauração serão discutidos em detalhes no Capítulo 11, Backup e Recuperação. Os parâmetros dbm cfg serão discutidos no Capítulo 4, Ambiente do DB2.
- Em ambos os casos não haverá nenhuma modificação na aplicação.

## 1.8 A manutenção do DB2 Express-C

Conforme discutido anteriormente, existem duas opções de suporte ao DB2 Express-C:

1. Comprar a licença de 12 meses. Esta opção fornece acesso, a qualquer hora, ao suporte técnico IBM DB2 e também inclui o acesso ao download das atualizações do software DB2 (também chamadas fixpacks).
2. Utilização do fórum on-line da comunidade DB2 Express-C. Esta opção é totalmente grátis porem não inclui o suporte oficial da IBM. Outro item que não está incluído nesta opção é o fornecimento de novas características e mudanças ou correções projetadas para corrigir um problema no software. O conceito de fixpack que será discutido no Capítulo 2, também não se aplica; porem, atualizações da imagem total do DB2 Express-C são disponibilizadas de tempos em tempos. A cada nova versão do DB2 são disponibilizadas novas versões do DB2 Express-C.

## 1.9 Softwares grátis relacionados ao DB2 Express-C

Todo software que está disponível para download a partir da página do DB2 Express-C ([www.ibm.com/db2/express/download.html](http://www.ibm.com/db2/express/download.html)) é grátis. Além das imagens do DB2 Express-C (para Linux e Windows, tanto 32 quanto 64 bits), existem outros softwares úteis que podem ser baixados gratuitamente:

- IBM Data Studio

- DB2 Net Search Extender
- DB2 Spatial Extender

Existem também outras ferramentas baseadas no DB2 Express-C e disponíveis para download a partir do site IBM Alphaworks ([www.alphaworks.ibm.com/datamgmt](http://www.alphaworks.ibm.com/datamgmt)) que podem ser úteis:

- Starter Toolkit for DB2 on Rails
- Web 2.0 Starter Toolkit for DB2

Se você está procurando um servidor de aplicação grátis, a IBM oferece:

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

### 1.9.1 IBM Data Studio

O IBM Data Studio é uma ferramenta baseada no Eclipse na qual pode ser realizada a modelagem, desenvolvimento, distribuição e gerenciamento de dados, bases de dados, e aplicações através do gerenciamento do ciclo de vida total do dado. O Data Studio substitui a antiga solução DB2 Developer Workbench 9.1.

O IBM Data Studio ajuda no desenvolvimento de funções definidas pelo usuário, procedimentos armazenados (stored procedures), XQuery, instruções SQL e possui um debugador integrado. Além disto, o Data Studio permite que você trabalhe com diagramas físicos que ajudam no entendimento do relacionamento entre as tabelas. Ele também pode ajudar no desenvolvimento e publicação dos dados na Web sem necessidade de programação. Iremos discutir o Data Studio no Capítulo 14, SQL PL stored procedures.

### 1.9.2 DB2 Net Search Extender

Com o DB2 Net Search Extender, você pode executar consultar rápidas e detalhadas em arquivos de texto, incluindo qualquer documento XML armazenado nativamente no DB2.

### 1.9.3 Starter Toolkit for DB2 on Rails

O [Starter Toolkit for DB2 on Rails](#) é um conjunto de produtos e tecnologia que habilita a rápida criação de um ambiente para construir aplicações Web usando a tecnologia Ruby on Rails e o DB2. Todos os softwares necessários para construção de uma aplicação Web estão incluídos: DB2 Express-C, driver DB2 para Ruby; adaptador DB2 para Rails; e também tutoriais, exemplos e outros materiais de estudo. Iremos discutir sobre o Ruby on Rails no Capítulo 17, Desenvolvendo em Java, PHP e Ruby.

### 1.9.4 Web 2.0 Starter Toolkit for DB2

O [Web 2.0 Starter Toolkit for DB2](#) é uma maneira fácil de começar a trabalhar com o DB2, PHP e Dojo. Ele te ajuda a distribuir o software, o leva aos tutoriais e inclui aplicações demonstrativas. Duas das aplicações demonstrativas são “Atom Feed Control Panel” (Painel de Controle de Alimentação do Átomo), a qual gera alimentação de átomos a partir de tabelas do DB2, e “Web Services Control Panel” (Painel de Controle de Serviços Web)

a qual cria serviços web em espera (REST web service) ao redor de tabelas DB2. Ambas as aplicações dependem de Dojo para desempenhar suas funções exemplares.

#### **1.9.5 WebSphere Application Server – Community Edition**

O IBM WebSphere Application Server Community Edition é uma versão mais leve do servidor de aplicação Java EE 5 o qual é distribuído gratuitamente. Construído com a tecnologia Apache Geronimo, este software contém as últimas inovações da comunidade de padrões abertos para desenvolver uma aplicação integrada, de fácil acesso e com uma fundação flexível para desenvolvimento e distribuição de aplicações Java. O suporte técnico o WAS CE está disponível através de uma assinatura anual.

# 2

## Capítulo 2 – Características e produtos relacionados

Este capítulo descreve as características do DB2 que estão inclusas na assinatura anual de licença do DB2 Express-C. Ele também descreve características que fazem parte de outras edições do DB2, que podem ser disponibilizadas através de um pequeno custo.

Características inclusas na edição grátis do DB2 Express-C:

- funcionalidade principal do DB2 (core)
- Control Center, Data Studio e outras ferramentas de gerenciamento
- pureXML
- Utilização em plataformas com até 2GB e 2 processadores
- Disponível em Linux, Windows e Solaris (x86)

Características não disponíveis na edição grátis do DB2 Express-C porém disponíveis com a compra da licença anual do DB2 Express-C:

- Pacotes de ajustes (Fix packs)
- Alta disponibilidade
- Replicação de Dados (SQL Homogêneo)
- Utilização em plataformas acima de 4GB e 4 processadores (em 2 dispositivos)

A tabela a seguir lista as características dos produtos e em quais edições as mesmas são disponibilizadas. As características que podem ser adquiridas separadamente estão listadas de acordo com a edição do DB2.

<b>Funcionalidade</b>	<b>Termo de Licença Fixa do DB2 Express-C</b>	<b>Edição DB2 Express</b>	<b>Edição DB2 Workgroup Server</b>	<b>Edição DB2 Enterprise Server</b>
Homogenous SQL Replication	Sim	Sim	Sim	Sim
Net Search Extender	Sim	Sim	Sim	Sim

<b>Funcionalidade</b>	<b>Termo de Licença Fixa do DB2 Express-C</b>	<b>Edição DB2 Express</b>	<b>Edição DB2 Workgroup Server</b>	<b>Edição DB2 Enterprise Server</b>
Spatial Extender	Sim	Sim	Sim	Sim
pureXML™ technology	Sim	pureXML Feature	pureXML Feature	pureXML Feature
High availability disaster recovery	Sim	High Availability Feature	Sim	Sim
Tivoli® System Automation	Sim	High Availability Feature	Sim	Sim
Advanced Copy Services	Não	High Availability Feature	Sim	Sim
Online reorganization	Não	High Availability Feature	Sim	Sim
Homogenous Federation	Não	Homogeneous Federation Feature	Homogeneous Federation Feature	Homogeneous Federation Feature
MQT	Não	Não	Query Optimization Feature	Sim
MDC	Não	Não	Query Optimization Feature	Sim
Query parallelism	Não	Não	Query Optimization Feature	Sim
Connection concentrator	Não	Não	Não	Sim
Table partitioning	Não	Não	Não	Sim

<b>Funcionalidade</b>	<b>Termo de Licença Fixa do DB2 Express-C</b>	<b>Edição DB2 Express</b>	<b>Edição DB2 Workgroup Server</b>	<b>Edição DB2 Enterprise Server</b>
DB2 Governor	Não	Não	Não	Sim
Compression: row level	Não	Não	Não	Storage Optimization Feature
Compression: backup	Não	Não	Não	Storage Optimization Feature
Label-based access control (LBAC)	Não	Não	Não	Advanced Access Control Feature
Geodetic Extender	Não	Não	Não	Geodetic Data Management Feature
Query Patroller	Não	Não	Não	Performance Optimization Feature
DB2 workload management	Não	Não	Não	Performance Optimization Feature
Performance Expert	Não	Não	Não	Performance Optimization Feature
Homogenous Q Replication	Não	Não	Não	Homogeneous Replication Feature
Database partitioning	Não	Não	Não	Não

**Table 2.1: DB2 Version 9.5 editions: feature and function support**

Características disponíveis em outras edições do DB2:

Características pagas da edição DB2 Express:

- pureXML
- Alta Disponibilidade
- Característica de Federação Homogênea

Características incluídas sem custo na Edição DB2 Workgroup:

- Alta Disponibilidade
- Disponível em AIX, Solaris, HP-UX além de Linux e Windows

Características pagas que podem ser obtidas na Edição DB2 Workgroup:

- pureXML
- Característica de Otimização de Consultas (MQT, MDC, paralelismo de consultas)
- Característica de Federação Homogênea

Características pagas que podem ser obtidas na Edição DB2 Enterprise:

- pureXML
- Característica de Otimização de Armazenamento (incluindo compressão de dados)
- Controle de Acesso Avançado (fina granularidade e segurança avançada)
- Otimização de Performance (Workload Management, Performance Expert, Query Patroller)
- Gerenciamento de Dados Geográficos (análise de localização geográfica)
- Característica de Federação Homogênea (acesso remoto ao DB2 e ao Informix® como se fossem tabelas locais)

A edição DB2 Enterprise também contém algumas funcionalidades incluídas sem custo como:

- Particionamento de tabelas (Range Partitioning)
- Consulta a Tabelas Materializadas (MQT)
- Multi-dimensional Clustering (MDC)
- Alta disponibilidade e Desastre e Recuperação (HADR), Automação no Sistema Tivoli
- Concentrador de Conexão

Produtos relacionados ao DB2 que podem ser comprados:

- DB2 Connect
- DB2 Warehouse Editions
- WebSphere® Federation Server
- WebSphere Replication Server

## 2.1 Características incluídas com a assinatura do DB2 Express-C

Esta seção detalha sobre os pacotes de correção do DB2 (Fix packs), HADR e replicação de SQL.

### 2.1.1 Pacotes de correção

Os pacotes de correção do DB2 são conjuntos de códigos aplicados em uma instalação do DB2 para corrigir diferentes problemas reportados depois do produto ser lançado. Através da assinatura de licença, é possível fazer o download dos pacotes de correção grátis e instalá-los. Os pacotes de correção são normalmente lançados a cada três meses.

Para fazer o download do último pacote de correção, acesse o suporte técnico do DB2 no site:

[http://www.ibm.com/software/data/db2/support/db2\\_9/](http://www.ibm.com/software/data/db2/support/db2_9/)

### 2.1.2 Alta Disponibilidade e Desastre e Recuperação (HADR - High Availability Disaster Recovery)

A característica de Alta Disponibilidade e Desastre e Recuperação (HADR) provê uma solução completa ou parcial para empresas que necessitam de Alta Disponibilidade e projetos de Desastre e Recuperação. Um ambiente HADR geralmente consiste de dois sites de dados, um primário e um secundário (os quais podem estar localizados em diferentes locais geográficos). O site primário é onde as bases de dados principais estão armazenadas e onde os dados são acessados pelas aplicações clientes. As transações são processadas no site primário e as logs das bases de dados são automaticamente enviadas para o site secundário através da rede. O site secundário tem uma cópia clonada da base de dados primária, usualmente criada a partir de um backup da base de dados primária e restauração no site secundário. Depois da aplicação das logs na base de dados primária elas são enviadas para a base de dados secundária e aplicadas da mesma maneira. Através da contínua aplicação das logs na base de dados secundária garante-se a sincronia com a base de dados primária, fazendo com que, no caso de falha da base de dados primária a secundária possa assumir sem problemas.

A solução completa de HADR do DB2 possui:

- Capacidade de execução de failover rapidamente e de maneira transparente para os usuários da aplicação
- Completa atomicidade das transações para prevenir a perda de dados
- Habilidade de upgrade dos sistemas ou aplicações sem que seja necessária uma interrupção no serviço
- Failover remoto, garantindo a recuperação total dos dados a partir de um desastre no local da unidade física de processamento
- Fácil gerenciamento através de ferramentas gráficas do DB2
- Todos os recursos disponíveis com o menor impacto possível na performance do sistema

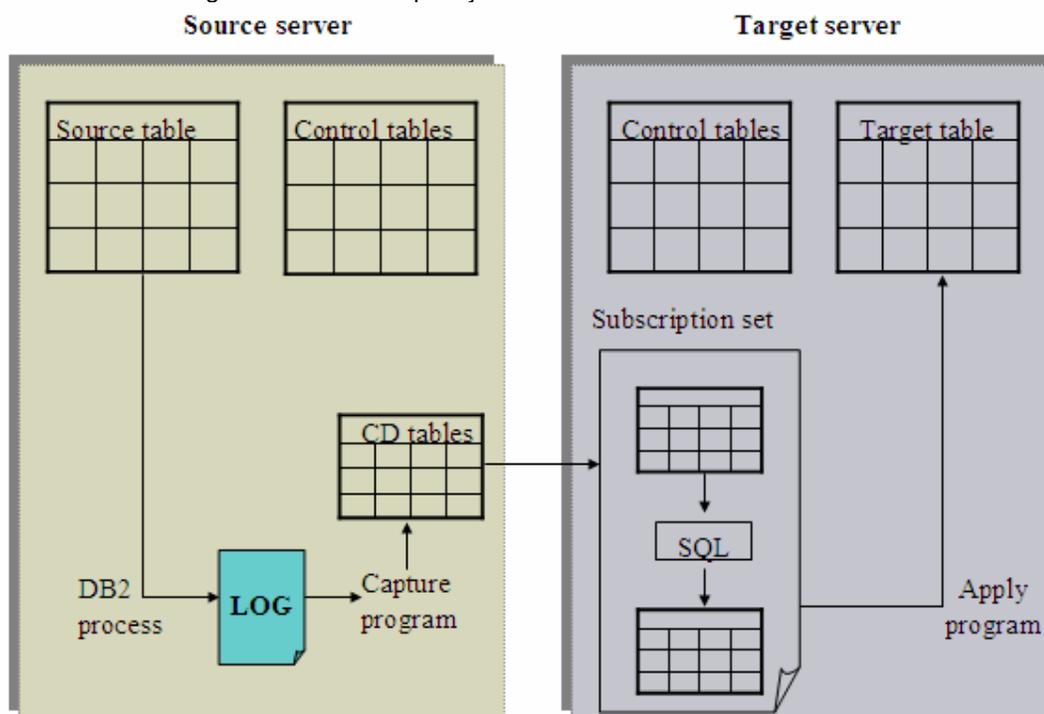
**Nota:**

Para visualizar uma demonstração de como funciona o HADR, por favor acesse o site:

<http://www-306.ibm.com/software/data/db2/express/demo.html>

### 2.1.3 Replicação de Dados

Esta característica permite a replicação de dados entre um servidor origem, onde os dados são capturados e um servidor destino, onde os dados são aplicados. A Figura 2.1 mostra uma visão geral de como a replicação funciona.



**Figura 2.1 – Replicação SQL**

Na Figura 2.1 existem dois servidores, um servidor de origem (source) e outro de destino (target). No servidor destino, o programa “Capture” captura as mudanças realizadas na base de dados. No servidor destino, o programa “Apply” aplica as mudanças na base de dados replicada. A replicação é útil para em várias situações onde é necessários dados replicados, inclusive no caso de problemas de capacidades de armazenamento, alimentação de aplicações data warehouses e data marts, e históricos de auditoria. Utilizando a característica de replicação SQL você pode replicar dados entre servidores DB2 Express-C e outras sistemas DB2 que estejam armazenados no Linux, UNIX, z/OS e i5/OS.

## 2.2 Características não disponíveis no DB2 Express-C

Esta seção descreve algumas características que não estão disponíveis na edição DB2 Express-C mas que podem ser encontradas nas outras edições do DB2.

### **2.2.1 Particionamento de base de dados**

A característica de particionamento de base de dados (DPF) só está disponível com a Edição DB2 Warehouse. O particionamento permite que a base de dados seja armazenada em múltiplas partições as quais podem estar em diferentes computadores. A característica DPF é baseada na arquitetura “shared-nothing” (nada dividido). Cada computador, onde a base de dados distribuiu as suas partições, traz ao sistema uma capacidade de processamento adicional, através de suas configurações de memória e processamento. A característica DPF é bastante útil em ambientes com bases de dados grandes como data warehouse onde há a necessidade de execução de consultas de suporte a decisão (DSS).

### **2.2.2 Concentrador de Conexões**

O concentrador de conexões é uma característica que permite o suporte a várias conexões concorrentes. Sem esta característica, cada conexão com o banco de dados exigia um agente de banco de dados. O concentrado de conexões introduziu o conceito de “agente lógico”, permitindo que um agente trate muitas conexões. Os agentes são discutidos em detalhe no Capítulo 6, Arquitetura do DB2.

### **2.2.3 Extensor Geodésico**

O Extensor Geodésico do DB2 está disponível como uma opção paga na Edição DB2 Enterprise. Com este extensor é possível o desenvolvimento de aplicações de inteligência de negócio e governamentais que precisam de um localizador geográfico. O Extensor Geodésico do DB2 pode construir uma representação tridimensional em qualquer escala. A maioria das informações são coletadas utilizando sistemas mundiais, como o posicionamento global dos satélites (GPS), e pode representar coordenadas de latitude/longitude (geocode). Dados de negócio, como endereços, podem ser convertidos para geocode pelo Extensor Geodésico do DB2 já que aplicações corporativas trabalham melhor quando os dados são mantidos desta maneira, deixando as projeções dos mapas na camada de apresentação, para que possam ser visualizados e impressos.

### **2.2.4 Gerenciamento do Balanço de Carga (Workload Management - WLM)**

Gerenciamento do balanço de carga do banco de dados baseado nas prioridades dos usuários e aplicações combinando a disponibilidade de recursos e os índices de balanços de carga. Este recurso permite que você regule o balanço de carga do banco de dados e das consultas fazendo com que consultas e tarefas importantes sejam executadas com prioridade, prevenindo que consultas “vulgares” monopolizem os recursos e assegurando que o seu sistema seja executado de maneira eficiente. O WLM é uma característica nova do DB2 9.5 e fornece meios mais poderosos do que as ferramentas que existiam nas versões anteriores do DB2 (Query Patroller e DB2 Governor).

## 2.3 Produtos pagos que estão relacionados aos DB2

### 2.3.1 DB2 Connect

O DB2 Connect é um software pago que permite que o DB2 no Linux, UNIX ou Windows se conecte ao DB2 nas plataformas z/OS ou DB2 i5/OS como mostrado na Figura 2.2. O DB2 Connect não é necessário quando a conexão ocorre no caso inverso; quando a conexão é a partir do DB2 nas plataformas z/OS ou DB2 i5/OS para DB2 no Linux, UNIX ou Windows. Existem duas edições do DB2 Connect: DB2 Connect Personal Edition, e DB2 Connect Enterprise Edition.

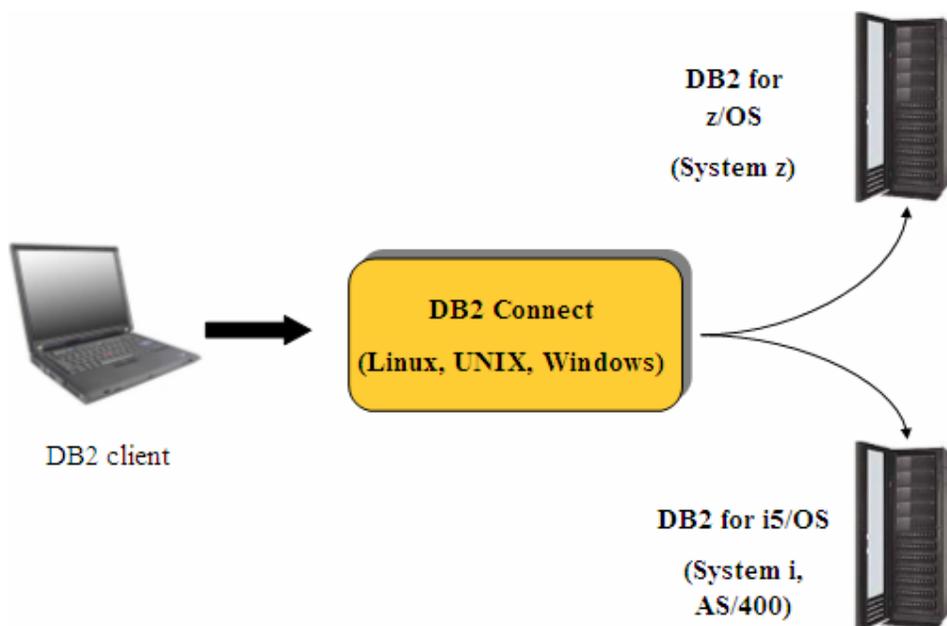


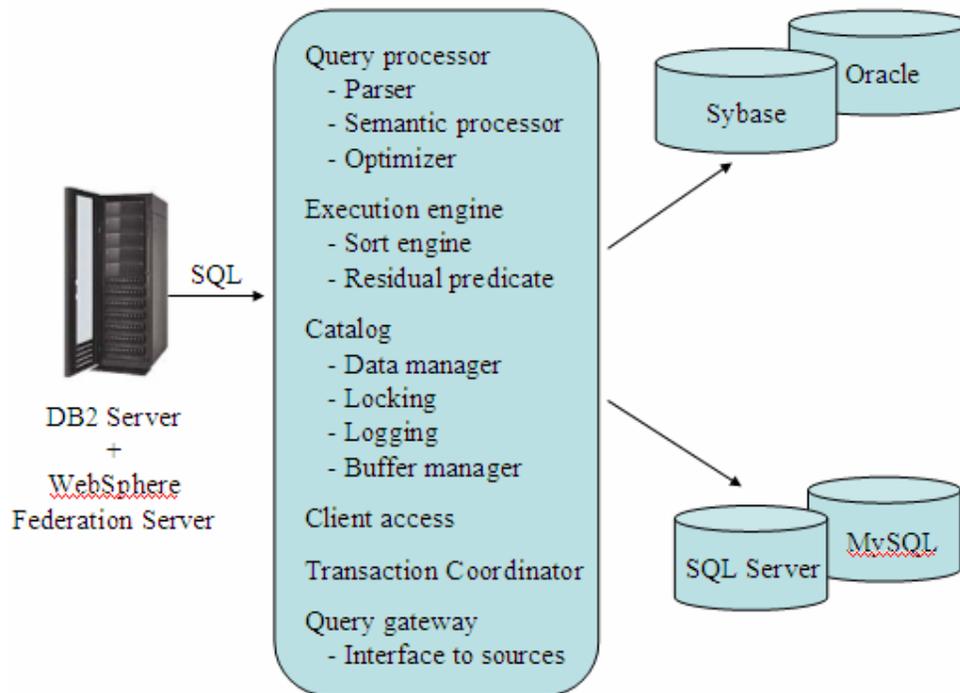
Figura 2.2 – DB2 Connect

### 2.3.2 WebSphere Federation Server

Anteriormente conhecido como WebSphere Information Integrator (para suporte a dados federados), o WebSphere Federation Server é utilizado em bancos de dados federados, onde você pode realizar consultas que utilizem objetos de diferentes bancos de dados relacionais. Por exemplo, se você comprar o WebSphere Federation Server, você pode executar a seguinte consulta:

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

A Figura 2.3 mostra um caso onde o WebSphere Federation Server é utilizado.



**Figura 2.3 – WebSphere Federation Server**

Em sistemas gerenciadores de banco de dados que fazem parte da família IBM, o suporte a federação é construída com o DB2 Express-C. Isto significa que o WebSphere Federation Server não é necessário, por exemplo, quando você quer executar uma consulta em duas bases de dados DB2 diferente, ou entre uma base de dados DB2 e outra Informix® (o Informix faz parte da família IBM).

### 2.3.3 WebSphere Replication Server

Anteriormente conhecido como WebSphere Information Integrator (para o suporte a replicação), o WebSphere Replication Server permite a replicação SQL de registros de base de dados não IBM. Este recurso também inclui a característica chamada Que-Replication ou replicação de dados utilizando filas de mensagens.



# 3

## Capítulo 3 – A instalação do DB2

Para instalar a edição DB2 Express-C no Linux ou no Wind, assegure-se que seu sistema satisfaça aos pré-requisitos de instalação.

### 3.1 Pré-requisitos de instalação

Com relação à versão do sistema operacional, o DB2 Express-C está disponível para Linux, Solaris (x64), e Windows 2003, 2000, XP, e Vista. A arquitetura de processador disponível é 32-bit, 64-bit e PowerPC (Linux). Se você precisa executar o DB2 em outras plataformas (como UNIX), você deve comprar uma das outras edições do DB2 previamente descritas neste livro. Todos os requisitos referentes ao sistema operacional necessário para instalar o DB2 estão descritas neste documento:

<http://www.ibm.com/software/data/db2/udb/sysreqs.html>

No que se refere a recurso de hardware, o DB2 Express-C pode ser instalado em sistema com qualquer quantidade de CPU e memória, porém, ele somente utilizará até 2 CPU's e 2GB de memória no caso da versão grátis, e até 4 CPU's e 4GB de memória na versão paga. Os sistemas podem ser físicos ou virtuais, criados a partir de partições ou sendo executados em uma máquina virtual. É claro que você pode executar o DB2 Express-C em sistemas menores, por exemplo, com um único processador e 1GB de memória.

Para conhecer as últimas informações sobre os requisitos de hardware do DB2 Express-C acesse o site: <http://www-306.ibm.com/software/data/db2/express/getstarted.html>

### 3.2 Regras de instalação relativas ao sistema operacional

Para instalar o DB2 Express-C no Linux ou no Windows, você deve usar o sistema operacional de acordo com as seguintes regras.

No **Linux**, você deve ser o usuário root (ou super usuário) para instalar o DB2 Express-C. Você também pode instalar o DB2 Express-C como um usuário não root; porém, você será limitado em relação ao o que você pode realizar com o produto. Por exemplo, através de uma instalação com um usuário não root, você não pode criar nenhuma outra instancias além da padrão.

No **Windows**, o usuário deve pertencer ao grupo Administrador da máquina onde você realizará a instalação. Alternativamente, um usuário não administrador pode ser usado, se um membro do grupo de Administradores do Windows fornecer os devidos privilégios para que este usuário possa realizar a instalação.

Com relação ao domínio das contas no Windows, para verificar os IDs dos usuários no servidor do DB2, o ID do usuário de instalação deve pertencer ao grupo de Administradores no mesmo domínio que as contas DB2 serão criadas. Você também pode usar a conta padrão Sistema Local para realizar a instalação de todos os produtos.

Este usuário também deve ter permissão para “Acessar este computador a partir da rede”.

**Nota:**

Assista o vídeo sobre a instalação do DB2 Express-C neste link:  
<http://www.channeldb2.com/video/video/show?id=807741:Video:4442>

### 3.3 O wizard de instalação

Existem vários métodos de instalação do DB2 Express-C, o mais fácil é utilizar a ferramenta GUI do wizard de instalação do DB2. Depois de realizar o download e a descompactação da imagem do DB2 Express-C, você pode chamar o wizard para realizar a instalação:

- Windows: execute o arquivo setup.exe que está no diretório EXP/image
- Linux: execute o comando db2setup no diretório exp/disk1

É muito fácil de instalar o DB2 Express-C seguindo as instruções do wizard de instalação do DB2. Na maioria dos casos, os valores padrão são suficientes, sendo assim, tudo o que você precisa fazer é aceitar a licença, clicar no botão “Next” repetidas vezes e clicar no botão “Finish”. Depois de alguns minutos, sua instalação estará completa e o DB2 estará rodando.

A Figura 3.1 mostra a tela “DB2 Setup Launchpad” (Lançamento da Configuração do DB2). Clique em “Install a Product” (Instale o Produto) e escolha “Install New” (Nova Instalação) para instalar uma nova cópia do DB2 Express-C em seu sistema.

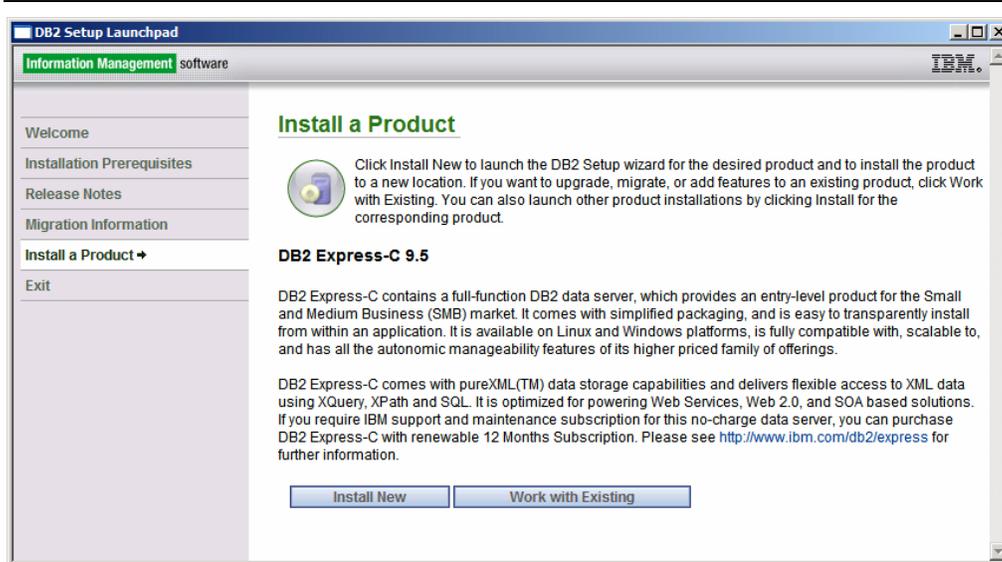


Figura 3.1 – Lançamento da Configuração do DB2

Depois de aceitar a licença, é aconselhado escolher a instalação típica (Typical) que é a padrão conforme mostre a Figura 3.2.

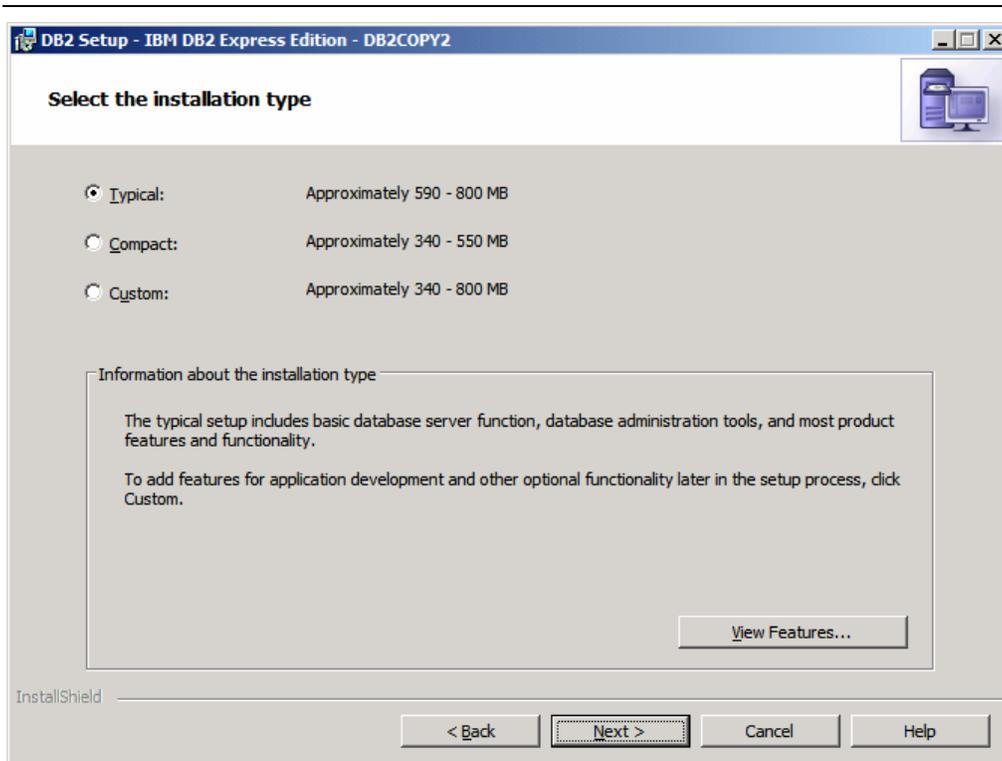
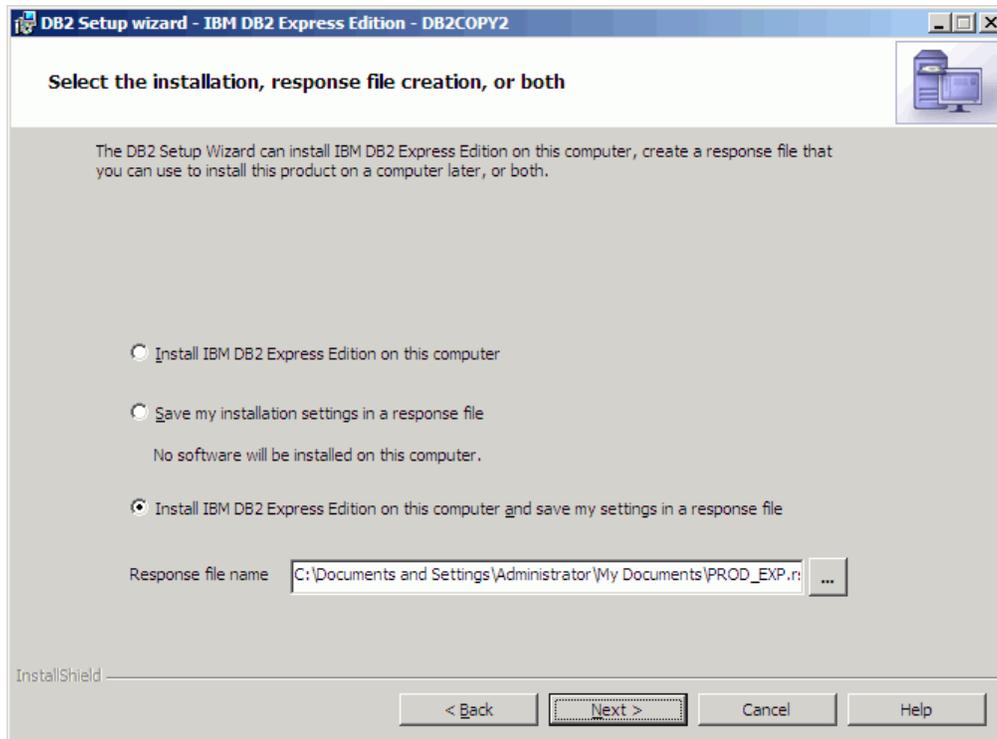


Figura 3.2 – Tipos de instalação

Na Figura 3.3, você tem as opções de instalação do produto, criando um arquivo de resposta ou ambas. O arquivo de resposta (response file) será discutido na seção 3.4, Instalação Silenciosa. Escolha o padrão (Install IBM DB2 Express Edition on this computer and save my settings in a response file).



**Figura 3.3 – Selecionando os tipos de instalação**

Escolha os valores padrão para as próximas telas. Quando você chegar na tela mostrada na Figura 3.4, você pode entrar com o nome do usuário que será utilizado para trabalhar com as instancias e outros serviços. Este usuário deve fazer parte do grupo de Administradores Locais do Windows. Se o ID do usuário que você inserir não existir, ele será criado como Administrador Local. Você pode deixar o campo domain (domínio) em branco se o ID do usuário não pertencer a nenhum domínio. O ID padrão do usuário criado no Windows é db2admin. No caso do Linux, o ID do usuário padrão é chamado db2inst1.

DB2 Setup - IBM DB2 Express Edition - DB2COPY2

### Set user information for the default DB2 instance

Specify the required user information that the DB2 instance, DB2\_01, will use to log on to your system.

It is highly recommended that you use a local user or domain user account instead of the LocalSystem account. Further details are available by clicking Help.

Local user or Domain user account

LocalSystem account

Use the same user name and password for the remaining DB2 services

InstallShield

< Back   Next >   Cancel   Help

**Figura 3.4 – Detalhando as informações do usuário na configuração de uma instancia padrão do DB2**

Finalmente, na Figura 3.5, o wizard de instalação mostra um resumo do que foi instalado, e todas as informações de configurações. Quando você clicar em “Finish” (Terminar), a instalação começará, e os arquivos de programa serão gravados no seu sistema.

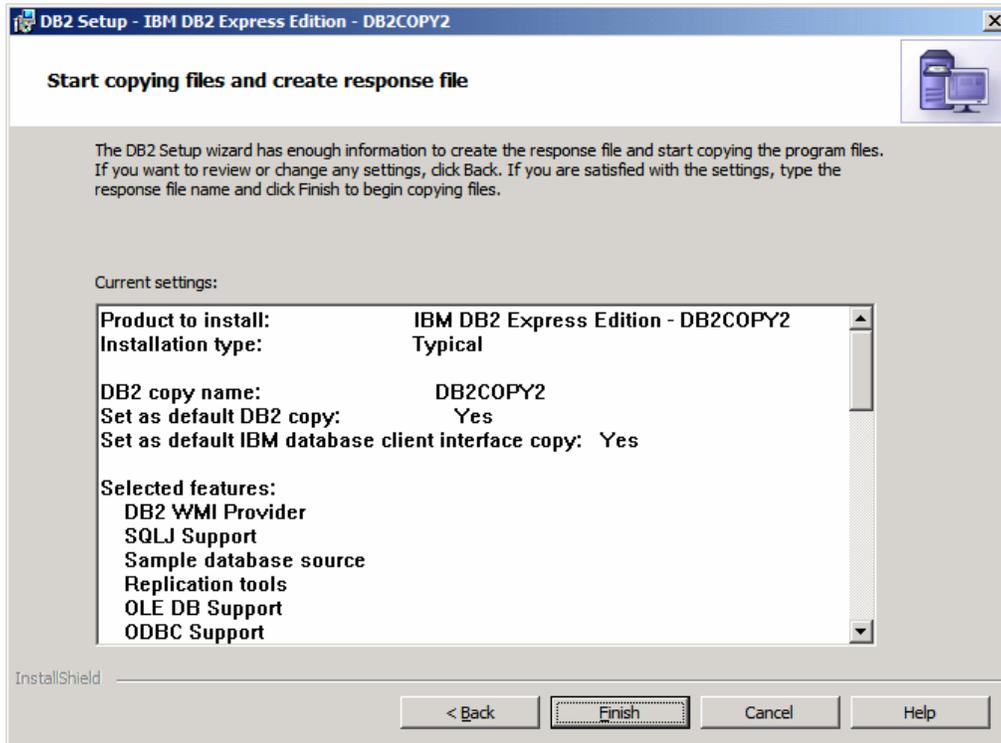


Figura 3.5 – Resumo de como será a instalação

### 3.4 Instalação Silenciosa

Existirão situações onde você deverá instalar o cliente do DB2 em diversos computadores; ou você terá que empacotar o servidor de dados do DB2 como parte da sua aplicação, fazendo com que ele seja instalado no mesmo processo de instalação da sua aplicação. Nestas situações, uma instalação silenciosa é o caminho ideal para instalar o DB2.

O DB2 permite instalações silenciosas através da utilização de um arquivo texto que tem todas as respostas que permitem realizar uma instalação completa. O texto abaixo mostra uma parte de um exemplo de arquivo de resposta.

```
Exemplo de parte de um 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
```

```
...
```

Existem várias maneiras de criar um arquivo de resposta.

- ▶ Instalar o DB2 Express-C em um computador utilizando o wizard de instalação do DB2. Uma das primeiras opções do wizard permite que você selecione uma opção que garante a criação do arquivo de resposta. No final do wizard, um arquivo de resposta será criado no diretório que você especificar. O arquivo gerado será um arquivo texto, sendo assim, você pode editá-lo, como você pode ver na Figura 3.3.
- ▶ Editar o arquivo de resposta que está incluído na imagem do DB2. Este arquivo de resposta (que possui a extensão .rsp) está localizado no diretório *db2/platform/samples/*
- ▶ No Windows, você também pode utilizar o comando de geração do arquivo de resposta:

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

Para instalar o DB2 utilizando o arquivo de resposta no Windows execute o seguinte comando:

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

No Linux execute o seguinte comando:

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

## QuickLab #1: Instalando o DB2 Express-C & criando a base de dados SAMPLE

### Objetivo

Antes que você possa começar a explorar todas as características e ferramentas incluídas no DB2 Express-C, você precisa instalá-lo em seu sistema. Neste Quicklab você realizará uma instalação básica do DB2 Express-C no Windows. O mesmo wizard de instalação está disponível no Linux e os passos são muito parecidos com os do Windows.

### Procedimentos

1. **Obtenha a imagem do DB2 Express-C.** Realize o download da imagem do DB2 Express-C, ou encomende o DVD Discovery Kit com as imagens do DB2 Express-C no site [ibm.com/db2/express](http://ibm.com/db2/express). Descompacte os arquivos em um diretório de sua escolha.
2. **Localize os arquivos.** Navegue nos diretórios (ou drives) que possuem os arquivos de instalação do DB2.
3. **Chame a tela inicial de instalação.** Inicie a instalação clicando duplo no arquivo `setup.exe`. No Linux, execute o comando `db2setup` com o usuário root. A partir da tela inicial, clique na opção `db2setup` no lado esquerdo da janela.
4. **Execute o wizard do DB2.** O wizard de instalação do DB2 verifica se todos os pré-requisitos de instalação são atendidos e verifica se existem instalações prévias do DB2. Clique no botão *Next* e continue a instalação.
5. **Revise o contrato de licença.** Leia e aceite o contrato de licença (selecione o botão “*I Accept...*”) e clique no botão *Next*.
6. **Escolha o tipo de instalação.** Neste exercício, selecione a opção *Typical* (esta é a opção padrão). A opção *Compact* realiza uma instalação básica, enquanto que a opção *Compact* permite que você customize as características que você deseja instalar. Clique no botão *Next* e continue a instalação.
7. **Selecione a pasta de instalação.** Esta janela permite que você escolha o drive e o diretório onde os binários do DB2 serão instalados no seu sistema. Utilize o drive padrão e o diretório que são mostrados abaixo:

**Drive:** C:

**Diretório:** C:\Program Files\IBM\SQLLIB

Clique no botão *Next* para continuar.

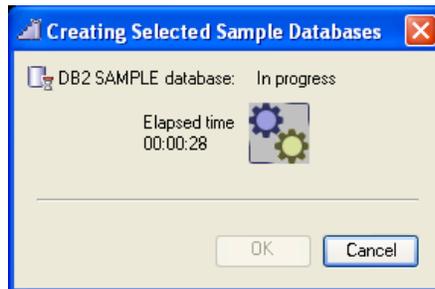
8. **Insira as informações do usuário.** Uma vez que o DB2 Express-C é instalado, alguns processos do DB2 são executados como serviços do sistema. Estes serviços precisam ser executados através de uma conta do sistema operacional. No ambiente Windows, é recomendado utilizar a conta **db2admin**. Se a conta não existir, o DB2 a cria no momento da instalação. Você também pode utilizar outra conta já existente, porém esta conta deve ter o privilégio de administrador local. Nós recomendamos a utilização dos valores padrão. Tenha certeza que você definiu uma senha para esta conta. No Linux, use o ID de usuário padrão **db2inst1** como dono da instancia, utilize o usuário **db2fenc1** para o usuário “fenced” e o usuário **dasusr1** como sendo o Administrador do DB2. Clique no botão *Next* para continuar.
  
9. Configure a instancia do DB2. A instancia do DB2 é o local onde as bases de dados são armazenadas. Uma instancia deve existir antes da criação da base de dados. Durante a instalação no ambiente Windows, uma instancia chamada DB2 é automaticamente criada. No ambiente Linux, o nome da instancia padrão é *db2inst1*. Nos trataremos este temas nos próximos capítulos deste livro.

O padrão é que as instancias do DB2 são configuradas para se comunicar via conexões TCP/IP na porta 50000. Tanto o protocolo quanto a porta padrões podem ser alterados clicando nos botões *Protocols* e *Startup*, respectivamente. Nós recomendamos a utilização dos valores padrão neste exercício. Clique no botão *Next* para continuar.

10. **Início da instalação.** Revise as opções de instalação previamente selecionadas. Clique no botão *Install* para começar a copiar os arquivos para o local de instalação. O DB2 também fará uma configuração inicial de alguns processos.
  
11. **Primeiros passos.** Depois que a instalação finalizar, outro utilitário, chamado “First Steps” é mostrado. Este utilitário também pode ser chamado em outros momentos através do comando `db2fs`.
  
12. A base de dados SAMPLE pode ser utilizada para testes. Esta base de dados é criada automaticamente depois da instalação do DB2. Verifique se a base de dados existe abrindo a ferramenta DB2 Control Center, escolhendo, a partir no Menu Iniciar do Windows: Iniciar -> Programas -> IBM DB2 -> *DB2COPY1 (Default)* -> *General Administration Tools* -> *Control Center*  
Você também pode iniciar o Control Center com o comando `db2cc`.
  
13. Se a base de dados SAMPLE é mostrada no Control Center você pode pular para o passo 16. Se você não conseguiu visualizar esta base de

dados, clique no item *Refresh* no Control Center e depois no menu *View* para atualizar as informações. Se a base de dados SAMPLE continua não aparecendo, então ela pode não ter sido criada. Você pode criá-la manualmente utilizando o utilitário First Steps. Escolha a aba “Database Creation” e siga as orientações do wizard para criar a base de dados SAMPLE. Assegure-se que a opção “XML and SQL objects and data” está selecionada e clique no botão OK. Esta última opção criará a base de dados UNICODE a qual é necessária, na versão 9, para suporte pureXML, porém não é necessária no DB2 9.5.

14. A seguinte tela será mostrada enquanto a base de dados esta sendo criada. (Este procedimento pode durar alguns minutos.) Quando a criação da base de dados terminar, clique no botão *OK* e fecha a ferramenta First Steps.



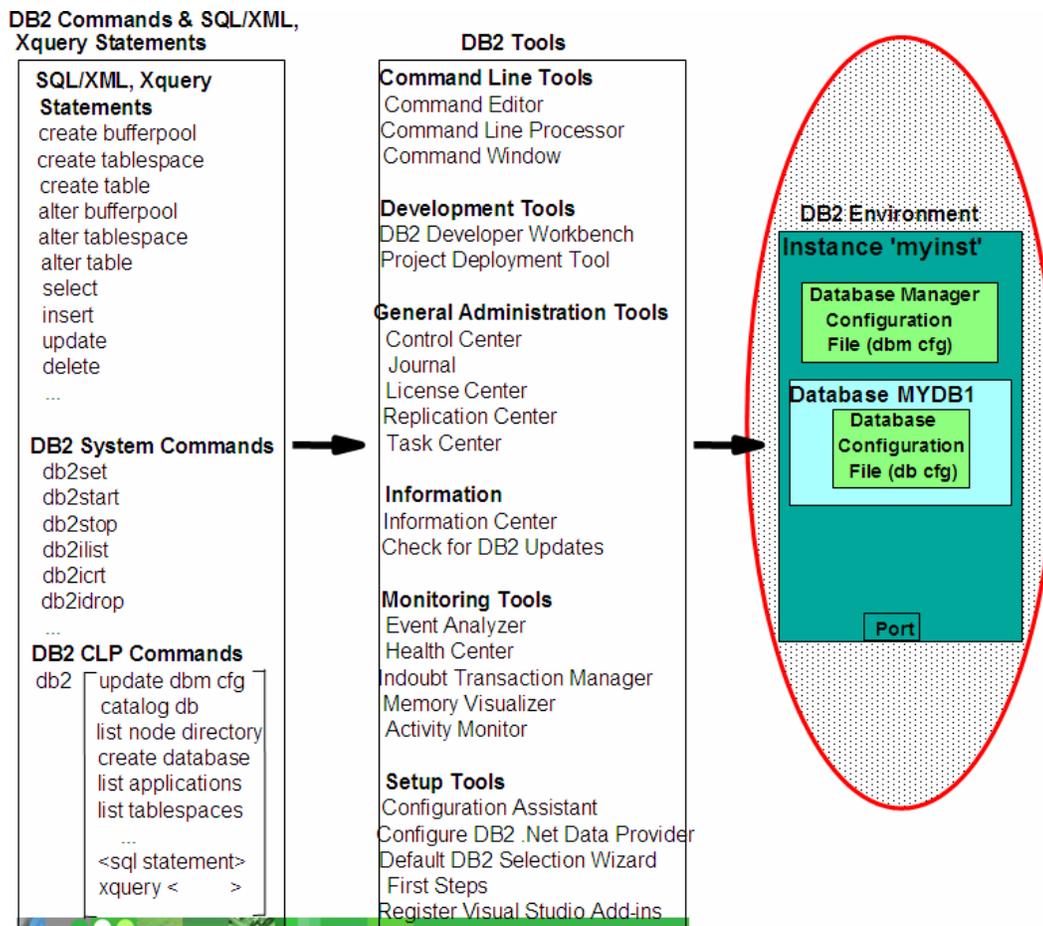
15. Volte ao Control Center e verifique que uma base de dados chamada SAMPLE aparece na árvore de objetos. Talvez você precise realizar a atualização da visão do Control Center para visualizar as alterações. Para fazer isto, selecione o item *Refresh* a partir do menu *View* do Control Center.
16. **Reinicialize o computador.** Embora este passo não seja mencionado na documentação oficial de instalação do DB2, nós recomendamos a reinicialização do sistema (se possível, pelo menos no ambiente Windows) para assegurar que todos os processos sejam iniciados com sucesso e para limpar os recursos de memória. Este passo é OPCIONAL.



# 4

## Capítulo 4 – Ambiente do DB2

Neste capítulo, discutiremos sobre o ambiente do DB2. A Figura 4.1 fornece um sumário sobre o DB2, e a elipse vermelha mostra a área de foco deste capítulo. O lado esquerdo da figura mostra os diferentes comandos de DB2, SQL, SQL/XML, e comandos Xquery que podem ser criados para interagir com um servidor DB2. O centro da figura mostra o nome de diferentes ferramentas usadas para interagir com um servidor DB2. O lado direito da figura mostra o ambiente básico do DB2, que consiste em uma instância, uma base de dados e os arquivos de configuração associados.



**Figura 4.1 – Visão geral do ambiente DB2**

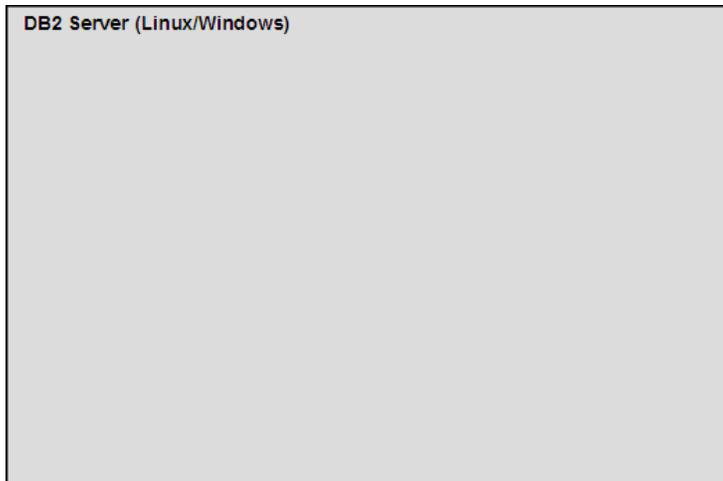
**Nota:**

Para maiores informações sobre o ambiente do DB2, assista aos vídeos:

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

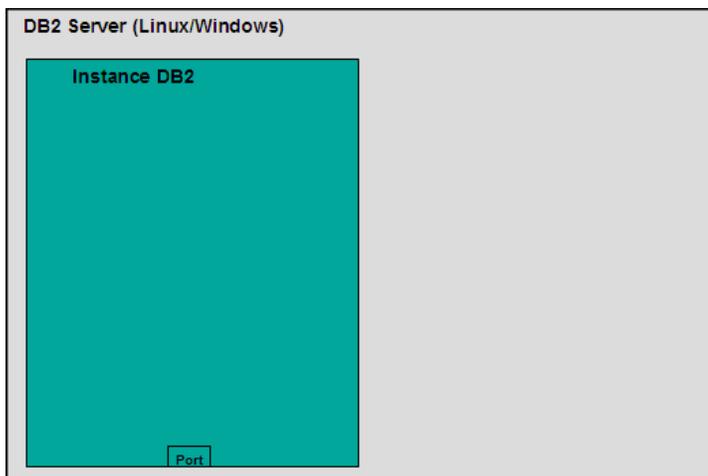
<http://www.channeldb2.com/video/video/show?id=807741:Video:4042>

Para mostrar o ambiente do DB2, vamos descrever cada elemento que o compõe passo a passo. A Figura 4.2 mostra uma representação de um servidor DB2 após a instalação do DB2 Express-C 9.5.



**Figura 4.2 – Representação de um Servidor DB2 após a instalação do DB2 Express-C 9.5**

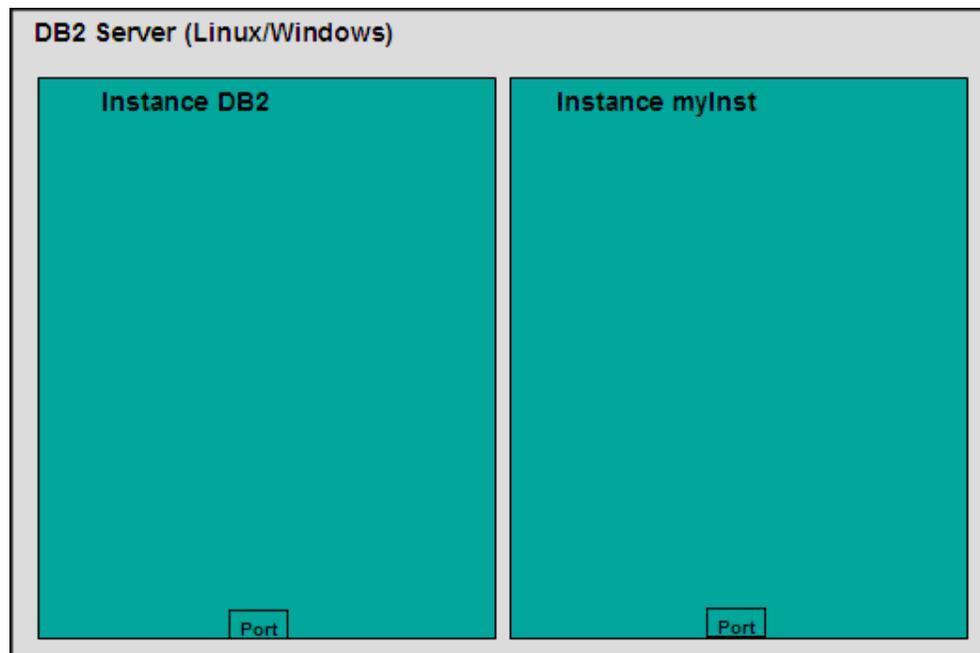
Como parte da instalação no Windows, uma instância padrão, chamada "DB2" ("db2inst1" no Linux) é criada. Isto é representado pelo retângulo verde da Figura 4.3. Uma instância é simplesmente um ambiente independente onde aplicações podem ser executadas e bases de dados podem ser criadas. Você pode criar várias instâncias em um servidor, e usá-las para diferentes propósitos. Por exemplo, uma instância pode ser usada para armazenar bases de dados de ambiente de produção, outra instância pode ser usada para bases de dados do ambiente de testes, e outra para um ambiente de desenvolvimento. Todas estas instâncias são independentes; isto é, as operações executadas em uma instância não afetam as outras instâncias.



**Figura 4.3 – A instância padrão criada pelo DB2**

Para criar uma nova instância no DB2, use o comando `db2icrt <instance name>`, onde `<instance name>` é substituído por um nome com 8 caracteres. Por exemplo, para criar a instância `myinst`, usamos o comando: `db2icrt myinst`.

A Figura 4.4 mostra uma nova instância chamada `myinst` em um retângulo verde separado.



**Figura 4.4 – Um servidor DB2 com duas instâncias**

Note que cada instância possui um número de porta único. Isto ajuda a distinguir as instâncias quando você deseja se conectar a uma base de dados em uma determinada instância a partir de um cliente remoto. Se você utilizar o DB2 Command Window, você pode tornar qualquer instância do DB2 ativa usando o seguinte comando do sistema operacional Windows:

```
set db2instance=myinst
```

Neste exemplo, se você criar agora uma base de dados a partir do Command Window, ela será criada na instância `myinst`.

Para listar as instâncias, execute o comando:

```
db2ilist
```

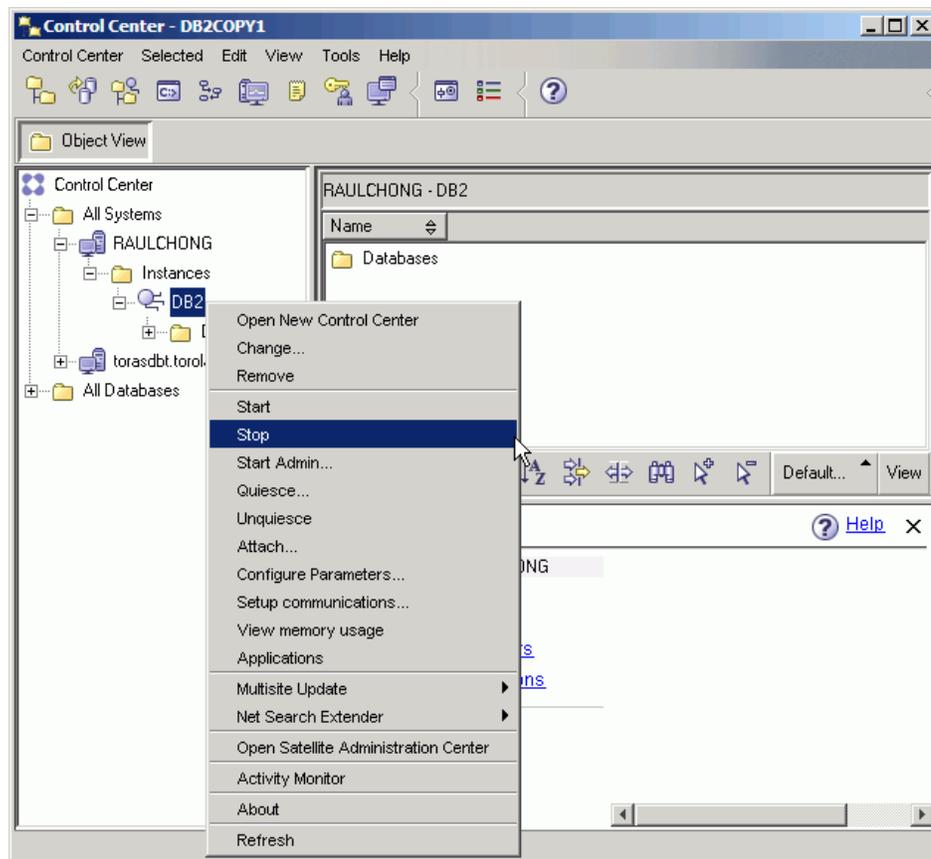
No Linux, uma instância deve corresponder a um usuário do sistema operacional Linux; portanto, para trocar de instância basta simplesmente trocar de usuário (com o comando `su`).

A Tabela 4.1 mostra alguns comandos úteis para se trabalhar em nível de instância.

Comando	Descrição
db2start	Inicia a instância atual
db2stop	Pára a instância atual
db2icrt	Cria uma nova instância
db2idrop	Exclui uma instância
db2ilist	Lista as instâncias que você possui em seu sistema
db2 get instance	Lista as instâncias ativas

**Tabela 4.1 – Comandos úteis do DB2 para se trabalhar em nível de instância**

Alguns dos comandos acima podem também ser executados através do Control Center. Por exemplo, no Control Center, se você expandir o diretório *Instances* e clicar com o botão direito na instância desejada, você pode escolher a opção *Start*, que é equivalente a executar o comando `db2start` no DB2 Command Window, ou a opção *Stop*, que é equivalente a executar o comando `db2stop`, como mostra a Figura 4.5.



**Figura 4.5 – Comandos de instância no Control Center**

Para criar uma base de dados na instância ativa, execute o comando abaixo no DB2 Command Window:

```
db2 create database mydb1
```

Para listar todas as bases de dados criadas, execute o comando:

```
db2 list db directory
```

Dentro de cada instância você pode criar várias bases de dados. Uma base de dados é uma coleção de objetos, como tabelas, visões, índices e outros. Bases de dados são unidades independentes, e, portanto, não compartilham objetos com outras bases de dados. A Figura 4.6 mostra uma representação de uma base de dados “MYDB1” criada dentro da instância “DB2”.

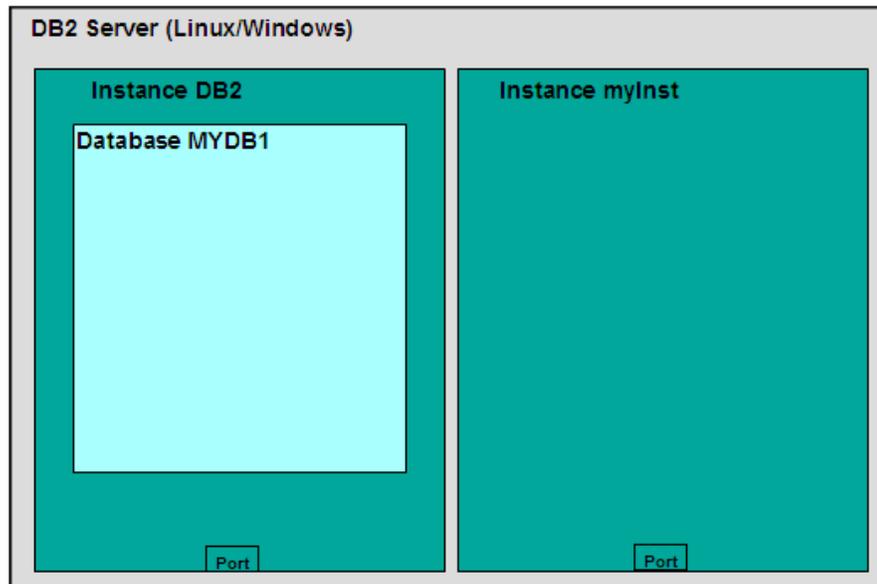


Figura 4.6 – Base de dados “MYDB1” criada na instância “DB2”

A Tabela 4.2 mostra alguns comandos que você pode usar em nível de base de dados.

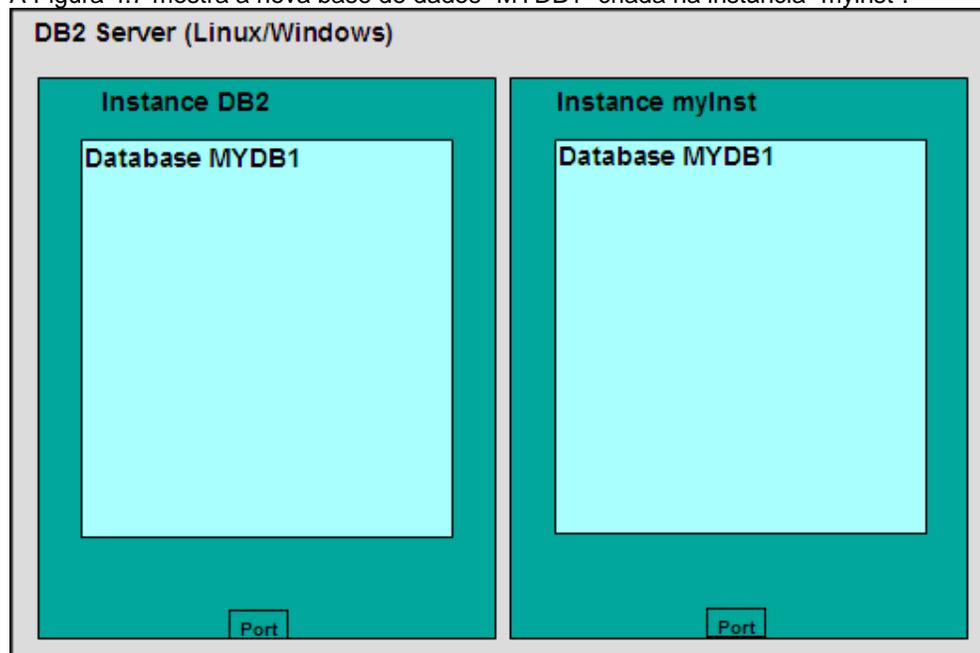
Comando SQL	Descrição
db2 create database	Cria uma nova base de dados
db2 drop database	Exclui uma base de dados
db2 connect to <database_name>	Conecta a uma base de dados
db2 create table/create visão/create index	Comandos SQL para criar tabelas, visões e índices, respectivamente

Tabela 4.2 – Comandos SQL em nível de bases de dados

Se quisermos criar uma outra base de dados com o mesmo nome (MYDB1), mas na instância “myinst”, os seguintes comandos devem ser executados no DB2 Command Window:

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

A Figura 4.7 mostra a nova base de dados “MYDB1” criada na instância “myInst”.



**Figura 4.7 – Base de dados “MYDB1” criada na instância “myInst”**

Quando uma base de dados é criada, vários objetos são criados por padrão: tablespaces, tabelas, um buffer pool e arquivos de log. A criação destes objetos leva um certo tempo, é por isso que o comando `create database` leva alguns minutos para ser processado. A Figura 4.8 mostra três tablespaces criadas por padrão. Tablespaces serão discutidas com mais detalhes no Capítulo 6, que trata da Arquitetura do DB2; mas, para o momento, pense em tablespaces como uma camada lógica entre tabelas e recursos físicos, como disco e memória.

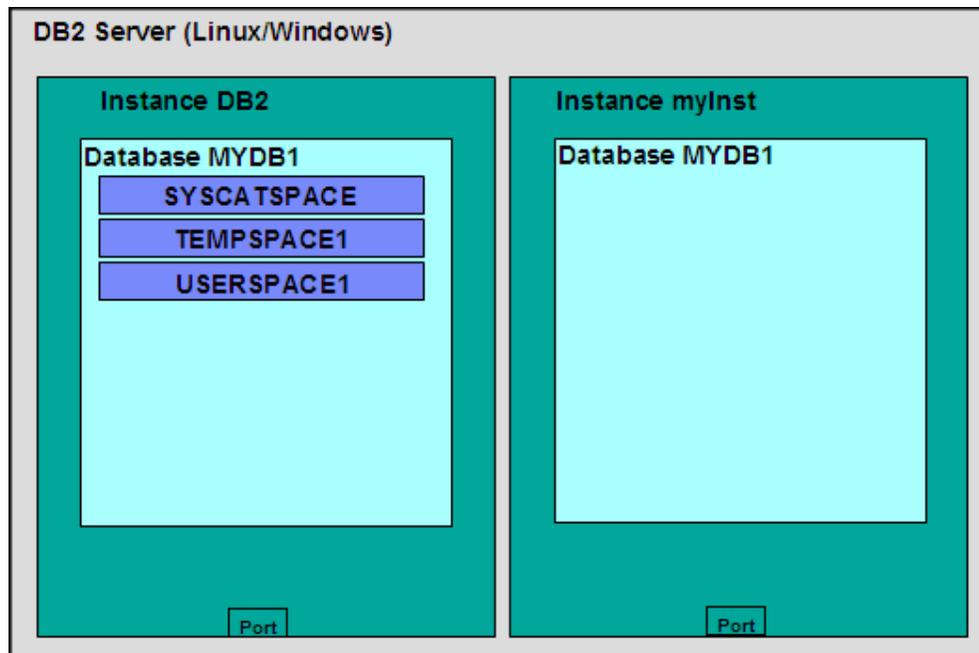


Figura 4.8 –Tablespaces criadas por padrão quando uma base de dados é criada

A tablespace SYSCATSPACE contém as tabelas de catálogo. O catálogo também é conhecido como dicionário de dados em outros sistemas gerenciadores de banco de dados relacional. Ele contém basicamente informações de sistema que não podem ser modificadas ou excluídas; senão a base de dados não funcionará corretamente. A tablespace TEMPSPACE1 é usada pelo DB2 quando é necessário um espaço adicional para a realização de algumas operações, como ordenações (*sorts*). A tablespace USERSPACE1 é geralmente usada para armazenar tabelas de usuários se não houver a especificação de uma tablespace no momento de criação de uma tabela.

Você também pode criar suas próprias tablespaces usando o comando CREATE TABLESPACE. A Figura 4.9 mostra a tablespace MYTBLS1 criada dentro da base de dados MYDB1 na instância DB2. Quando se cria uma tablespace, deve-se especificar quais discos e memória (buffer pool) devem ser usados. Portanto, se você tem uma "hot" table, isto é, uma tabela que é usada constantemente, você pode alocar os discos mais rápidos e a maior quantidade de memória se atribuir à tabela uma tablespace com estas características.

Na Figura 4.9 são mostrados dois outros objetos criados por padrão: um buffer pool chamado IBMDEFAULTBP e os arquivos de log.

Um buffer pool é basicamente um cache usado pela base de dados. Você pode criar um ou mais buffer pools, mas sempre deve haver um buffer pool com o tamanho de página correspondente ao tamanho de página dos tablespaces existentes. Páginas e tamanho de páginas serão discutidos com mais detalhes no Capítulo 6, Arquitetura do DB2.

Os arquivos de log são usados para recuperação. Quando se trabalha com uma base de dados, não são só informações desta base que são armazenadas nos discos. Enquanto se trabalha com esta base, arquivos de log armazenam todas as operações executadas sobre os dados. Pense em logs como arquivos temporários onde existe uma operação que salva automaticamente o arquivo. Logs são discutidos com mais detalhes no Capítulo 11: Backup e Recuperação.

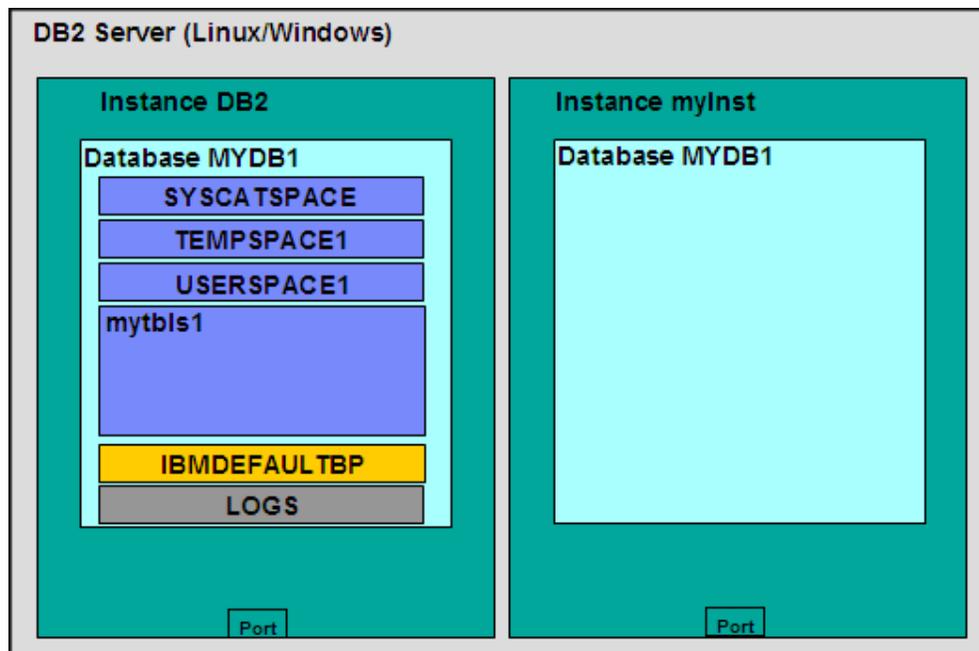


Figura 4.9 – Buffer pool e logs criados por padrão

Anteriormente discutimos que instâncias são ambientes independentes e, portanto, uma base de dados com o mesmo nome pode ser criada em diversas instâncias. Como as instâncias, bases de dados também são unidades independentes; logo, objetos em uma base de dados não possuem nenhuma relação com objetos em outras bases de dados. A figura 4.10 mostra a tablespace “mytbls1” em ambas as bases de dados MYDB1 e SAMPLE, dentro da instância DB2. Isto é válido pelo fato das bases de dados serem unidades independentes. Observe que a Figura 4.10 não mostra os outros objetos padrão para a base de dados SAMPLE devido a restrições de espaço na figura.

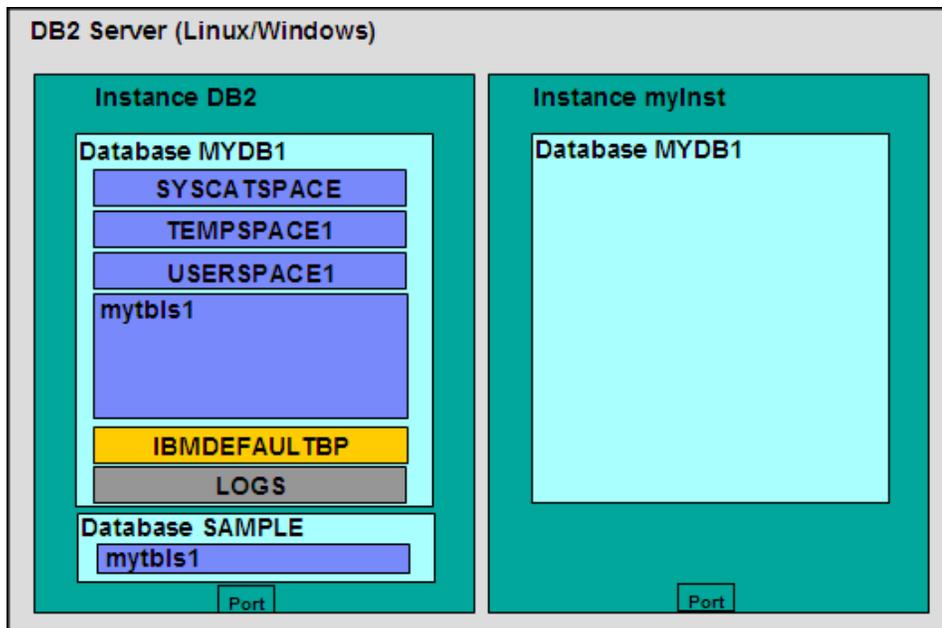


Figura 4.10 – Tablespaces com o mesmo nome em diferentes bases de dados

Uma vez que o tablespace foi criado, pode-se criar objetos dentro do tablespace, como tabelas, visões e índices. Isto é ilustrado na Figura 4.11.

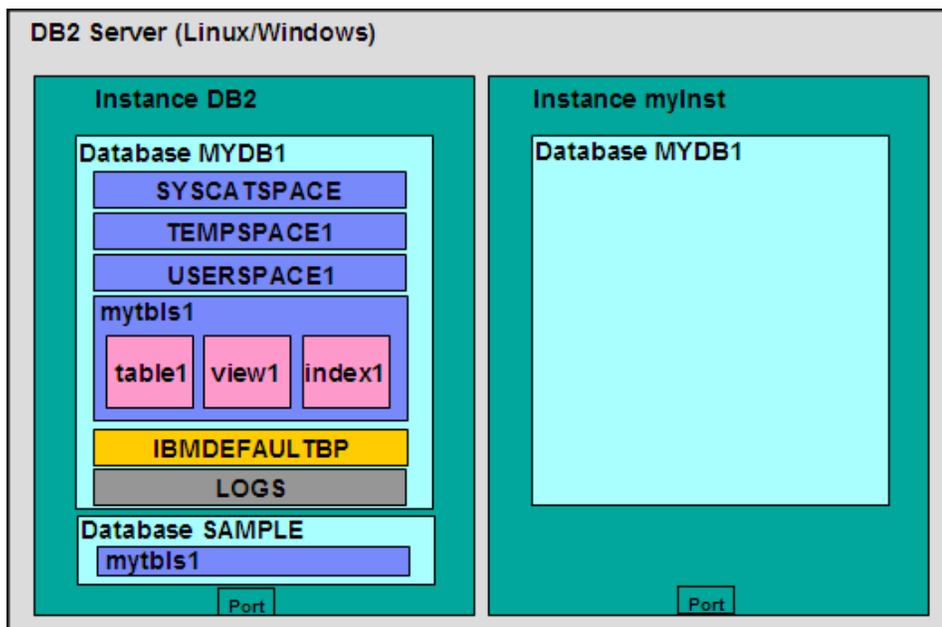


Figura 4.11 – Tabelas, visões, índices criados dentro do tablespace

## Quicklab #2 – Criando uma nova base de dados

### Objetivo

Neste Quicklab, você criará uma nova base de dados usando o Create Database wizard, no Control Center.

### Procedimento

1. A partir da janela Object Tree do Control Center, clique com o botão direito no diretório *All Databases*, selecione a opção *Create Database*, e escolha o item *With Automatic Maintenance*. Isto abrirá o Create Database Wizard.

2. Especifique o nome da base de dados e local na página *Name* do wizard. Use os seguintes valores:

Nome da base de dados:	EXPRESS
Drive padrão(Windows):	C:
Caminho padrão (Linux):	/home/db2inst1
Alias:	Será EXPRESS se for deixado em branco
Comentário:	É opcional e pode ser deixado em branco

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

3. Na página *Specify where to store your data* não faça nenhuma modificação e clique *Next*.
4. Na página *Select your maintenance strategy*, deixe o padrão (*Yes, I can specify an offline ...*), e clique *Next*.
5. Especificar a janela de manutenção de tempo na página *Timing* do wizard. Especifique duas ou mais horas por semana quando o DB2 pode executar tarefas de manutenção automática para preservar a operacionalidade de sua base de dados.
6. Configurar notificação na página *Mail Server* do wizard. O DB2 pode mandar um e-mail ou uma mensagem para um pager automaticamente se um problema ou uma condição não-operacional for detectada. Se você quiser configurar isso, indique um servidor SMTP disponível para que o DB2 possa utilizar para enviar e-mails. Para este quicklab, nós não usaremos um servidor SMTP, então deixe esta opção em branco e clique *Next*.
7. Revise as opções selecionadas na página *Summary* do wizard. Clique no botão *Finish* para iniciar o processo de criação da base de dados. A

criação da base de dados geralmente leva alguns minutos, neste tempo, uma barra de indicação de progresso é mostrada na tela.

#### 4.1 Configuração do DB2

Os parâmetros do DB2 podem ser configurados usando a ferramenta Configuration Advisor. Para acessar esta ferramenta pelo Control Center clique com o botão direito em uma base de dados e escolha a opção “Configuration Advisor”. Baseado em suas respostas para algumas questões sobre recursos e carga de trabalho do seu sistema, o configuration advisor irá prover uma lista de parâmetros do DB2 que devem ser modificados com valores sugeridos. Se você deseja saber mais detalhes sobre a configuração do DB2, continue lendo; caso contrário, use o Configuration Advisor e você está pronto para trabalhar com o DB2!

Um servidor DB2 pode ser configurado em quatro níveis diferentes:

- Variáveis de ambiente
- Arquivo de configuração do gerenciador de base de dados (dbm cfg)
- Arquivo de configuração da base de dados (db cfg)
- Registro de perfil do DB2 (DB2 profile registry)

Isto é mostrado na Figura 4.12. Nesta figura, observe a localização de cada retângulo. Por exemplo, as variáveis de ambiente estão localizadas no nível do sistema operacional do servidor, enquanto os parâmetros do arquivo de configuração do gerenciador de base de dados estão localizados no nível de instância. Parâmetros de configuração da base de dados estão localizados no nível da base de dados, e o registro de perfil do DB2 está localizado tanto no nível de sistema operacional quanto no nível de instância.

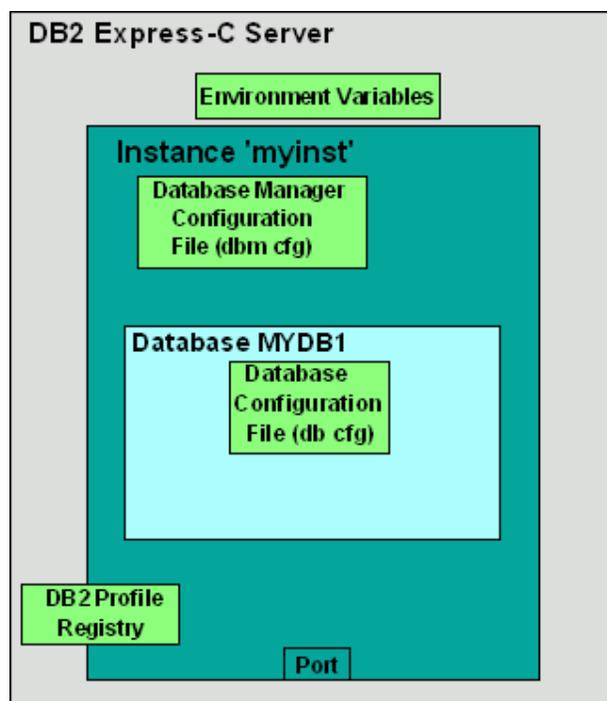


Figura 4.12 – Configuração do DB2

#### 4.1.1 Variáveis de ambiente

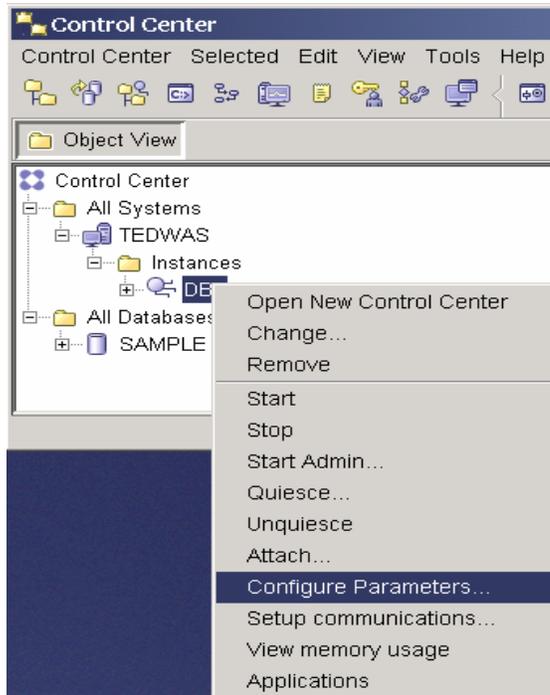
Variáveis de ambiente estão localizadas no nível do sistema operacional. Uma variável de ambiente importante é DB2INSTANCE. Ela indica a instância ativa, a qual você está utilizando no momento que seus comandos DB2 são aplicados. Por exemplo, para tornar ativa a instância “myinst” no Windows, você pode executar o seguinte comando de sistema operacional:

```
set db2instance=myinst
```

#### 4.1.2 Arquivo de configuração do gerenciador de base de dados (dbm cfg)

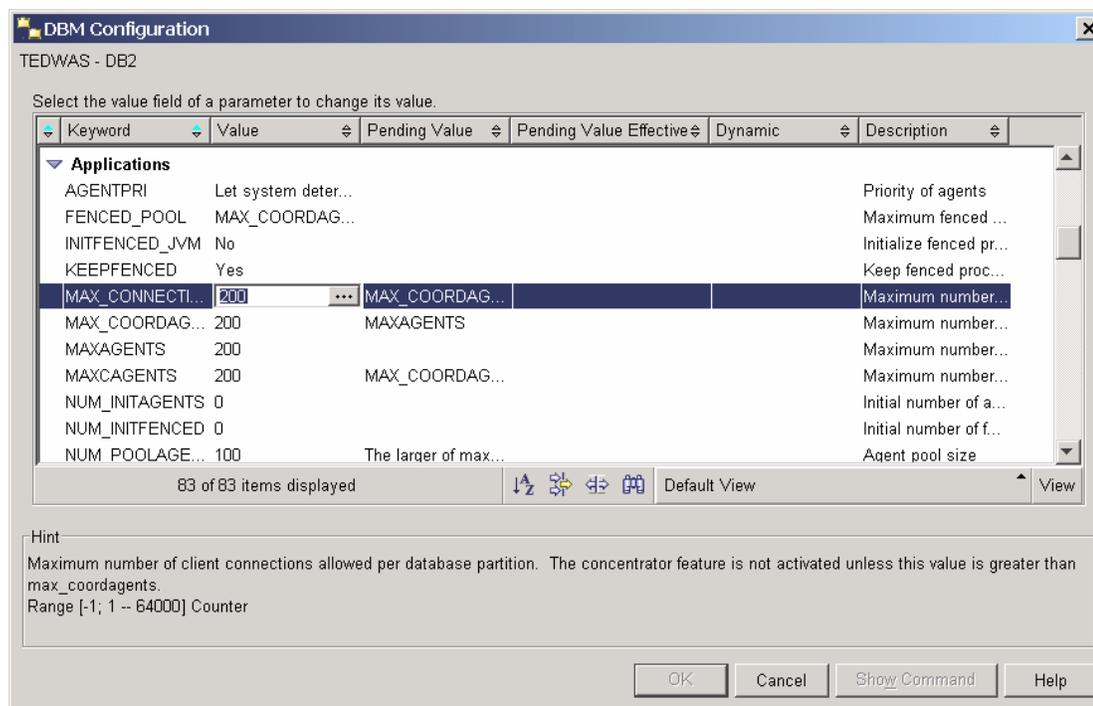
O arquivo de configuração do gerenciador de base de dados (*database manager configuration file - dbm cfg*) inclui parâmetros que afetam a instância e todas as bases de dados nela contidas. O arquivo de configuração do gerenciador de base de dados pode ser visto ou modificado usando linha de comando, ou através do DB2 Control Center.

Para trabalhar com o DBM CFG no Control Center, selecione um objeto de instância a partir do diretório de instâncias do control center, clique com o botão direito para abrir um menu pop up e selecione a opção Configure Parameters, como mostrado na Figura 4.13.



**Figura 4.13 – Configurando o dbm cfg a partir do Control Center**

Depois de escolher a opção *Configure Parameters*, será mostrada a tela da Figura 4.14 com a lista de parâmetros do dbm cfg.



**Figura 4.14 – A caixa de diálogo do dbm cfg**

Muitos parâmetros são dinâmicos, ou seja, as mudanças feitas nestes têm efeito imediato; contudo, a modificação de alguns parâmetros pode requerer que a instância seja parada e iniciada novamente. A partir da linha de comando, isso pode ser feito utilizando os comandos `db2stop` e `db2start`.

Antes de uma instância ser parada, todas as aplicações devem ser desconectadas. Se você deseja forçar a parada da instância, você pode utilizar o comando `db2stop force`.

Uma instância também pode ser parada através do Control Center, clicando no objeto instância e selecionando Stop ou Start.

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

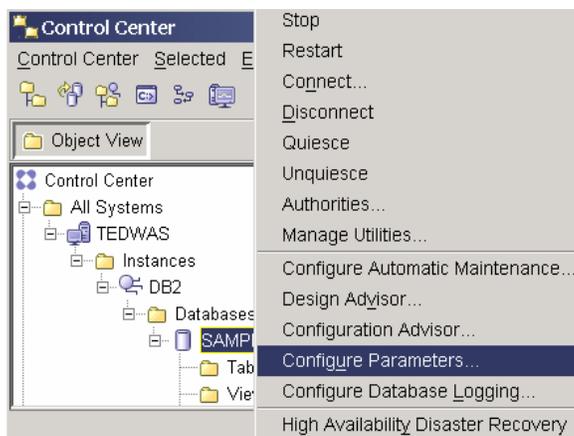
Comando	Descrição
<code>db2 get dbm cfg</code>	Recupera informação sobre o dbm cfg
<code>db2 update dbm cfg using &lt;parameter_name&gt; &lt;value&gt;</code>	Atualiza o valor de um parâmetro do dbm cfg

**Tabela 4.3 - Comandos para manipular o dbm cfg**

### 4.1.3 Arquivo de configuração da base de dados (db cfg)

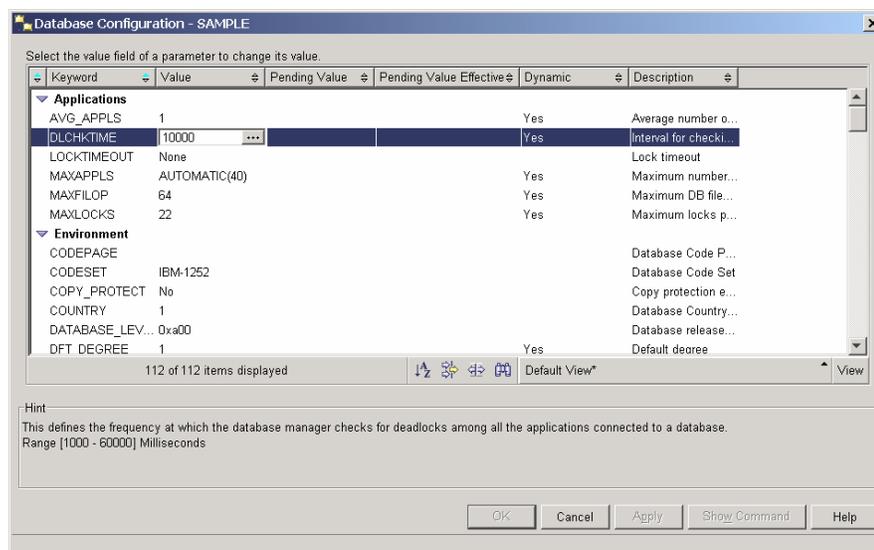
O arquivo de configuração da base de dados (*database configuration file* – dbm cfg) inclui parâmetros que afetam uma determinada base de dados. O arquivo de configuração da base de dados pode ser visto ou modificado usando linha de comando ou através do DB2 Control Center.

Para trabalhar como DB CFG no Control Center, selecione o objeto de base de dados a partir do diretório de base de dados do Control Center, clique com o botão direito para abrir um menu popup e selecione a opção Configure Parameters, como mostrado na Figura 4.15.



**Figura 4.15 – Configurando o db cfg a partir do Control Center**

Depois de escolher a opção Configure Parameters, será mostrada a tela da Figura 4.16 com a lista de parâmetros do db cfg.



**Figura 4.16 – O 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 &lt;database_name&gt;</code>	Recupera informação sobre o db cfg para uma determinada base de dados
<code>update db cfg for &lt;database_name&gt; using &lt;parameter_name&gt; &lt;value&gt;</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 (DB2 profile registry)

As variáveis do registro de perfil do DB2 incluem parâmetros que podem ser específicos para uma plataforma e podem ser alterados globalmente (afetando outras instâncias), ou em nível de instância (afetando uma instância em particular).

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 configuradas no momento
<code>db2set -lr</code>	Lista todas as variáveis do registro de perfil do DB2
<code>db2set &lt;parameter&gt;=&lt;value&gt;</code>	Atribui um determinado valor a um parâmetro

**Tabela 4.5 - Comandos para manipular o registro de perfil do DB2**

A Tabela 4.6 mostra algumas variáveis de registro mais usadas no DB2.

Variável de Registro	Descrição
DB2COMM	Especifica os gerenciadores de comunicação que são inicializados quando o gerenciador de base de dados é iniciado.
DB2_EXTSECURITY	No Windows, previne acesso não autorizado ao DB2, travando os arquivos de sistema do DB2.
DB2_COPY_NAME	Armazena o nome da cópia do DB2 que está em uso. Para mudar para uma cópia diferente do DB2, rode o comando <code>installpath\bin\db2envvars.bat</code> . Não se pode alterar esta variável para modificar a cópia do DB2.

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

Por exemplo, para permitir a comunicação usando TCPIP, configure a variável de registro DB2COMM como é mostrado abaixo:

```
db2set db2comm=tcPIP
```

## 4.2 O DB2 Administration Server

O DB2 Administration Server (DAS) é um processo daemon que roda no servidor DB2 para permitir que clients remotos possam administrar graficamente o servidor DB2. Há apenas um DAS por computador físico, como mostrado na Figura 4.16.

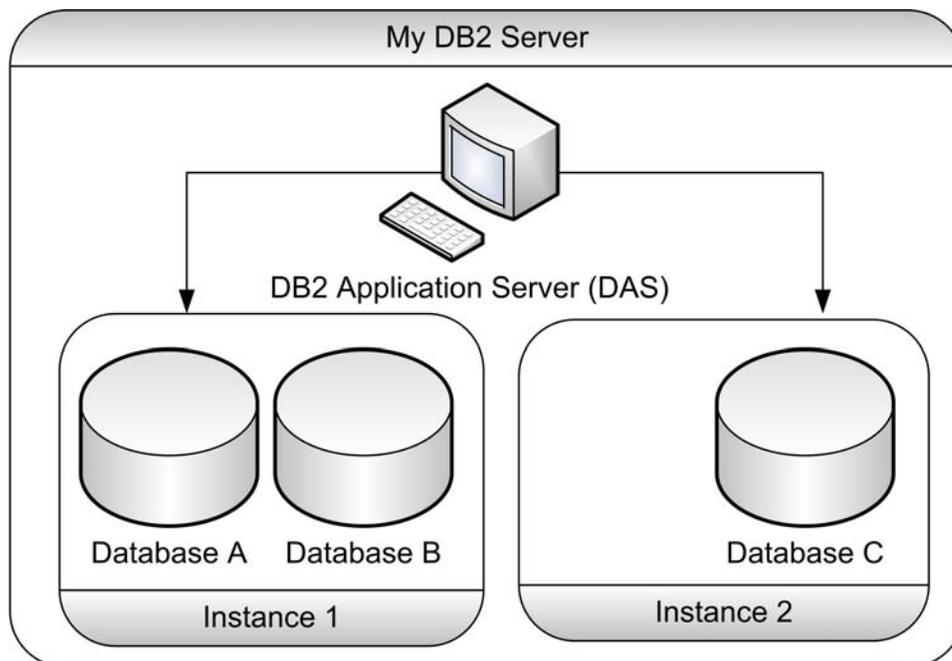


Figura 4.16 – O DB2 Administration Server (DAS)

## Quicklab #3 – Trabalhando com instâncias, bases de dados e configuração

### Objetivo

Neste Quicklab, você irá criar uma nova instância, uma base de dados e alterar parâmetros de configuração em um servidor DB2 no Windows. Você pode fazer isso tanto pelo Control Center quanto pelo Command Window. As instruções apresentadas abaixo são referentes ao uso do Command Window.

### Procedimento

1. No Command Window, crie uma nova instância chamada *newinst*  
`db2icrt newinst`
2. Na nova instância *newinst*, crie uma base de dados *newdb* com os valores padrão  
`set db2instance=newinst`  
`db2start`  
`db2 create database newdb`
3. Liste todas as instâncias do seu servidor  
`db2ilist`
4. Altere para a instância *DB2* e confirme se a instância atual foi realmente alterada  
`set db2instance=db2`  
`db2 get instance`
5. Mude o parâmetro *FEDERATED* do *dbm cfg* para o valor *YES* e verifique se a mudança ocorreu.  
`db2 update dbm cfg using FEDERATED YES`  
`db2 force applications all`  
`db2 terminate`  
`db2stop`  
`db2start`  
`db2 get dbm cfg`
6. Conecte à base de dados *SAMPLE* com o usuário e senha que você está logado no sistema operacional  
`db2 connect to sample user <userID> using <psw>`
7. Liste as aplicações que estão rodando em sua instância atual  
`db2 list applications show detail`

8. Abra outra janela do DB2 Command Window e conecte novamente à base de dados SAMPLE sem especificar usuário e senha. Agora liste novamente as aplicações conectadas à instância.  

```
db2 connect to sample
db2 list applications
```
9. Desconecte uma das conexões do DB2 Command Window  

```
db2 force application (application handle da aplicação "db2bp.exe"
obtido a partir do comando db2 list applications)
```
10. Exclua a instância *newinst*  

```
db2idrop newinst
```
11. Exclua e recrie o DAS, e inicialize-o.  

```
db2admin stop
db2admin drop
db2admin create
db2admin start
```
12. Configure a variável de registro DB2COMM para tcpip e npipe em sua instância  

```
db2set db2comm=tcpip,npipe
db2stop
db2start
```
13. Remova o valor da variável de registro DB2COMM  

```
db2set db2comm=
db2stop
db2start
```
14. Verifique o valor atual do parâmetro LOGSECOND do db cfg, e modifique-o para 5. Verifique se o novo valor foi aplicado  

```
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

Neste capítulo são descritas algumas das ferramentas que podem ser usadas com o DB2. A elipse vermelha da Figura 5.1 mostra a área de foco deste capítulo.

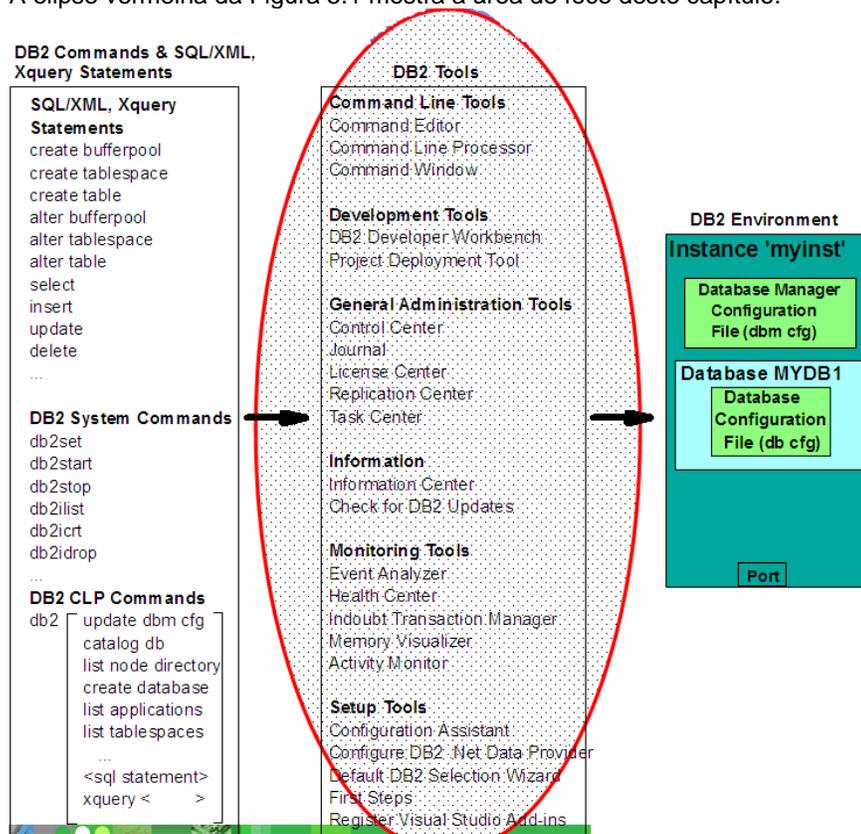


Figura 5.1 – Visão geral do DB2: ferramentas do DB2

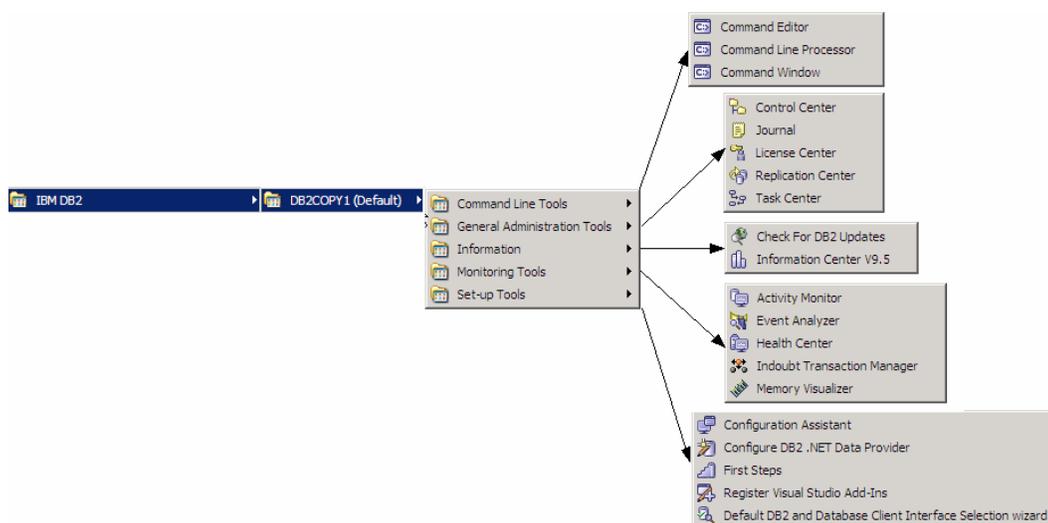
**Nota:**

Para maiores informações sobre ferramentas do DB2 e scripts, assista aos vídeos:

<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 pelo atalho do IBM DB2 a partir do menu Iniciar. A maioria destas ferramentas existem tanto no Linux quanto no Windows.



**Figura 5.2 – Ferramentas do DB2 a partir da opção IBM DB2 do menu Iniciar**

A Tabela 5.1 provê uma lista de atalhos de comandos que podem ser usados para inicializar algumas ferramentas tanto no Linux quanto no Windows.

Nome da ferramenta	Comando
Command Editor	db2ce
Command Line processor	db2
Command Window (Only on Windows platforms)	db2cmd
Control Center	db2cc
Task Center	db2tc
Health Center	db2hc
Configuration Assistant	db2ca
First Steps	db2fs

**Tabela 5.1 – Atalhos de comandos para inicializar algumas ferramentas do DB2**

## 5.1 Control Center

A ferramenta primária do DB2 para administração de base de dados é o Control Center, ilustrado na Figura 5.3.

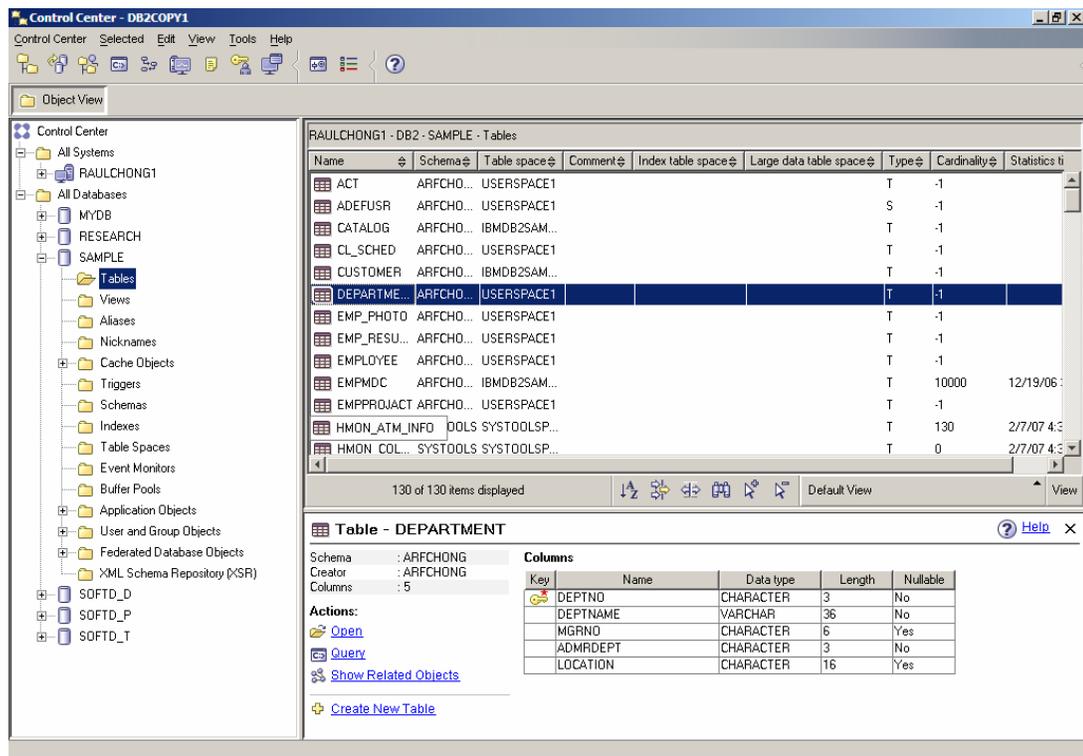


Figura 5.3 - O Control Center do DB2

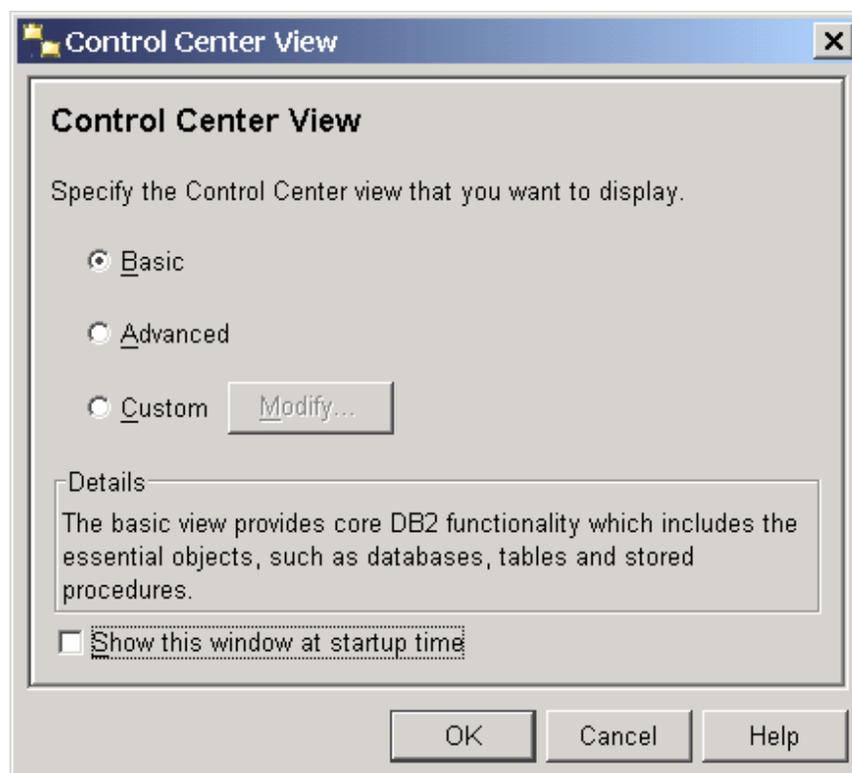
O Control Center é uma ferramenta de administração centralizada que permite que você:

- Liste seus sistemas, instâncias, bases de dados e objetos de bases de dados;
- Crie, modifique e gerencie bases de dados e objetos de bases de dados;
- Abra outras ferramentas gráficas do DB2.

A seção do lado esquerdo do Control Center mostra uma hierarquia visual dos objetos de bases de dados existentes em seu(s) sistema(s), provendo um “diretório” para tabelas (diretório Tables), visões (diretório Views), etc. Quando se clica duas vezes em um diretório (por exemplo no diretório Tables, como mostrado na Figura 5.3), a seção superior à direita irá listar todos os objetos relacionados; neste caso, todas as tabelas associadas com a base de dados SAMPLE. Se você selecionar uma determinada tabela na seção superior direita, a seção inferior direita irá fornecer informações mais específicas sobre tal tabela.

Ao clicar com o botão direito em diferentes diretórios/objetos da árvore de objetos, menus aplicáveis para tal diretório ou objeto são abertos. Por exemplo, clicando com o botão direito em uma instância e escolhendo a opção “Configure parameters” permitirá que você veja e atualize o arquivo de configuração do gerenciador de base de dados daquela instância. Da mesma forma, se você clicar com o botão direito em uma base de dados e escolher a opção “Configure parameters”, você poderá ver e atualizar o arquivo de configuração da base de dados. O ambiente do DB2 e parâmetros de configuração são discutidos com mais detalhes no Capítulo 4, Ambiente do DB2.

Da primeira vez que você executa o Control Center, uma janela pede para escolher qual visão você gostaria de usar. A escolha da visão determina quais tipos de opções e objetos de bases de dados são apresentados. A Figura 5.4 mostra a caixa de diálogo de visão do Control Center.



**Figura 5.4 – A caixa de diálogo de visão do Control Center**

A visão básica provê as principais funcionalidades do DB2.

A visão avançada mostra algumas opções e propriedades.

A visão customizada permite que você escolha propriedades, opções e objetos específicos que você deseja ver.

Para abrir novamente a caixa de diálogo de visão do Control Center, selecione a opção “Customize Control Center” a partir do menu Tools, como mostrado na Figura 5.5.

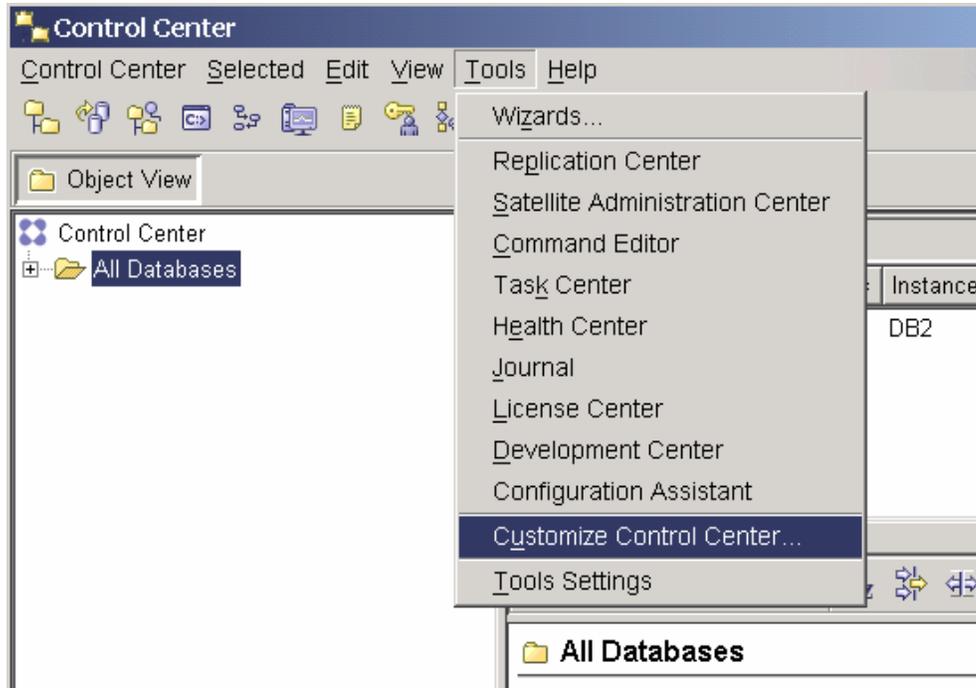


Figura 5.5 – Customizando o Control Center

#### Executando o Control Center

Há muitas formas de executar o Control Center:

- Navegando através do menu *Iniciar* do Windows
- Executando o comando *db2cc* em um prompt de comando
- Clicando no ícone do Control Center  na barra de ferramentas de qualquer ferramenta gráfica do DB2
- Clicando no ícone do DB2 presente na barra de tarefas do Windows, como mostrado na Figura 5.6 (clique com o botão direito no ícone verde do DB2 e selecione a opção DB2 Control Center)

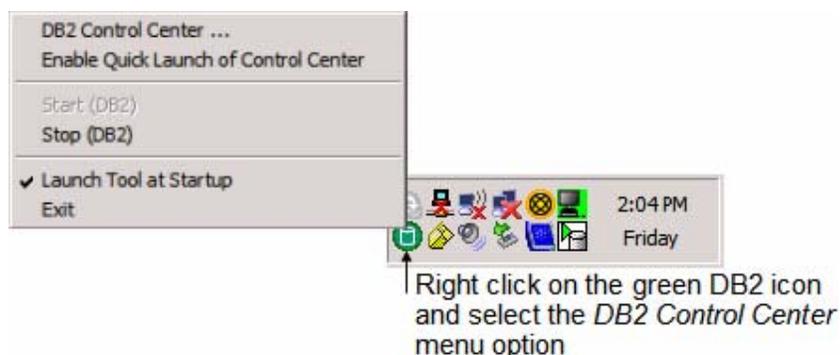


Figura 5.6 – Executando o DB2 Control Center a partir da barra de ferramentas do Windows

## 5.2 Command Editor

Usando o DB2 Command Editor, você pode executar comandos DB2, SQL e XQuery, analisar o plano de execução de uma instrução e ver ou atualizar os resultados de uma consulta.

A Figura 5.7 mostra o Command Editor com uma descrição de seus elementos.

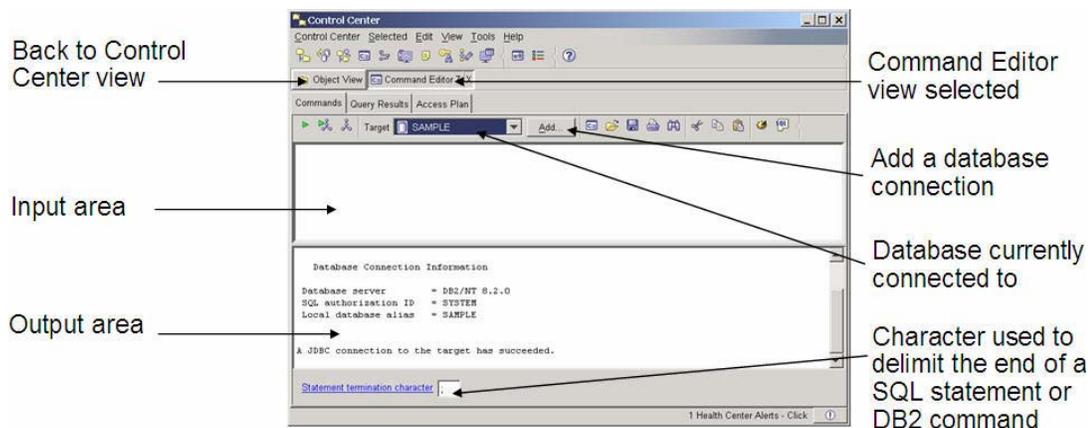
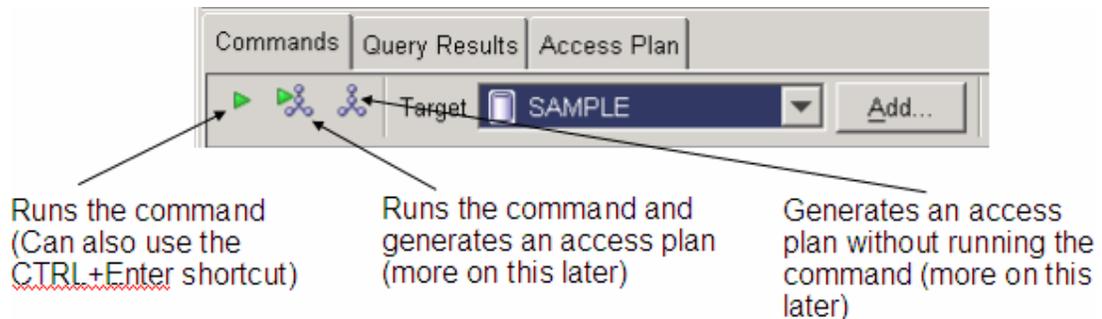


Figura 5.7 – Command Editor do DB2

Na *input area*, você pode inserir várias instruções, desde que cada uma delas seja finalizada com um caractere de terminação. Se você pressionar o botão de execução (veja Figura 5.8), as instruções serão executadas uma após outra. Se você selecionar explicitamente uma instrução em particular, somente a instrução selecionada será executada. Uma conexão com uma base de dados deve existir para que as instruções sejam executadas, contudo, uma das instruções pode ser uma instrução de conexão.



**Figura 5.8 – O Command Editor – aba Commands**

### Executando Command Editor

Você pode executar o Comand Editor de diversas formas:

- ▶ A partir do menu Iniciar do Windows:  
*Iniciar -> Programas -> IBM DB2 -> DB2COPY1 (Default) -> Command Line Tools -> Command Editor*
- ▶ Em um prompt de comandos, digite db2ce
- ▶ No menu Tools no Control Center
- ▶ Embutido no Control Center
  - Clique com o botão direito no ícone da base de dados SAMPLE da árvore de objetos do Control center e selecione o item *Query*
  - Sempre que um objeto que pode ser consultado é selecionado (base de dados, tabela, etc.) você pode executar o Command Editor, clicando no link *Query* na seção de detalhes de um objeto do Control Center.
- ▶ No Control Center, clique no ícone do Command Editor  na barra de ferramentas do Control Center, conforme mostra a Figura 5.9.



**Figura 5.9 – O ícone do Command Editor no Control Center**

### Adicionando uma conexão com uma base de dados

Para adicionar uma conexão com uma base de dados, clique no botão Add (veja Figura 5.7). Uma caixa de diálogo irá aparecer, conforme mostra a Figura 5.10.

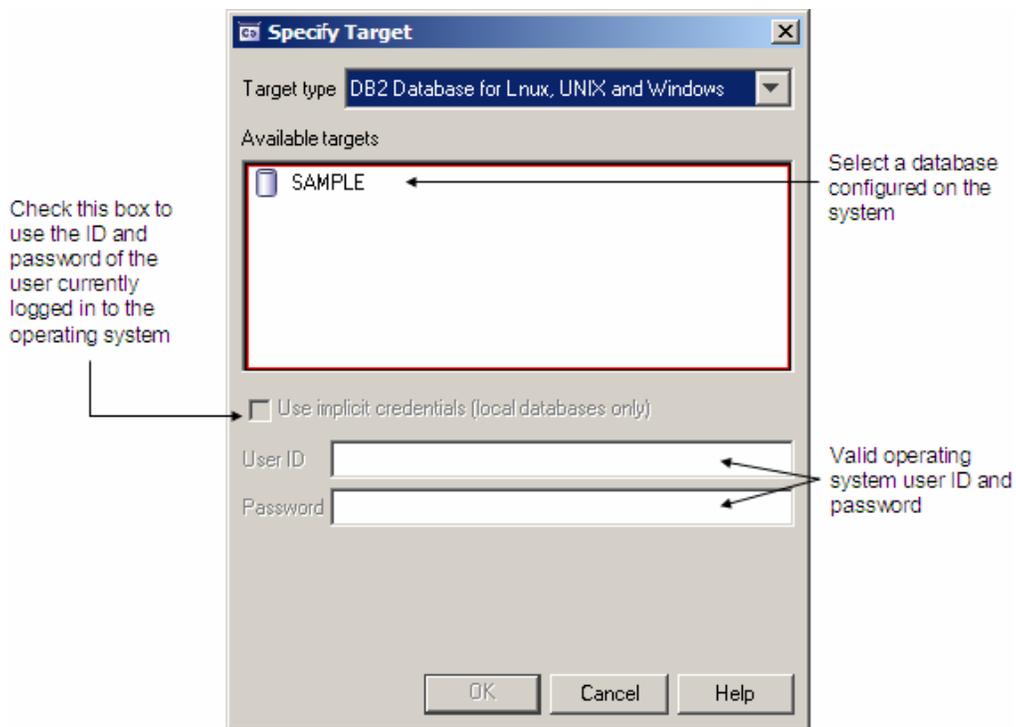
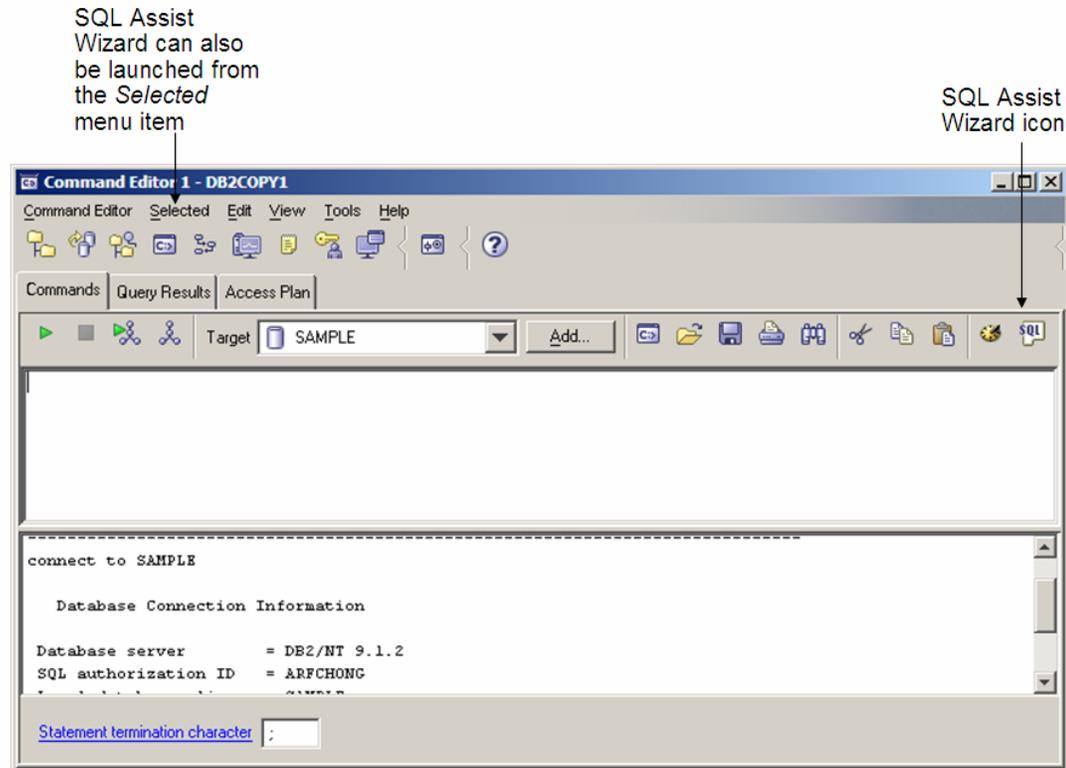


Figura 5.10 – Adicionar uma conexão com uma base de dados

### 5.3 SQL Assist Wizard

Se você não está familiarizado com a linguagem SQL e gostaria de usar um assistente ou wizard para gerar o código SQL, o SQL Assist Wizard está disponível no Command Editor para ajudá-lo. Como mostra a Figura 5.11, você invoca-o a partir do Command Editor, clicando no último item com símbolo SQL.



**Figura 5.11 – Invocando o SQL Assist Wizard**

A Figura 5.12 mostra o SQL Assist wizard. Ele é bem simples de usar. Primeiramente, indique o tipo de instrução SQL que você precisa de auxílio (SELECT, INSERT, UPDATE, DELETE). Dependendo de qual instrução você escolher, diferentes opções aparecerão. Na parte inferior da janela você verá como a instrução SQL é construída conforme você seleciona diferentes opções no wizard.

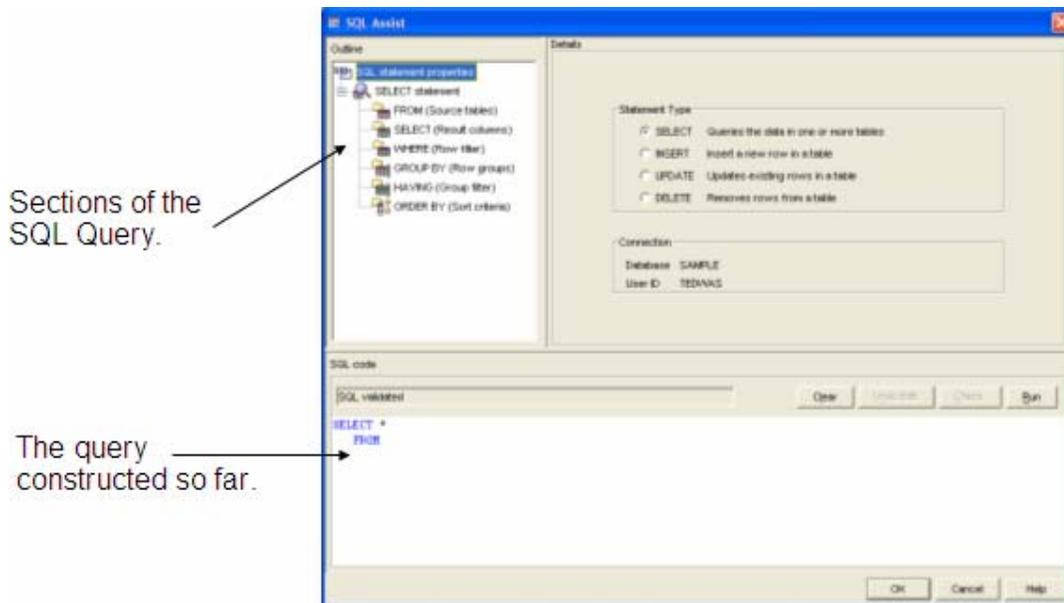


Figura 5.12 – O SQL Assist wizard

#### 5.4 Botão Show SQL

A maioria das ferramentas gráficas e wizards do DB2 permite que você revise o comando ou instrução SQL que é criado como um resultado do uso da ferramenta ou wizard para executar uma ação. Para ver isto, clique no botão Show SQL da ferramenta com a qual você está trabalhando, como mostram a Figura 5.13 e Figura 5.14.

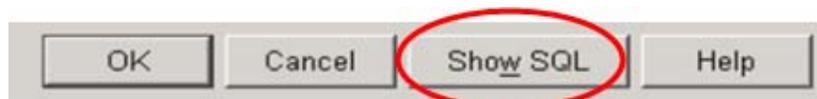


Figura 5.13 – O botão Show SQL

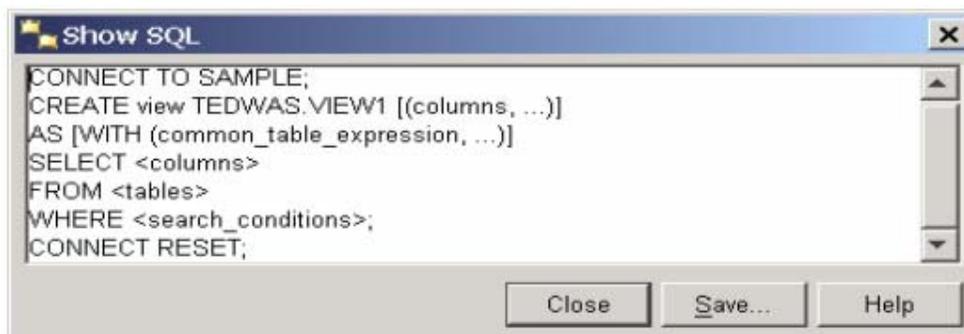


Figura 5.14 – Resultado do uso do botão Show SQL

O recurso de revisão de instruções SQL e comandos é bastante útil para o aprendizado da sintaxe SQL, e também para armazenar os comandos ou instruções em um arquivo para uso futuro. Além disso, você pode construir scripts, reusando os comandos e instruções gerados.

## Quicklab #4: Populando a base de dados EXPRESS usando scripts

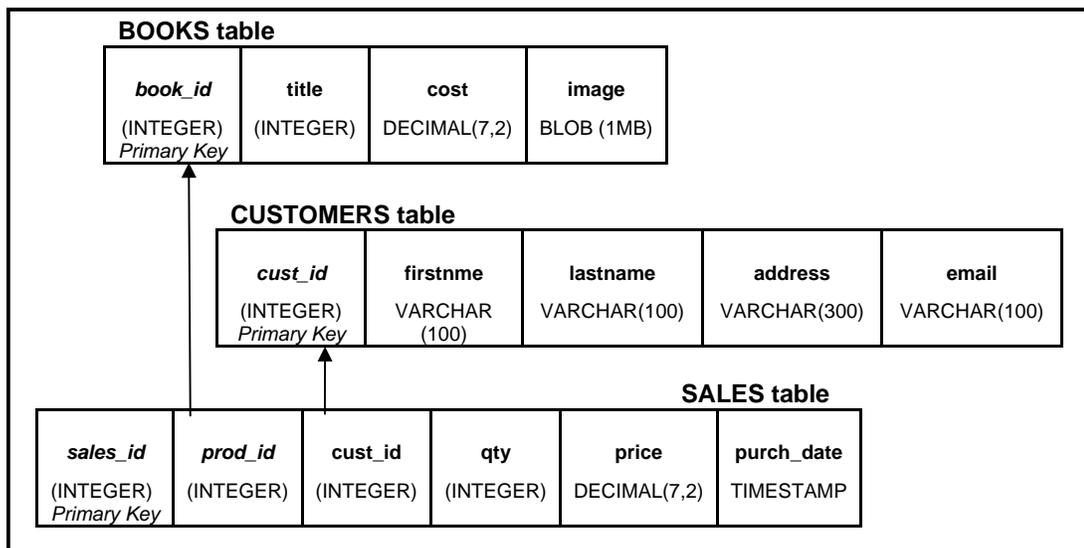
### Objetivo

Neste Quicklab, você irá popular a base de dados EXPRESS utilizando o Command Editor e dois scripts fornecidos.

### Procedimentos

1. No Quicklab #2 você criou a base de dados EXPRESS; agora você precisa populá-la com algumas tabelas e dados. Por conveniência, dois scripts, chamados `quicklab4.db2` e `quicklab4.dat` foram criados para fazer isto para você. O script `quicklab4.db2` contém os comandos usados para criar as tabelas e portanto deve ser executado primeiro. O script `quicklab4.dat` contém instruções que inserem dados dentro das tabelas. Ambos os scripts podem ser encontrados no arquivo zip chamado `quicklabs`, que acompanha este livro. Para executar estes scripts, abra o Command Editor. Garanta que a nova base de dados que você criou está selecionada na lista drop-down da barra de ferramentas. Se a nova base de dados não aparece na lista, adicione uma conexão para a base usando o botão *Add*.
2. Clique *Selected* → *Open* no menu do Command Editor e navegue até o diretório onde os scripts estão armazenados. Selecione o arquivo `quicklab4.db2` e clique no botão *OK*. O conteúdo do arquivo deve ser mostrado na área de texto editável do Command Editor. Clique no botão *Run* para executar o script. Verifique se nenhum erro foi encontrado durante a execução do script.
3. Repita o passo (2) para o arquivo `quicklab4.dat`.

A nova base de dados que você criou é de uma simples livraria na Internet. A tabela *BOOKS* contém todas as informações sobre os livros que a livraria possui. A tabela *CUSTOMERS* contém informações sobre cada cliente da loja. Finalmente, a tabela *SALES* contém dados sobre vendas. Quando um cliente compra um livro, um registro é armazenado na tabela *SALES*. O diagrama abaixo mostra o projeto e o relacionamento entre as tabelas.



## 5.5 Scripts

É sempre útil criar arquivos de script que executam vários comandos do DB2 ou instruções SQL repetidamente. Por exemplo, um administrador de bases de dados (*Database Administrator – DBA*) pode querer executar um determinado script todo dia para verificar o número de linhas de tabelas importantes.

Há duas formas gerais de scripts:

1. Scripts SQL
2. Scripts de sistema operacional (shell).

### 5.5.1 Scripts SQL

Scripts SQL incluem instruções de consulta e comandos de base de dados. Estes scripts são relativamente simples de entender e são independentes de plataforma. Contudo, variáveis ou parâmetros não são suportados.

Por exemplo, os seguintes comandos são salvos em um arquivo chamado `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;
```

#### Arquivo `script1.db2`

No script acima, todas as instruções são instruções SQL e cada uma delas está separada por um delimitador, neste caso, o ponto-e-vírgula. O nome do arquivo não precisa ter a extensão “db2”. Qualquer extensão pode ser utilizada.

#### Executando scripts SQL

Um script SQL pode ser executado tanto pelo Command Editor quanto pelo DB2 Command Window no Windows ou através de um shell no Linux. Para executar o script anterior a partir do DB2 Command Window ou de um shell no Linux, você pode usar o seguinte comando:

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

ou:

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

Neste comando:

- t indica que as instruções usam o caracter de terminação padrão (ponto-e-vírgula).
- v indica o modo verbose; fazendo com que o db2 to escreva na tela os comandos que estão sendo executados.
- f indica que o nome do arquivo especificado é um arquivo de script.
- z indica que a mensagem de saída de tela será escrita no final do arquivo cujo nome está especificado (isto é opcional, mas recomendado).

Quando a opção **-t** é usada e nenhum delimitador de linha é especificado, o ponto-e-vírgula é assumido como delimitador das instruções. Pode haver situações em que outro delimitador é necessário. Por exemplo, um script contendo código SQL PL necessita que um caractere de terminação de instruções diferente seja usado, porque o ponto-e-vírgula é usado entre as definições de objetos SQL PL para finalizar instruções procedimentais.

Por exemplo, no arquivo de script abaixo, chamado "functions.db2", o qual contém a Linguagem de Definição de Dados (DDL) necessária para criar uma função, o ponto-e-vírgula é necessário ao final da sintaxe do SELECT requerida na função. Para a instrução CREATE FUNCTION, estamos usando o ponto de exclamação como terminador. Se usássemos o ponto-e-vírgula como terminador de instruções, haveria conflito no tempo de execução no script, resultando em um erro reportado pelo DB2.

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

#### Arquivo functions.db2

Para informar ao DB2 que um caractere de terminação diferente está sendo usado, use a opção **-d**, seguida do caractere de terminação desejado, como mostrado abaixo:

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

A descrição de outras opções que podem ser usadas pode ser visualizada através do Command Window ou do shell do Linux, usando o comando:

```
db2 list command options
```

#### 5.5.2 Scripts de sistema operacional (shell)

Scripts de sistema operacional fornecem grande flexibilidade e poder, pois eles dão a possibilidade de incluir lógica de programação adicional. Eles são dependentes de plataforma, mas suportam parâmetros e variáveis. Abaixo é apresentado um exemplo de um script simples do sistema operacional Windows.

```
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 esquema.log triggers.log app_objects.log

db2 set esquema user1

db2 -t -v -f esquema.db2 -z esquema.log

db2 -td@ -v -f triggers.db2 -z triggers.log

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

#### Arquivo create\_database.bat

Para executar este script de sistema operacional a partir da linha de comando, você deve executar os seguintes comandos no Windows:

```
create_database.bat db2admin ibmdb2
```

No Windows, a extensão “bat” informa ao sistema operacional que o arquivo é executável em lote (batch).

No Linux, você precisa mudar o modo do arquivo para indicar que ele é executável, usando, por exemplo, o comando `chmod +x`. Posteriormente, você pode executá-lo da mesma maneira listada acima.

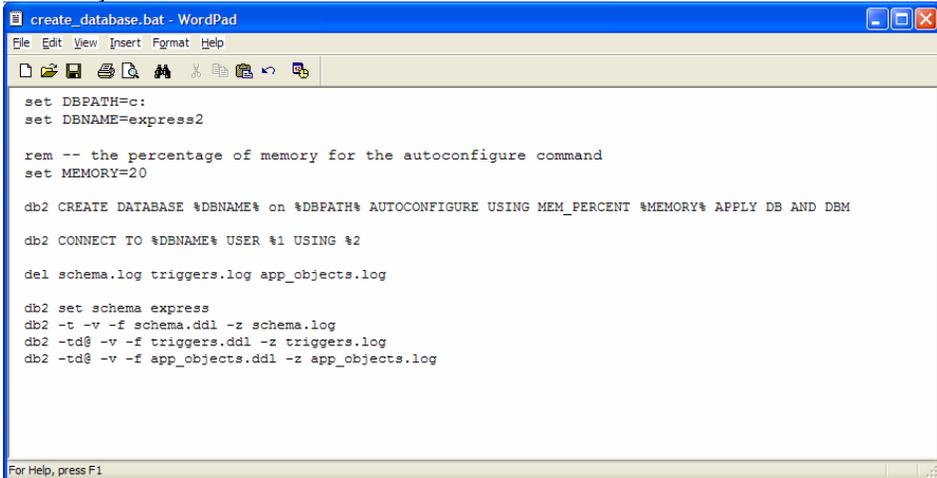
## Quicklab #5: Criando um script de instalação para a base de dados EXPRESS

### Objetivo

Scripts são um mecanismo poderoso para executar tarefas repetitivas como coletar estatísticas de uma base de dados, backups, e para distribuir uma base de dados nos clientes. Scripts de sistema operacional têm a vantagem de suportar parâmetros, sendo mais flexíveis. Neste quicklab, você criará um script de sistema operacional para fazer a entrega da base de dados *EXPRESS* como *EXPRESS2*. O script chamará scripts SQL anteriores para objetos da base de dados. Para economizar espaço, este quicklab mostra os scripts e comandos específicos da plataforma Windows. Se você preferir trabalhar no Linux, faça as modificações apropriadas nas instruções abaixo.

### Procedimento

1. Abra um editor de textos, como Notepad ou Wordpad e adicione as informações como mostrado abaixo.

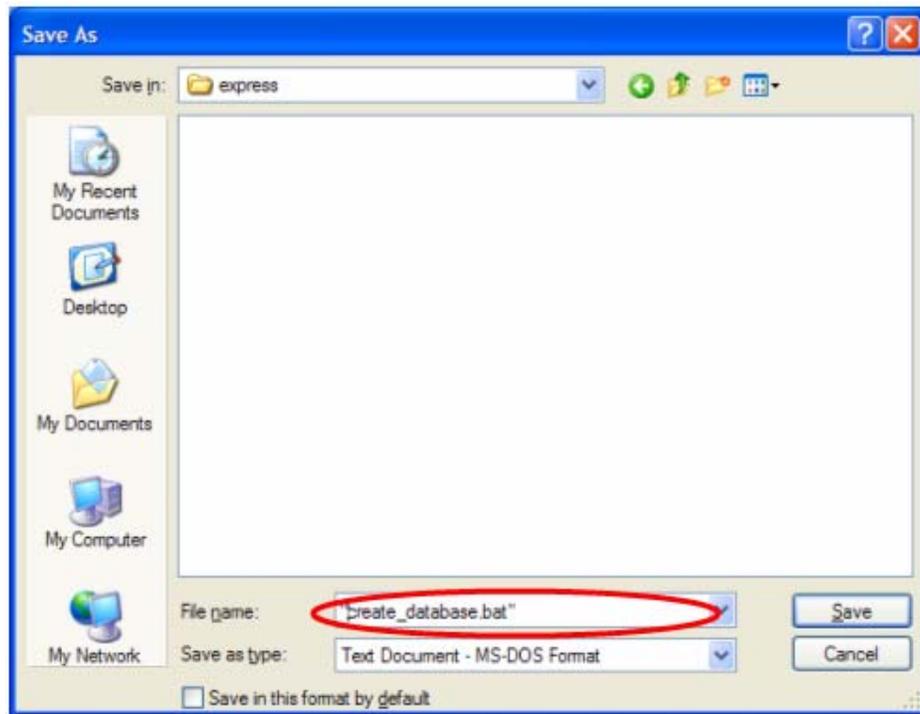


```
create_database.bat - WordPad
File Edit View Insert Format Help
set DBPATH=c:
set DBNAME=express2

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

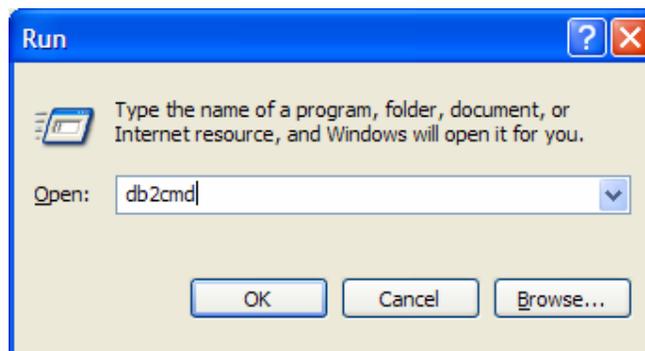
db2 CREATE DATABASE %DBNAME% on %DBPATH% AUTOCONFIGURE USING MEM_PERCENT %MEMORY% APPLY DB AND DBM
db2 CONNECT TO %DBNAME% USER %1 USING %2
del schema.log triggers.log app_objects.log
db2 set schema express
db2 -t -v -f schema.ddl -z schema.log
db2 -td@ -v -f triggers.ddl -z triggers.log
db2 -td@ -v -f app_objects.ddl -z app_objects.log
For Help, press F1
```

2. Salve o arquivo de scripts como `create_database.bat`. Na caixa de diálogo `Save As`, garanta que você escolheu a opção de formato do MS-DOS (no Wordpad). Se você salvar o arquivo em um formato diferente, o Wordpad pode introduzir caracteres invisíveis que podem causar problemas na execução do script. Além disso, coloque aspas no nome do arquivo para se certificar que o Windows não coloque uma extensão `.TXT`, como mostrado na figura abaixo.



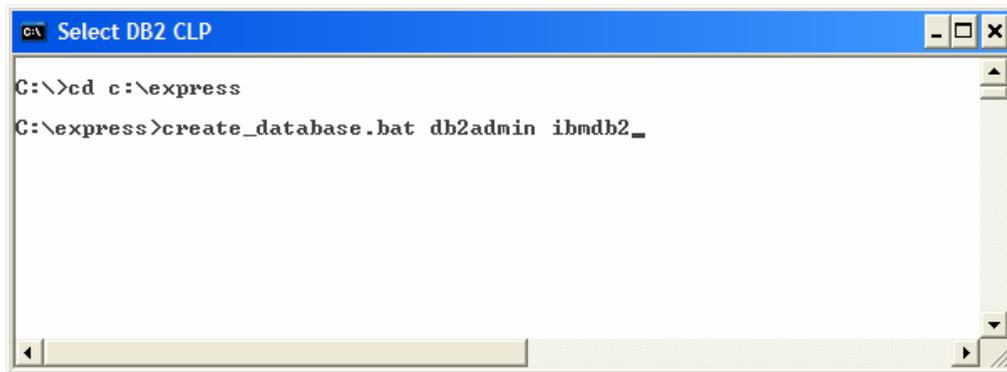
3. Para executar scripts que interajam com o DB2, você deve ter o ambiente de linha de comandos do DB2. Para abrir uma janela do DB2 Command Window, vá para *Iniciar > Arquivos de Programas > IBM DB2 > DB2COPY1 (default) > Command Line Tools > Command Window*.

Alternativamente, você pode usar *Iniciar > Executar*, digite `db2cmd` e pressione *enter* como mostrado abaixo.



4. Depois, para executar o script, digite os comandos:

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



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

5. Familiarize-se com o script criado. Você entende o que está acontecendo em cada linha?
6. Tente responder às seguintes questões:
  - a. Onde foi estabelecida a conexão com a base de dados?
  - b. O que significam o %1 e %2?
  - c. O que a seguinte linha de código faz? Onde é usada? Para quê?  
SET DBPATH=C:
  - d. O que a seguinte linha de código faz?  
del esquema.log, triggers.log, app\_objects.log
  - e. O que ocorre quando o script não é chamado com nenhum parâmetro?
  - f. Por que os scripts SQL são chamados dentro de instruções CONNECT TO? Como eles se conectam à base de dados?

## 5.6 Task Center

A ferramenta gráfica Task Center permite que você crie tarefas: um conjunto de operações, como executar comandos DB2, comandos de sistema operacional ou scripts. Ações subseqüentes podem ser desempenhadas se a tarefa falha ou é executada com sucesso. Por exemplo, se uma tarefa que envolve fazer backup de uma base de dados importante às 3:00 da madrugada, ocorre com sucesso, um e-mail pode ser enviado avisando o DBA. Mas, se o backup falhar, o Task Center pode enviar um pager ao DBA. A Figura 5.15 mostra o Task Center.

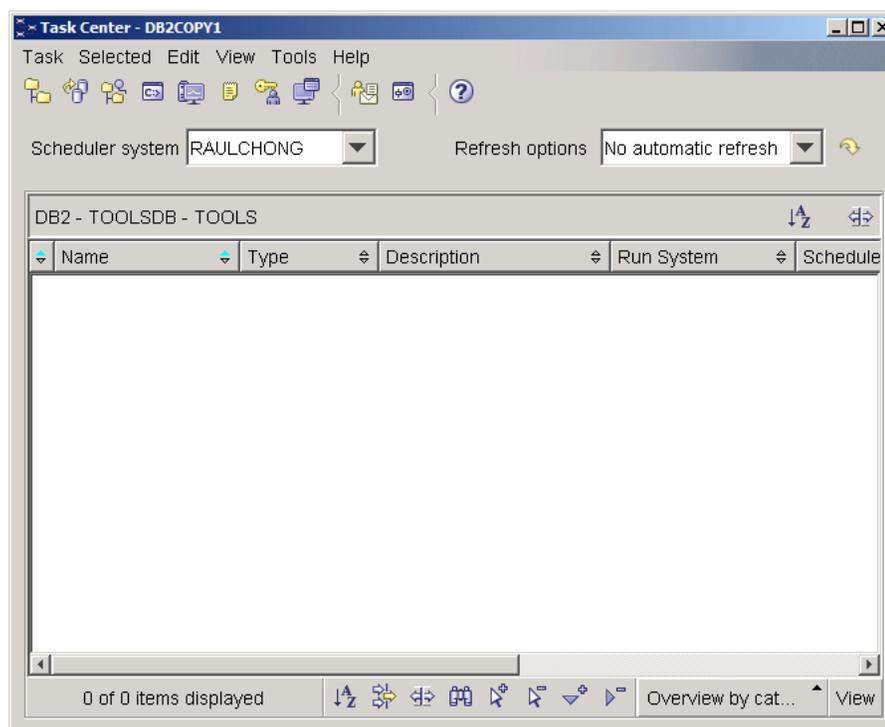


Figura 5.15 – O Task Center

### 5.6.1 Uma base de dados Tools Catalog

Todos os detalhes sobre suas tarefas são armazenados em uma base separada do DB2 chamada de Tools Catalog. Esta base de dados deve existir para que as tarefas possam ser agendadas. Para criar uma base de dados Tools Catalog, você pode usar o comando:

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

No exemplo acima, `systools` é o nome do esquema de todas as tabelas da base de dados, e o nome da base de dados é `toolsdb`. Iremos falar mais sobre esquemas no Capítulo 8, Trabalhando com objetos da base de dados.

### Executando o Task Center

Você pode executar o Task Center a partir do Control Center, clicando em *Tools > Task Center*, como mostrado na Figura 5.16. Alternativamente, você pode iniciar esta ferramenta a partir do menu Iniciar do Windows: *Iniciar > Programas > IBM DB2 > DB2COPY1 > General Administration Tools > Task Center*.

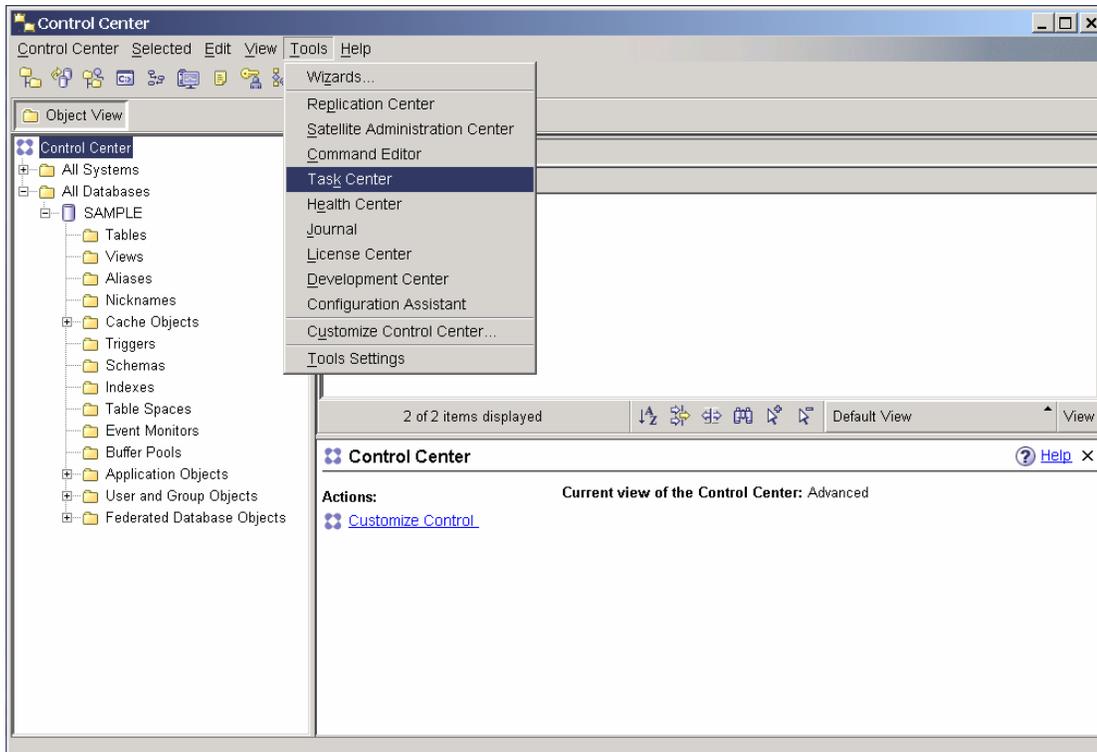


Figura 5.16 – Executando o Task Center

### Agendando tarefas com o Task Center

Qualquer tipo de script (criado ou não através de uma ferramenta gráfica do DB2) pode ser agendado usando o Task Center. Tarefas são executadas no sistema onde você criou a base de dados tools catalog e no horário que foram agendadas. Nós estimulamos que você explore o Task center. Criar uma tarefa é bem simples.

## 5.7 Journal

A ferramenta gráfica DB2 Journal oferece ao DBA um histórico de atividades. A Figura 5.17 mostra o Journal e a Tabela 5.2 mostra informações que você pode obter no Journal.

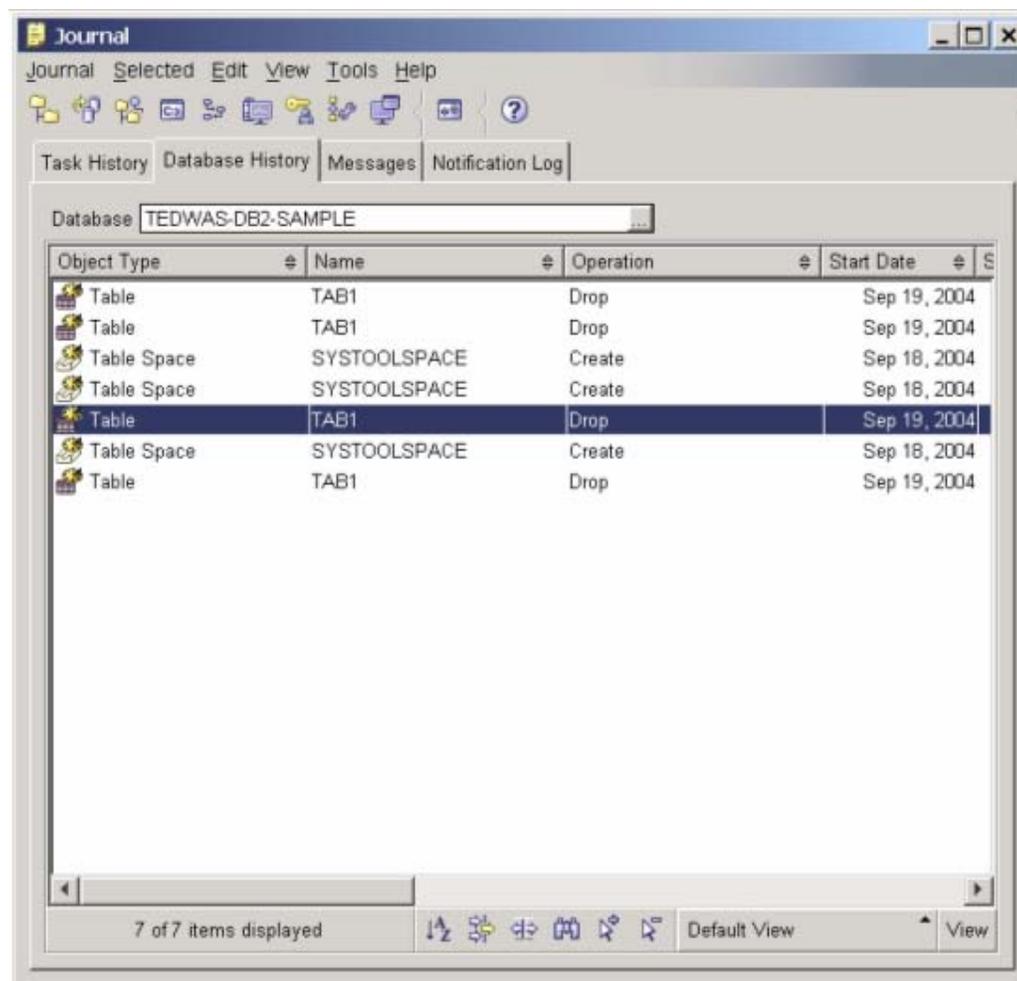


Figura 5.17 –O Journal

Tipo de Informação	Descrição
Histórico de tarefas	Todas as tarefas agendadas e seus status (sucesso ou falha).
Histórico da base de dados	Um registro de atividades desempenhadas em uma base de dados (backup, restore, REORGs, etc.)
Mensagens	Histórico de mensagens retornadas por ferramentas do DB2. Isto é útil se você deseja executar uma operação novamente e comparar mensagens de erro antigas, ou se você fechar uma janela de diálogo muito rápido ou por acidente.
Log de notificação	Contém mensagens em nível de sistema. Erros críticos são armazenados aqui.

Tabela 5.2 - Informação oferecida no Journal

### Executando o Journal

Você pode executar o Journal a partir do Control Center, clicando em *Tools > Journal*, como mostrado na Figura 5.18. Alternativamente, você pode iniciar esta ferramenta no menu Iniciar do Windows: *Iniciar > Programas > IBM DB2 > DB2COPY1 > General Administration Tools > Journal*.

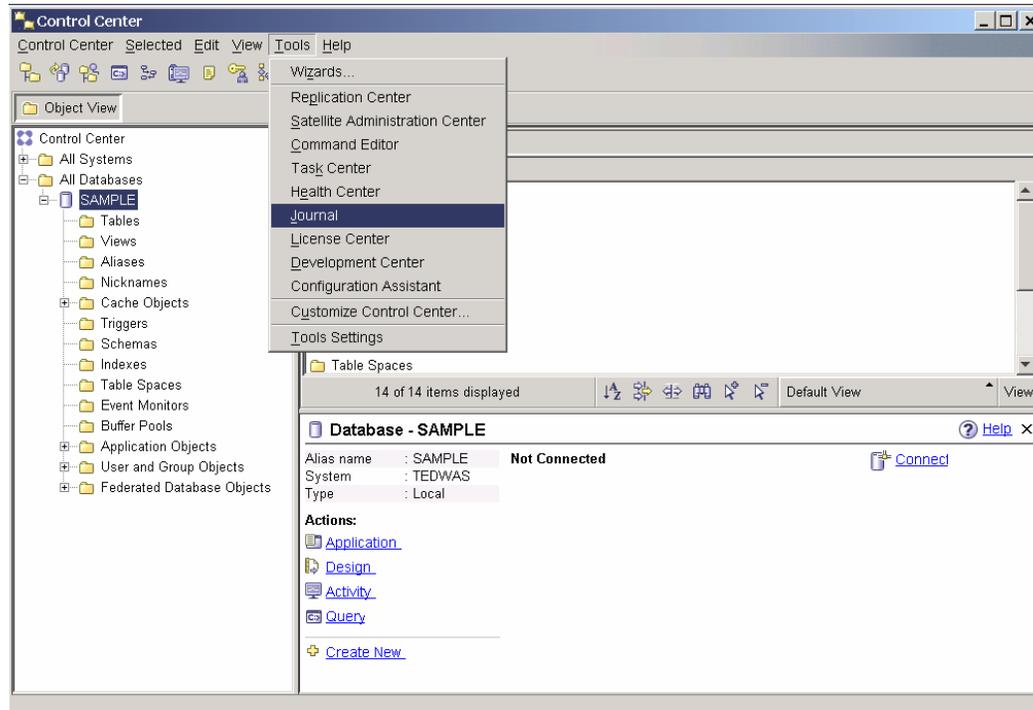


Figura 5.18 – Executando o Journal

## 5.8 Health Monitor

O Health Monitor é um agente padrão que executa dentro do DB2 Engine, monitorando todos os aspectos de operacionalidade de base de dados (memória, gerenciamento de espaço, atividades automatizadas previamente definidas, etc.). Quando algum aspecto do DB2 está operando fora dos parâmetros configurados, uma exceção é lançada e mostrada ao DBA. Existem três tipos de estados de alerta:

- ▶ Atenção: um estado não-normal.
- ▶ Warning: um estado não-crítico que não requer atenção imediata, mas pode indicar um sistema não-ótimo.
- ▶ Alarme: uma condição crítica, requerendo uma ação imediata.

O Health Monitor pode ser ativado ou desativado usando o parâmetro HEALTH\_MON do dbm cfg.

### 5.8.1 Health Center

O Health Center é uma ferramenta gráfica para interação com o Health Monitor. O Health Center exibe alertas de um sistema em nível de instância, base de dados e tablespaces. A Figura 5.19 mostra o Health Center.

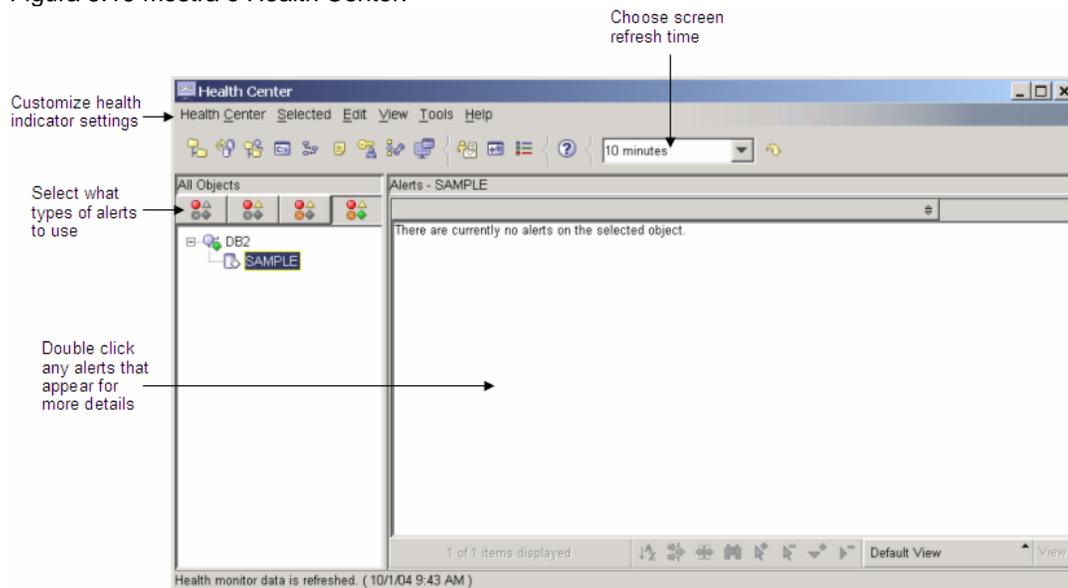


Figura 5.19 – O Health Center

#### Executando o Health Center

Você pode executar o Health Center a partir do Control Center, clicando no menu Tools e escolhendo a opção Health Center, como mostrado na Figura 5.20. Você também pode iniciar esta ferramenta em *Iniciar > Programas > IBM DB2 > DB2COPY1 >Monitoring Tools > Health Center*.

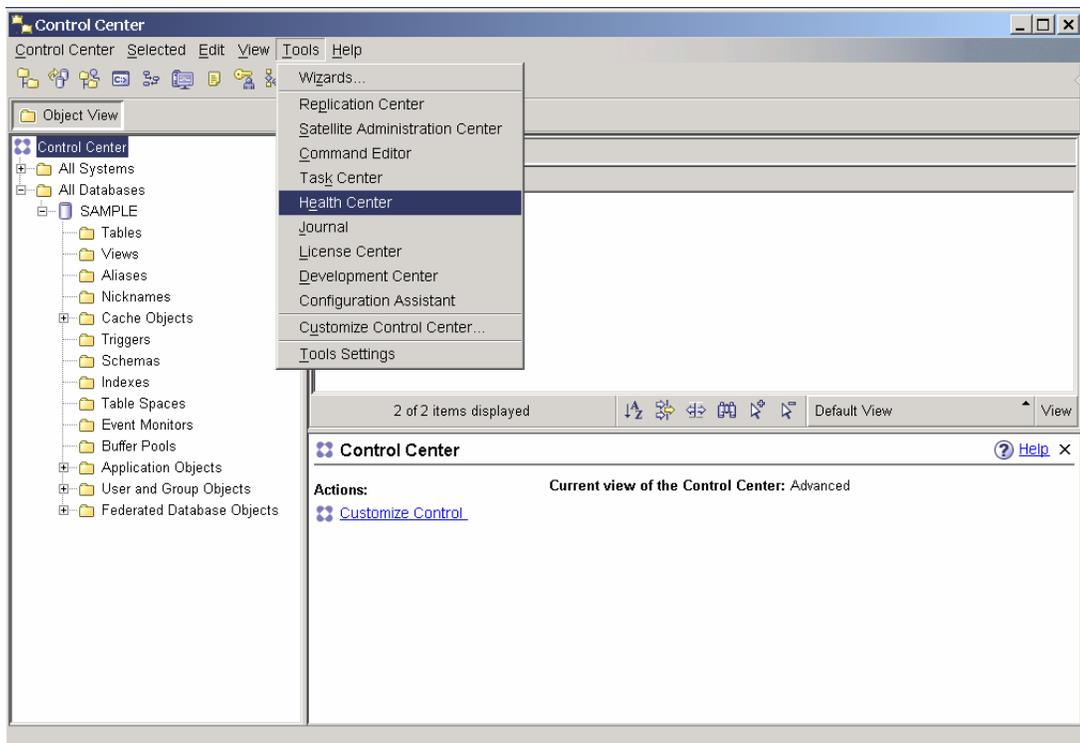
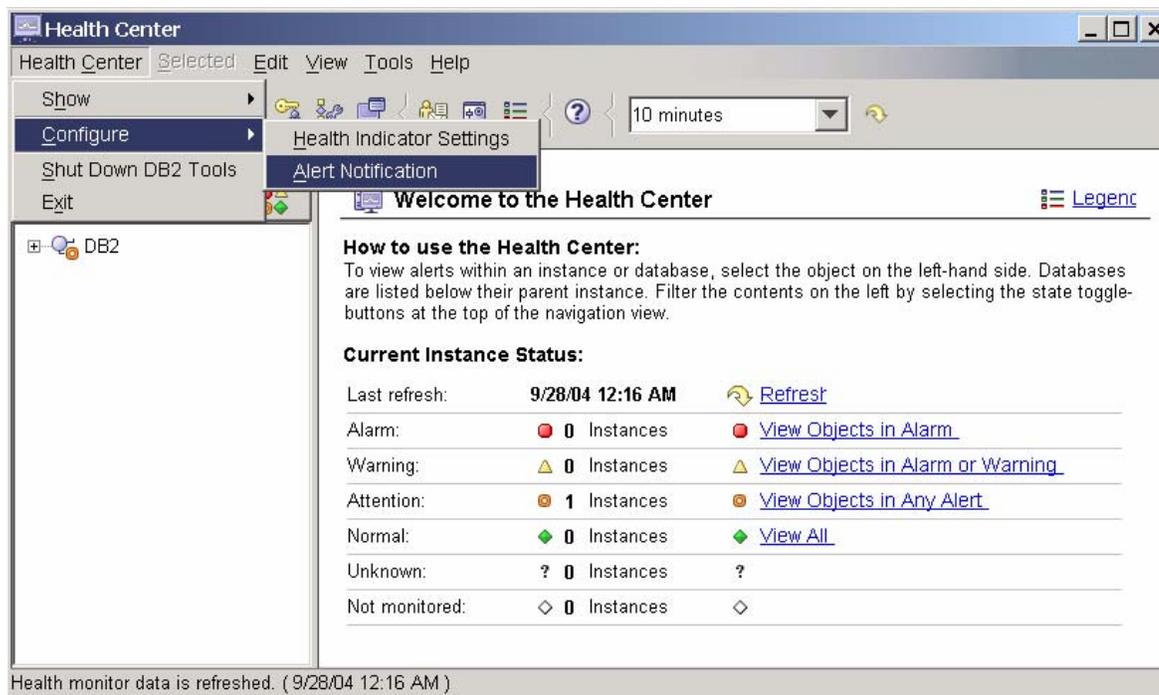


Figura 5.20 – Executando o Health Center

### Configurando Notificações do Health Alert

Com o Health Center iniciado, você pode configurar notificações de alerta clicando em *Health Center menu > Configure > Alert Notification*, como mostrado na Figura 5.21. Notificações de alerta permitem que você insira nomes de contato como o endereço de e-mail ou o número do pager de pessoas que serão contatadas caso um alerta ocorra.



**Figura 5.21 – Notificações de alerta**

**PARTE II – CONHECENDO O DB2:  
ADMINISTRAÇÃO DE BANCOS DE  
DADOS**



# 6

## Capítulo 6 – Arquitetura do DB2

Neste capítulo, discutiremos brevemente sobre a arquitetura do DB2.

- O Modelo de Processo do DB2
- O Modelo de Memória do DB2
- O Modelo de Armazenamento do DB2

**Nota:**

Para mais informações sobre a arquitetura DB2, assista ao vídeo abaixo:  
<http://www.channeldb2.com/video/video/show?id=807741:Video:4482>

### 6.1 Modelo de Processo do DB2

A Figura 6.1 retrata o Modelo de Processo do DB2. Nesta figura, os retângulos representam os processos enquanto os círculos representam as tarefas. O principal processo do DB2 é chamado db2sysc. Na seqüência desse processo existem várias tarefas, a principal delas também é chamada de db2sysc. Esta é a principal tarefa que gera outras tarefas. Quando uma aplicação remota tenta se conectar ao servidor utilizando uma instrução SQL CONNECT, os ouvintes remotos para o protocolo de comunicação receberão esta solicitação e farão contato com um agente coordenador DB2 (db2agent). O agente DB2 é como um pequeno trabalhador que executa operações em nome do DB2. Quando a aplicação for local, como esta, e estiver sendo executada em um mesmo servidor que o DB2, as etapas são muito semelhantes, porém um agente db2ipccm gerencia o pedido, em vez das tarefas db2tcpcm. Em alguns casos, como quando o paralelismo é ativado, o agente db2 pode gerar outros agentes que aparecem como tarefas db2agntp. Outros agentes mostrados na figura, como db2pfchr, db2loggr, db2dlock também podem ser utilizados para diferentes propósitos. Os processos mais comuns são descritos na Tabela 6.1 e as tarefas mais comuns são descritas na Tabela 6.2

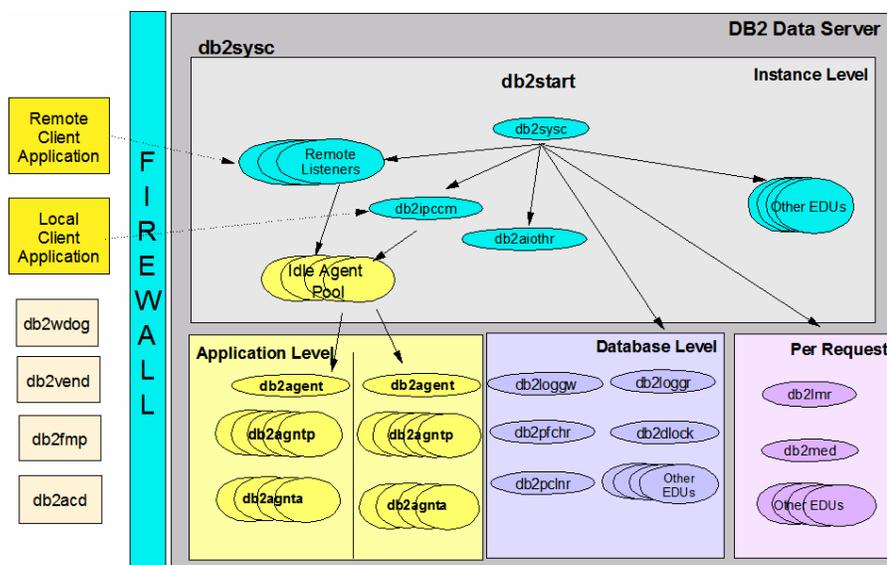


Figura 6.1 O Modelo de Processo DB2

Nome do Processo	Descrição
db2sysc (Linux) db2syscs (Win)	O principal sistema controlador ou engine do DB2. O DB2 9.5, tem apenas uma multitarefa principal no processo engine para toda a partição. Toda Engine Dispatchable Units (EDUs) são tarefas dentro deste processo. Sem este processo, o servidor da base de dados não funcionará.
db2acd	Computação autônoma daemon. É utilizada para executar tarefas automáticas para o cliente, tais como a monitoração, a manutenção automática dos utilitários e o cronograma do admin. Esse processo era chamado db2hmon.
db2wdog	O DB2 watchdog. O watchdog é o pai do principal processo da engine, db2sysc. Ele limpa os recursos se o processo db2sysc finalizar anormalmente.
db2vend	O fenced vendor process foi introduzido no DB2 9.5. Todo código não IBM é executado neste processo fora da engine. Este código de terceiros são programas não IBM que podem interagir com o DB2, como, por exemplo, o arquivo de log pode ser gerenciado por código não IBM através da especificação de um parâmetro de saída de uma rotina que aponte para este código.
db2fmp	Processos “fenced” que executam códigos do usuário, no servidor, fora do firewall para stored procedures e funções definidas pelo usuário. Este processo substitui tanto o processo db2udf quanto o db2dari que foram utilizados nas versões anteriores do DB2.

Tabela 6.1 – Processos DB2 Comuns

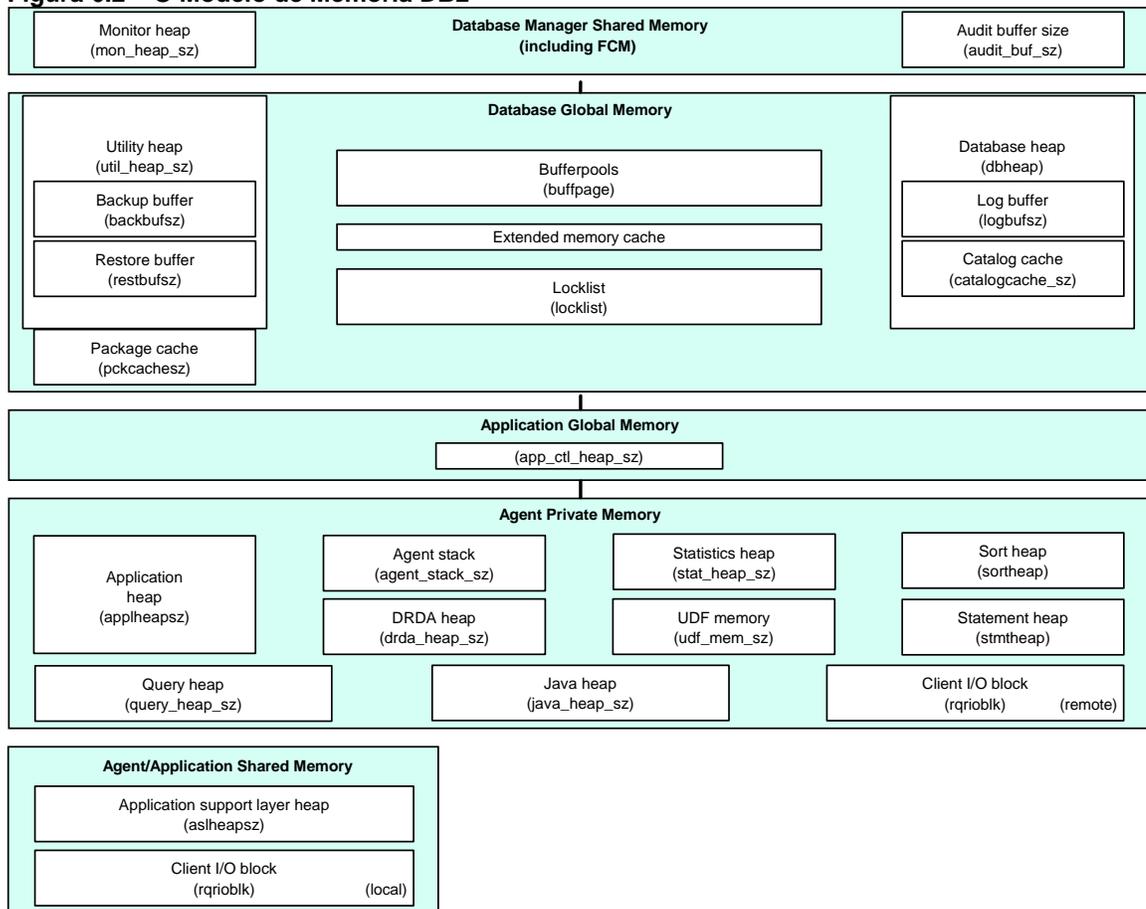
Nome das Tarefas	Descrição
db2sysc	Sistema controlador de tarefa. Esta tarefa é responsável pela inicialização, finalização e gerenciamento da instância.
db2tccpm	Tarefa listener de comunicação TCP/IP
db2agent	Agente coordenador que executa as operações da base de dados em nome das aplicações (pelo menos 1 por conexão, dependendo se o Connection Concentrator estiver ativado).
db2agntp	Ativa o sub agente gerado se INTRA_PARALLEL estiver definido como YES. Executará as operações da base de dados para a aplicação. O db2agent coordenará o trabalho entre os diferentes sub agentes db2agntp.
db2pfchr	DB2 asynchronous I/O data prefetcher (NUM_IOSERVERS)
db2pclnr	DB2 asynchronous I/O data writer (NUM_IOCLEANERS)

Tabela 6.2 –Tarefas DB2 Comuns

## 6.2 O modelo de Memória DB2

O modelo de memória do DB2 consiste em diferentes áreas na memória no nível de instância, de base de dados e da aplicação, e o nível do agente como mostrado na figura 6.2. Não explicaremos em detalhes cada uma das diferentes áreas da memória neste livro, mas somente proporcionaremos uma breve visualização.

Figura 6.2 – O Modelo de Memória DB2



Quando uma instância é iniciada, o gerenciador da base de dados compartilha a memória alocada. Isto normalmente não utiliza muito espaço. Quando você se conecta pela primeira vez na base de dados, a “Database Global Memory” é alocada. Neste bloco, o buffer pool é uma das partes mais importantes, especialmente por questões de performance. O tamanho do buffer pool determinará o tamanho total da “Database Global Memory”.

A “agent private memory” é a memória utilizada por cada agente DB2. Sem utilizar o connection concentrator, cada conexão requer um agente. Tipicamente um agente pode utilizar aproximadamente 3 a 5 MB. Com o connection concentrator, várias conexões podem utilizar um agente reduzindo, portanto, a necessidade de mais memória física.

### 6.3 O Modelo de Armazenamento DB2

Nesta seção descreveremos os seguintes conceitos:

- Páginas e Extents
- Buffer pool
- Tablespace

#### 6.3.1 Páginas e Extents

A página é a unidade mínima de armazenamento no DB2. Os tamanhos de página permitidos são: 4K, 8K, 16K e 32K. O extent é um agrupamento de páginas. Se o DB2 trabalhasse com uma página por vez o custo de performance seria muito alto, portanto, o DB2 trabalha com extents. O tamanho da página e do extent são definidos quando trabalhamos com buffer pools e tablespaces, como veremos nas próximas seções.

#### 6.3.2 Buffer pools

O buffer pool é uma memória cache para dados de tabelas e de índices. Ele melhora a performance reduzindo a direct sequencial I/O e promovendo leitura assíncrona (pre-fetching) e escrita. Isto é, o DB2 antecipa o que as páginas irão precisar, recuperando-as antecipadamente do disco para o buffer pool, deixando-as prontas para o uso.

Os buffer pools são alocados em páginas com unidades de memória de 4K, 8K, 16K, e 32K. Deve haver pelo menos um buffer pool por base de dados e pelo menos um buffer pool para uma tablespace de determinado tamanho da página.

#### Criando um Buffer Pool

Para criar um buffer pool você pode utilizar a opção `CREATE BUFFERPOOL`. Alternativamente, utilizando o Control Center, você pode clicar com o botão direito na pasta Buffer Pool dentro de uma base de dados e escolher a opção `Create` como mostrado na Figura 6.3

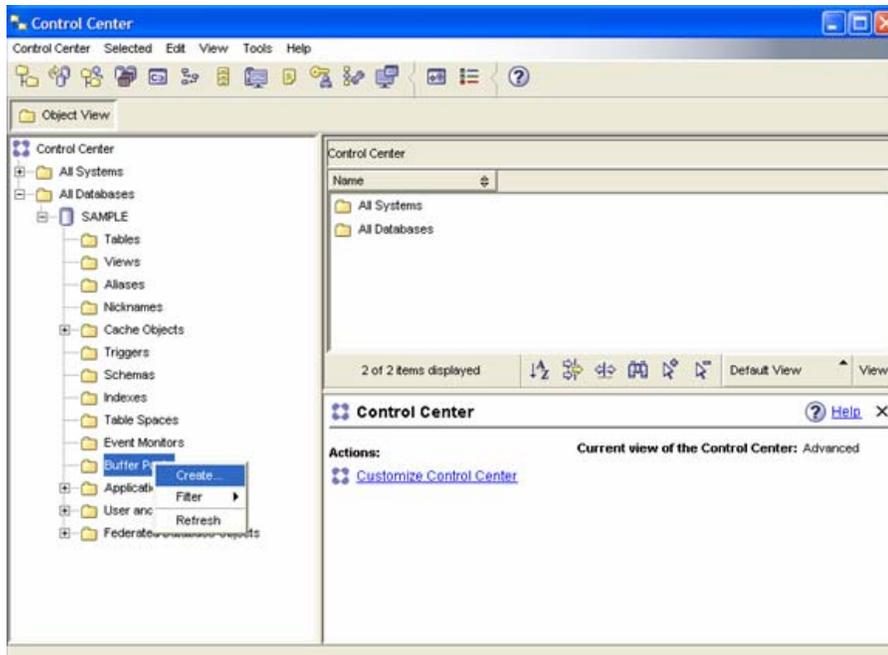


Figura 6.3 – Criando o Buffer Pool

Após clicar na opção `Create`, a caixa de diálogo `Create Buffer Pool` aparecerá como mostra a Figura 6.4

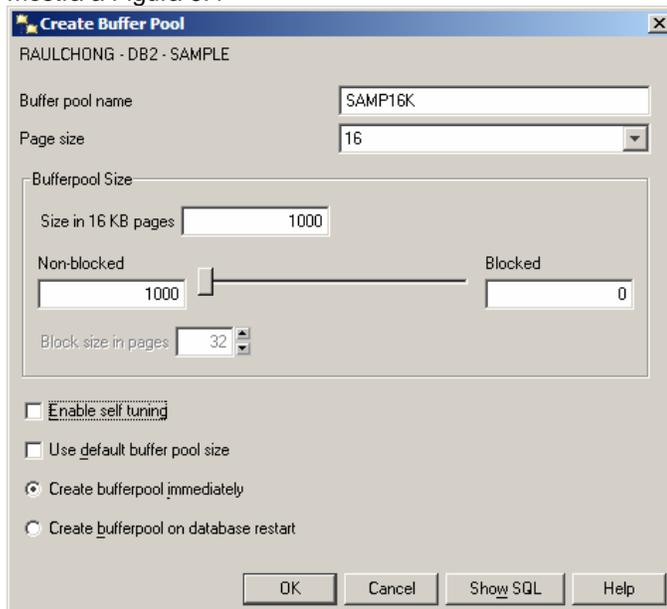


Figura 6.4 – Criando a caixa de diálogo do buffer pool

A maioria dos campos da Figura 6.4 são auto-explicativos. Os campos “Non-blocked” and “Blocked” referem-se ao número de páginas que deveriam existir como desbloqueados e bloqueados. Buffer pools bloqueados asseguram que páginas contínuas no disco são deslocadas para a área do buffer pool, também continuamente em uma área bloqueada. Isto pode melhorar a performance. O número de páginas não deve ser maior que 98% do número de páginas para o buffer pool. Especificando o valor 0, o block I/O é desativado.

Uma vez que o buffer pool tenha sido criado, este poderá ser mostrado no Control Center conforme mostra a Figura 6.5

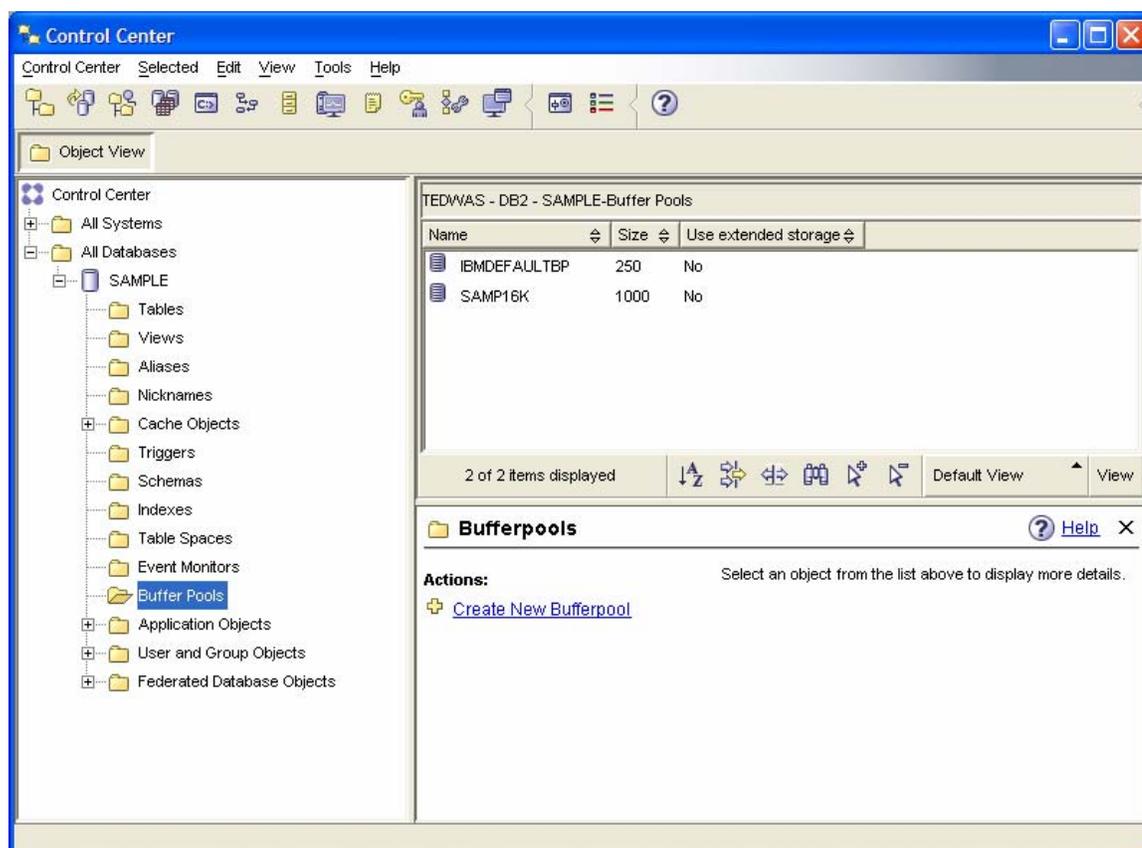


Figura 6.5 – O Control Center após a criação do buffer pool “SAMP16K”

### 6.3.2 Tablespaces

Tablespaces são interfaces lógicas entre as tabelas lógicas e a memória física do sistema (buffer pool) e os containers (discos). Utilize a instrução `CREATE TABLESPACE` para criar uma tablespace. Você pode especificar as seguintes opções:

- O tamanho de página para a tablespace (4KB, 8KB, 16KB ou 32KB). O tamanho de página da tablespace deve corresponder ao mesmo tamanho de página de um buffer pool.
- O buffer pool associado a esta tablespace.
- Um tamanho de extent
- Um tamanho de pre-fetch

### **Tipos de tablespaces**

Existem três tipos de tablespaces:

- Regular  
São utilizadas para tabelas de usuário. Por exemplo, a tablespace USERSPACE1 é uma tablespace regular.
- Grande (Large)  
São utilizadas, opcionalmente, para separar dados LOB em seus próprios tablespaces. Também é utilizada para armazenar dados XML em bases de dados criadas com o suporte para pureXML – quando a base de dados foi criada como UNICODE e utilizou o tipo de dados XML para as colunas. Este tipo de tablespace é default.
- Temporária:  
Existem dois tipos de tablespaces temporárias:
  - ▶ System temporary  
São utilizadas pelo DB2 para operações internas, como as ordenações. Por exemplo, a tablespace TEMSPACE1, criada quando se cria uma nova base de dados, é uma tablespace temporária do sistema.
  - ▶ User temporary  
São utilizadas para criar tabelas User Defined Global Temporary (tabelas temporárias em memória). São geralmente confundidas com tablespaces temporárias do sistema.

### **Gerenciamento de Tablespaces**

As tablespaces podem ser classificadas de acordo com a maneira como elas são gerenciadas. Esta opção pode ser especificada na instrução CREATE TABLESPACE.

#### **Gerenciadas pelo Sistema**

Este tipo de tablespace é conhecida System Managed Storage (SMS). Neste caso, o sistema operacional gerenciará o armazenamento. Elas são simples de gerenciar, e os repositórios são diretórios dos filesystems. O espaço não é pré-alocado, mas os arquivos crescem dinamicamente. Caso você especifique os repositórios, eles serão fixos em tempo de criação e outros repositórios não poderão ser adicionados posteriormente, ao menos que um redirected restore seja utilizado. Ao utilizar tablespaces SMS as tabelas de dados, índices e dados LOB não poderão ser distribuídos através de diferentes tablespaces.

#### **Gerenciadas pela base de dados**

Este tipo de tablespace é conhecida como Database Managed Storage (DMS). Neste caso, o DB2 gerenciará o armazenamento. O gerenciamento das tablespaces necessitará de mais intervenção manual de uma DBA. Os repositórios podem ser arquivos ou raw devices pré-alocados. Para raw devices, os dados são escritos diretamente sem O/S caching. Os repositórios podem ser adicionados, removidos ou reajustados. As tablespaces DMS são as melhores no ponto de vista de performance. Tabelas de dados, índices e dados LOB podem ser divididos em diferentes tablespaces, o que aumenta a performance.

#### **Gerenciadas pelo armazenamento automático**

Este tipo de tablespace, gerenciada pelo armazenamento automático, tem o mesmo benefício de fácil utilização, similar as tablespaces SMS, porém apresenta melhor performance e flexibilidade do que as tablespaces DMS. Portanto, ao iniciar com DB2 9, este é o tipo padrão de tablespace. Para estas tablespaces, o usuário especifica um grupo lógico de dispositivos de armazenamento. Nenhuma definição explícita de repositório é necessária. Os repositórios são automaticamente criados através dos caminhos de armazenamento. O crescimento de repositórios existentes e a criação de novos são completamente gerenciados pelo DB2.

Para utilizar o gerenciamento automático, você precisa criar a base de dados com a opção de gerenciamento automático habilitado (é a opção padrão) e associar um conjunto de caminhos de armazenamento a ele. Após a criação, se necessário, você pode redefinir os caminhos de armazenamento utilizando a operação de RESTORE da base de dados. Após isso você poderá criar tablespaces que utilizam o gerenciamento automático.

#### **Exemplo de armazenamento automático**

Primeiramente, crie a base de dados com o armazenamento automático habilitado, como nos exemplos abaixo:

O armazenamento automático é habilitado como opção padrão:

```
CREATE DATABASE DB1
```

O armazenamento automático é explicitamente especificado:

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

O armazenamento automático é habilitado como opção padrão, mas os caminhos de armazenamento são informados:

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

O armazenamento automático é explicitamente desabilitado:

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

Após isso, crie uma tablespace com o de armazenamento automático habilitado, como nos exemplos abaixo:

O armazenamento automático também pode ser habilitado como opção padrão na criação da tablespace:

```
CREATE TEMPORARY TABLESPACE TEMPTS
```

O armazenamento automático é explicitamente especificado na criação da tablespace:

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

O armazenamento automático é implicitamente especificado, quando o tamanho inicial é alocado e de acordo com o crescimento da tablespace, até que o tamanho máximo seja atingido.

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

**Como os dados são armazenados em tablespaces**

Por padrão, o DB2 escreverá nos extents de discos através dos repositórios. Por exemplo, se você tiver uma tablespace de 4KB com um extent de tamanho 8 e utilizar 3 repositórios em uma tablespace DMS, isto significa que 32KB de dados (4K x 8 páginas por extent = 32KB) serão escritos em um disco antes de escrever em outro. Isto é mostrado na Figura 6.6. Note que as tabelas não compartilham extents.

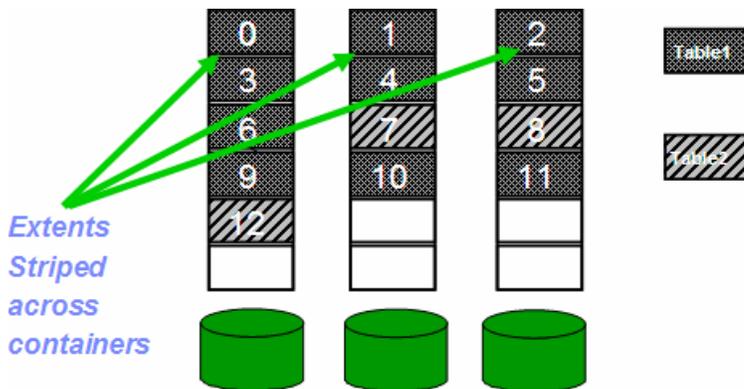


Figura 6.6 – Escrevendo dados em tablespaces

**Criando uma tablespace utilizando o Control Center**

Para criar uma tablespace através do Control Center, clique com o botão direito na pasta Tablespaces de uma determinada base de dados e selecione *Create*, como mostrado na Figura 6.7. O assistente “Create Tablespace Wizard” aparecerá, como mostrado na Figura 6.8.

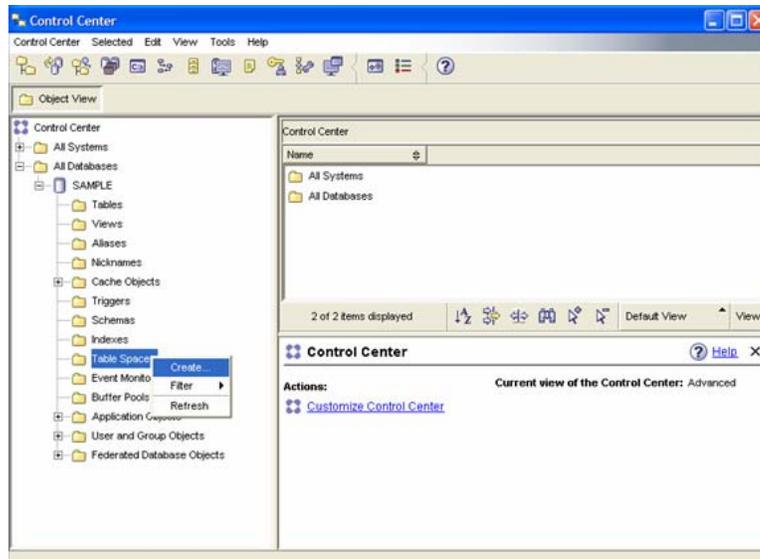


Figura 6.7 - Criando uma tablespace utilizando o Control Center



Figura 6.8 – O assistente ‘Create Tablespace’

# 7

## Capítulo 7 – Conectividade do DB2 Cliente

Neste capítulo discutiremos as configurações necessárias para realização da conexão de um cliente DB2 para um servidor DB2 utilizando TCPIP. O servidor DB2 possui um componente cliente, permitindo que ele se comporte como um cliente ao conectar em outro servidor DB2. Existem muitas maneiras de configurar a conectividade do cliente DB2, contudo discutiremos apenas a maneira mais simples, utilizando o *Configuration Assistant*.

**Nota:**

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

### 7.1 Diretórios do DB2

Os diretórios DB2 são arquivos binários que armazenam informações sobre cada base de dados que você pode conectar de sua máquina. Existem 4 diretórios:

#### 1. Diretório base de dados do sistema

Este diretório é similar a um índice de um livro. Ele mostra as bases de dados, remotas ou locais, que você pode se conectar. Para bases de dados locais existirá um ponteiro para o *diretório base de dados local*. Para bases de dados remotas, existirá um ponteiro para uma entrada no *diretório nó*. Para visualizar o conteúdo desse diretório execute o comando `list db directory`.

#### 2. Diretório base de dados local

Este diretório possui informações das bases de dados que você pode conectar e que estão em sua máquina. Para visualizar o conteúdo desse diretório execute o comando `list db directory on <drive/path>`.

#### 3. Diretório nó

Este diretório possui informações sobre como conectar em determinada base de dados. Por exemplo, se o protocolo TCP/IP for utilizado, uma entrada no nó TCP/IP incluiria o endereço IP do servidor onde a base de dados DB2, que você pretende conectar, e a porta da instância onde a base de dados está localizada. Para visualizar o conteúdo desse diretório execute o comando `list node directory`.

#### 4. Diretório DCS

Este diretório aparecerá somente se você tiver instalado o software DB2 Connect para conectar em DB2 em z/OS (mainframe) ou DB2 em i5/OS. Para visualizar o conteúdo desse diretório execute o comando `list dcs directory`.

A visualização e atualização do conteúdo desses diretórios podem ser realizadas através da ferramenta *Configuration Assistant*.

## 7.2 Configuration Assistant

Utilizando o *Configuration Assistant*, você pode facilmente configurar a conectividade entre um cliente DB2 e um servidor DB2.

Para carregar o *Configuration Assistant* no Windows, você pode selecionar Iniciar > Programas > IBM DB2 > DB2COPY1 > Set up Tools > Configuration Assistant.

Através da linha de comando, você pode carregar a ferramenta utilizando o comando `db2ca`. A Figura 7.1 mostra o Configuration Assistant.

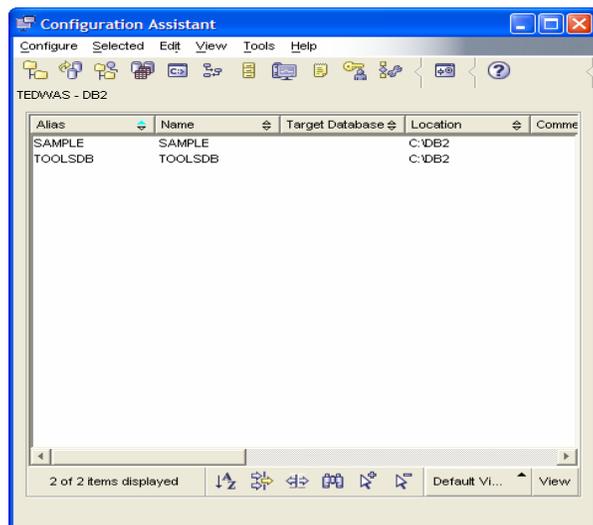


Figura 7.1 – O Configuration Assistant

### 7.2.1 Configurações necessárias no servidor

Existem duas coisas que você precisa configurar no servidor.

- 1) DB2COMM

A variável de registro determina quais *listeners* do protocolo de comunicação deveriam monitorar requisições de clientes. Usualmente TCP/IP é o protocolo de comunicação mais utilizado. Ao alterar este parâmetro, a reinicialização é necessária. Para visualizar e alterar o valor DB2COMM no *Configuration Assistant*, selecione *Configure > DB2 Registry* como mostrado nas figuras 7.2 e 7.3

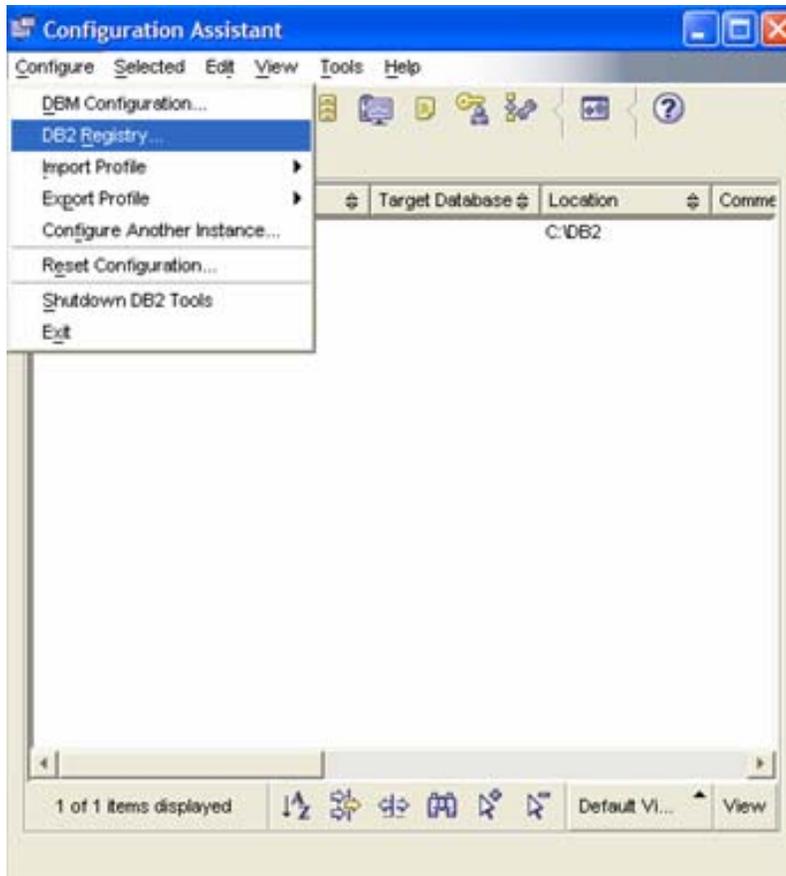


Figura 7.2 – Acessando o DB2 Registry

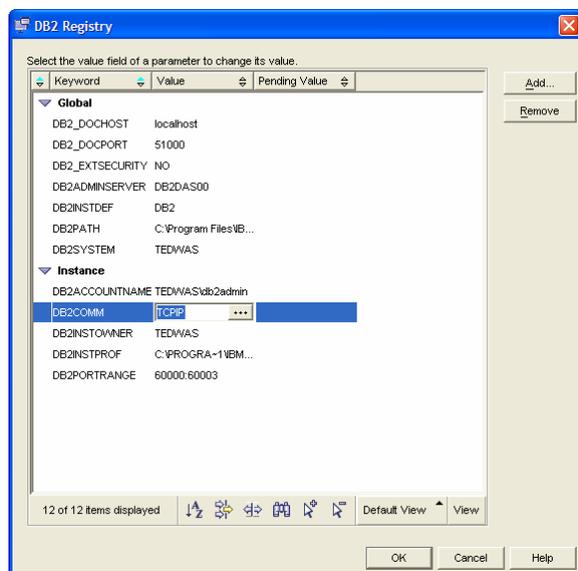


Figura 7.3 –Verificando o valor da variável DB2 Registry do DB2COMM

## 2) SVCENAME

Este parâmetro dbm cfg deveria ser definido para o nome do serviço (como definido no arquivo de serviços TCP/IP) ou número da porta para usar quando você quiser acessar bases de dados na instância. No *Configuration Assistant*, selecione *Configure > DBM configuration*, como mostrado na figura 7.4.

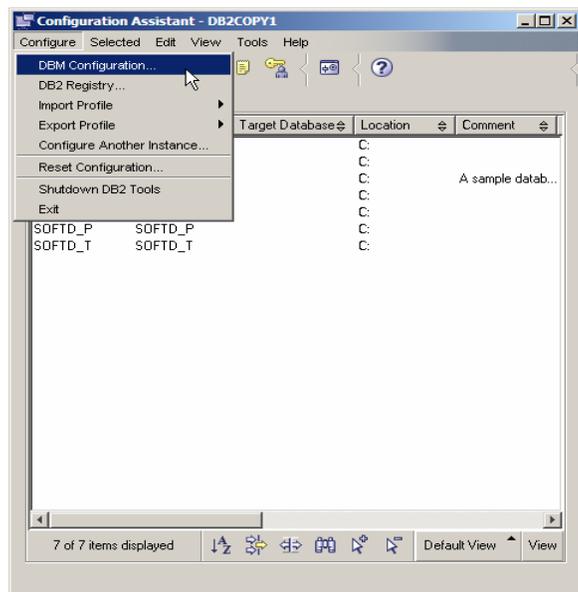


Figura 7.4 –Visualizando o dbm cfg no Configuration Assistant

Uma vez que você estiver na janela *DBM Configuration*, encontre a seção *Communications* e procure por *SVCENAME*. Você pode alterar o valor para um texto ou também para o número de uma porta, se necessário. Como mostrado na Figura 7.5.

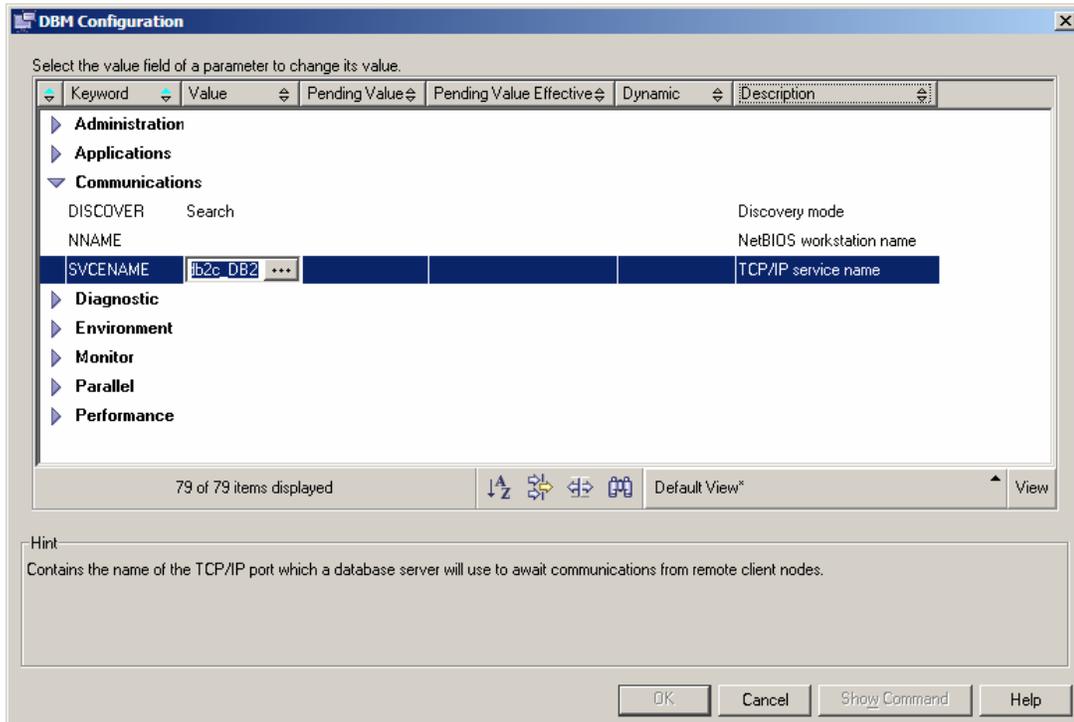


Figura 7.5 – Visualizando o parâmetro *SVCENAME* do *dbm cfg*

### 7.2.2 Configurações necessárias no cliente

No cliente você precisa primeiro saber as seguintes informações.

1. O nome da base de dados que você quer conectar.
2. O número da porta da instância DB2 no servidor onde a base de dados está instalada. Você pode utilizar um nome de serviço, desde que exista uma entrada no arquivo de serviços TCP/IP.
3. O ID de usuário do sistema operacional e senha para conectar na base de dados. Este ID de usuário deve ser configurado previamente no servidor.

A informação acima pode ser a entrada do cliente DB2 utilizando o *Configuration Assistant*. Inicialmente, carregue o assistente *Add database* selecionando *Selected -> Add Database Using Wizard*, como mostrado na Figura 7.6.

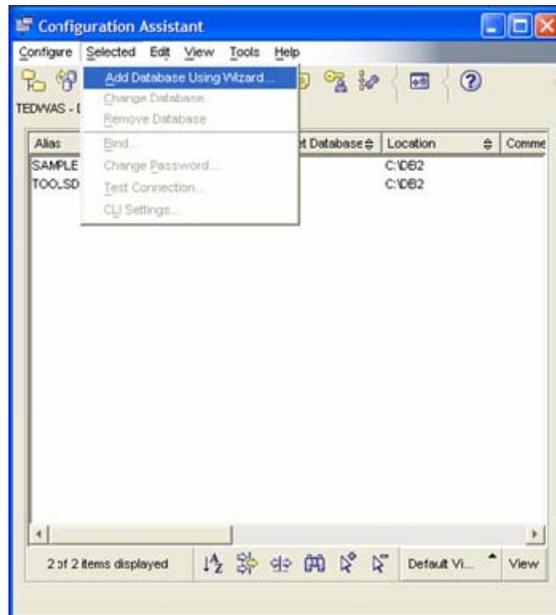


Figura 7.6 – Carregando o assistente *Add Database*

Você pode também acessar este assistente clicando com o botão direito do mouse, no espaço em branco no *Configuration Assistant*, e selecionando o assistente *Add Database*.

Figura 7.7 mostra o assistente *Add Database*.

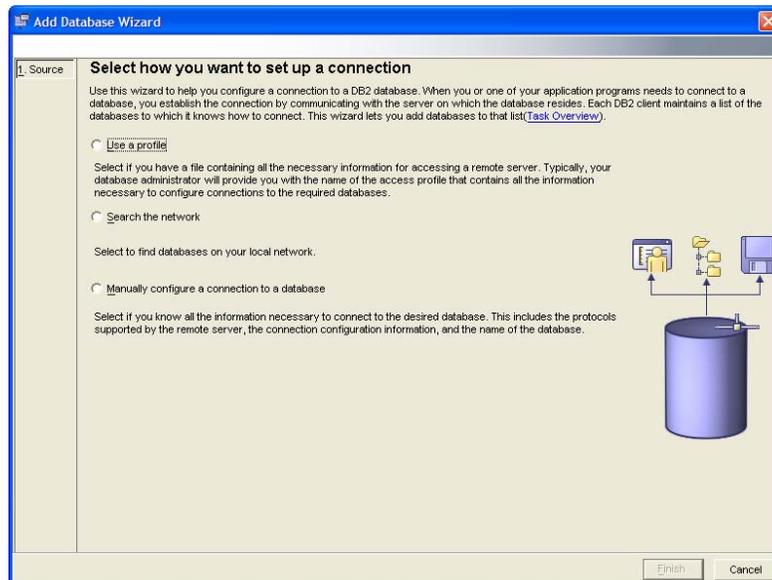


Figura 7.7 – Assistente *Add Database*

No assistente *Add Database* existem 3 opções.

### Utilize um perfil

Poderão existir situações onde você precisará configurar muitos clientes para conectar no mesmo servidor DB2. Nestas situações, é conveniente executar todas as configurações de um cliente, e armazenar estas configurações em um arquivo de perfil. Com este arquivo, você pode carregar todas as informações diretamente em outro cliente. Na Figura 7.7, se você pode selecionar “*Use a Profile*”, você carregará a informação de um perfil existente. Adiante neste capítulo serão mostrados mais detalhes, descrevendo como criar os perfis de cliente e servidor.

### Localizando na rede

Este método, também conhecido como “Descoberta”, diz ao DB2 para procurar na rede um determinado servidor, instância ou base de dados. Para este método funcionar, o DAS deve estar rodando em cada servidor DB2 onde as bases de dados estão para serem descobertas. Com este método, existem 2 caminhos para executar a pesquisa.

- Search: Procura na rede toda. Isto não é recomendado se a rede é grande e com muitos *hubs*, por que isto demoraria muito tempo para recuperar dados de cada sistema.
- Known: Procura na rede por um servidor conhecido pelo endereço que você informou.

Os dois métodos são ilustrados na Figura 7.8.

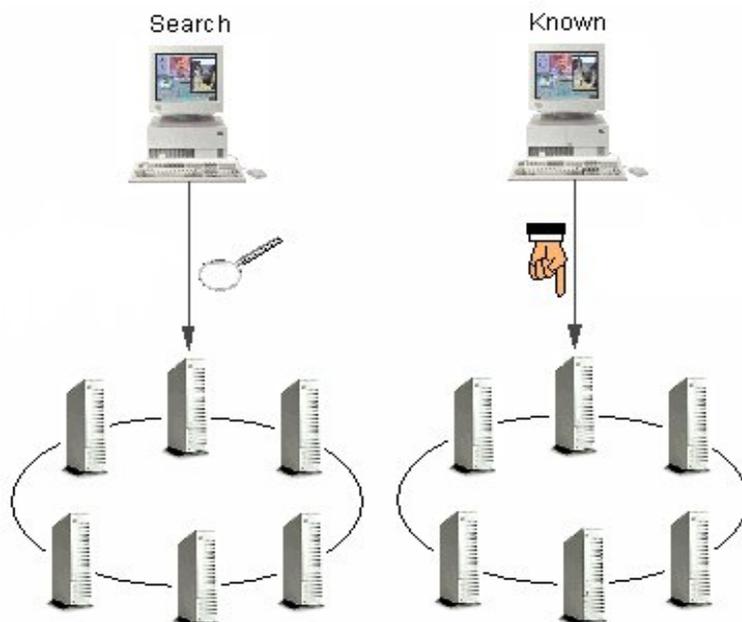
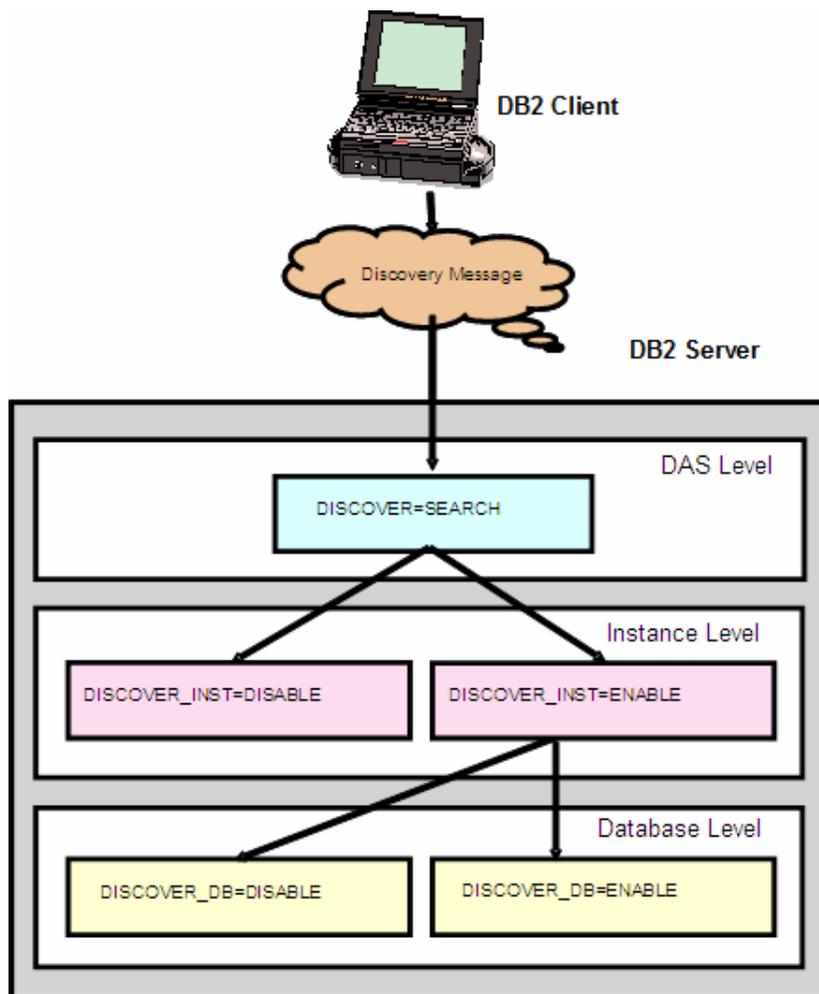


Figura 7.8 Os métodos de procura Search e Known (ou Discovery)

Existirão circunstâncias onde um administrador não gostaria que clientes procurassem por bases de dados com informações confidenciais na rede. Isto pode ser evitado com o DAS, no nível de instância ou base de dados. A Figura 7.9 informa detalhes sobre isto.



**Figura 7.9 – Configurando parâmetros para permitir o *discovery***

A Figura 7.9 mostra os diferentes níveis em que você pode habilitar ou desabilitar o *discovery*. No nível DAS, você pode informar um valor *SEARCH* ou *KNOWN* para o parâmetro *DISCOVERY*. No nível de instância o parâmetro *DISCOVER\_INST* do *dbm cfg*, pode ser definido para Ativado ou Desativado. Finalmente, no nível da base de dados, o parâmetro *DISCOVER\_DB* pode também ser definido como Ativado ou Desativado. Definindo esses parâmetros, conseqüentemente disponibiliza para você a granularidade para a descoberta de bases de dados.

### **Configurando manualmente uma conexão para a base de dados**

Utilizando este método você manualmente adiciona o *hostname*, números de porta e informações de bases de dados no *Configuration Assistant*, o qual criará comandos de catálogo para executar a configuração de conectividade. O *Configuration Assistant* não verificará se essa informação está correta, você saberá que é incorreto, se você não conseguir conectar no servidor. Também garanta que o ID de usuário e a senha que você informou para se conectar na base de dados remota estão corretos. Por padrão, a autenticação será executada no servidor DB2 que você está tentando se conectar, portanto, você deverá informar o ID de usuário e a senha definida naquele servidor.

#### **7.2.3 Criando perfis de cliente e de servidor**

Se você está configurando um grande número de clientes ou servidores, ao invés de definir cada um individualmente, você pode definir um e exportar um perfil (i.e. arquivo de configuração) dele, e então aplicar este perfil para outros clientes/servidores. Isto economiza um grande tempo ao configurar o ambiente.

Para criar um perfil personalizado no *Configuration Assistant*, clique no menu *Configure* então selecione *Export Profile => Customize*, como mostrado na Figura 7.10.

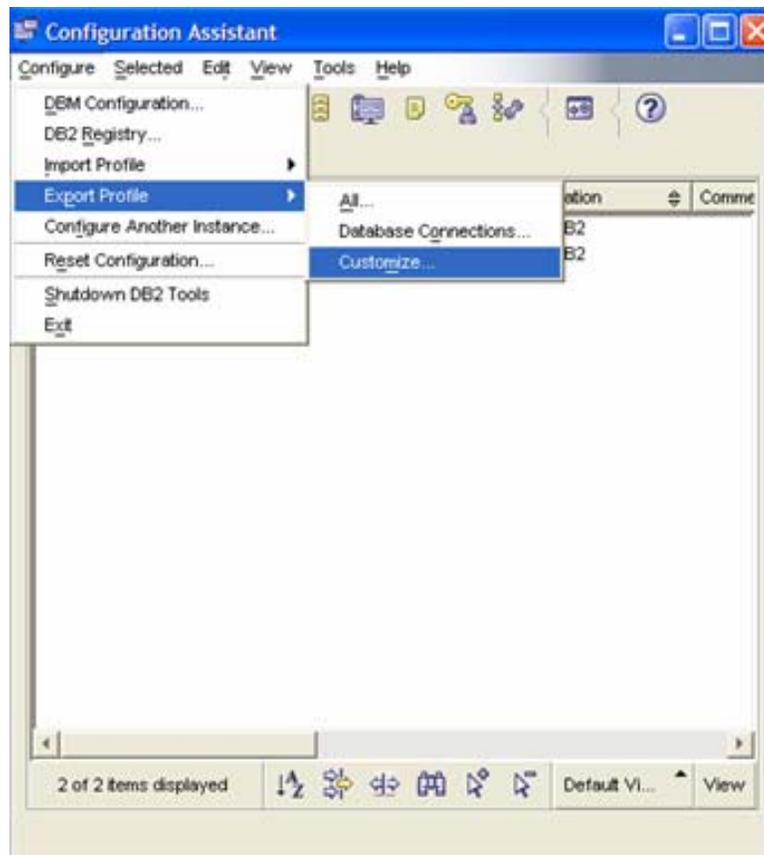


Figura 7.10 – Exportando um perfil

Figura 7.11 mostra os campos que precisam ser preenchidos para exportar o perfil.

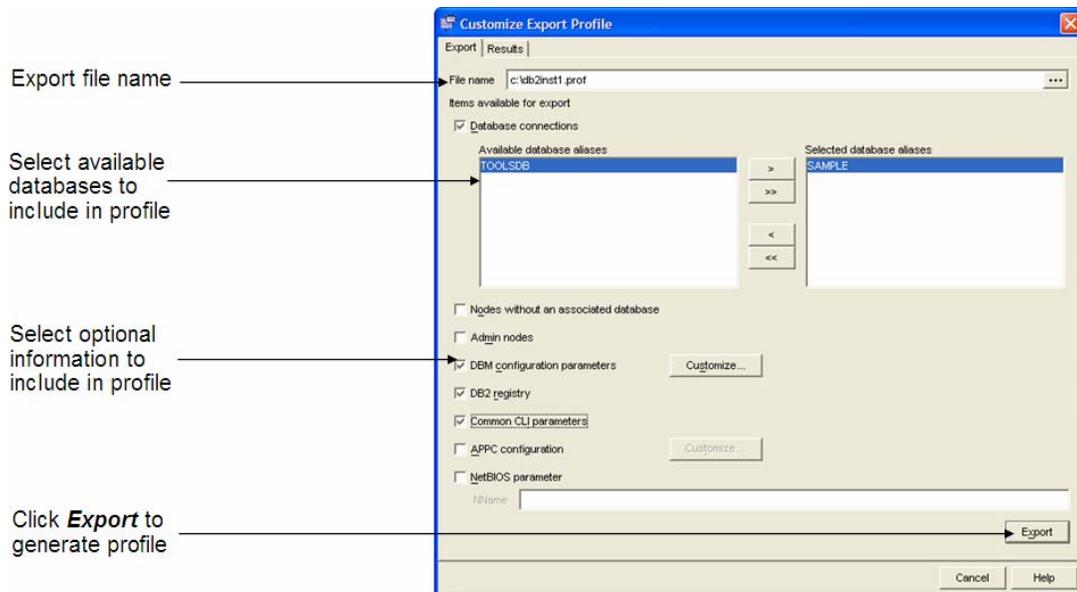


Figura 7.11 – Janela de diálogo *Customize Export Profile*

Figura 7.12 mostra o resultado após clicar em *Export*, na janela de diálogo *Customize Export Profile*.

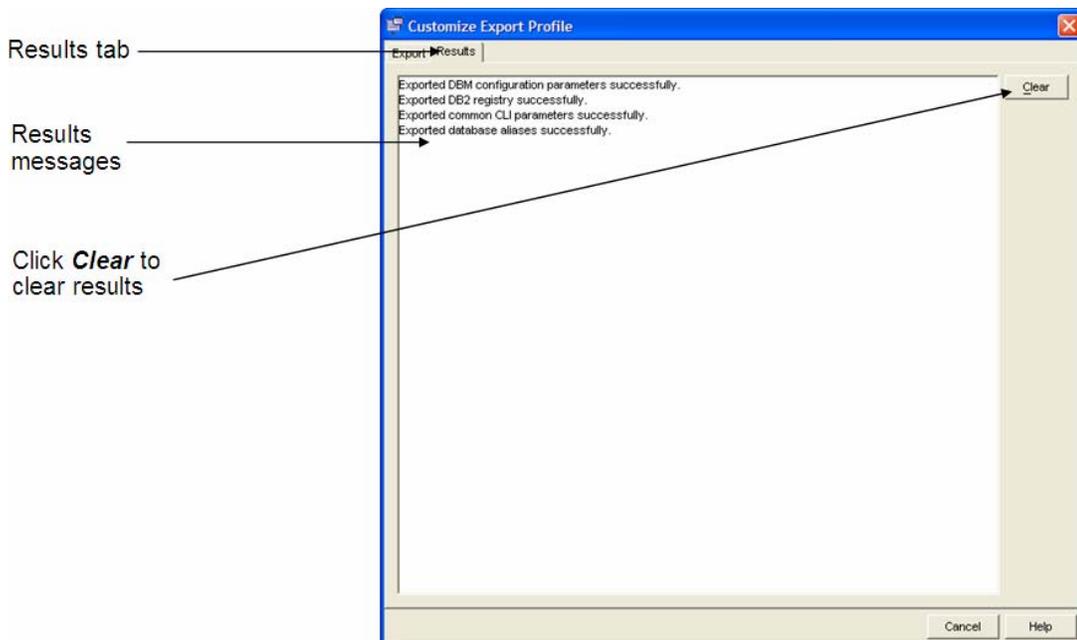


Figura 7.12 – Resultados da exportação de perfis

Para importar um perfil personalizado através do *Configuration Assistant*, clique no menu *Configure*, então selecione *Import Profile => Customize*, como mostrado na figura 7.13.

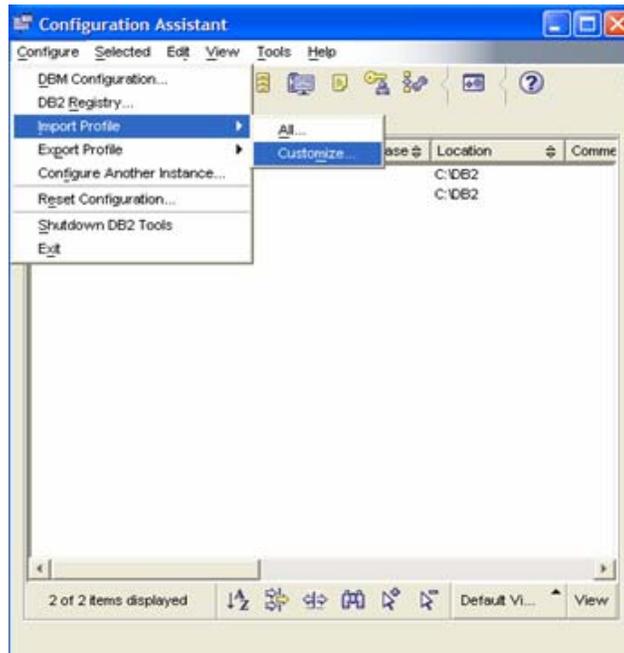


Figura 7.13 – Importando um perfil

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

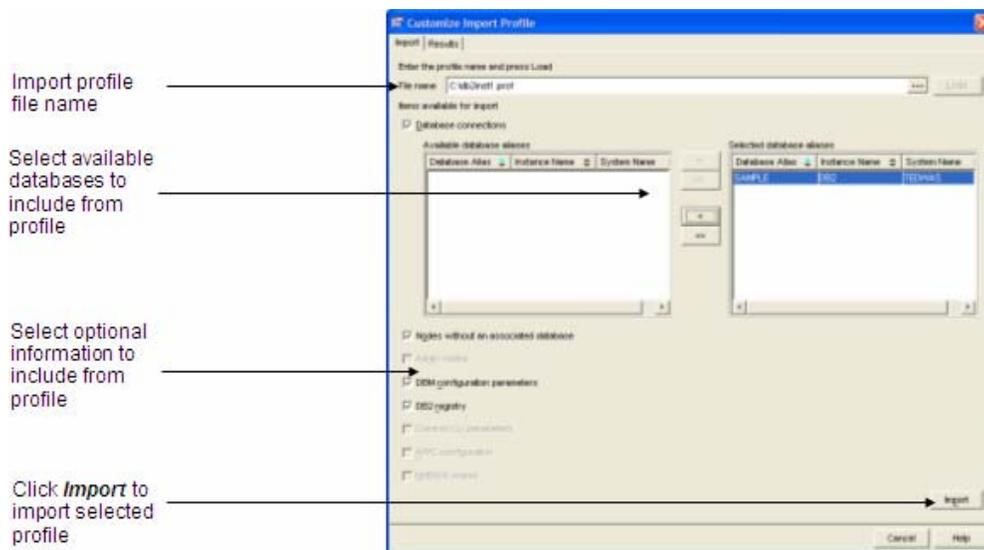


Figura 7.14 – Importando um perfil personalizado



5. Na página *Protocol* do assistente, selecione a opção TCP/IP. Clique no botão *Next* para avançar para a próxima página do assistente.
6. Na página TCP/IP do assistente, informe no hostname completo ou o endereço IP que você descobriu no passo 1. Informe o número da porta do passo 1. Clique no botão *Next* para avançar para a próxima página do assistente.

Nota: A opção para *Service Name* pode ser utilizada se você tiver uma entrada no arquivo *Local Services* com um número de porta definido, o qual corresponde ao número da porta em que o servidor remoto estiver escutando. Quando você utilizar esta opção, DB2 procurará no arquivo *Services* na máquina local e não no servidor. Você deve adicionar uma entrada neste arquivo, se você desejar utilizar esta opção.

7. Na página *Database* do assistente, informe o nome da base de dados definida no servidor remoto, que você descobriu no passo 1, no campo *Database Name*. Note que o campo *Database Alias* é automaticamente preenchido com o mesmo valor. O *Alias* da base de dados é o nome que as aplicações locais utilizarão para conectar nesta base de dados. Caso você já tenha uma base de dados local chamada *SAMPLE*, o DB2 não deixará você catalogar outra base de dados com o mesmo nome. Você deve usar um *Alias* diferente. Por exemplo, altere o *Alias* da base de dados para *SAMPLE1*. Se desejar, informe um comentário opcional sobre esta base de dados. Clique no botão *Next* para avançar para a próxima página do assistente.
8. Na página *Data Source* do assistente, você pode opcionalmente registrar a nova base de dados (fonte de dados) com uma fonte de dados ODBC. Isso automaticamente o registra para você, no *Windows ODBC Manager*. Neste exemplo, desmarque o *Register this database for ODBC*, já que você não utilizará ODBC. Clique no botão *Next* para avançar para a próxima página do assistente.
9. Na página *Node Options* do assistente, especifique o sistema operacional do servidor da base de dados remota. Considerando que todas as estações de trabalho neste laboratório utilizam Microsoft Windows, tenha certeza que o item *Windows* da lista está selecionado. O campo *Instance Name* deverá ser definido para DB2, caso ele não esteja definido. Clique no botão *Next* para avançar para a próxima página do assistente.
10. A página *System Options* do assistente dá a oportunidade de garantir que o nome do sistema e o hostname estão corretos, e verificar se o sistema operacional foi selecionado. Clique no botão *Next* para avançar para a próxima página do assistente.

11. A página *Security Options* do assistente permite que você especifique aonde você deseja executar a autenticação do usuário e qual método você deseja utilizar. Selecione a opção *Use authentication value in server's DBM Configuration*. Isto fará com que o método utilize o parâmetro `AUTHENTICATION` no arquivo de configuração da instância remota. Clique no botão *Finish* para catalogar a base de dados remota e finalizar o assistente. Uma caixa de confirmação deverá aparecer. Clique no botão *Test Connection* para garantir que você poderá conectar com sucesso na base de dados. Também verifique se o nome de usuário e senha que você informou foram definidos no servidor remoto (desde que o parâmetro `AUTHENTICATION` no servidor esteja definido para `SERVER` ). Se o teste de conexão for realizado com sucesso, você terá catalogado corretamente a base de dados remota. Se isso não acontecer, revise os valores informados nos passos anteriores do assistente (clique no botão *Change*).
12. Abra o *Control Center* e tente acessar as tabelas da nova base de dados remota catalogada.
13. Volte ao assistente e tente catalogar uma base de dados diferente deste, utilizando a opção *Search the Network*. Siga os passos que você utilizou para configurar a conexão manualmente. Note que em redes grandes, a pesquisa deverá levar um tempo muito grande para retornar resultados.



# 8

## Capítulo 8 – Trabalhando com objetos da base de dados

Neste capítulo estudaremos os objetos da base de dados, esquemas, tabelas, visões, índices, sequences, etc. Alguns objetos de base de dados mais avançados como triggers, funções definidas pelo usuário (UDFs) e stored procedures serão discutidos no capítulo 14, SQL PL stored procedures, e capítulo 15, Inline SQL PL, UDF's, triggers.

**Nota:**

Para mais informações sobre como trabalhar com objetos da base de dados, assista ao vídeo:

<http://www.channeldb2.com/video/video/show?id=807741:Video:4242>

### 8.1 Esquema

Esquemas são “*name spaces*” para uma coleção de objetos da base de dados. Eles são usados para:

- Indicar o responsável por um objeto ou relacionamento com uma aplicação
- Agrupar os objetos relacionados juntos

Todos os objetos da base de dados DB2 possuem um nome qualificado dividido em duas partes, sendo o esquema a primeira parte dele:

<esquema\_name>.<object\_name>

O nome qualificado do objeto deve ser único. Ao conectar em uma base de dados e criar ou referenciar um objeto sem especificar o esquema, o DB2 assumirá o ID do usuário conectado na base de dados como nome do esquema. Por exemplo, se você conectar na base de dados SAMPLE com o usuário 'arfchong' e criar uma tabela utilizando a instrução CREATE TABLE:

```
CREATE TABLE artists
```

O nome qualificado da tabela será arfchong.artists.

## 8.2 Tabelas

Uma tabela é uma coleção de dados relacionados, organizados logicamente em colunas e linhas. A instrução abaixo mostra um exemplo de como criar a tabela utilizando 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
```

Nas seções abaixo, descreveremos as partes principais da instrução CREATE TABLE.

### 8.2.1 Tipos de dados

A Figura 8.1 lista os tipos de dados suportados pelo DB2

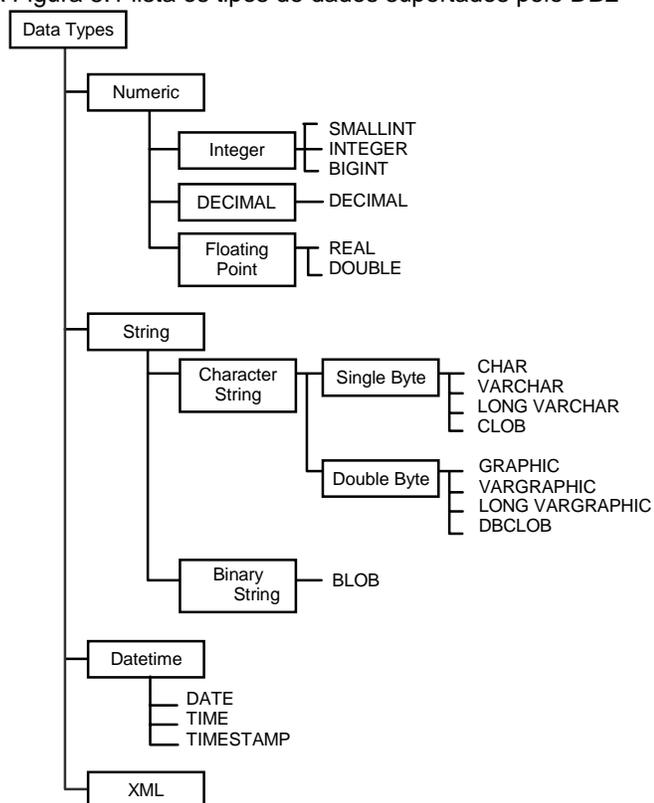


Figura 8.1 – Tipos de dados DB2

### Tipos de dados Large Object (LOB)

Tipos de dados Large Objects são usados para armazenar textos grandes, textos binários grandes ou arquivos, como mostrado na Figura 8.2.

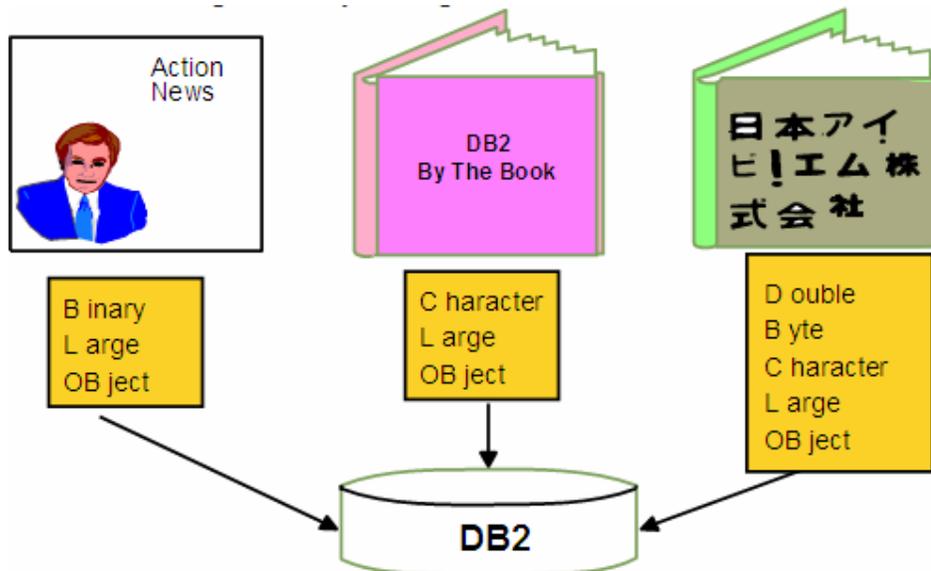


Figura 8.2 – Tipos de dados LOBs

Esses objetos binários grandes têm seus nomes abreviados por conveniência, sendo objeto binário grande um BLOB, objeto texto grande um CLOB e um objeto texto de duplo byte grande um DBCLOB.

### User-defined types

O DB2 permite que seus usuários definam seus próprios tipos de dados baseados em tipos de dados existentes. Estes são conhecidos como tipos de dados definidos pelo usuário (UDT). UDT's são úteis quando:

- Existe a necessidade de estabelecer contexto para os valores
- Existe a necessidade do DB2 forçar os dados informados

As instruções abaixo ilustram como criar e utilizar os UDTs.

```
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 )
```

Neste exemplo, 2 UDTs são criados. POUND e KILOGRAM. Ambos são baseados no tipo de dados INTEGER. A cláusula WITH COMPARISONS definida como parte da sintaxe, define que funções de modelagem com o mesmo nome dos tipos de dados também serão criados.

A tabela `person` utiliza o dois novos UDTs na colunas `weight_p` e `weight_k`, respectivamente. Se você executar o seguinte comando:

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

Você receberá uma mensagem de erro informando que 2 colunas com tipos de dados diferentes foram comparadas. Apesar de `weight_p` e `weight_k` serem dos tipos de dados POUND e KILOGRAM respectivamente, ambos baseados no tipo de dado INTEGER, você impossibilita a comparação desses dados. É exatamente isto que você precisa, pois em uma situação real, o que significaria a comparação entre libras e quilograma? Realmente não faria sentido.

No próximo exemplo, você gostaria de comparar a coluna `weight_p` com um inteiro, entretanto esses tipos de dados são diferentes, e você receberia uma mensagem caso não utilizasse a função de modelagem.

Como você pode observar na instrução abaixo, nós utilizamos a função de modelagem POUND() para tornar essa comparação possível. Como indicado anteriormente, a função de modelagem POUND() foi criada com o UDT quando utilizamos a cláusula WITH COMPARISONS da instrução CREATE DISTINCT TYPE.

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

### Valores nulos

Um valor nulo representa um estado desconhecido. Contudo, a instrução CREATE TABLE pode definir uma coluna utilizando a cláusula NOT NULL, garantindo assim, que a coluna possua valores desconhecidos. Você também pode especificar um valor padrão para a coluna se NOT NULL não estiver definido. A próxima instrução nos mostra um exemplo 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
)
```

### 8.2.2 Colunas Identity

Uma coluna identity é uma coluna numérica que automaticamente gera um número único para cada registro inserido. Deve existir apenas uma coluna identity por tabela.

Existem duas maneiras de gerar valores para uma coluna identity, dependendo como ela foi definida.

- **Generated always:** os valores sempre são gerados pelo DB2. Aplicações não podem definir um valor explícito.
- **Generated by default:** valores podem ser definidos explicitamente por uma aplicação, ou o DB2 definirá o valor caso nenhum dado seja informado. Neste caso o DB2 não pode garantir a unicidade do valor. Esta opção é destinada para propagação de dados, e para a carga e descarga de uma tabela.

Vamos analisar o exemplo abaixo:

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

Neste exemplo, a coluna subscriberID é um inteiro definido como uma coluna identity que sempre gera valores. O valor gerado iniciará em 100 e será incrementado em 100.

### 8.2.3 Objetos SEQUENCE

Objetos sequence geram números únicos na base de dados. Diferentemente de colunas identity, as sequences são independentes de tabelas. A instrução abaixo nos mostra 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
```

PREVVAL retorna o valor atual da sequence, enquanto NEXTVAL retorna o próximo valor.

O exemplo acima também usa SYSIBM.SYSDUMMY1. Este é uma tabela do catálogo do sistema que possui uma coluna e uma linha. É utilizado em situações onde uma consulta necessita retornar uma saída baseada em apenas um valor. Tabelas do catálogo do sistema serão descritas na próxima seção.

## 8.2.4 Tabelas do catálogo do sistema

Cada base de dados possui seu próprio catálogo de tabelas e visões do sistema. Esses objetos armazenam *metadados* dos objetos da base de dados. Você pode consultar esses objetos como uma tabela qualquer da base de dados.

3 esquemas são utilizados para identificar tabelas do catálogo do sistema.

- SYSIBM: Tabelas bases, otimizadas para uso do DB2.
- SYSCAT: visões baseadas nas tabelas de SYSIBM, otimizadas para facilitar o uso
- SYSSTAT: estatísticas da base de dados

Abaixo seguem alguns exemplos das visões do catálogo:

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

### 8.2.5 Tabelas temporárias

Tabelas temporárias são tabelas criadas em memória e utilizadas por uma aplicação, assim que a aplicação for encerrada, elas serão removidas. Essas tabelas podem ser acessadas apenas pelas aplicações que as criaram. Nenhuma entrada existirá nas tabelas de catálogo do DB2.

O acesso a essas tabelas é muito eficiente porque não existe uma contenção no catálogo, nenhum bloqueio de registros, nenhuma opção padrão de logging (logging é opcional) e nenhuma checagem de autoridade. Existe o suporte de índices para tabelas temporárias, o que significa que qualquer índice pode ser criado em tabelas temporárias.

Você pode executar RUNSTATS nessas tabelas.

As tabelas temporárias são alocadas em um tablespace temporário do usuário, o qual deve ser definido anteriormente para a criação de tabelas temporárias. As instruções a seguir disponibilizam exemplos de como criar 3 tabelas temporárias.

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

```
DECLARE GLOBAL TEMPORARY TABLE tempemployees
  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;
```

Quando uma tabela temporária é criada, seu esquema será SESSION, e deverá ser especificado. O ID de usuário utilizado para criar a tabela temporária terá todos os privilégios na tabela. Cada aplicação que cria uma tabela temporária terá sua própria cópia, como mostrado na Figura 8.5.



Figura 8.5 – Escopo de tabelas temporárias globais

### 8.3 Visões

Uma visão é a representação dos dados de uma tabela. Os dados de uma visão não são armazenados separadamente, mas obtidos quando a visão é requisitada. Visões aninhadas são visões baseadas em outras visões. Todas as informações sobre visões são armazenadas nas seguintes visões de catálogo: SYSCAT.VIEWS, SYSCAT.VIEWDEP e SYSCAT.TABLES. Abaixo segue um exemplo de como criar e usar uma visão.

```
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
...		

### 8.4 Índices

Um índice é um conjunto ordenado de chaves, cada um referenciando um registro de uma tabela. Um índice permite a unicidade e melhora o desempenho. Algumas das características sobre índices são:

- A ordenação de um índice pode ser crescente ou decrescente
- As chaves de um índice podem ser únicas ou não-únicas
- Várias colunas podem ser utilizadas para um índice (também conhecido como índice composto)
- Se o índice e os dados físicos são “clusterizados” em uma seqüência de índice similar, eles serão um índice cluster.

Como exemplo:

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

### 8.4.1 Design Advisor

O *Design Advisor* é uma excelente ferramenta para orientar você no design de otimização da sua base de dados para um determinado trabalho. O *Design Advisor* pode auxiliá-lo no design dos seus índices, consultas em tabelas materializadas (MQTs), clustering multi-dimensional (MDC) e a característica de particionamento de base de dados. O *Design Advisor* pode ser carregado no *Control Center*, através do clique com o botão direito na base de dados e selecionando “Design Advisor”, como mostrado na Figura 8.6.

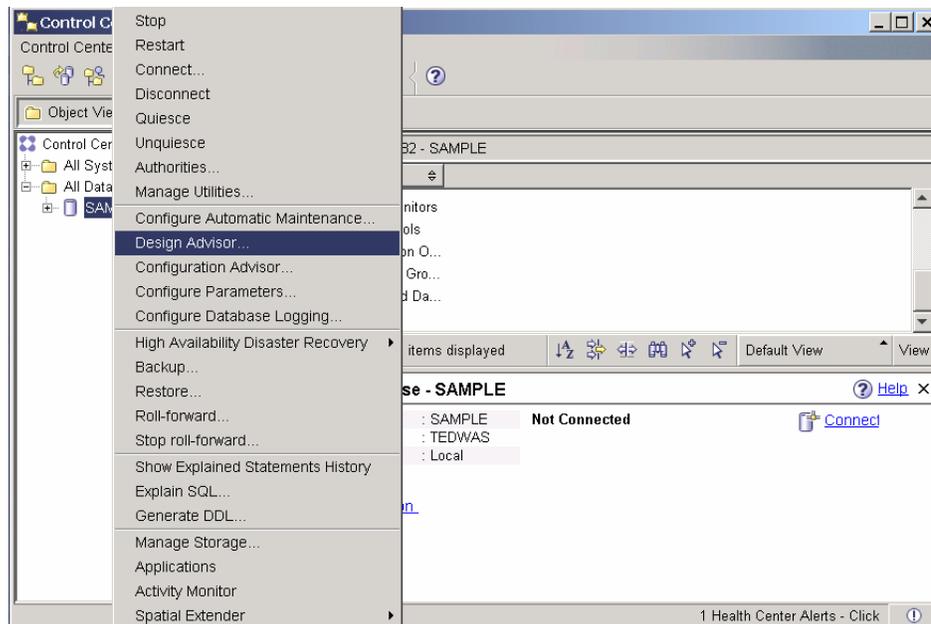


Figura 8.6 – Carregando o *Design Advisor* no *Control Center*

Figura 8.7 mostra o *Design Advisor*. Siga os passos do assistente para obter maiores recomendações sobre DB2.



Figura 8.7 –O Design Advisor

## 8.5 Integridade Referencial

A integridade referencial permite que sua base de dados gerencie o relacionamento entre tabelas. Você pode estabelecer relacionamentos do tipo pai-filho entre tabelas, como mostrado na Figura 8.8. Na Figura existem 2 tabelas, DEPARTMENT e EMPLOYEE, relacionadas pelo número do departamento. A coluna WORKDEPT na tabela EMPLOYEE só pode conter números de departamentos que já existem na tabela DEPARTMENT. Isto só acontece, porque nesse exemplo a tabela DEPARTMENT é a tabela pai e a tabela EMPLOYEE é a tabela filho ou tabela dependente. A Figura também mostra a instrução CREATE TABLE necessária para estabelecer o relacionamento.

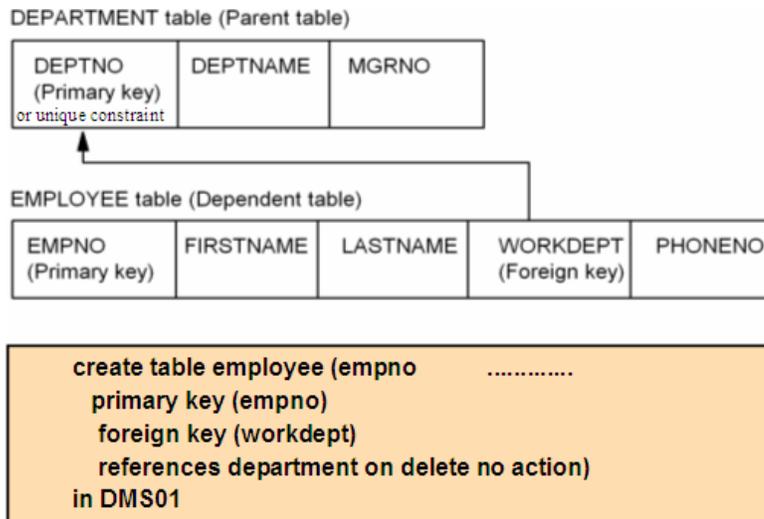


Figura 8.8 –Um exemplo de integridade referencial entre tabelas

Na integridade referencial, os seguintes conceitos são utilizados.

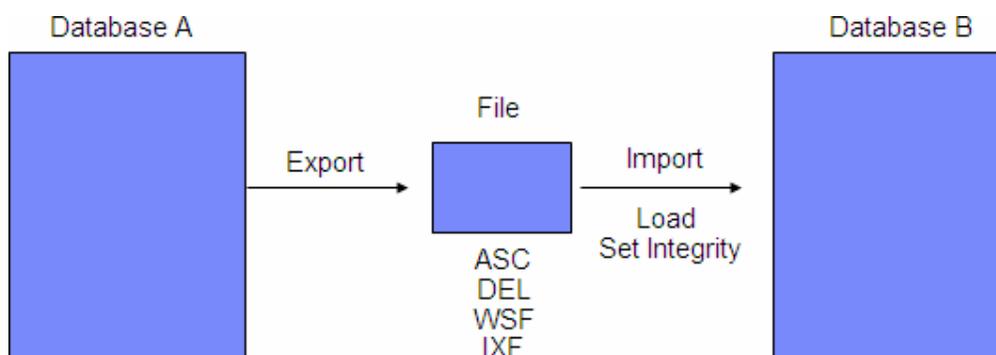
Conceito	Descrição
Tabela-pai	Uma tabela controladora de dados, onde as chaves-pai existem.
Tabela dependente	Uma tabela dependente dos dados da tabela-pai. Ela contém também uma chave estrangeira. Para um registro existir na tabela dependente, um registro igual deve existir na tabela-pai.
Chave primária	Define a chave-pai da tabela-pai. Ela não pode conter valores nulos e os valores devem ser únicos. Uma chave primária é composta de uma ou mais colunas de uma tabela.
Chave Estrangeira	Referencia a chave primaria de uma tabela-pai.

Dados em tabelas podem estar relacionados com uma ou mais tabelas através da integridade referencial. *Constraints* (restrições) também podem ser aplicados em valores de dados para que eles estejam de acordo com uma propriedade ou regra de negócio. Por exemplo, se a coluna de uma tabela armazena o sexo de uma pessoa, uma *constraint* pode garantir que os únicos valores permitidos são 'M' para masculino e 'F' para feminino.

# 9

## Capítulo 9 – Utilitários de movimentação de dados

As ferramentas ou comandos descritos nesta seção são usados para movimentar dados na mesma base de dados ou entre de bases de dados na mesma ou em diferentes plataformas. A Figura 9.1 fornece um resumo dos utilitários de movimentação de dados.



**Figura 9.1 – Utilitários de movimentação de dados**

Na Figura 9.1 existem duas bases de dados, base de dados A e B. Através do utilitário EXPORT, pode-se exportar os dados de uma tabela para um arquivo. O arquivo pode ter os seguintes formatos:

ASC = ASCII

DEL = ASCII delimitado

WSF = Formato de planilha

IXF = Formato de troca integrado

Arquivos ASC e DEL são arquivos texto que podem ser lidos e revisados em qualquer editor de texto. WSF é um formato que pode mover os dados para planilhas como Excel or Lotus® 1-2-3. IXF é um formato que inclui os dados e a Linguagem de Definição de Dados (DDL) da tabela em questão. Isto será conveniente caso a tabela precise ser reconstruída, porque é possível fazê-lo diretamente de um arquivo com o formato IXF, enquanto utilizando outros formatos tal ação não seria possível.

Uma vez que os dados foram exportados para o arquivo, o utilitário IMPORT pode ser utilizado para importar os dados do arquivo para outra tabela. A tabela já deve existir

quando for utilizado os formatos ASC, DEL e WSF, porém, se for utilizado o formato IXF a tabela não precisa ser criada antes da importação. Outra maneira de carregar os dados em uma tabela é utilizar o utilitário LOAD. O utilitário LOAD é rápido e acessa diretamente as páginas da base de dados, sem interagir com a *engine* do DB2; contudo, este método não realiza a verificação das *constraints*, assim como *triggers* não serão disparadas. Para garantir a consistência dos dados carregados pelo utilitário LOAD, o comando SET INTEGRITY é frequentemente utilizados após a carga.

As próximas seções descreverão os utilitários EXPORT, IMPORT e LOAD em mais detalhes.

**Nota:**

Para mais informações sobre como trabalhar com os Utilitários de Movimentação de dados, assista ao vídeo:

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

## 9.1 Utilitário EXPORT

O utilitário EXPORT é usado para extrair dados de uma tabela para um arquivo, como visto anteriormente. Internamente uma operação SQL SELECT será executada. O exemplo abaixo exporta para o arquivo *employee.ixf*, no formato IXF, 10 linhas da tabela *employee*.

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

Recomendamos que você teste o exemplo acima. A tabela *employee* faz parte da base de dados SAMPLE, portanto você precisa se conectar nesta base de dados criada no capítulo anterior.

Se você preferir trabalhar com ferramentas GUI, o utilitário EXPORT também pode ser encontrado no *Control Center*, como mostrado na Figura 9.2.

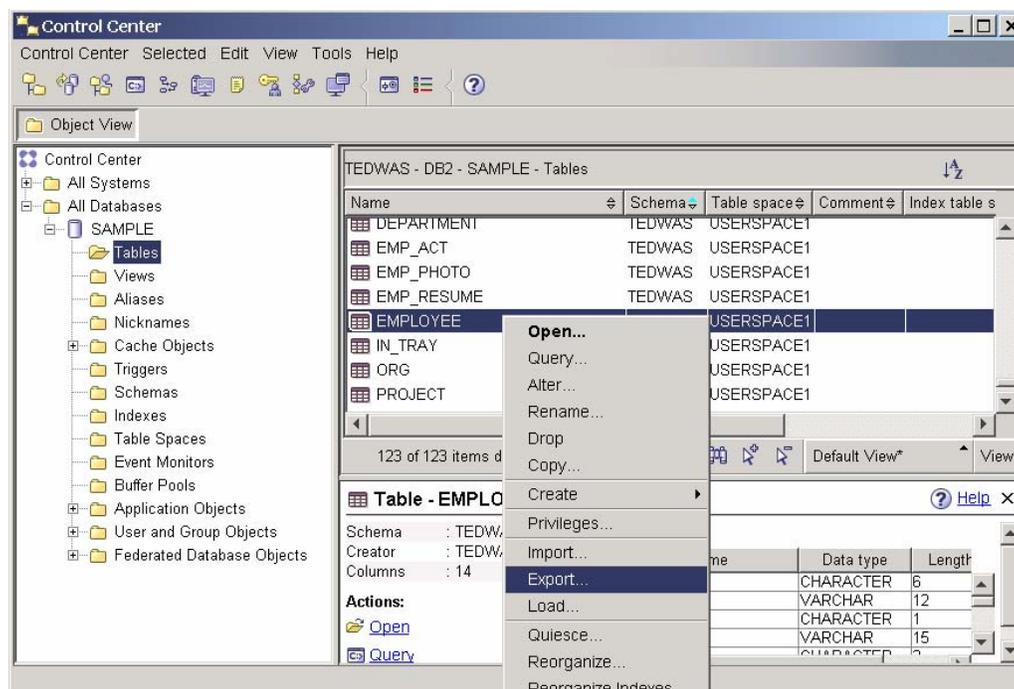


Figura 9.2 – Carregando a caixa de diálogo EXPORT

Como mostrado na figura, você deve primeiro selecionar a tabela *employee*, clicando sobre ela uma vez, e então clicando com o botão direito sobre a tabela, um menu pop-up será exibido, onde você poderá escolher a opção “Export”. Após selecionar esta opção, um assistente aparecerá. Siga os passos fornecidos pelo assistente para finalizar a operação.

## 9.2 IMPORT utility

O utilitário IMPORT é usado para carregar os dados de um arquivo para uma tabela. Internamente uma operação SQL INSERT é executada. Como uma operação INSERT é executada, todas as *triggers* são ativadas, todas as *constraints* são verificadas imediatamente e o *bufferpool* da base de dados é utilizado. O exemplo a seguir carrega todos os dados do arquivo IXF *employee.ixf* na tabela *employee\_copy*. A opção REPLACE\_CREATE é uma das muitas opções disponíveis para o utilitário IMPORT. Esta opção substituirá o conteúdo da tabela *employee\_copy*, caso ela exista antes da execução do utilitário IMPORT, ou criará e carregará os dados se a tabela não existir. Recomendamos que você teste o exemplo abaixo, porém primeiro execute o utilitário EXPORT da seção anterior.

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

Se você preferir trabalhar pelo Control Center, você poderá carregar o utilitário EXPORT, selecionado qualquer tabela com o clique do botão direito, e escolher a opção IMPORT, como mostrado na Figura 9.3.

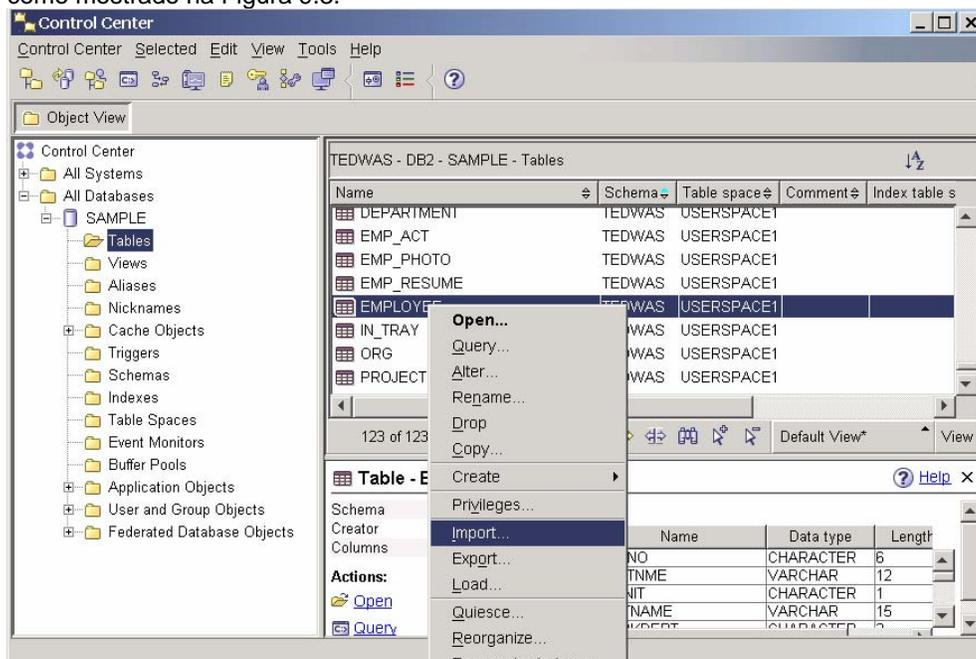


Figura 9.3 – Carregando a caixa de diálogo IMPORT

### 9.3 LOAD

O utilitário LOAD é a maneira mais rápida de carregar dados de um arquivo para uma tabela. Como visto anteriormente, o utilitário LOAD não utiliza a *engine* do DB2, portanto triggers não são ativados, o *bufferpool* não é utilizado e as *constraints* podem ser verificadas somente em uma etapa diferente. Por outro lado, a operação LOAD é mais rápida que a operação IMPORT em função de ser um carregador de dados de baixo nível, que acessa diretamente as páginas de dados no disco. Este processo acontece em três fases: LOAD, BUILD e DELETE.

O exemplo a seguir carrega todos os dados do arquivo IXF *employee.ixf* na tabela *employee\_copy*. O REPLACE é uma das opções disponíveis do utilitário LOAD. Neste caso é utilizado para SUBSTITUIR todo o conteúdo da tabela *employee\_copy*.

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

Após executar o comando acima (o qual você pode testar), a *tablespace* a qual pertence a tabela, pode ser colocado no estado CHECK\_PENDING. Isso significa que você precisará executar o comando SET\_INTEGRITY para verificar a consistência dos seus dados, conforme o exemplo abaixo.

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

Se você preferir trabalhar com o Control Center, você pode carregar os utilitários LOAD e SET INTEGRITY, como mostrado nas Figura 9.4 e 9.5, respectivamente.

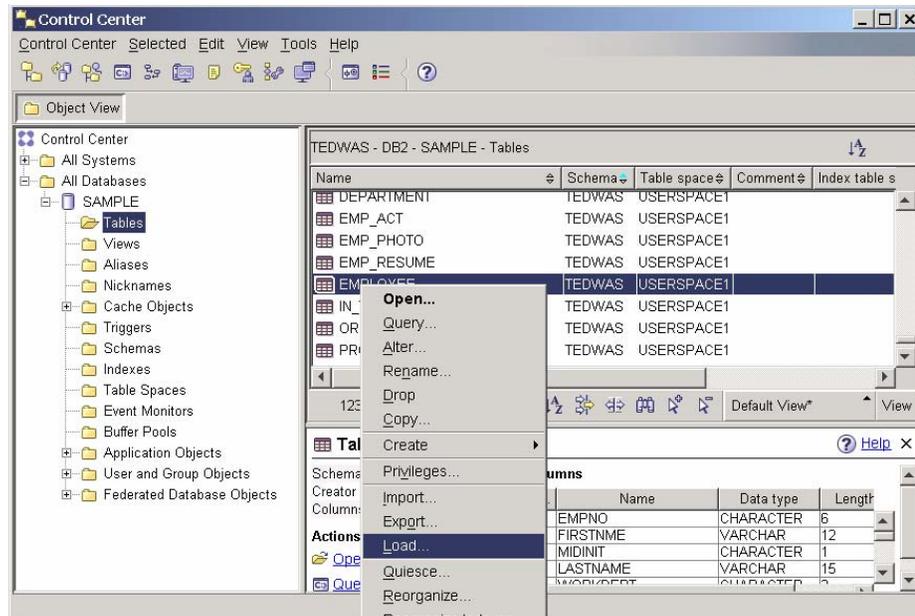


Figura 9.4 – Carregando o assistente LOAD

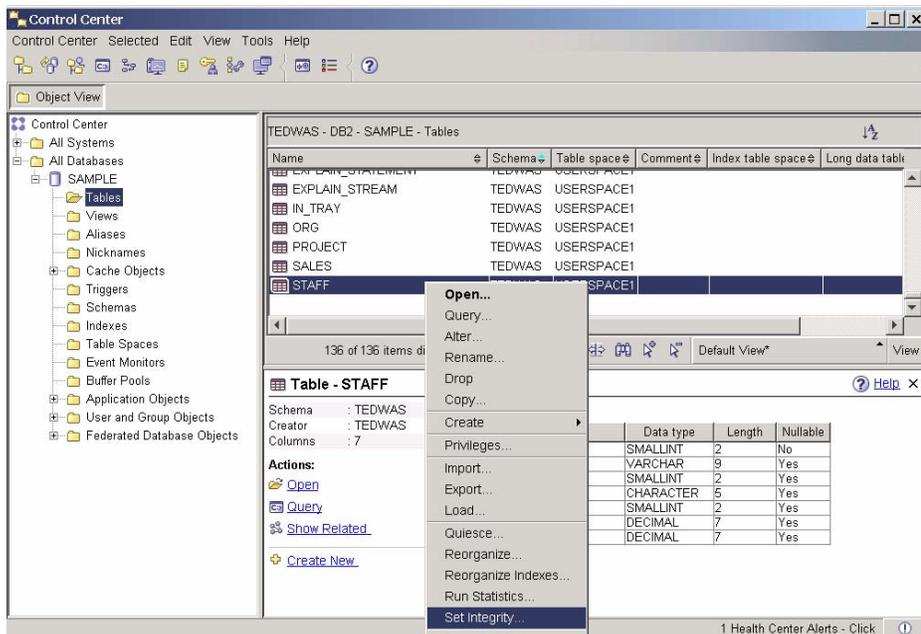


Figura 9.5 – Carregando o assistente SET INTEGRITY

## 9.4 O utilitário db2move

Os utilitários EXPORT, IMPORT e LOAD trabalham com uma tabela por vez. Embora você pudesse escrever um script para gerar os comandos acima para cada tabela na base de dados, outro utilitário chamado *db2move*, pode fazer isso por você. O utilitário *db2move* trabalha somente com os arquivos IXF e os nomes dos arquivos serão automaticamente gerados pelo *db2move*. Os exemplos abaixo mostram como executar o *db2move* com as opções EXPORT e IMPORT respectivamente, usando a base de dados SAMPLE.

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

O Control Center não tem uma opção para o *db2move*.

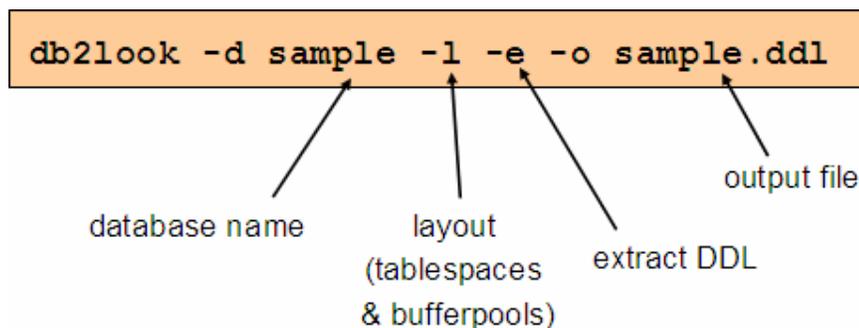
## 9.5 O utilitário db2look

Enquanto os utilitários EXPORT, IMPORT, LOAD e *db2move* permitem você movimentar dados de uma tabela para outra, seja em uma base de dados ou através de diversas bases de dados, o utilitário *db2look* pode ser utilizado para extrair as instruções DDL, estatísticas de bases de dados e características de tablespaces de uma base de dados e armazená-las em um arquivo de script que poderá ser executado em um outro sistema. Por exemplo, se você quisesse clonar uma base de dados de um servidor DB2 em Linux para um servidor DB2 em Windows; você poderia inicialmente executar o utilitário *db2look*

no servidor DB2 em Linux para obter a estrutura da base de dados e armazenar a estrutura em um arquivo de script. Você então copiaria o arquivo de script para o servidor DB2 em Windows e executaria o script para iniciar a construção da base de dados clonada. Neste momento, a estrutura da base de dados seria clonada. O próximo passo seria executar o utilitário *db2move* com a opção EXPORT no servidor DB2 em Linux e então copiar todos os arquivos gerados para o servidor DB2 em Windows. Após isso executaria o *db2move* com as opções IMPORT ou LOAD. Uma vez feito isso, sua base de dados estaria completamente clonada de um servidor para outro em plataformas diferentes.

O cenário abaixo pode ser necessário ao trabalhar com base de dados e diferentes plataformas, como Linux e Windows. Se ambos os servidores são executados na mesma plataforma você poderia usar os comandos BACKUP e RESTORE, o que torna este processo mais fácil e mais direto. Os comandos BACKUP e RESTORE são discutidos em mais detalhes em um próximo capítulo do livro.

O exemplo abaixo extrai o layout da tablespace e do bufferpool, juntamente com as instruções DDL da base de dados SAMPLE, e as armazena no arquivo sample.ddl. Recomendamos que você execute o comando abaixo e analise a saída do arquivo texto "sample.ddl".



O comando *db2look* tem muitas opções para serem descritas no livro, contudo você pode utilizar o indicador *-h* para obter uma breve descrição das opções disponíveis.

```
db2look -h
```

O utilitário *db2look* também pode ser iniciado através do Control Center mostrado na Figura 9.6.

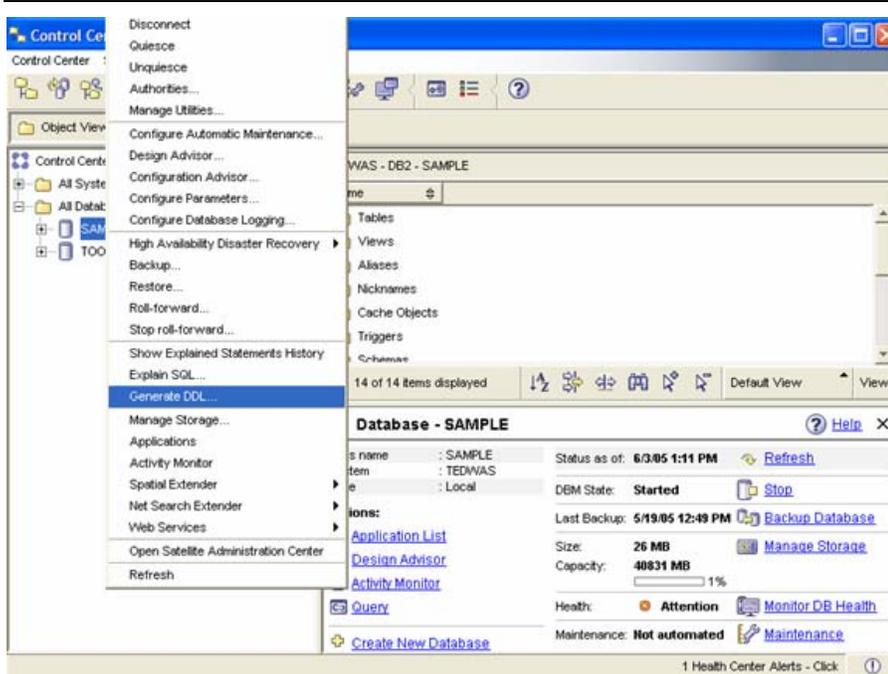
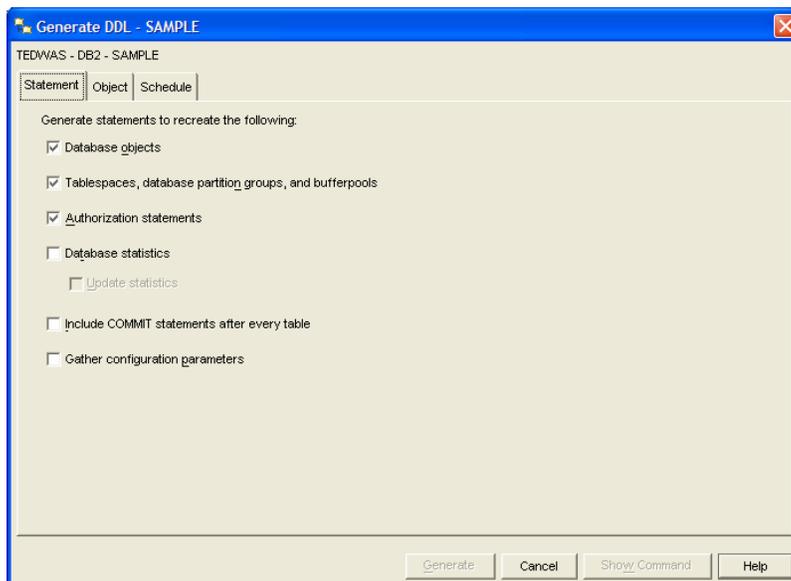


Figura 9.6 – Extraindo DDL através do Control Center

Na Figura 9.6, selecione a base de dados da qual você deseja obter o DDL, clicando com o botão sobre ela, e escolhendo “Generate DDL”. A janela do “Generate DDL” aparece mostrando várias opções de extração, conforme Figura 9.7.



**Figura 9.7 – Extraíndo DLL através do Control Center**

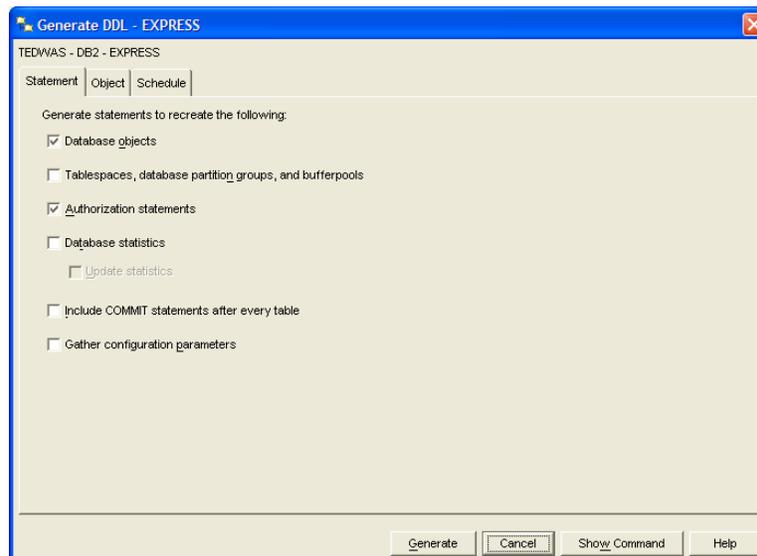
## Quicklab #8 – Extrair DDL da base de dados EXPRESS

### Objetivo

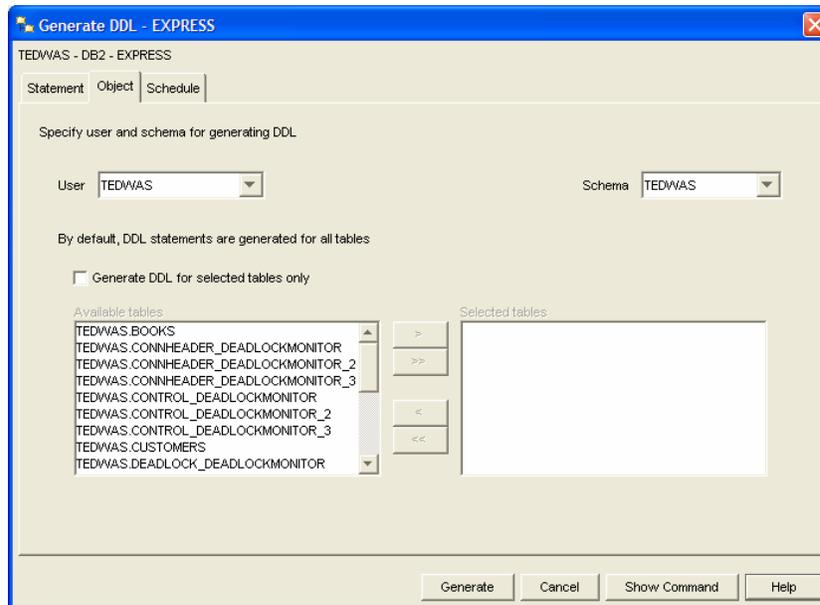
Quando você clona a base de dados, seu objetivo é realizar a recriação da base de dados, tão direta e repetidamente quanto possível. Isto é usualmente feito utilizando scripts SQL, os quais podem ser imediatamente executados após o DB2 ter sido instalado. Neste Quicklab você extrairá as definições do objeto da base de dados EXPRESS (criada no Quicklab #2) usando o *Control Center*.

### Procedimento

1. Abra o Control Center
2. Na árvore de objetos, clique com o botão direito sobre a base de dados EXPRESS e selecione o item de menu “Generate DDL”. Isto carrega a janela de diálogo “Generate DDL”.
3. Na janela “Generate DDL” especifique as opções para o DDL gerado, como mostrado abaixo. Se você criou objetos adicionais no seu ambiente, como tablespaces, bufferpools, etc, você selecionaria eles aqui. Desde que você não tivesse criado esses tipos de objetos, desmarque a caixa. Estatísticas de base de dados não foram incluídas porque o ambiente de produção terá um conjunto de estatísticas diferente do ambiente de desenvolvimento. Similarmente, os parâmetros de configuração serão diferentes também. No seu próprio ambiente, se tudo estiver configurado exatamente da maneira como será implantado, você deveria incluir aquelas opções adicionais.



4. Alterne para a guia *Object*. Você está apto para escolher apenas os objetos que você deseja gerar o DDL. Neste caso, selecione o usuário e o esquema que você usou para criar todos os objetos, e gerar o DDL para todos os objetos naquele esquema. Clique no botão "Generate" para iniciar a criação do DDL.



5. Revise o DDL gerado. O resultado do passo anterior é um único script com todas as instruções SQL para os objetos escolhidos. Agora você organizará o script em agrupamentos lógicos.
6. Crie um diretório chamado "C:\express" no sistema de arquivos e salve neste diretório o arquivo DDL gerado para um arquivo chamado *schema.ddl*. (Clique no botão "Save").
7. Abra o novo arquivo salvo no Command Editor. (Dica: No Command Editor, escolha File => Open).
8. Embora nós tenhamos apenas escolhido o DDL de tabelas, você perceberá que o DDL para outros objetos da base de dados também foram incluídos. Mova todas as instruções CREATE TRIGGER para um novo arquivo chamado "triggers.ddl". Embora nós apenas tenhamos a criação de um *trigger*, geralmente é uma boa prática separar objetos por tipos.
9. Recomendamos também remover todos as:
  - a. Instruções CONNECT TO
  - b. Instruções DISCONNECT

Você deveria ter dois scripts neste momento.

DDL para tabelas, visões, índices e *constraints*.  
C:\express\schema.ddl

DLL para *triggers*  
C:\express\triggers.ddl

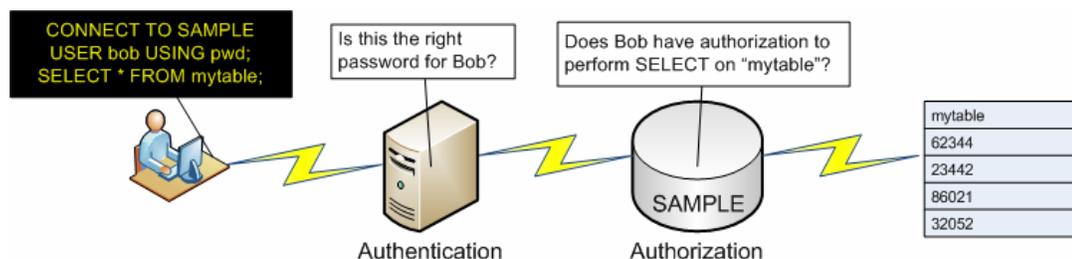
10. Refine o script para implantação:
  - a. Remova comentários desnecessários (por exemplo – CONNECT TO...)
  - b. Separe as funções e procedimentos nos seus próprios arquivos (útil quando existem muitas funções e procedimentos). Você deve também querer agrupar eles por função ou aplicação. (por exemplo billing.ddl, math.dll, stringfunc.ddl, etc)
11. Verifique que um caractere especial é utilizado para delimitar o final dos *triggers*, funções e procedimentos ( @ ). Isto é necessário ao delimitar o final da instrução *CREATE <object>* em oposição ao final da instrução procedural no objeto.



# 10

## Capítulo 10 – Segurança da base de dados

Este capítulo aborda como a segurança é tratada no DB2. A Figura 10.1 fornece uma visão básica:



**Figura 10.1 – Resumo da segurança no DB2**

Conforme a figura acima, a segurança no DB2 acontece em dois momentos:

### Autenticação

Este é o processo onde o usuário é validado. A autenticação é executada por um recurso de segurança externo ao DB2 (tipicamente pelo sistema operacional, algum método de autenticação da rede ou algum plug-in de autenticação). A autenticação pelo Sistema Operacional é a opção default. Ao utilizar a autenticação pelo sistema operacional, a identificação do usuário e a senha são enviadas ao servidor (parte da instrução de conexão). O servidor de base de dados então invoca a autenticação do sistema operacional para validar a identificação do usuário e a senha.

### Autorização

Nesta etapa, o DB2 verifica se o usuário autenticado pode executar a operação solicitada. A informação de autorização é armazenada no catálogo do DB2 e no arquivo de configuração DBM.

Por exemplo, na Figura 10.1, o usuário "bob" se conecta na base de dados SAMPLE com a instrução:

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

Tanto “bob” quanto “pwd” são enviados ao sistema operacional ou algum mecanismo externo de autenticação para aprovar a autenticação, verificando se o nome de usuário “bob” está definido e se a senha informada pertence ao usuário. No sucesso da execução da autenticação o sistema operacional retornará o controle de segurança para o DB2. Após isso, quando o usuário “bob” executa uma instrução como:

```
SELECT * FROM mytable
```

Agora o DB2 assume o controle de segurança para verificar a autorização e confirmar se o usuário “bob” possui privilégio de executar instruções SELECT na tabela “mytable”. Se a verificação da autorização falhar, o DB2 retornará uma mensagem de erro, caso contrário a instrução será executada na tabela “mytable”.

**Nota:**

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

## 10.1 Autenticação

Embora a autenticação seja executada pelo sistema operacional (ou outro mecanismo de segurança), o DB2 decide o nível em que esta autenticação ocorre.

O parâmetro AUTHENTICATION do DBM CFG definido no servidor tem um intervalo de possíveis valores. Por exemplo, quando o parâmetro é definido para SERVER (o padrão), a autenticação é executada pelo sistema operacional/mecanismo externo de segurança no servidor. Contudo, se AUTHENTICATION é definida para CLIENT, a autenticação é executada pelo sistema operacional/mecanismo externo de segurança no cliente, conforme a Figura 10.2.

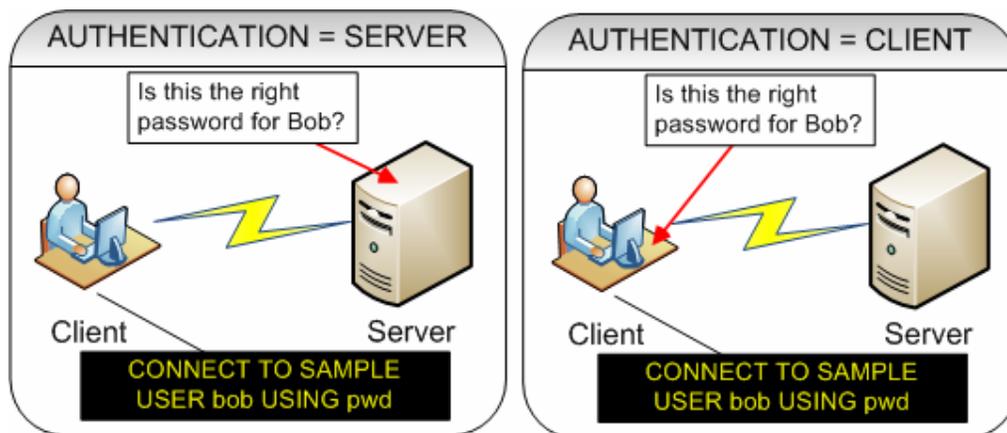


Figura 10.2 – Aonde a autenticação Where é realizada

O parâmetro AUTHENTICATION pode ter os valores listados na tabela 10.1.

Comando	Descrição
SERVER (default)	Autenticação realizada no servidor.
CLIENT	Autenticação realizada no cliente.
SERVER_ENCRYPT	Similar a autenticação SERVER, porém com o usuário e senha criptografados.
KERBEROS	Autenticação realizada pelo mecanismo de segurança Kerberos.
SQL_AUTHENTICATION_DATAENC	Autenticação SERVER com conexão de dados criptografados.
SQL_AUTHENTICATION_DATAENC_CMP	Similar a autenticação acima, porém a criptografia de dados utilizada apenas quando disponível.
GSSPLUGIN	Autenticação utiliza um plug-in de segurança baseado na API GSS

**Tabela 10.1 – Valores válidos para o parâmetro AUTHENTICATION**

## 10.2 Autorização

A autorização consiste dos privilégios e autorizações que são armazenadas nas tabelas de sistema do DB2 e gerenciadas pelo DB2.

Um privilégio permite o usuário executar uma simples operação na base de dados como CREATE, UPDATE, DELETE, INSERT, etc.

Uma autorização é um papel predefinido que consiste de vários privilégios. A Figura 10.3 mostra as diferentes autorizações e privilégios do DB2.

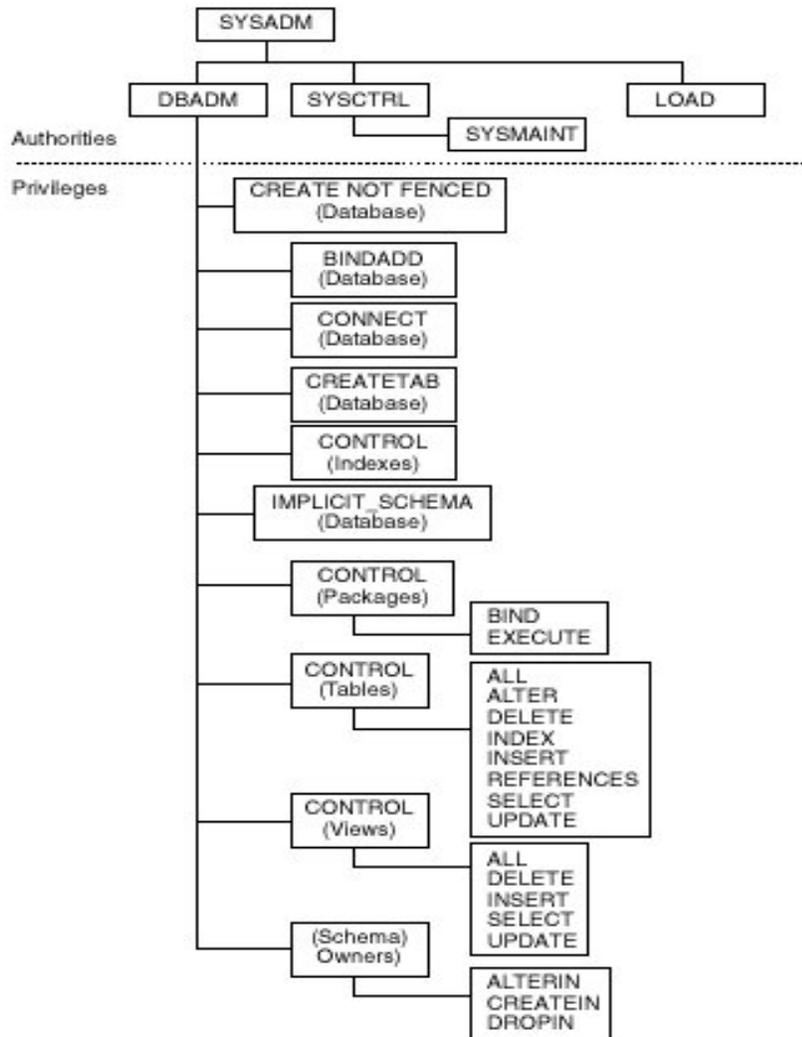


Figura 10.3 – Autorizações e privilégios

A tabela 10.2 mostra as diferentes funções que cada autorização pode executar. Como você pode ver, SYSADM possui o maior número de autorizações, enquanto SYSMON o menor.

<i>Função</i>	<i>SYSADM</i>	<i>SYSCTRL</i>	<i>SYSMAINT</i>	<i>SYSMON</i>	<i>DBADM</i>	<i>LOAD</i>
<b>Update DBM CFG</b>	Y					
<b>Grant/revoke DBADM</b>	Y					
<b>Establish/change SYSCTRL</b>	Y					
<b>Establish/change SYSMAINT</b>	Y					
<b>Establish/change SYSMON</b>	Y					
<b>Force users off database</b>	Y	Y				
<b>Create/drop database</b>	Y	Y				
<b>Restore to new database</b>	Y	Y				
<b>Update DB CFG</b>	Y	Y	Y			
<b>Backup database/table space</b>	Y	Y	Y			
<b>Restore to existing database</b>	Y	Y	Y			
<b>Perform roll-forward recovery</b>	Y	Y	Y			
<b>Start/stop instance</b>	Y	Y	Y			
<b>Restore table space</b>	Y	Y	Y			
<b>Run trace</b>	Y	Y	Y	Y		
<b>Obtain monitor snapshots</b>	Y	Y	Y			
<b>Query table space state</b>	Y	Y	Y			
<b>Prune log history files</b>	Y	Y	Y			
<b>Quiesce table space</b>	Y	Y	Y		Y	Y
<b>LOAD tables</b>	Y				Y	Y
<b>Set/unset check pending state</b>	Y				Y	
<b>Create/drop event monitors</b>	Y				Y	

---

**Tabela 10.2 – Autorizações e privilégio do DB2**

Ao conceder as autorizações SYSADM, SYSCTRL ou SYSMANT para um grupo, os parâmetros SYSADM\_GROUP, SYSCTRL\_GROUP e SYSMANT\_GROUP do DBM CFG podem ser atribuídos a um grupo do sistema operacional.

Por exemplo, para atribuir a autorização SYSADM para o grupo do sistema operacional “db2admns”, você pode executar este comando:

```
update dbm cfg using SYSADM_GROUP db2admns
```

Cada instância do DB2 tem seu próprio grupo de autorizações definidos.

No Windows, estes parâmetros não são definidos por padrão, o que significa que o grupo de administradores local do Windows será o SYSADM. No Linux o grupo do responsável pela instância será o grupo padrão.

### 10.3 Autorização DBADM

A autorização DBADM (administrador de base de dados) é um super usuário da base de dados. Não é uma autorização em nível de instância, portanto não é listada na seção anterior. Ao permitir autorização DBADM, utilize a instrução mostrada no exemplo abaixo.

```
connect to sample  
grant DBADM on database to user <userid>
```

No exemplo acima, você precisa primeiro conectar na base de dados, neste caso a base de dados “sample”, e então você pode conceder a autorização DBADM para o usuário. Você precisa ser SYSADM para permitir a autorização DBADM.

Note que o DBADM não pode criar tablespaces embora elas sejam objetos da base de dados porque tablespaces trabalham com containeres (disco) e buffer pools (memória), os quais são recursos físicos do sistema.

### 10.4 O grupo PUBLIC

O DB2 define um grupo interno chamado PUBLIC. Todo usuário identificado pelo sistema operacional ou serviço de autenticação da rede é implicitamente um membro do grupo PUBLIC. Quando uma base de dados é criada, alguns privilégios são permitidos automaticamente para PUBLIC:

- CONNECT;
- CREATETAB;
- IMPLICIT SCHEMA;
- BINDADD

Por segurança, nós recomendamos retirar 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 sentenças GRANT e REVOKE

As instruções GRANT e REVOKE fazem parte dos padrões SQL, e são utilizadas para conceder ou remover privilégios de um usuário ou grupo. Abaixo temos alguns exemplos dessas instruções:

**Para conceder o privilégio SELECT na tabela T1 para o usuário USER1**

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

**Para conceder todos os privilégios na tabela T1 para o grupo GROUP1:**

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

**Para conceder todos os privilégios na tabela T1 para o grupo GROUP1:**

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

**Para conceder o privilégio EXECUTE no procedimento p1 para o usuário USER1:**

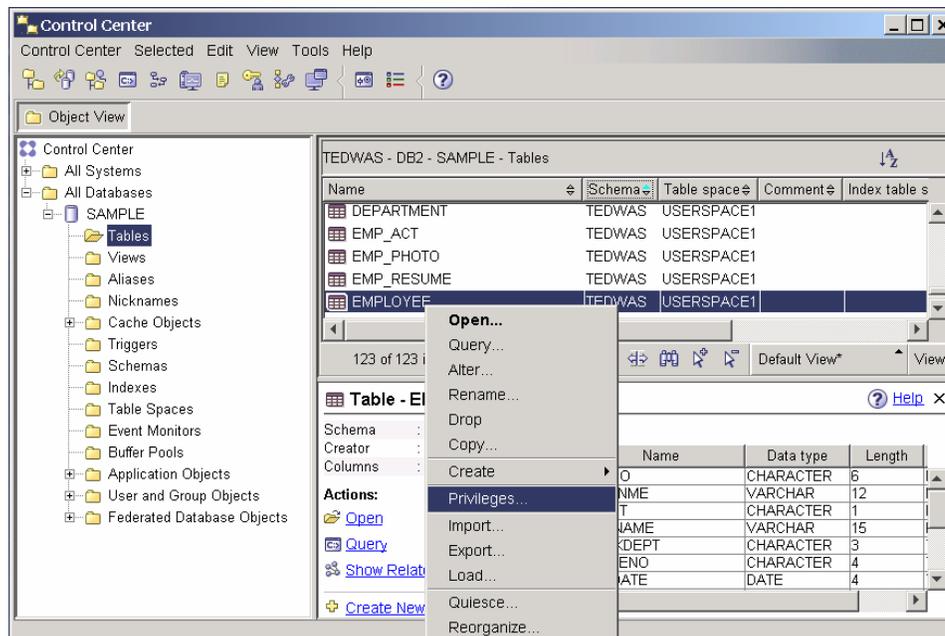
```
GRANT EXECUTE ON PROCEDURE p1 TO USER user1
```

**Para retirar o privilégio EXECUTE no procedimento p1 do usuário USER1:**

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

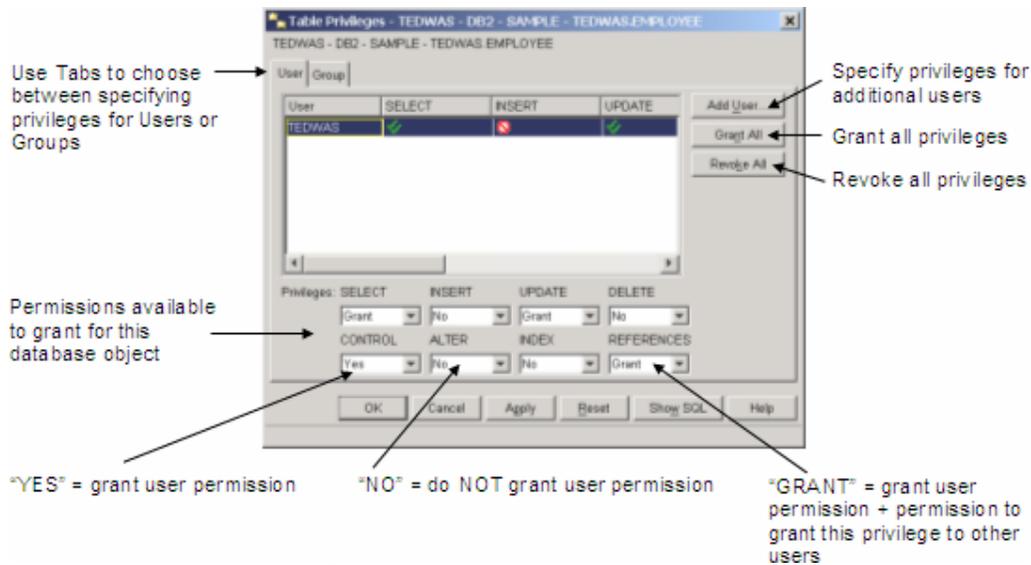
## 10.6 Verificação de autorização e privilégio

O modo mais fácil de verificar a autorização e privilégio é através do Control Center. A Figura 10.4 mostra como carregar a caixa de diálogo “Table Privileges” da tabela EMPLOYEE no Control Center.



**Figura 10.4 – Carregando a caixa de diálogo “Table Privileges”**

Conforme a Figura 10.4, você seleciona a tabela desejada, clica com o botão direito sobre ela, e escolhe “Privileges”. Uma vez selecionada a opção, a caixa de diálogo “Table Privileges” aparece, como mostrado na figura 10.5. Esta figura também explica os diversos campos e elementos da caixa de diálogo.



**Figura 10.5 – A caixa de diálogo “Table Privileges”**

Alternativamente, você pode consultar as visões do catálogo SYSCAT do DB2, as quais contêm as informações de autorização. Por exemplo, você gostaria de saber se o usuário DB2ADMIN tem privilégio de SELECT na tabela T2, e gostaria de saber quem concedeu este privilégio. Você poderia executar a seguinte consulta:

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

GRANTOR	GRANTEE	SELECTAUTH
ARFCHONG	DB2ADMIN	Y

No exemplo acima, o usuário ARFCHONG permitiu o privilégio GRANT para o usuário DB2ADMIN.

### 10.7 Considerações sobre grupo de privilégio

Para facilitar a administração do DB2, defina usuários em grupos e então conceda os privilégios necessários aos grupos.

Quando um grupo recebe privilégios, os membros do grupo recebem implicitamente os privilégios, herdando através do relacionamento.

Quando um usuário é removido de um grupo, ele perde os privilégios implícitos do grupo, mas ainda possuirá os privilégios explicitamente concedidos anteriormente. Privilégios explicitamente concedidos a um usuário devem ser removidos explicitamente.

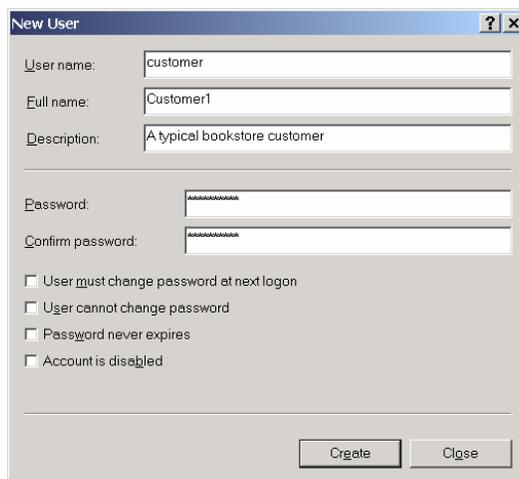
## Quicklab #9 – Concedendo e removendo permissões de usuário

### Objetivo

Você utiliza a conta do administrador da instância (SYSADM) para executar todos os comandos da base de dados. Esta conta possui acesso total a todos os utilitários, dados e objetos da base de dados. Portanto, é muito importante tomar cuidado com esta conta a fim de evitar acidentes ou perda de dados. Na maioria dos casos, você criará diferentes contas de usuário ou grupos, com um conjunto limitado de permissões. Neste exercício você criará uma nova conta de usuário e então atribuirá privilégios específicos.

### Procedimento

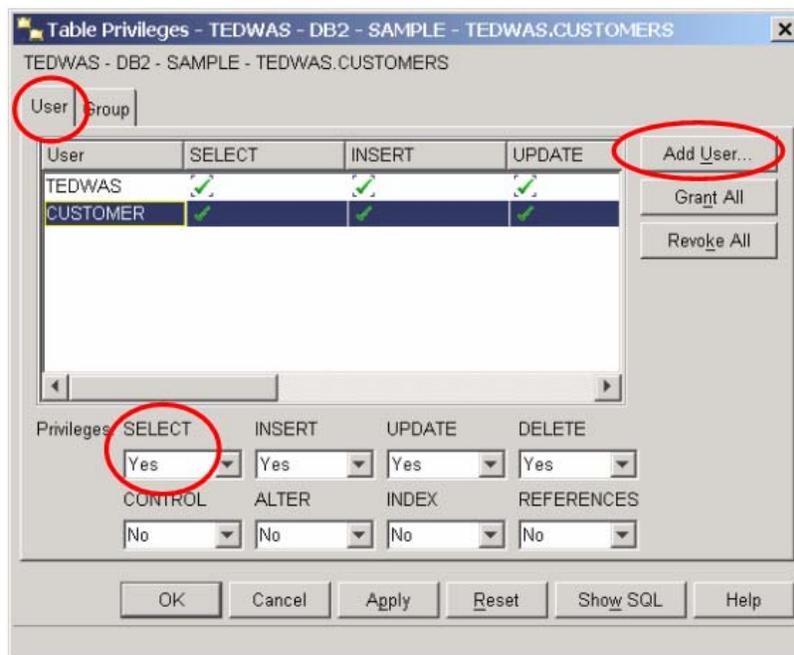
1. Abra o Gerenciador do Computador do Windows, clicando com o botão direito no ícone do “Meu Computador” na área de trabalho, e selecione “Gerenciar” no menu.
2. Expanda “Ferramentas do Sistema” na árvore no painel da esquerda da janela e expanda a pasta “Usuário e grupos locais”. Clique com o botão direito na pasta “Usuários” e selecione o item “Novo usuário...”.
3. Na janela de diálogo “Novo usuário”, insira as informações a seguir: no campo “Nome do usuário” insira “customer” e no campo “Nome completo” insira “Customer1”. No campo descrição insira “A typical bookstore customer”. Nos campos “Senha” e “Confirmar senha” insira “ibmdb2”. Desmarque a opção “O usuário deve alterar a senha no próximo logon” e clique no botão “Criar” para criar o novo usuário.



4. Garanta que a visão avançada é utilizada no Control Center do DB2. Para alternar para a visão avançada, selecione o item de menu “Customize Control Center” do

menu “Control Center Tools”. Selecione a opção “Advanced” e clique no botão “OK”.

5. Expanda a árvore de objeto do Control Center no painel esquerdo da árvore de objetos para “All Databases” > EXPRESS > Tables.
6. Conceda os privilégios necessários para o novo usuário criado. Na lista de tabelas da base de dados EXPRESS, clique com o botão direito na tabela CUSTOMERS e selecione o item “Privileges” para visualizar a janela de diálogo dos privilégios da tabela.
7. Clique no botão “Add User” e selecione o usuário “customer” criado. Clique no botão “OK” na caixa de diálogo “Add User”.
8. Você verificará que o usuário “customer” foi adicionado a lista de usuários, mas não tem privilégios atribuídos. Para atribuir os privilégios SELECT, INSERT, UPDATE e DELETE para o usuário, altere cada caixa drop down para Yes. Um cliente da Internet deveria poder ver/adicionar/atualizar/remover os dados da sua conta. Não concedemos ao usuário outras permissões porque elas não são necessárias. Clique no botão “OK” para fechar a janela de diálogo “Table Privileges” e aceite as alterações realizadas.



9. Repita os passos 7-9 para as tabelas “BOOKS” e “SALES”. Para a tabela “BOOKS” apenas conceda o privilégio SELECT porque o cliente não deve estar apto a modificar nenhum dos dados do inventário de armazenamento. Para a

tabela "SALES" apenas conceda os privilégios SELECT e INSERT. O cliente não deve ter o privilégio de remover ou atualizar, porque apenas funcionários cadastrados deveriam ter acesso para modificar as transações de vendas.

10. Conecte na base de dados usando o usuário "customer" criado acima. Tente selecionar os dados das tabelas de clientes. O que acontece? Tente remover ou atualizar os dados da tabela "SALES". O que acontece?

Neste Quicklab nós criamos somente um usuário; entretanto sua aplicação pode possuir diferentes tipos de usuário. Experimente criar outros usuários e defina seus privilégios. Você poderá criar grupos de usuários e definir os privilégios para esses grupos, ao invés de especificar a cada usuário individualmente.

# 11

## Capítulo 11 – Backup e Recuperação

Neste capítulo discutiremos o sistema de log do DB2, como fazer cópia completa ou parcial da sua base de dados usando o utilitário “BACKUP” e como recuperar seus dados através do utilitário “RESTORE”.

**Nota:**

Para maiores informações sobre o sistema de log, backup e recuperação, assista o vídeo: <http://www.channeldb2.com/video/video/show?id=807741:Video:4282>

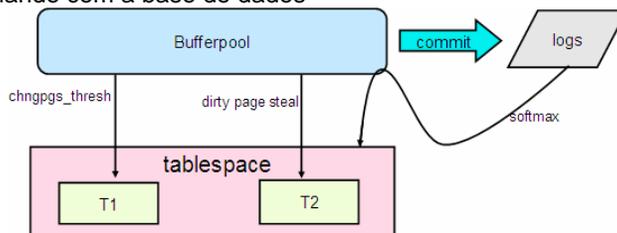
### 11.1 Log de base de dados

Se você estiver trabalhando com um editor de texto, toda vez que quiser garantir que seu documento está salvo, você clica no botão “Salvar”. No mundo da base de dados, a instrução “COMMIT” realiza isso. Toda vez que a instrução “COMMIT” é executada, você garante que qualquer alteração realizada nos dados é salva em algum lugar.

De uma maneira parecida, quando você trabalhar com um documento texto, algumas vezes você verá no lado inferior direito a breve mensagem informando “auto-saving”. No mundo dos bancos de dados isso também acontece, porque qualquer operação executada nos dados, como UPDATE, INSERT ou DELETE, será salva em algum lugar assim que você as executar.

Esse “algum lugar” do parágrafo anterior refere-se aos logs da base de dados. Os logs da base de dados são armazenados em um disco e são usados para gravar ações ou transações. Se existir um problema no sistema ou na base de dados, logs são usados para refazer transações consolidadas, durante a recuperação.

A Figura 11.1 disponibiliza uma visão gráfica do que acontece no sistema de log quando você está trabalhando com a base de dados



**Figura 11.1 – Log da base de dados**

Na Figura 11.1, você percebe tablespaces e logs. Ambos residem no disco, embora recomendamos que não devam ser mantidos no mesmo disco. Quando se executa uma operação UPDATE, as páginas para as linhas em questão serão enviadas para o buffer pool (memória). As alterações serão executadas no buffer pool e os valores antigos e novos serão armazenados nos arquivos de log, algumas vezes imediatamente ou quando o buffer estiver cheio. Se a operação COMMIT for executada após a operação UPDATE os valores antigos e novos serão armazenados nos arquivos de log imediatamente. Este processo é repetido por muitas outras operações SQL que são executadas na base de dados. Apenas quando certas condições são encontradas, como alcançar o número de páginas alteradas, especificado no parâmetro CHNGPGS\_THRES, as páginas no buffer pool serão “externalizadas” ou escritas na tablespace em disco. O parâmetro CHGPGS\_THRES indica a porcentagem do buffer pool com páginas “sujas”, isto é, páginas que contêm mudanças.

Analisando o desempenho, não faz sentido executar duas escritas para cada operação COMMIT: uma para escrever nos logs e outra para escrever na tablespace em disco, isto porque a “externalização” dos dados para a tablespace em disco apenas acontece quando os valores de parâmetros como “chgpgs\_thres” são atingidos.

## 11.2 Tipos de logs

Existem dois tipos de logs.

### Logs Primários

Eles são pré-alocados e o número de logs primários disponíveis é determinado pelo parâmetro LOGPRIMARY do DB CFG.

### Logs Secundários

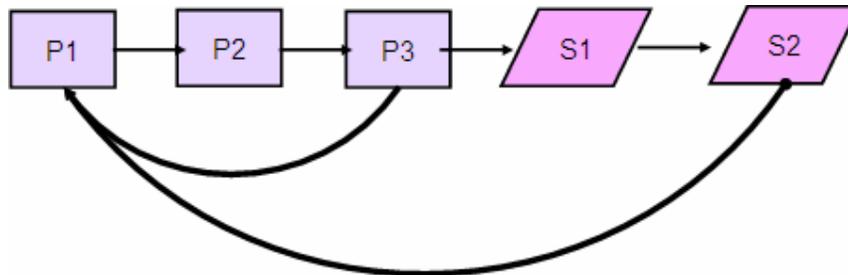
São alocados dinamicamente quando necessário pelo DB2. O número máximo de logs secundários é definido pelo parâmetro LOGSECONDD do DB CFG. Alocar dinamicamente um log é uma tarefa cara; portanto, para operações diárias, garanta que você utilize a alocação de logs primários. Os logs secundários são removidos quando todas as conexões com a base de dados são fechadas.

É possível utilizar logs infinitos caso você defina LOGSECONDD para o valor -1; contudo, isto não é recomendado, e você pode ter problemas de falta de espaço no sistema de arquivos.

### 11.3 Tipos de logging

Existem dois tipos de logging: logging circular (default) e logging de arquivamento.

A Figura 11.2 demonstra como a logging circular funciona.



**Figura 11.2 – Trabalhando com logs primários e secundários**

Na Figura 11.2 existem 3 logs primários, portanto podemos assumir que o LOGPRIMARY foi definido para 3. Para simplificar, vamos considerar que apenas uma transação é executada neste exemplo. Assim que a transação é executada, começa-se a preencher o arquivo de log P1 e então P2. Se a consolidação ocorre e a informação é externalizada para a tablespace em disco, então P1 e P2 podem ser sobrescritos, porque a informação não é mais necessária para recuperação em caso de problema (como discutimos anteriormente neste capítulo). Se, por outro lado, a transação é tão longa que precise utilizar P1, P2 e P3 e ainda necessitar mais espaço em log porque a transação não foi consolidada nem externalizada, então o log secundário (S1 na figura) seria dinamicamente alocado. Se a transação ainda prosseguir, outros logs secundários serão alocados até que o número de LOGSECOND de logs sejam alocados. Se mais logs forem necessários, uma mensagem de erro, indicando que uma condição de log cheio foi alcançada, será retornada ao usuário e a transação será desfeita.

#### 11.3.2 Log de arquivamento ou log de retenção

No log de arquivamento, também conhecido como log de retenção, os arquivos de log não são sobrescritos, mas são mantidos online ou offline. Logs de arquivamento online são mantidos com os logs ativos, os quais ainda serão necessários para a recuperação. Arquivos de logs offline são movidos para outra mídia, como uma fita, e isso pode ser feito utilizando rotinas USEREXIT. Para habilitar o log de arquivamento, defina o parâmetro LOGRETAIN do DB CFG para Yes.

Logs de arquivamento são usados normalmente em sistemas em produção, e, como os logs são mantidos, é possível fazer a recuperação da base de dados desde o ponto mais recente até o ponto mais antigo que esta no arquivo de log. Com log de arquivamento, o DBA pode recuperar erros humanos. Por exemplo, se o usuário de um sistema inadvertidamente inicia a execução de uma transação incorreta que demora alguns dias, caso o problema seja identificado tardiamente, o DBA pode recuperar o sistema para o momento em que o problema foi iniciado. Entretanto, existem algumas manipulações manuais necessárias para re-executar a transação corretamente.

O log de arquivamento é necessário para a recuperação roll-forward e backup on-line. A figura 11.3 retrata o processo do log de arquivamento.

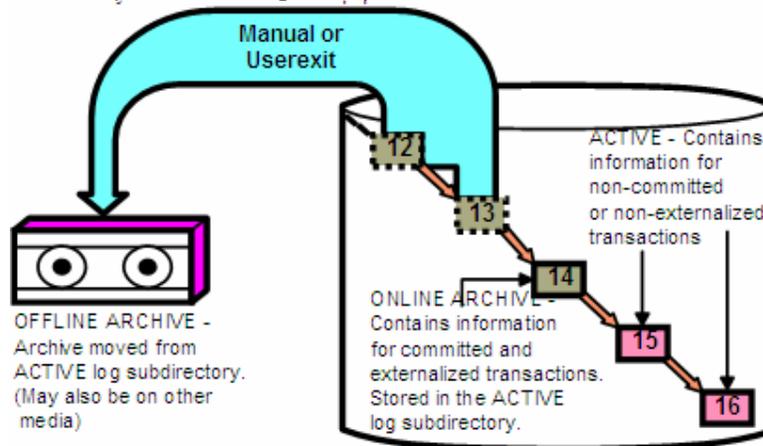


Figura 11.3 – Log de arquivamento

## 11.4 Log da base de dados no Control Center

Você pode configurar o log da base de dados no Control Center, clicando com o botão direito na base de dado em questão e escolhendo “Configure Database Logging”, conforme retratado na Figura 11.4.

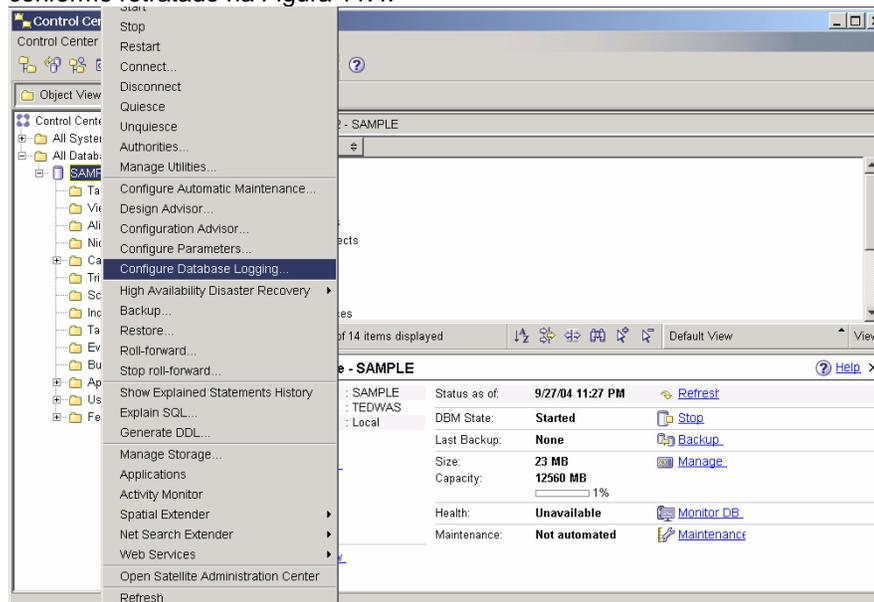


Figura 11.4 – Configurando o log da base de dados no Control Center

A figura 11.5 mostra o assistente de log da base de dados, onde você pode escolher entre o log circular e log de arquivamento.

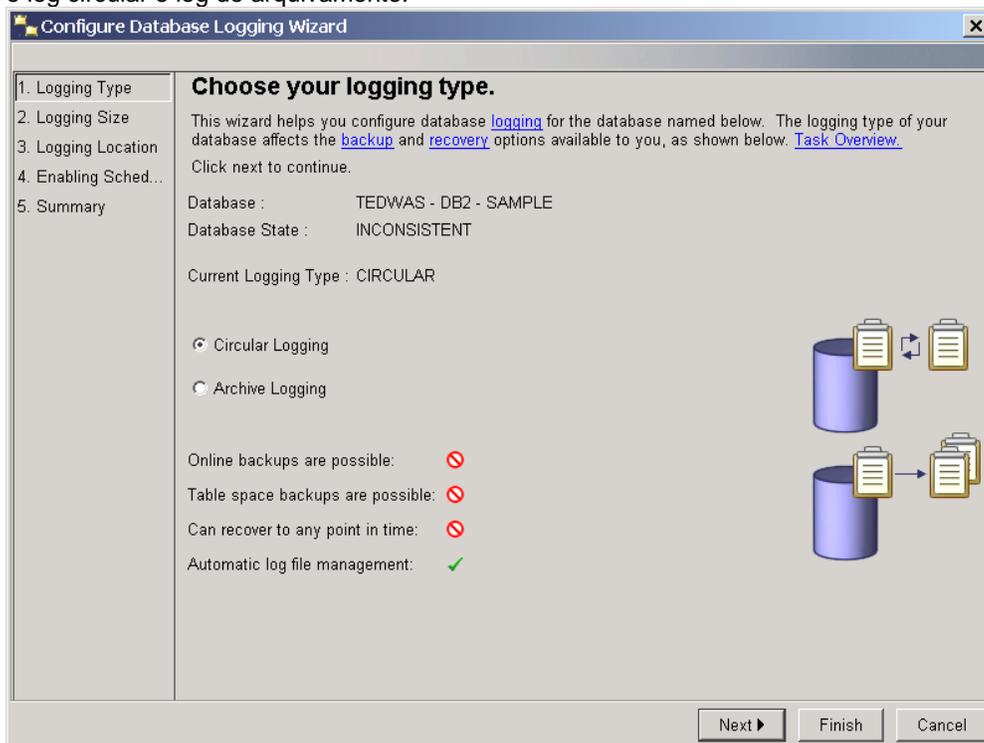


Figura 11.5 – Assistente de log da base de dados

## 11.5 Parâmetros de log

Existem alguns parâmetros no DB CFG relacionados ao log. A tabela 11.1 lista os parâmetros principais.

Parâmetro	Descrição
logbufsz	Quantidade de memória a ser utilizada com um buffer para o log, antes de escrever os dados no disco.
logfilsz	Tamanho de cada log configurado, com 4KB de páginas.
logprimary	Número de logs primários de tamanho <b>logfilsz</b> que serão criados.
logsecond	Número de logs secundários que serão criados, caso necessário, na recuperação dos dados.
logpath/newlogpath mirrorlogpath	Local onde os atuais e futuros arquivos de log estão localizados. Para proteger os logs de uma falha de disco ou remoção por acidente, você pode especificar que um conjunto idêntico de logs seja mantido em um local secundário (espelho).
loghead	gNome do arquivo de log ativo.
userexit	Permite que o programa <b>userexit</b> copie logs offline.
softmax	Limita o custo da recuperação
logretain	Habilita o modo de log de arquivamento
overflowlogpath	Similar a opção OVERFLOW LOG PATH do comando ROLLFORWARD; entretanto, ao invés de especificar a opção OVERFLOW LOG PATH para cada comando ROLLFORWARD, você pode definir esta configuração apenas uma vez no parâmetro.
blk_log_dsk_ful	Deve-se definir esta opção para prevenir erros de disco cheio quando o DB2 não puder criar um novo arquivo de log no caminho de logs atual. Ao invés disso, o DB2 tentará criar o arquivo de log a cada 5 minutos, até que isso seja possível. Instruções SQL não-bloqueadas, e SQL read-only podem continuar sua execução.
max_log	Porcentagem do espaço máximo do log ativo por transação
num_log_span	Número de arquivos de logs para 1 UOW ativo

Tabela 11.1 – Parâmetros de log

## 11.6 Backup da base de dados

O comando backup do DB2 permite que você tire uma imagem (cópia) dos dados no momento que o comando é executado. A sintaxe mais simples para executar o programa é:

```
BACKUP DATABASE <dbname> [ TO <path> ]
```

A maioria dos comandos pode ser executada online ou offline. Online permite que outros usuários estejam conectados e estejam executando operações na base de dados enquanto você executa o comando. Offline indica que nenhum outro usuário estará

conectado na base de dados, enquanto você executa a operação. Para permitir uma operação online, adicione a palavra-chave ONLINE na sintaxe do comando, senão o comando assumirá que você está executando o comando offline.

Por exemplo, se você quer realizar o backup da base de dados SAMPLE para o diretório C:\BACKUPS, você pode executar o seguinte comando no console do Windows/Linux:

```
db2 BACKUP DB sample TO C:\BACKUPS
```

Verifique que o diretório C:\BACKUPS deve existir antes da execução do comando. Garanta também que não existem conexões na base de dados enquanto você executa o comando acima, caso contrário você receberá uma mensagem de erro informando que um backup offline não pode ser executado quando existem conexões.

Para descobrir se existem conexões na base de dados, execute o comando abaixo no console do Windows/Linux:

```
db2 list applications
```

Para forçar todas as conexões da base de dados, execute o comando abaixo no console do Windows/Linux:

```
db2 force applications all
```

Você não deve executar este comando em um ambiente de produção com muitos usuários conectados pois você pode receber muitos telefonemas de colegas furiosos! Verifique também que este ultimo comando é executado de maneira assíncrona. O que significa que quando você tentar executar backup logo após de ter o comando acima, ele pode ainda não funcionar. Espere alguns segundos, e repita o comando backup.

Depois da correta execução do backup, um novo arquivo contendo backup da imagem base de dados é criado. O nome do arquivo segue a seguinte convenção, conforme figura 11.6.

### Linux/UNIX/Windows

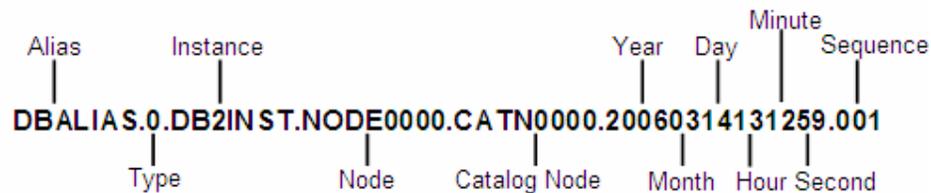


Figure 11.6 – Convenção de nome para imagem de Backup

O tipo (type) 0 significa que este backup é um backup completo (*backup full*). Um tipo “3”, por exemplo, significa que é um backup de tablespace. O nó (*node*) NODE0000 é fixo para bases de dados não particionadas, que é o caso de base de dados criado utilizando as todas as versões exceto a DB2 Enterprise que possui a característica DPF. O nó do

catálogo também é fixo e possui o valor CATN0000. Leia maiores informações sobre este tópico no manual do DB2.

Quando muitos backups são realizados e armazenados no mesmo local, a data e a hora (timestamp) no final do arquivo são utilizadas para se distinguir as diversas imagens do backup. Como veremos na próxima seção, o comando RESTORE pode ser utilizado em conjunto com o timestamp para restaurar a base de dados de acordo com determinada data/hora.

## Quicklab #10 – Agendando um backup

### Objetivo

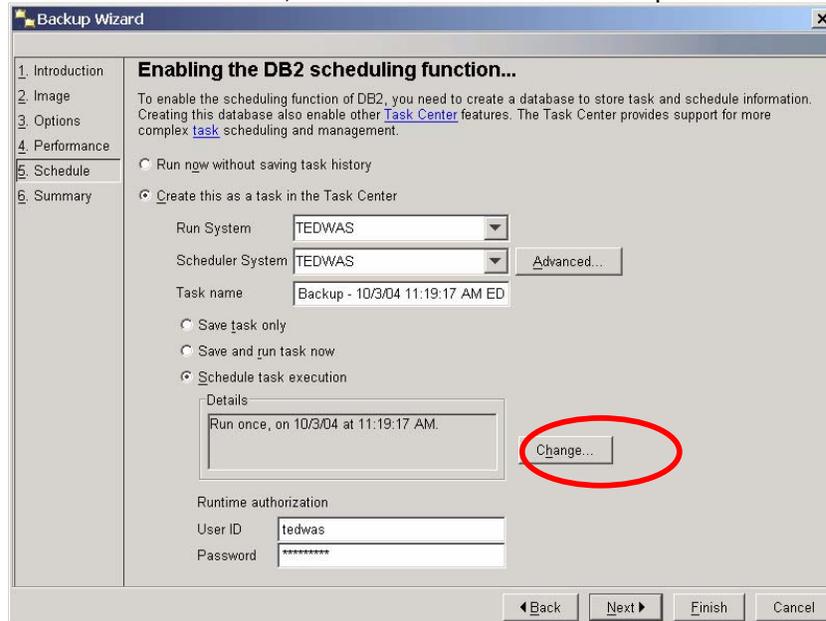
Embora o DB2 possa realizar automaticamente muitas atividades de manutenção de base de dados, você poderá customizar certas atividades. Neste Quicklab você criará um backup customizado para ser executado a noite na base de dados *EXPRESS*.

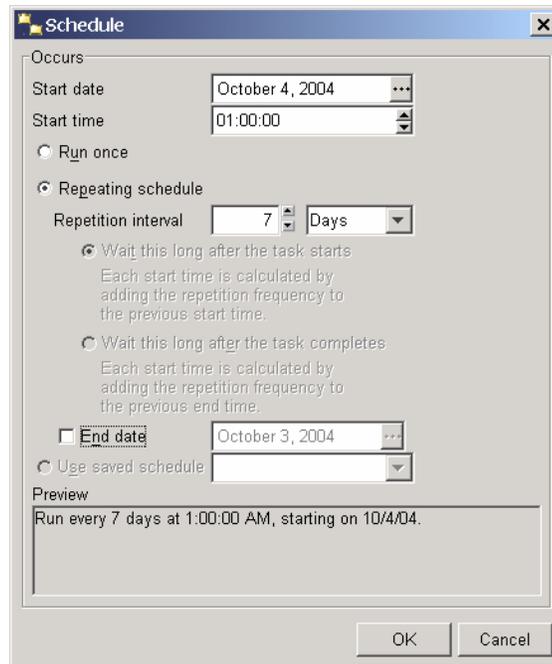
### Procedimentos

1. A partir da árvore de objetos do Control Center, escolha Control Center => All Databases. Clique com o botão da direita na base de dados *EXPRESS* e selecione o item Backup. O *Backup Wizard* será aberto.
2. A página *Introduction* mostra o estado atual da base de dados incluindo a data do último backup e o método de log utilizado. Clique no botão *Next* para ir para a próxima página.
3. Na página *Image* do wizard, selecione o destino da imagem do backup. Você deverá utilizar um drive diferente do que é utilizado para armazenar sua base de dados. Por enquanto, crie uma nova pasta no filesystem chamada: `C:\db2backup`, e especifique esta pasta como o destino do backup. No wizard, selecione the *File System* na lista do menu suspenso *Media Type*. Clique no botão *Add*, selecione a pasta que você acabou de criar e clique no botão *OK*. Clique no botão *Next* para ir para a próxima página.
4. Você pode explorar as páginas *Options* e *Performance* porém as opções default são suficientes para que o DB2 execute o backup de uma forma otimizada. Navegue para a página *Schedule*.
5. Na página *Schedule*, se o agendamento não estiver disponível, habilite o mesmo neste momento. Selecione “create the tools catalog” e escolha a base de dados *EXPRESS*. O catálogo de ferramentas (tools catalog) armazena meta-dados sobre todos as tarefas de agendamento. Clique no botão *OK* para continuar. Clique no botão *NEXT* para ir para a próxima página onde o catálogo de ferramentas (tools catalog) será criado.



6. Na página *Schedule*, escolha “Schedule task execution” para o backup ser executado todos os dias, iniciando a 1h da manhã. Clique no botão *Next*.





7. Na página *Summary*, você pode revisar as tarefas agendadas. Após a revisão, clique no botão *Finish* para criar as tarefas.
8. Abra o Task Center para visualizar e/ou modificar a nova tarefa de backup que você acabou de criar.

## 11.7 Recuperação de base de dados

Uma recuperação de base de dados implica em restaurar sua base de dados a partir do backup e/ou das logs. Se você somente restaura os dados a partir do backup, você estará recriando a base de dados a partir do momento no qual o backup foi realizado.

Se as logs (archival logging) foram habilitadas antes do backup, você pode restaurar a base de dados usando também as logs. Como veremos nesta seção, uma restauração “roll-forward” permite que você recupere a base de dados utilizando o backup e depois aplicando as logs (roll-forward), ou especificando um determinado ponto no tempo.

Verifique que o termo “recuperação” (“recovery”) é usado com frequência nesta seção porem o comando que deve ser executado é chamado ‘RESTORE’ (restaurar).

### 11.7.1 Tipos de Recuperação

Existem três tipos de recuperação:

- **Recuperação Rápida ou Reiniciada**

Assuma que você está trabalhando em um computador de mesa que está executando muitas transações importantes na base de dados DB2. De repente, houve uma queda de energia ou alguém, acidentalmente, retirou os cabos de energia: o que acontecerá na base de dados?

Na próxima vez que você reiniciar seu computador, e reiniciar o DB2, haverá uma recuperação rápida dos dados automaticamente. Na recuperação rápida, o DB2 executará automaticamente o comando RESTART DATABASE o qual irá ler e re-fazer ou desfazer as transações que estavam gravadas nos arquivos de log. Quando este comando finalizar, você terá a garantia que sua base de dados estará em estado consistente, o que significa que todos os comandos executados finalizados com COMMIT estão gravados e os que não foram finalizados com COMMIT serão revertidos ao estado anterior.

- **Imagem de Recuperação ou Versão de Recuperação**

Este tipo de recuperação indica que você recuperará os dados somente a partir da imagem do backup; o que significa que a sua base de dados voltará ao estado que estava no momento de realização do backup. Todas as transações efetivadas após a realização do backup serão perdidas.

- **Recuperação Roll-forward**

Com este tipo de recuperação, além de você executar o RESTORE da imagem do backup você também executa o comando ROLLFORWARD que aplica as logs que fazem com que você consiga recuperar a sua base de dados para qualquer ponto do passado (recover to a specified point in time). Este tipo de recuperação minimiza a perda de dados.

### 11.7.2 Restore da base de dados

Utilize o comando RESTORE para recuperar uma base de dados a partir da sua imagem de backup. A seguinte sintaxe é mais simples que pode ser usada para este comando:

```
RESTORE DATABASE <dbname> [from <path>] [taken at <timestamp>]
```

Por exemplo, se você tem o seguinte arquivo de imagem de backup da base de dados *sample*:

```

Alias      Instance      Year  Day  Minute  Sequence
|         |         |    |   |      |
SAMPLE.0.DB2INST.NODE0000.CATN0000.20060314131259.001
          |         |         |         |   |   |   |
          Type     Node     Catalog Node  Month Hour Second

```

Você poderia executar o seguinte comando:

```
RESTORE DB sample FROM <path> TAKEN AT 20060314131259
```

### 11.8 Outras operações que utilizam BACKUP e RESTORE

A seguinte lista mostra alguns exemplos do que você pode fazer com os comandos BACKUP e RESTORE. Por favor, consulte os manuais do DB2 para obter maiores detalhes.

- Realizar o backup de uma base de dados que possui uma instancia 32-bits e restaurar em uma instancia 64-bits.
- Restaurar sobre uma base de dados já existente.
- Utilizar a restauração redirecionada quando é necessário recuperar os dados em um sistema que usa um número diferente de discos do número de discos da origem do backup.
- Realizar o backup ou a restauração de somente uma tablespace ao invés da base de dados toda.
- Backups incrementais ou “delta” são permitidos; “delta” backups gravam somente os dados que estão diferentes da ultima imagem do backup; backups incrementais gravam todas as alterações e inserções que foram realizadas desde a ultima imagem de backup realizada.
- Backup a partir de “flash copy” (é necessário hardware específico).
- Recuperação de tabela removidas (dropped tables), caso esta opção esteja habilitada para determinada tabela.
- Backup a partir de uma plataforma (por exemplo Windows) e recuperação em outra plataforma (por exemplo Linux) não é possível. Utilize db2look e db2move nestes casos.



# 12

## Capítulo 12 – Tarefas de Manutenção

Este capítulo discute algumas tarefas necessárias para manter a sua base de dados bem administrada. A orientação geral do DB2 é automatizar a maiorias destas tarefas. A versão DB2 Express-C, assim como todas as versões do DB2, possibilitam esta automatização de tarefas. Esta capacidade de auto-gerenciamento é um grande beneficio para empresas pequenas e médias que não podem contratar um DBA para administrar seus servidores de dados. Por outro lado, se existe um DBA, ele ou ela podem ter mais tempo livre para desempenhar outras atividades que adicionem valor para a empresa, se programarem automaticamente as tarefas de manutenção de base de dados.

**Nota:**

Para maiores 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

Existem 3 principais tarefas de manutenção no DB2, como mostrado na figura 12.1: REORG, RUNSTATS e REBIND.

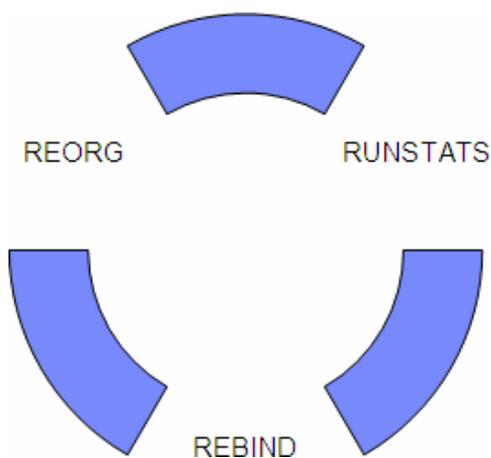


Figure 12.1 – Tarefas de Manutenção: REORG, RUNSTATS, REBIND

A figura 12.1 mostra que as tarefas de manutenção são realizadas de maneira circular. Se o REORG é realizado, é recomendado que também seja realizado o RUNSTATS, seguido do REBIND. Depois de algum tempo, as tabelas de uma base de dados serão modificadas devido as operações de UPDATE, DELETE e INSERT. E, neste momento, o ciclo deverá reiniciar com a tarefa de REORG.

### 12.1.1 O comando REORG

Com o decorrer do tempo, a medida que você executa operações de UPDATE, DELETE e INSERT na sua base de dados, seus dados começam a ficar fragmentados. O comando REORG recupera o espaço perdido e reorganiza os dados de uma maneira mais eficiente. As tabelas que são mais modificadas serão as maiores beneficiadas em uma operação de REORG. Você pode executar o REORG em índices ou tabelas, assim como o REORG pode ser executado on-line ou off-line.

O REORG off-line é mais rápido e mais eficiente mas, não permite o acesso a tabela. Enquanto o REORG on-line permite o acesso a tabela, consumindo muitos recursos do sistema, sendo melhor para utilização em tabelas pequenas.

**Sintaxe:**

```
REORG TABLE <tablename>
```

**Exemplo:**

```
REORG TABLE employee
```

O comando REORGCHK pode ser usado antes do REORG para determinar se uma tabela ou índice precisa ser desfragmentado.

### 12.1.2 O comando RUNSTATS

O otimizador do Db2 é o “cérebro” do DB2. Ele calcula o modo mais eficiente de acesso e recuperação de dados. O otimizador utiliza as estatísticas dos objetos da base de dados que estão armazenadas nas tabelas de catálogo do DB2 para maximizar a performance do banco de dados. Por exemplo, as tabelas do catálogo possuem estatísticas sobre quantas colunas existem nas tabelas, quantas linhas existem nas tabelas, quantos e quais são os índices disponíveis para uma tabela e assim por diante.

As informações estatísticas não são atualizadas dinamicamente. Isto acontece porque não seria conveniente realizar a atualização das estatísticas constantemente para cada operação realizada na base de dados visto que poderia afetar negativamente na performance do banco de dados. Devido a este fato, o DB2 possui o comando RUNSTATS que faz a atualização das estatísticas. É muito importante que as estatísticas estejam sempre atualizadas. O otimizados do DB2 pode fazer mudanças radicais no modo de acesso caso ele “pense” que uma tabela possui 1 linha ao invés de 1 milhão de linhas. Quando as estatísticas da base de dados estão atualizadas, o DB2 pode escolher um modo de acesso melhor. A frequência com que as estatísticas são recuperadas deve ser determinada de acordo com a frequência de que as tabelas são alteradas.

**Sintaxe:**

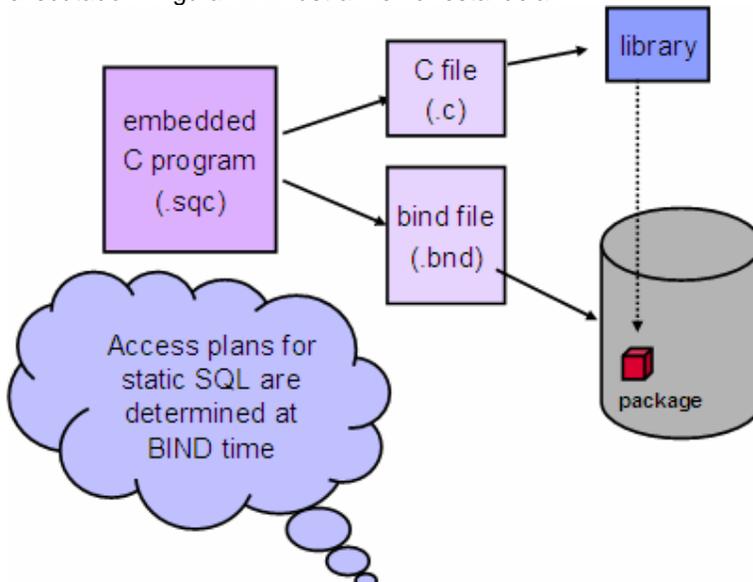
```
RUNSTATS ON TABLE <schema.tablename>
```

**Exemplo:**

```
RUNSTATS ON TABLE myschema.employee
```

### 12.1.3 BIND / REBIND

Depois de varias vezes de execução do comando RUNSTATS, nem todas as consultas utilizarão a ultima estatística. O plano de execução de um comando SQL é determinado quando você executa o comando BIND, sendo assim, as estatísticas utilizadas no momento do BIND podem não ser as mesmas do momento que o comando SQL é executado. A figura 12.2 ilustra melhor esta idéia.



**Figura 12.2 – Processo BIND no SQL estático**

Na figura 12.2 um programa C encapsulado (armazenado como um arquivo com extensão “sqc”) é pré-compilado. Depois da compilação, dois arquivos são gerados, um arquivo “.c” que contem o código C puro, e os comandos SQL comentados; e, outro arquivo “.bnd” que contem as instruções SQL. O arquivo C com a extensão “.c” é compilado normalmente com um compilador C, criando uma “library”, como é mostrado no topo do desenho a direita. O arquivo “.bnd” é ligado (bound), de uma maneira parecida, gerando um pacote que é armazenado na base de dados. O processo de “bind” é equivalente ao processo de compilação porem o “bind” compila instruções SQL, incluindo o melhor plano de acesso determinado no momento em que aquela instrução foi gerada e armazenando na base de dados como um pacote.

Agora, o que acontece se 1 milhão de linhas forem inseridas na tabela utilizando um programa C encapsulado ou embutido? Depois da inserção, se for executado o comando RUNSTATS, as estatísticas estarão atualizadas, porem, o pacote não será atualizado automaticamente com relação ao método de acesso. Neste caso, o comando db2rbind pode ser usado para realizar o “rebind” de todos os pacotes, fazendo com que eles contem as informações atualizadas das tabelas.

**Sintaxe:**

```
db2rbind database_alias -l <logfile>
```

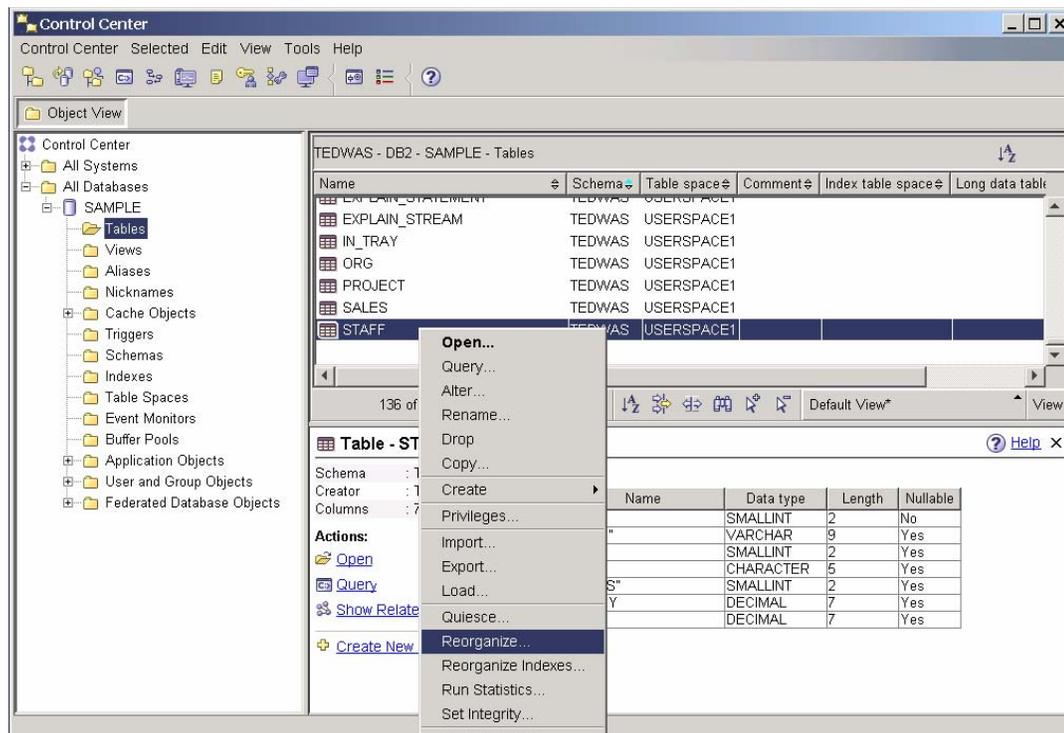
**Exemplo:**

Para realizar o “rebind” de todos os pacotes da base de dados *sample* e armazenar a saída no arquivo *mylog.txt*, execute o comando:

```
db2rbind sample -l mylog.txt
```

**12.1.4 Tarefas de Manutenção no Control Center**

A partir do Control Center você pode executar o REORG e o RUNSTATS conforme mostra a figura 12.3 shows.



**Figura 12.3 – Executando REORG e RUNSTATS a partir do Control Center**

Escolha a tabela onde você gostaria de executar a operação, selecione-a com o botão direito do mouse e escolha Reorganize (para REORG) ou Run Statistics (para RUNSTATS).

**A visão operacional da base de dados**

Quando você escolhe um “database” aparecerá a visão operacional desta base de dados do lado direito da janela do Control Center, mostrando informações como o tamanho da base de dados, quando foi realizado o backup pela ultima vez, quais atividades de manutenção automática estão agendadas, etc. Esta visão permite que você consiga visualizar rapidamente as atividades de manutenção necessárias na sua base de dados, conforme você pode verificar na figura 12.4.

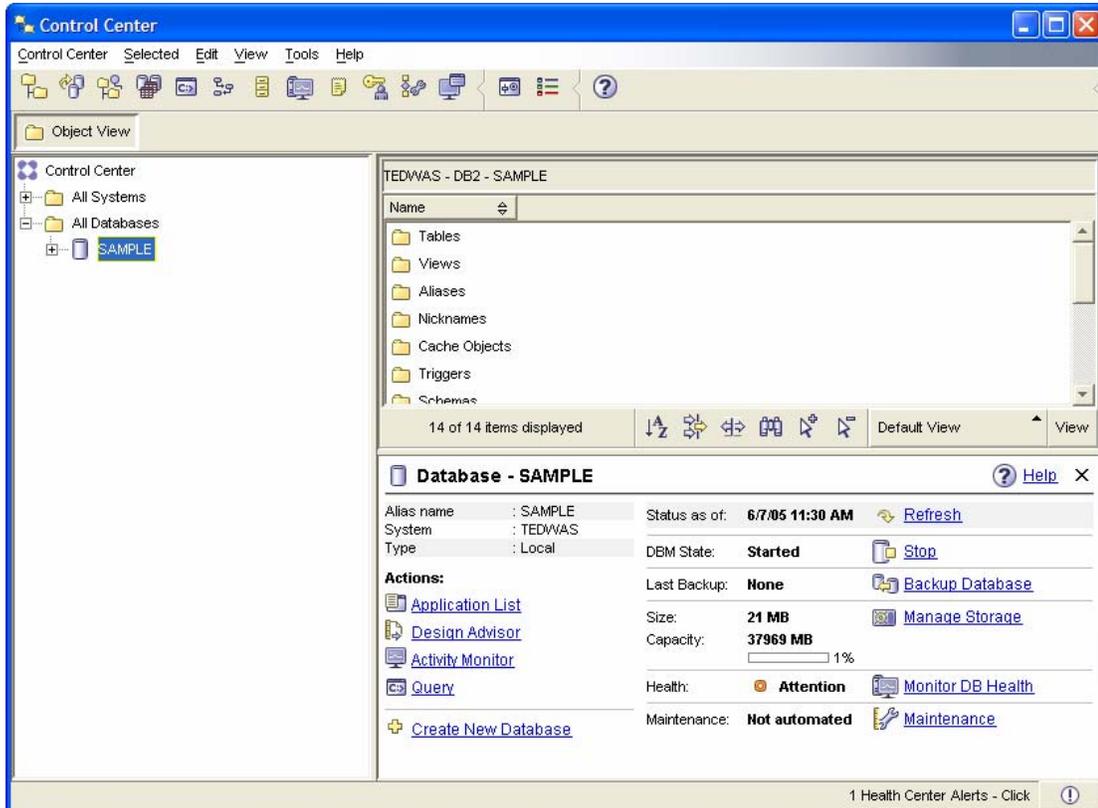


Figura 12.4 – A visao operacional da base de dados a partir do Control Center

## 12.2 Opções de Manutenção

Existem três maneiras de realizar as tarefas de manutenção:

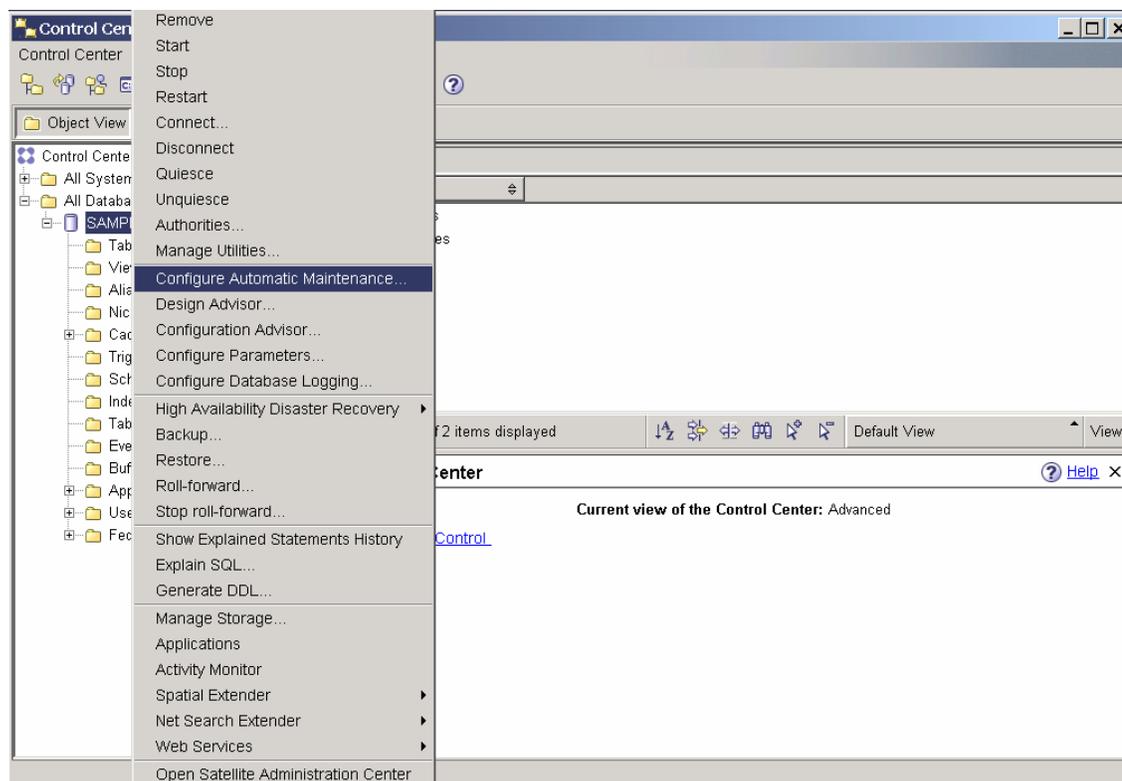
1. **Manutenção Manual**  
Você realiza as tarefas de manutenção manualmente, de acordo com a necessidade.
2. **Criar scripts para realizar as tarefas de manutenção**  
Você cria scripts com os comandos de manutenção e os agenda para execução regularmente.
3. **Manutenção Automática**  
Você deixa que o DB2 realize as tarefas de manutenção (REORG, RUNSTATS, BACKUP)

Nesta seção nos concentraremos nas atividades de manutenção.

A manutenção automática consiste no seguinte:

- O usuário define uma “janela de manutenção” onde as tarefas podem ser executadas com mínimo impacto a base de dados. Por exemplo, se o sistema tem a menor utilização no Domingo das 2h as 4h das madrugada, então este período poderia ser utilizado como uma “janela de manutenção”.
- Existem duas “janelas de manutenção”: uma para as operações on-line e outra para as operações off-line.
- O DB2 realizará estas operações automaticamente somente quando for necessário e dentro da “janela de manutenção”.

A partir do Control Center, você pode chamar “Configure Automated Maintenance Wizard” como mostrado na Figura 12.5.



**Figura 12.5 – Chamando o “Configure Automated Maintenance Wizard”**

A Figure 12.6 mostra o “Configure Automated Maintenance Wizard”.

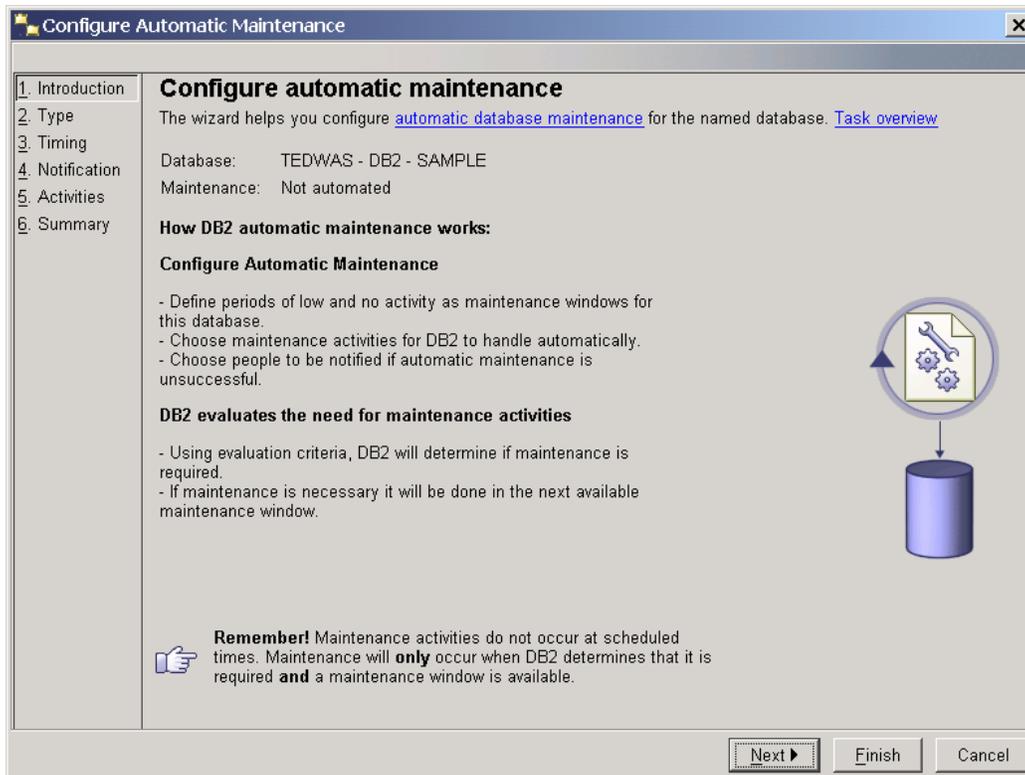


Figura 12.6 – Configure Automated Maintenance Wizard

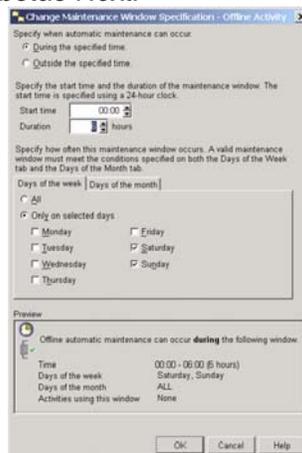
## Quicklab #11 – Configurando a manutenção automática

### Objetivo

Neste Quicklab, em poucos passos você irá configurar tarefas de manutenção automática utilizando a base de dados SAMPLE.

### Procedimentos

1. A partir da árvore de objetos do Control Center, clique com o botão da direita na base de dados SAMPLE e selecione o item *Configure Automatic Maintenance*.
2. A tela *Introduction* do wizard mostrará o cenário atual de manutenção. Se você criou a base de dados com a opção “automated maintenance” então todas as tarefas de manutenção já estão configuradas. Você poderá usar o wizard para re-configurar as opções de manutenção. Clique no botão *Next* para mover para a próxima página.
3. Na tela *Type* do wizard você pode escolher desabilitar ou mudar as tarefas de manutenção. Selecione “change the current automated maintenance settings” para alterá-las e clique no botão *Next*.
4. A tela *Timing* do wizard pede para você especificar a janela de manutenção. Configure a janela off-line para todos os Sábados e Domingos a partir das 6h, conforme mostrado na figura abaixo. Clique no botão *Change* ao lado da janela de manutenção off-line e escolha o horário desejado. Depois disto, clique no botão *OK* para retornar ao wizard. Não é necessário definir a janela on line visto que ela pode ocorrer a qualquer momento. Clique no botão *Next*.



5. Na tela *Notification* você pode definir um contato para ser acionado no caso de falha na tarefa de manutenção. Pule este passo por enquanto.
6. Na tela *Activities*, você pode escolher as atividades que deseja automatizar assim como se quer ser notificado ou não em caso de falha. Neste exemplo, assegure-se que todas as atividades estão marcadas como “*Automate*” e que não existe nenhuma marcação em *Notify*. . Clique no botão *Next*.
7. Antes de continuar, você deve configurar o local onde o backup será armazenado. O ideal é armazenar o backup em disco diferente do que a base de dados está salva, para evitar problemas em caso de perda do disco. A partir da janela *Activities*, selecione a opção *Backup* e clique no botão *Configure Settings*.
8. Na aba *Backup Criteria* da janela de diálogo *Configure Settings*, escolha a opção *Balance Database Recoverability with Performance*. Na aba *Backup Location*, selecione local onde você está armazenando o backup e clique no botão *Change*. Especifique uma pasta diferente para armazenar o backup (certifique-se que há espaço suficiente para armazenar o arquivo de backup). Na aba *Backup Mode*, verifique se a opção *Offline Backup* está selecionada. Clique no botão OK para fechar a aba *Backup Criteria*. Clique no botão *Next*.
9. A tela *Summary* do *Configure Automated Maintenance* wizard contém um resumo das opções que foram selecionadas. Clique no botão *Finish* para aceitar as opções e implementá-las.



# 13

## Capítulo 13 – Concorrência e Bloqueios

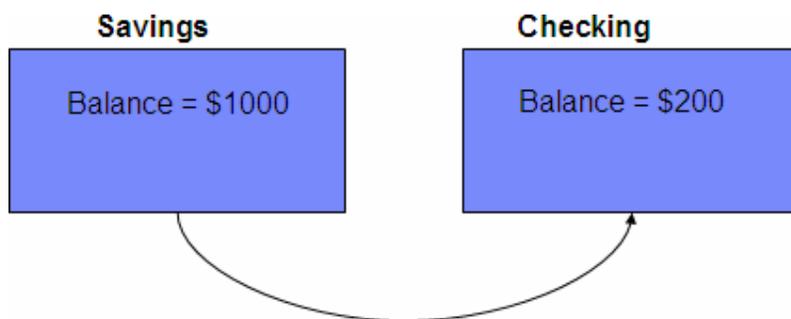
Este capítulo explica como permitir que múltiplos usuários acessem uma base de dados ao mesmo tempo sem que um interfira na ação do outro e mantendo os dados consistentes. Nós iremos mostrar os conceitos de transação, concorrência e bloqueio (locking).

### Nota:

Para mais informação 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). Uma transação finaliza com a instrução COMMIT, o que significa também que uma nova transação pode começar. A figura 13.1 mostra um exemplo de transação.



Transfer \$100 from Savings to Checking:

- Debit \$100 from Savings account
- Credit \$100 to Checking account

**Figura 13.1 – Um exemplo de transação**

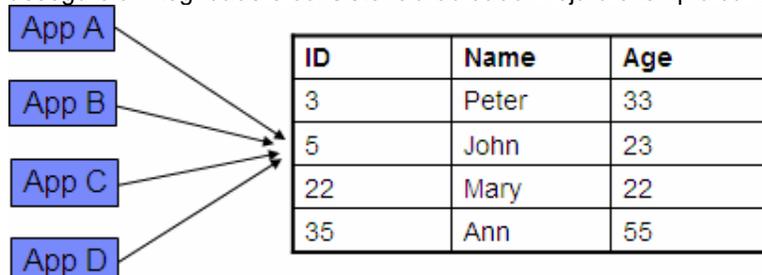
Na figura 13.1 mostra um exemplo onde você quer transferir 100 dólares da sua conta poupança para a sua conta corrente. A seguinte seqüência de eventos é necessária para que esta transação se complete:

Débito de US\$100 da conta de poupança  
Crédito de US\$100 na conta corrente

Se a seqüência de eventos acima não for tratada como uma unidade de trabalho, ou seja, uma transação, imagine o que poderia acontecer se houvesse uma queda no sistema depois de ter sido realizado o débito da sua conta poupança e antes de ser creditado os US\$100 na sua 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 mesmo objeto da base de dados. O acesso ao dado deve ser coordenado corretamente e de uma maneira transparente utilizando um mecanismo que assegure a integridade e consistência do dado. Veja o exemplo da figura 13.2.



**Figura 13.2 – Um exemplo de concorrência e da necessidade de controle do acesso aos dados**

Na figura 13.2, existem quatro aplicações, App A, App B, App C, e App D que estão tentando acessar a mesma linha de uma tabela (linha 2). Se não houvesse nenhum mecanismo de controle de concorrência, todas as aplicações poderiam executar operações na mesma linha da tabela. Se assumirmos que todas as aplicações estão atualizando a coluna *Age* (Idade) na linha 2, com diferentes valores, a aplicação que fizer a atualização por último agirá como a “vencedora” nesta situação. Com este exemplo conseguimos visualizar como é necessário que exista um controle de transações que garantam o resultado final. Este controle de concorrência é baseado em bloqueios (locks).

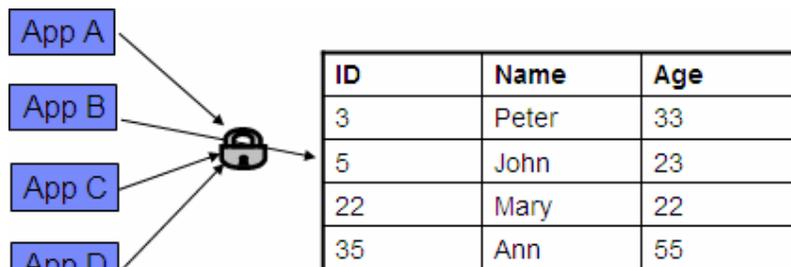
Os conceitos de concorrência e bloqueio caminham juntos. O bloqueio para, temporariamente, outras operações até que a primeira operação termine. Quanto mais bloqueios houver no sistema, menor será a concorrência. Por outro lado, quanto menor o número de bloqueios, maior a probabilidade de existir concorrência.

Os bloqueios acontecem automaticamente de acordo com a necessidade de garantir a segurança da transação; os bloqueios são liberados quando a transação termina (usando

os comandos COMMIT ou ROLLBACK). Os bloqueios podem acontecer nas tabelas ou nas linhas de uma tabela. Existem dois tipos básicos de bloqueio:

- Bloqueios Divididos (Share locks - S locks) – acontecem quando uma aplicação quer realizar a leitura e impede outras aplicações de fazer a alteração (UPDATE) da mesma linha.
- Bloqueios Exclusivos (Exclusive locks X locks) – acontecem quando uma aplicação quer atualizar, inserir ou remover uma linha.

Verifique a Figura 13.3, similar a Figura 13.2 porém mostra a ocorrência do bloqueio.



**Figura 13.3 – Exemplo de concorrência e da necessidade dos bloqueios**

Tomando como exemplo a figura 13.2, se a aplicação B é a primeira a acessar a linha 2 e esta realizando um UPDATE, a aplicação B mantém um bloqueio X na linha. Quando as aplicações A, C e D tentam acessar a mesma linha, elas não poderão realizar o UPDATE por causa do bloqueio X. Este controle permite a consistência e a integridade do dado.

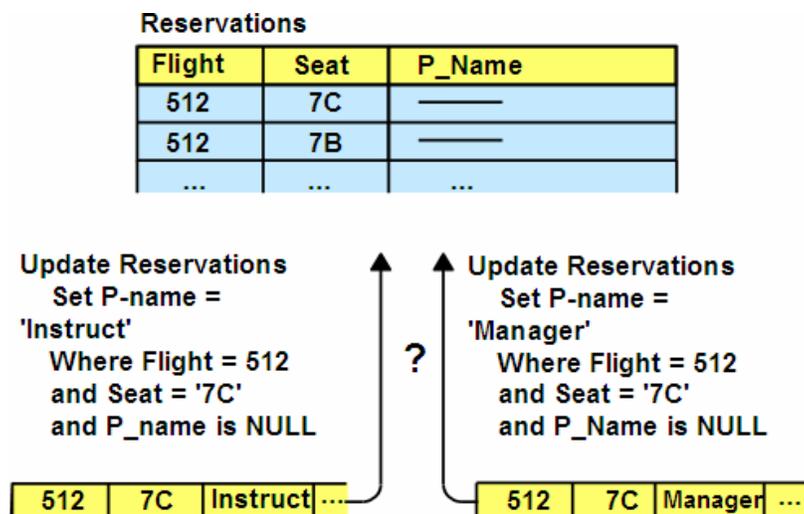
### 13.3 Problemas ocasionados pela falta de controle de concorrência

Sem o controle de concorrência, os seguintes problemas podem acontecer:

- ▶ Operações de atualização (UPDATE) perdidas
- ▶ Leitura de dados que não tenham sido finalizados (Uncommitted read)
- ▶ Leitura de dados não repetidos (Non-repeatable read)
- ▶ Leituras Fantasmas (Phantom read)

#### 13.3.1 Operações de atualização (UPDATE) perdidas

Operações de atualização (UPDATE) perdidas é um problema parecido com o que foi explanado anteriormente nesta seção onde a aplicação que está realizando o ultimo UPDATE é a “campeã”.



**Figura 13.4 – 1 Operação de atualização (UPDATE) perdida**

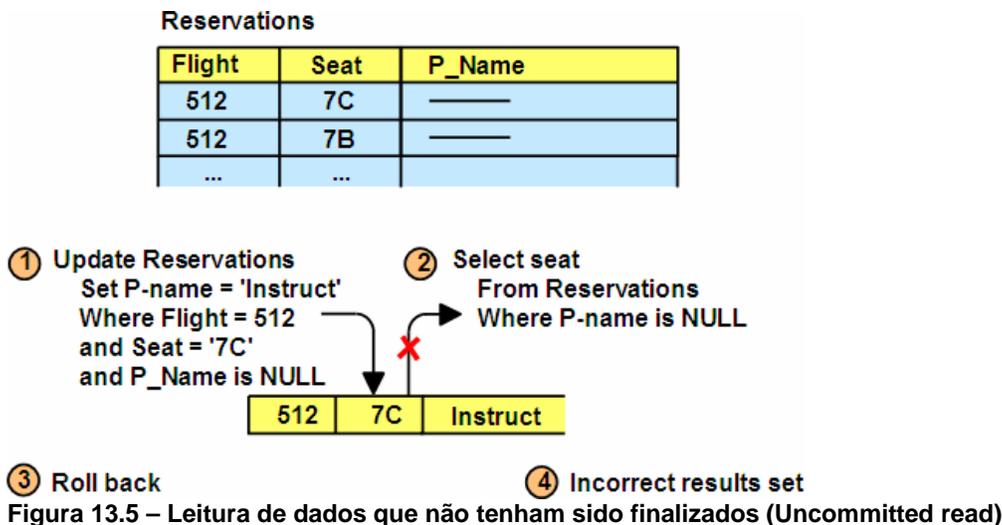
Na Figura 13.4 existem duas aplicações tentando atualizar a mesma linha. A aplicação da esquerda é chamada App1, e a da direita App2. A seqüência dos eventos é:

1. App1 atualiza a linha
2. App2 atualiza a mesma linha
3. App1 executa o COMMIT
4. App2 executa o COMMIT

A atualização da App1 é perdida quando App2 realiza a atualização, por isto este problema é chamado “Operação de atualização (UPDATE) perdida”.

### 13.3.2 Leitura de dados que não tenham sido finalizados (Uncommitted read)

A leitura de dados que não tenham sido finalizados (Uncommitted read) ou “leitura suja” (“dirty read”) permite que uma aplicação leia dados que não tenham sido finalizados pelo comando COMMIT e por isto este dado não é consistente.



A Figura 13.5 mostra a seguinte seqüência de eventos:

1. App1 atualiza uma linha
2. App2 lê o novo valor da linha
3. App1 executa o ROLLBACK (não finaliza a atualização da linha, voltando o dado ao estado anterior)

A aplicação App2 esta lendo dados que não foram finalizados pelo comando COMMIT, portanto este dado é invalido, por isto este problema é chamado “Leitura de dados que não tenham sido finalizados (Uncommitted read)”.

### 13.3.3 Leitura de dados não repetidos (Non-repeatable read)

A leitura de dados não repetidos (Non-repeatable read) significa que você não pode obter o mesmo resultado depois de realizar 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 de dados não repetidos (Non-repeatable read)**

Na Figura 13.6, considere que você está tentando fazer uma reserva de um voo de Dallas para Honolulu. A seqüência de eventos é:

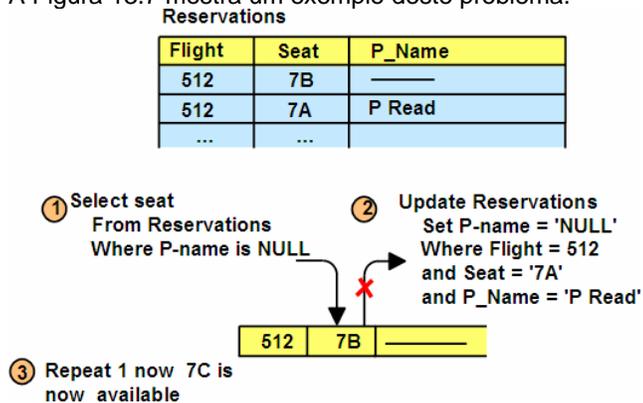
1. App1 abre um cursor (também conhecido como result set) e obtém o que você vê na Figura 13.6
2. App2 apaga a linha que está armazenada no cursor (por exemplo, a linha com destino "San Jose")
3. App2 executa um COMMIT
4. App1 fecha e re-abre o cursor

Neste caso, como App1 não vai obter o mesmo dado mesmo realizando uma leitura repetida, ela não pode reproduzir estes dados; é por este motivo que este problema é chamado "Leitura de dados não repetidos (Non-repeatable read)".

### 13.3.4 Leituras Fantasmas (Phantom read)

O problema de Leituras Fantasmas (Phantom read) é parecido ao problema de Leitura de dados não repetidos (Non-repeatable read), a diferença é que em *fetches* subseqüentes você pode obter linhas adicionais, ao invés de um menor número de linhas.

A Figura 13.7 mostra um exemplo deste problema.



**Figura 13.7 – Leitura Fantasma (Phantom read)**

A Figura 13.7 mostra a seguinte seqüência de eventos:

1. App1 abre um cursor
2. App2 adiciona uma linha na base de dados a qual é qualificada pelo cursor
3. App2 executa o COMMIT
4. App1 fecha e re-abre o cursor

Neste caso, a App1 não irá recuperar os mesmos dados se realizar uma leitura repetida mas sim recuperará mais dados, por isto este problema é chamado "Leitura Fantasma (Phantom read)".

## 13.4 Níveis de Isolamento

Você pode pensar que níveis de isolamento são políticas de bloqueamento onde, dependendo do nível de isolamento escolhido, você tem um comportamento diferente da base de dados com relação aos bloqueios de dados.

O DB2 possui diferentes níveis de isolamento de dados:

- ▶ Leitura de Dados não finalizados (Uncommitted Read UR)
- ▶ Estabilidade de Cursor (Cursor Stability CS)
- ▶ Estabilidade de Leitura (Read Stability RS)
- ▶ Leitura Repetida (Repeatable Read RR)

### 13.4.1 Leitura de Dados não finalizados (Uncommitted Read UR)

A Leitura de Dados não finalizados (Uncommitted Read UR) é também chamada de Leitura Suja (dirty read). É o menor nível de isolamento e fornece o maior nível de concorrência. Não há bloqueio de linha quando são executadas operações de leitura, exceto quando existem operações de “DROP” ou “ALTER TABLE” (remoção ou alteração na estrutura da tabela). As operações de UPDATE são executadas da mesma maneira que no nível de isolamento “Estabilidade de Cursor (Cursor Stability CS)”.

Os problemas que podem ocorrer com este nível de isolamento são:

- ▶ Leitura de Dados não finalizados (Uncommitted read)
- ▶ Leitura de dados não repetidos (Non-repeatable read)
- ▶ Leitura Fantasma (Phantom read)

O problema que é evitado neste nível de isolamento é:

- ▶ Perda na Atualização (Loss of update)

### 13.4.2 Estabilidade de Cursor (Cursor Stability CS)

Estabilidade de Cursor (Cursor Stability CS) é o nível de isolamento padrão pois fornece o menor nível de bloqueamento. Basicamente, neste nível de isolamento, a linha corrente do cursor é bloqueada. Se a linha é somente necessária para leitura, a linha é bloqueada até que ser capturada (fetched) ou até que a transação termine. Se a operação que a ser realizada é o UPDATE, a linha é bloqueada até que a transação termine.

Os problemas que podem ocorrer com este nível de isolamento são:

- ▶ Leitura de dados não repetidos (Non-repeatable read)
- ▶ Leitura Fantasma (Phantom read)

Os problemas que são evitados neste nível de isolamento são:

- ▶ Perda na Atualização (Loss of update)
- ▶ Leitura de Dados não finalizados (Uncommitted read)

### 13.4.3 Estabilidade de Leitura (Read Stability RS)

Com a Estabilidade de Leitura (Read Stability RS), todas as linhas que fazem parte da operação solicitada são bloqueadas até que a transação termine. Todas as linhas qualificadas por determinado cursor são bloqueadas até que a transação termine. Por exemplo, se você tem uma tabela que contém 10.000 linhas e executa uma operação de consulta que retorna 10 linhas então, estas 10 linhas da tabela serão bloqueadas. O nível de isolamento “Estabilidade de Leitura (Read Stability RS)” usa um grau moderado de bloqueio.

Os problemas que podem ocorrer com este nível de isolamento são:

- ▶ Leitura Fantasma (Phantom read)

Os problemas que são evitados neste nível de isolamento são:

- ▶ Perda na Atualização (Loss of update)
- ▶ Leitura de Dados não finalizados (Uncommitted read)
- ▶ Leitura de dados não repetidos (Non-repeatable read)

### 13.4.4 Leitura Repetida (Repeatable Read RR)

O nível de isolamento “Leitura Repetida (Repeatable Read RR)” é o maior nível de isolamento pois provê o maior grau de bloqueamento e o menor nível de concorrência. Os bloqueios são realizados em todas as linhas processadas; sendo assim, o bloqueio ocorre em todas as linhas, mesmo que as mesmas não façam parte do resultado final da consulta. Nenhuma outra aplicação poderá realizar operações de UPDATE, DELETE ou INSERT até que a primeira transação finalize o seu processamento. Leituras repetidas garantirão que a mesma consulta retorne sempre o mesmo resultado.

- ▶ Não há nenhum problema neste nível de isolamento.

Os problemas que são evitados neste nível de isolamento são:

- ▶ Perda na Atualização (Loss of update)
- ▶ Leitura de Dados não finalizados (Uncommitted read)
- ▶ Leitura de dados não repetidos (Non-repeatable read)
- ▶ Leitura Fantasma (Phantom read)

### 13.4.5 Comparando os níveis de isolamento

A Figura 13.8 compara os diferentes níveis de isolamento em relação ao “fetch”. Você pode visualizar que quando o nível de isolamento é “Leitura de Dados não finalizados (Uncommitted Read UR)” não existe nenhum bloqueio (lock). O nível “Estabilidade de Cursor (Cursor Stability CS)” realiza o bloqueio de 1 linha, no momento do fetch, mas o libera assim que é necessário fazer o fetch da linha 2, e assim por diante. Nos níveis de isolamento “Estabilidade de Leitura (Read Stability RS)” e “Leitura Repetida (Repeatable Read RR)”, qualquer linha necessária no fetch será bloqueada, e o bloqueio não será liberado até o final da transação (quando for realizado o COMMIT).

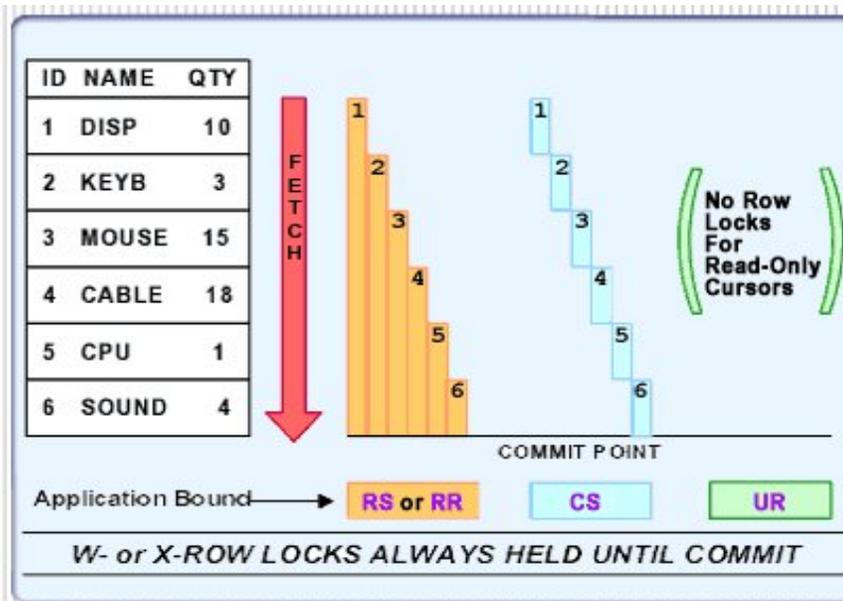


Figura 13.8 – Comparando os níveis de isolamento em relação ao fetch

### 13.4.6 Definindo o nível de isolamento

O nível de isolamento pode ser especificado em vários níveis:

- ▶ Sessão (aplicação)
- ▶ Conexão
- ▶ Instrução

O nível de isolamento é normalmente definido no nível de Aplicação ou Sessão. Se nenhum nível de isolamento for especificado, o padrão será utilizar a estabilidade de cursor. Por exemplo, a tabela 13.1 mostra os níveis de isolamento possíveis para programas .NET ou JDBC e como estas propriedades funcionam no nível de isolamento do DB2.

DB2	.NET	JDBC
Uncommitted Read (UR)	ReadUncommitted	TRANSACTION_READ_UNCOMMITTED
Cursor Stability (CS)	ReadCommitted	TRANSACTION_READ_COMMITTED
Read Stability (RS)	RepeatableRead	TRANSACTION_REPEATABLE_READ
Repeatable Read (RR)	Serializable	TRANSACTION_SERIALIZABLE

---

**Tabela 13.1 – Comparação da Terminologia de nível de isolamento**

O nível de isolamento pode ser aplicado através da cláusula WITH {isolation level}. Por exemplo:

```
SELECT ... WITH {UR | CS | RS | RR}
```

Exemplo:

Uma aplicação precisa fazer a contagem do número de registros em uma tabela e necessita de uma boa performance nesta consulta. O nível de isolamento “Estabilidade de Cursor (Cursor Stability CS)” pode ser usado nesta instrução SQL:

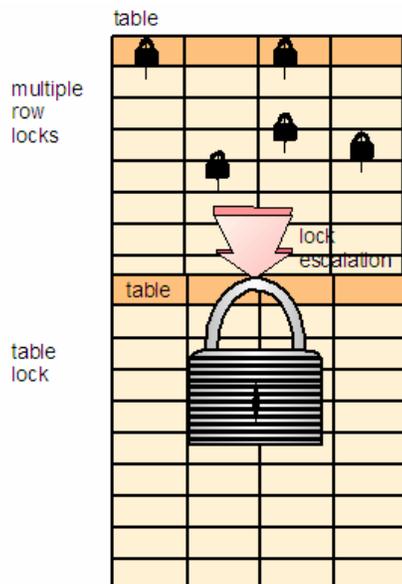
```
SELECT COUNT(*) FROM tabl WITH UR
```

Em instruções SQL embutidas (embedded SQL), o nível de isolamento é definido no momento do “bind”; em instruções SQL dinâmicas, o nível de isolamento é definido no momento que se executa a instrução.

A escolha do nível de isolamento depende do comportamento da sua aplicação. Se sua aplicação não precisa de instruções de “count” exatas, como no exemplo acima, escolha o nível de isolamento “Leitura de Dados não finalizados (Uncommitted Read UR)”. Se sua aplicação requer um controle rígido dos dados, escolha o nível de isolamento “Leitura Repetida (Repeatable Read RR)”.

### 13.5 Escalada do Bloqueio (Lock Escalation)

Cada bloqueio realizado pelo DB2 consome memória. Quando o otimizador define que é melhor realizar um bloqueio na tabela toda, ao invés de bloquear várias linhas, acontece a escalada do bloqueio (lock escalation), conforme ilustra a figura 13.9.



**Figura 13.9 – Escalada do Bloqueio (Lock Escalation)**

Existem dois parâmetros principais de banco de dados relacionados a escalada do bloqueio (Lock Escalation):

- ▶ **LOCKLIST** – Uma quantidade de memória (em páginas de 4k) é reservada para gerenciar os bloqueios em todas as aplicações. O padrão é 50 \* 4k (200K) páginas no Windows.
- ▶ **MAXLOCKS** – é o maior percentual que um bloqueio pode usar. O default é 22% do total de memória.

Sendo assim, se o valor padrão for utilizado, relacionados a escalada do bloqueio (Lock Escalation) ocorre quando uma transação necessita de mais de 44k de memória para bloqueio  $200\text{K} * 22\% = 44\text{K}$ . Se a escalada do bloqueio ocorrer com frequência, aumente o valor do parâmetro MAXLOCKS. Escaladas do bloqueio não é um bom sinal pois degradam a performance e reduzem a concorrência. O arquivo de diagnostico do DB2 (db2diag.log, o qual normalmente é gravado na pasta the C:\Program Files\IBM\SQLLIB\DB2) pode ser usado para verificar quando está ocorrendo a escalada do bloqueio (Lock Escalation).

### 13.6 Monitoração de Bloqueio

Você pode monitorar a ocorrência de bloqueios utilizando a aplicação DB2 lock snapshots. Para habilitar o DB2 lock snapshots, utilize o seguinte 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 de saída da aplicação lock snapshot.

```

Application Lock Snapshot

Snapshot timestamp           = 11-05-2002 00:09:08.672586
Application handle          = 9
Application ID              = *LOCAL.DB2.0089C5050843
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                   = 0x050007000480010000000000052
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
Table Name                  = T2
Mode                        = X

```

Figura 13.9 – Lock Snapshot

### 13.7 Bloqueio em Espera (Lock wait)

Quando duas ou mais aplicações precisam realizar uma operação no mesmo objeto, uma delas terá que esperar para acionar o bloqueio. De acordo com o padrão, uma aplicação poderá esperar indefinidamente. O tempo que uma aplicação espera por um bloqueio é controlado pelo parâmetro LOCKTIMEOUT da configuração do banco de dados. O valor padrão deste parâmetro é -1 (espera infinita).

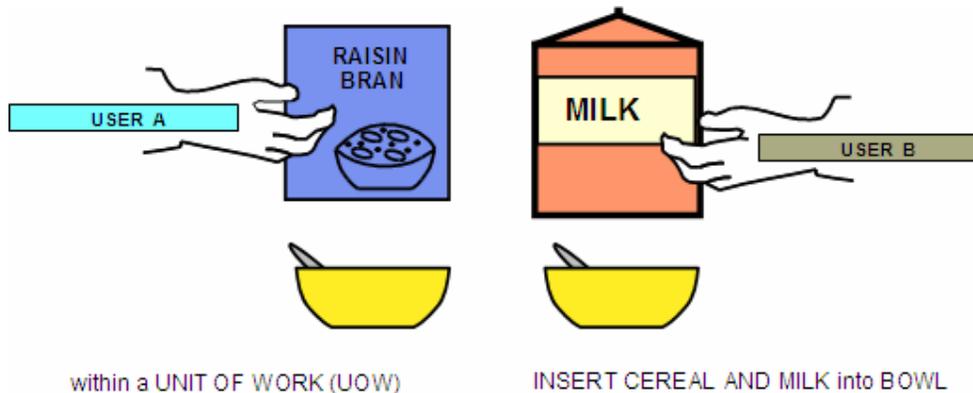
O parâmetro CURRENT LOCK TIMEOUT pode ser usado para definir o tempo de espera de bloqueio para uma determinada conexão. O valor padrão deste parâmetro é o mesmo valor do parâmetro LOCKTIMEOUT. Utilize a instrução SET LOCK TIMEOUT para mudar este valor. Uma vez que este parâmetro tenha sido alterado para determinada sessão ele persistirá para todas transações desta sessão:

Exemplo:

```
SET LOCK TIMEOUT=WAIT n
```

### 13.8 Causas e como descobrir um deadlock

Um deadlock ocorre quando duas ou mais aplicações conectadas na mesma base de dados esperam indefinidamente por um recurso. A espera não é resolvida porque cada aplicação “prendendo” um recurso que a outra precisa. Deadlocks, na maioria das vezes, são problemas relacionados ao desenho da aplicação. A Figura 13.10 ilustra um deadlock.



**Figura 13.10 – Cenário de Deadlock**

Na Figura 13.10, o usuário A está “segurando” os cereais e não irá “soltar” até que ele obtenha o leite. Por outro lado, o usuário B está “segurando” o leite e não irá “soltar” enquanto não receber o cereal. Neste caso, temos uma situação de deadlock.

Para simular uma situação de deadlock no DB2, realize os seguintes passos:

1. Abra duas janelas do “DB2 Command Windows” (as quais iremos chamar de “CLP1” e “CLP2” respectivamente), representando duas diferentes aplicações conectando-se a mesma base de dados.
2. A partir de CLP1 execute os seguintes comandos:

```
db2 connect to sample
db2 +c update employee set firstnme = 'Mary' where empno =
'000050'
```

Primeiro estamos conectando a base de dados SAMPLE, e realizando uma instrução UPDATE na linha que possui “empno = 50000” na tabela employee. A opção “+c” indica que não queremos que o DB2 execute a instrução COMMIT automaticamente. Estamos fazendo isto para que possamos simular a espera por um bloqueio.

3. A partir de CLP2 execute os seguintes comandos:

```
db2 connect to sample
db2 +c update employee set firstnme = 'Tom' where empno =
'000030'
```

Na janela CLP2, que representa a segunda aplicação, nós também estamos nos conectando a base de dados SAMPLE, mas estamos atualizando outra linha na mesma tabela employee.

4. A partir de CLP1 execute:

```
db2 +c select firstnme from employee where empno = '000030'
```

Após pressionar Enter, execute a instrução acima, o SELECT pode parecer “travado”. Na verdade, ele não está travado mas sim esperando a liberação do bloqueio que foi necessário quando a aplicação CLP2 executou o comando mostrado no passo 3. neste momento, se o valor de LOCKTIMEOUT foi deixado no padrão -1, a aplicação CLP1 irá esperar indefinidamente.

5. A partir de CLP2 execute:

```
db2 +c select firstnme from employee where empno = '000050'
```

Executando a instrução SELECT acima não estamos criando um deadlock. Esta instrução SELECT também parecerá travada, pois está esperando a liberação do bloqueio exclusivo que foi gerado pela aplicação CLP1 no passo 2.

No cenário de deadlock acima, O DB2 irá verificar a configuração do parâmetro DLCHKTIME. Este parâmetro define o intervalo de tempo para os deadlocks. Por exemplo, se este parâmetro estiver definido para 10 segundos, o DB2 irá verificar a cada 10 segundos se existe a ocorrência do deadlock. Se um deadlock acontecer, o DB2 utilizará um algoritmo interno para determinar qual das duas transações deverá ser revertida (rolled back) e qual continuará.

Se você está com problemas decorrentes de deadlocks, você deve re-examinar suas transações e verificar se há possibilidade de alterá-las.

### 13.9 Boas práticas de concorrência e bloqueio

Os itens abaixo são dicas que podem ser seguidas para obter a melhor situação de concorrência:

1. Mantenha as transações o mais leve possível. Isto pode ser atingido se você incluir freqüentes instruções COMMIT (inclusive para transações somente de leitura).

2. Armazene logs das transações somente quando necessário.

3. Execute operações de limpeza, usando:
 

```
ALTER TABLE ACTIVATE NOT LOGGED INITIALLY WITH EMPTY TABLE
```

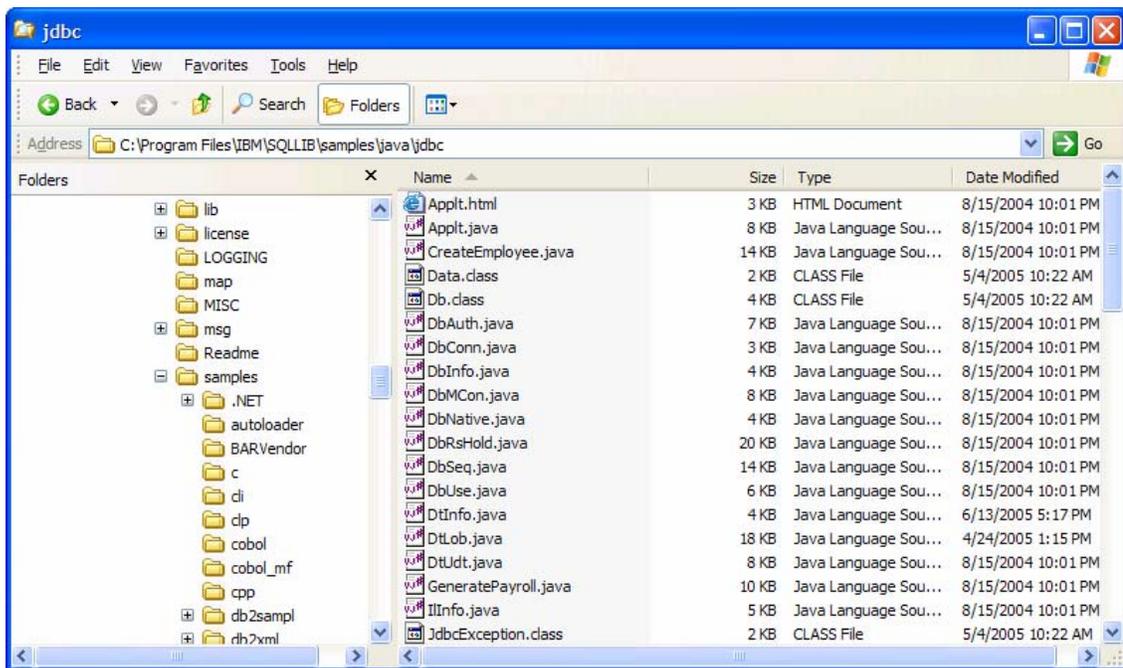
4. Realize operações de modificação em grupo (batch). Por exemplo:
 

```
DELETE FROM (
  SELECT * FROM tedwas.t1 WHERE c1 = ... FETCH FIRST 3000 ROWS
  ONLY)
```

5. Utilize as características de concorrência encontradas nas ferramentas do DB2.
6. Defina o parâmetro de nível de isolamento LOCKTIMEOUT (os tempos sugeridos são entre 30 e 120 segundos). Não o deixe com o valor padrão -1. Você pode também utilizar “session-based lock timeout” (tempo de sessão baseado nos bloqueios).
7. Não recupere mais dados do que o necessário. Por exemplo, use “FETCH FIRST n ROWS ONLY” nas suas instruções SELECT.

**PARTE III – CONHECENDO O DB2:  
DESENVOLVIMENTO DE  
APLICAÇÕES**

Na parte II deste livro, discutimos em profundidade objetos de aplicação da base de dados como stored procedures, user defined functions (UDFs) e triggers. Note que você pode aprender como programar com diferentes linguagens utilizando o DB2 como servidor de dados, apenas visualizando as aplicações de exemplo que acompanham a instalação do servidor DB2 no diretório SQLLIB\samples. A figura abaixo mostra os programas exemplo em Java, disponibilizados com o DB2, em uma plataforma Windows.



**Programas exemplo em Java que acompanham o DB2**

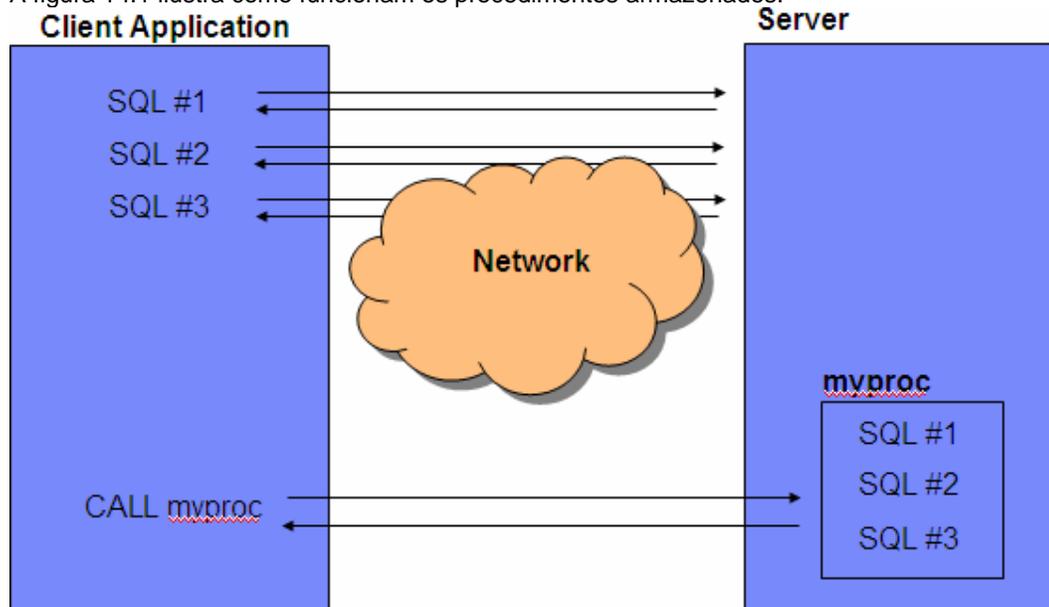
# 14

## Capítulo 14 – Procedimentos Armazenados SQL PL

Neste capítulo nós discutiremos os procedimentos armazenados (stored procedures). Um procedimento armazenado é um objeto de aplicação da base de dados que pode englobar instruções SQL e lógicas de negócio. Manter uma parte da lógica da aplicação na base de dados gera um melhor desempenho assim como a quantidade do tráfego da rede entre a aplicação e a base de dados é reduzida consideravelmente. Além disso, procedimentos armazenados geram uma localização centralizada para armazenar o seu código, sendo assim, outras aplicações podem reutilizar os mesmo procedimentos.

Os procedimentos armazenados no DB2 podem ser escritos utilizando SQL PL, C/C++, Java, Cobol, CLR (Linguagem Comum Runtime) linguagens suportadas e OLE. Neste capítulo, nos focamos em procedimentos SQL PL devido a sua popularidade e simplicidade.

A figura 14.1 ilustra como funcionam os procedimentos armazenados.



**Figura 14.1 – Redução do tráfego da rede com procedimentos armazenados**

No canto superior esquerdo da figura, você vê várias instruções SQL executadas uma após a outra. Cada SQL é enviado do cliente para o servidor, e o servidor envia o resultado para o cliente. Se muitas instruções SQL são executadas como estas, o tráfego da rede aumenta. Por outro lado, no canto inferior direito, você pode ver o procedimento armazenado “myproc” ocupando o servidor, que contém o mesmo SQL e, em seguida, no cliente (no lado esquerdo), a instrução CALL é utilizada para chamar o procedimento. Este segundo método de chamar o procedimento é mais eficiente, pois uma única instrução que vai se conectar, através da rede, e somente um resultado é enviado para o cliente.

Procedimentos armazenados podem também ser úteis por questões de segurança da sua base de dados. Por exemplo, você pode permitir que acessem ou visualizem tabelas somente através de procedimentos armazenados, isto ajuda a garantir a segurança de acesso ao servidor e faz com que os usuários não acessem informações que não foram autorizadas. Isto é possível porque os usuários não precisam ter privilégios explícitos em tabelas ou visões quando eles acessam os dados através de procedimentos armazenados; eles somente precisam ter permissão para executar os procedimentos armazenados.

**Nota:**

Para mais informações sobre procedimentos armazenados SQL PL, assista a este vídeo: <http://www.channeldb2.com/video/video/show?id=807741:Video:4343>

**14.1 O IBM Data Studio**

O IBM Data Studio é uma solução que ajuda você a desenvolver e gerenciar aplicações da base de dados durante todo o ciclo de vida do sistema. Algumas características que podem ser executadas com o Data Studio são:

- Criar, modificar e remover objetos da base de dados DB2 (com análise de impacto)
- Explorar e editar dados relacionais e XML
- Construir visualmente SQL e instruções XQuery
- Otimizar consultas utilizando o Visual Explain
- Desenvolvimento, depuração e implantação de procedimentos armazenados SQL e Java™
- Desenvolvimento utilizando funções definidas pelo usuário (UDFs)
- Desenvolvimento de aplicações SQLJ
- Desenvolvimento de consultas e rotinas em pureXML
- Executar tarefas de movimentação de dados
- Colaborar e dividir partes do projeto com os membros da equipe
- Construir rapidamente SOAP e REST Web Services
- Descobrir objetos da base de dados relacionados com modelos de dados físicos (diagramas)
- Visualizar a distribuição dos dados nas tabelas

O IBM Data Studio é baseado na plataforma Eclipse. É outra imagem (ou seja, não faz parte da instalação da imagem DB2), mas também é gratuita. É possível fazer o download

do Data Studio através da aba “Download” no site [ibm.com/db2/express](http://ibm.com/db2/express). A Figura 14.2 mostra o IBM Data Studio.

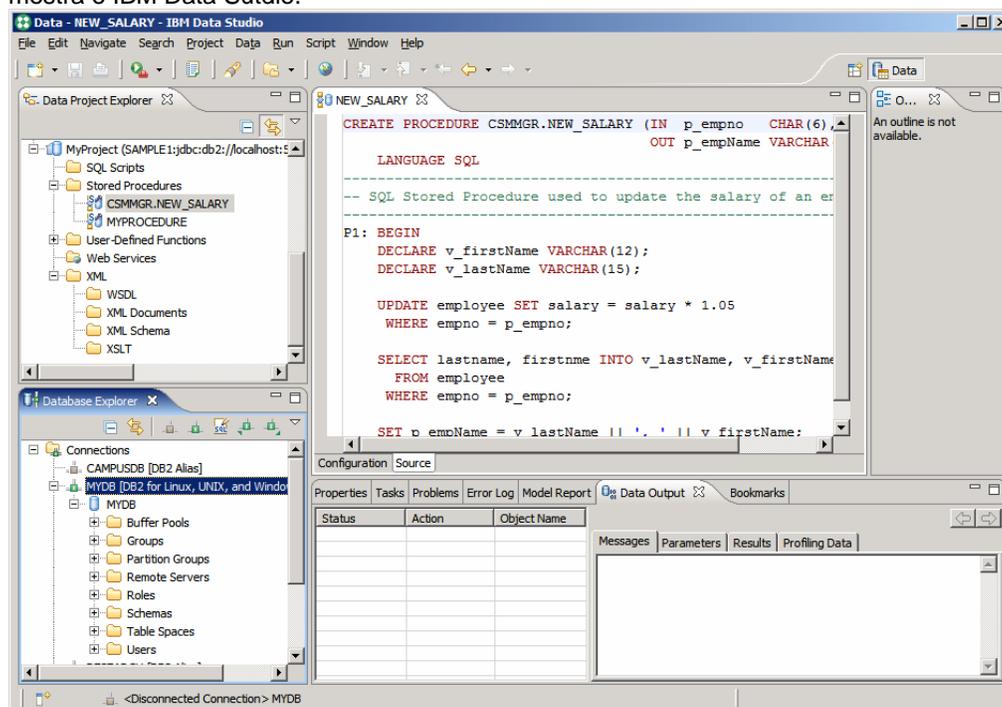


Figura 14.2 – O IBM Data Studio

### 14.1.2 Criar um procedimento armazenado no studio de dados

Para criar um procedimento armazenado Java ou SQL PL no Data Studio, siga os seguintes passos. Note que os procedimentos armazenados escritos em outras linguagens não podem ser criados no Data Studio.

#### Primeira Etapa: Criar um projeto Data Studio

No menu Data Studio, escolha *File -> New -> Project* e escolha *Data Development Project*. Como mostrado na Figura 14.3

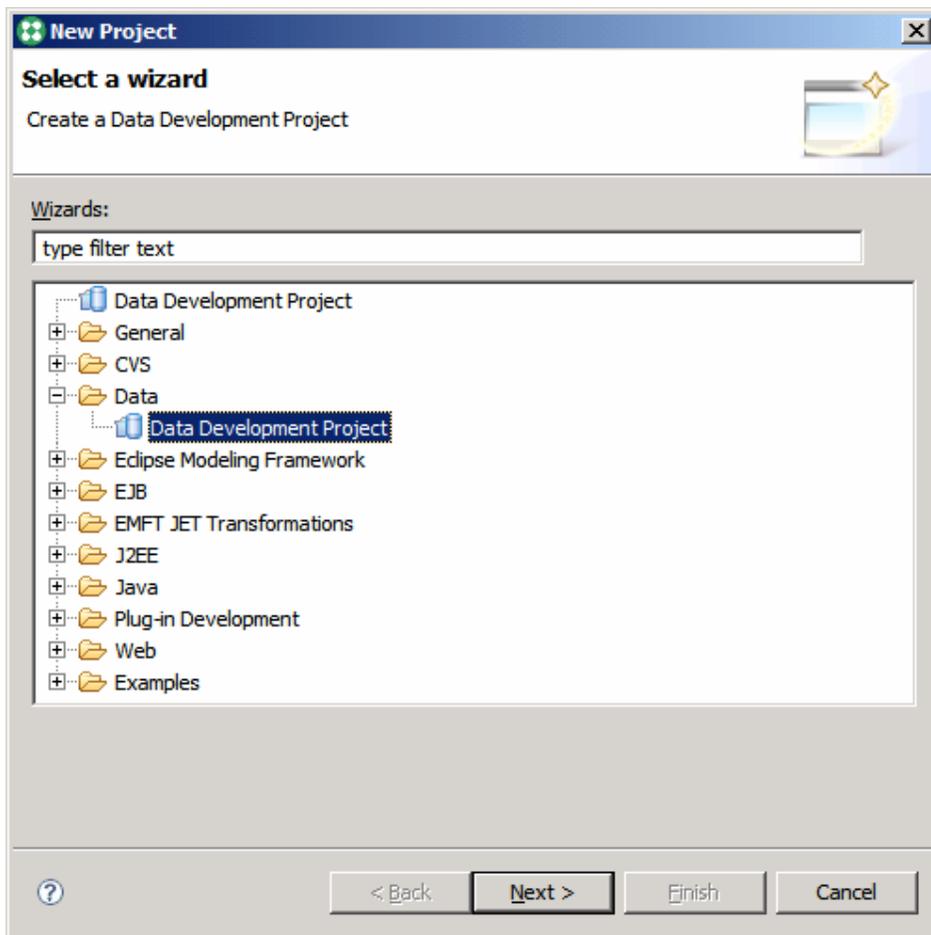


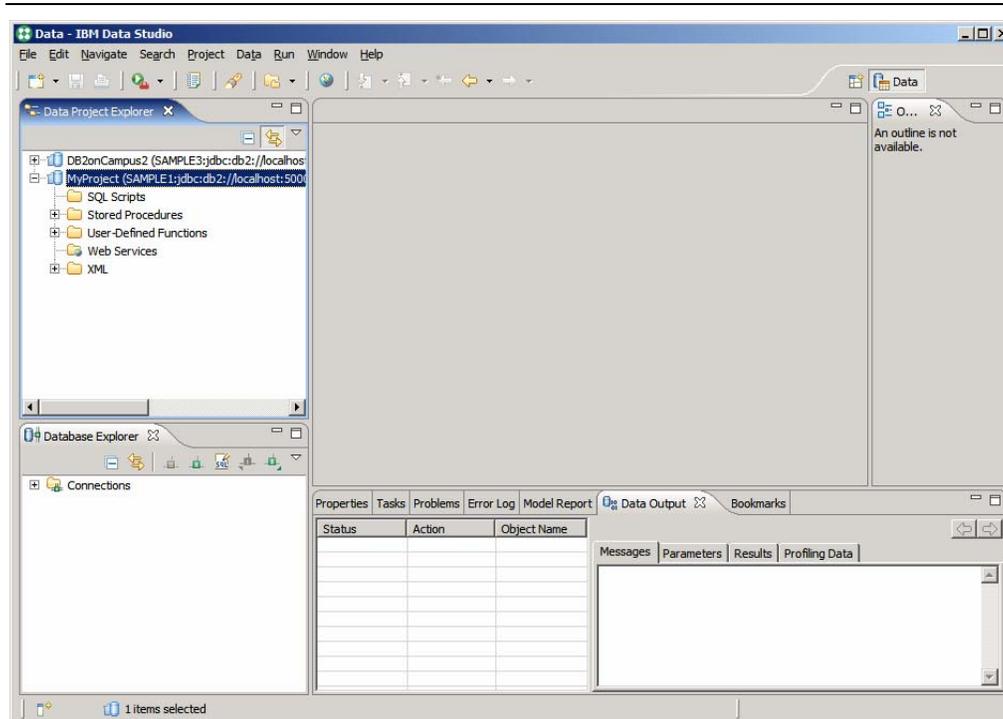
Figura 14.3 – O projeto de desenvolvimento de dados

Siga os passos do assistente para inserir um nome para o seu projeto, indique qual base de dados você deseja conectar parte do seu projeto e especifique o diretório JDK (ele é fornecido por padrão).

#### **Segunda Etapa: Criar um procedimento armazenado**

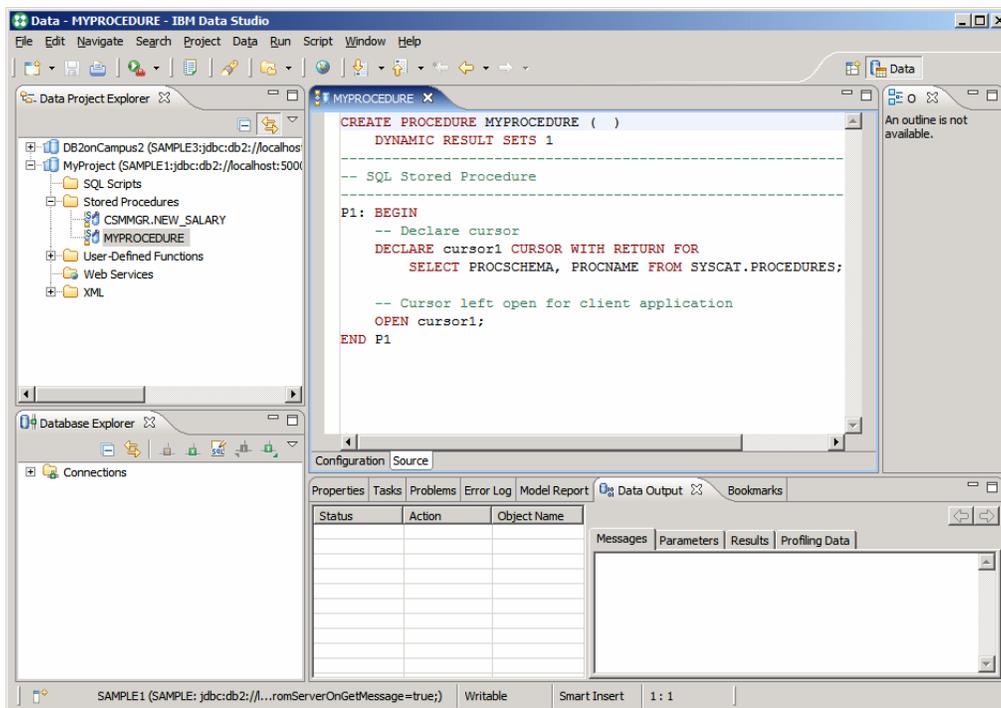
Quando o projeto é criado, o lado esquerdo da perspectiva dos dados irá mostrar seu projeto.

Na Figura 14.4 você pode ver em “myProject” o projeto criado e desenvolvido.



**Figura 14.4 – O projeto “myProject”**

A Figura 14.4 mostra pastas diferentes para o seu projeto. Quando você desejar criar um procedimento armazenado, clique com o botão direito na pasta *Stored Procedures* e escolha *New -> Stored Procedure*. Complete a informação solicitada pelo assistente “New Stored Procedure” com o projeto para associar com o procedimento, o nome e a linguagem do procedimento (Note que somente SQL PL e Java são suportados dentro do IBM Data Studio), e as instruções SQL utilizadas no procedimento. De maneira padrão, o Data Studio te dá um exemplo de instrução SQL. Neste ponto, você pode clicar em *Finish* e um procedimento armazenado é criado utilizando um modelo de código e a instrução SQL anteriormente utilizada como exemplo. Isto é mostrado na Figura 14.5.

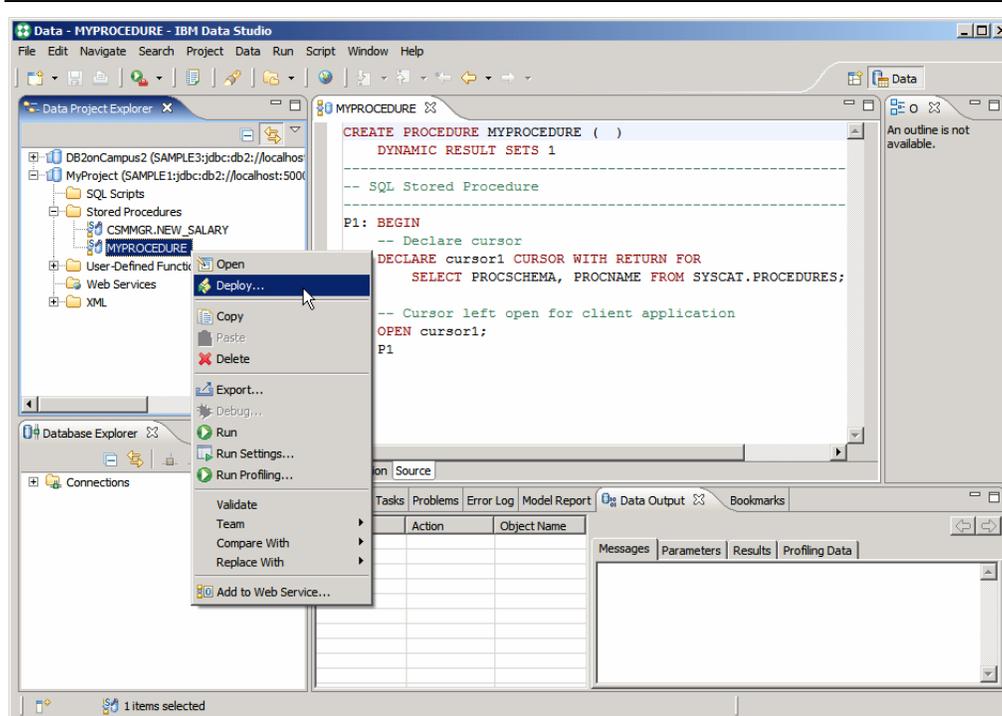


**Figura 14.5 – Um exemplo de procedimento armazenado**

Na figura 14.5, o código para o exemplo de procedimento armazenado “MYPROCEDURE” foi criado. Você pode substituir todo o código com seu próprio código. Para simplificar, nós continuaremos neste livro utilizando o exemplo de procedimento armazenado acima como se tivéssemos escrito.

**Terceira Etapa: Compile (implante) um procedimento armazenado**

Depois que o procedimento armazenado é criado, compile e implante-o clicando com o botão direito no painel esquerdo e escolha a opção “Deploy”. A figura 14.6 ilustra esta etapa.



**Figura 14.6 – Implantando um procedimento armazenado**

#### **Passo 4: Execute um procedimento armazenado**

Após o procedimento armazenado ser implantado, você pode executá-lo clicando com o botão direito sobre ele e selecionando 'Run'. Os resultados aparecerão na aba "Results" no lado inferior direito da tela.

Para executar um procedimento armazenado a partir da linha de comando ou no Command Editor, você pode usar a instrução `CALL <procedure name>`.

## **14.2 Fundamentos dos procedimentos armazenados SQL PL**

A linguagem procedural SQL (SQL PL) é fácil de aprender e é utilizada para criar procedimentos armazenados, os quais têm boa performance no DB2. Procedimentos armazenados SQL PL (ou simplesmente SQL stored procedures) são o foco deste capítulo.

### **14.2.1 A estrutura de um procedimento armazenado**

A sintaxe básica de um procedimento armazenado é:

```
CREATE PROCEDURE proc_name [( {optional parameters} )]
    [optional procedure attributes] <statement>
```

Onde <statement> é uma instrução única ou um conjunto de instruções agrupadas com `BEGIN [ATOMIC] ... END`

### 14.2.2 Atributos opcionais de procedimentos armazenados

A seguir são listados alguns dos atributos opcionais dos procedimentos armazenados.

- `LANGUAGE SQL`  
Este atributo indica a linguagem de programação que O procedimento armazenado utilizará. `LANGUAGE SQL` é o valor default. Para outras linguagens, como Java ou C, utilize `LANGUAGE JAVA` ou `LANGUAGE C`, respectivamente.
- `RESULT SETS <n>`  
Atributo necessário caso o procedimento armazenado retorne `n` conjuntos de resultados.
- `SPECIFIC my_unique_name`  
Este é um nome único que pode ser atribuído para a procedure. Um procedimento armazenado pode ser sobreposto, ou seja, vários procedimentos armazenados podem ter o mesmo nome, porém com diferentes tipos de parâmetros. Ao definir a palavra-chave `SPECIFIC`, você informa um nome único para cada um desses procedimentos armazenados. Por exemplo, para remover um procedimento armazenado utilizando a palavra-chave `SPECIFIC`, você pode executar a instrução `DROP SPECIFIC PROCEDURE`. Se a palavra-chave `SPECIFIC` não for definida, você deverá usar a instrução `DROP PROCEDURE` e informar o nome do procedimento armazenado com os parâmetros, assim o DB2 saberá qual dos procedimentos armazenados sobrepostos você deseja remover.

### 14.2.3 Parâmetros

Abaixo estão listados os três tipos de parâmetros dos procedimentos armazenados SQL PL.

- `IN` – Parâmetro de entrada
- `OUT` – Parâmetro de saída
- `INOUT` – Parâmetro de entrada e saída

Por exemplo:

```
CREATE PROCEDURE proc(IN p1 INT, OUT p2 INT, INOUT p3 INT)
```

Ao chamar a procedimento armazenado, todos os parâmetros devem ser informados na instrução `CALL`. Por exemplo, para executar a procedimento armazenado acima, você deve especificar:

```
CALL proc (10,?,4)
```

O ponto de interrogação ( ? ) é utilizado para os parâmetros de saída na instrução `CALL`.

Abaixo segue outro exemplo de um procedimento armazenado com parâmetros.

```
CREATE PROCEDURE P2 ( IN    v_p1 INT,
                    INOUT v_p2 INT,
                    OUT   v_p3 INT)

LANGUAGE SQL
SPECIFIC myP2
BEGIN
    -- my second SQL procedure
    SET v_p2 = v_p2 + v_p1;
    SET v_p3 = v_p1;
END
```

Para executar a procedimento armazenado a partir do Command Editor, utilize:

```
call P2 (3, 4, ?)
```

#### 14.2.4 Comentários em um procedimento armazenado SQL PL

Existem duas maneiras de especificar comentários em um procedimento armazenado SQL PL.

- Utilizando dois traços. Por exemplo:  
-- This is an SQL-style comment
- Utilizando o formato similar ao utilizado na linguagem C. Por exemplo:  
/\* This is a C-style coment \*/

#### 14.2.5 Instruções compostas

Uma instrução composta e um procedimento armazenado é uma instrução composta de várias instruções procedurais e instruções SQL encapsuladas pelas palavras-chave BEGIN e END. Quando a palavra-chave ATOMIC segue a palavra-chave BEGIN, as instruções compostas são tratadas como uma unidade, ou seja, todas as instruções e procedimentos da instrução composta devem ser executados com sucesso para que toda a instrução composta tenha sucesso. Caso uma das instruções não obtenha sucesso, tudo será desfeito. A Figura 14.7 mostra a estrutura de uma instrução composta.

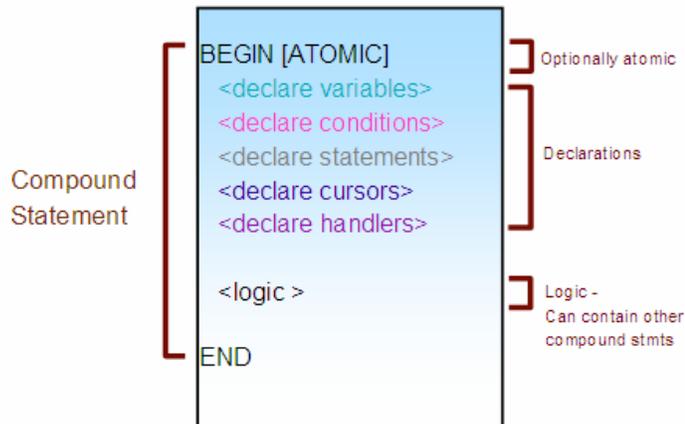


Figura 14.7 – Instruções compostas

### 14.2.6 Declaração de variáveis

Para declarar uma variável, utilize a instrução DECLARE.

```
DECLARE var_name <data type> [DEFAULT value];
```

Alguns exemplos:

```

DECLARE temp1 SMALLINT DEFAULT 0;
DECLARE temp2 INTEGER DEFAULT 10;
DECLARE temp3 DECIMAL(10,2) DEFAULT 100.10;
DECLARE temp4 REAL DEFAULT 10.1;
DECLARE temp5 DOUBLE DEFAULT 10000.1001;
DECLARE temp6 BIGINT DEFAULT 10000;
DECLARE temp7 CHAR(10) DEFAULT 'yes';
DECLARE temp8 VARCHAR(10) DEFAULT 'hello';
DECLARE temp9 DATE DEFAULT '1998-12-25';
DECLARE temp10 TIME DEFAULT '1:50 PM';
DECLARE temp11 TIMESTAMP DEFAULT '2001-01-05-12.00.00';
DECLARE temp12 CLOB(2G);
DECLARE temp13 BLOB(2G);
  
```

### 14.2.7 Instruções de atribuição

Para definir um valor para uma variável, utilize a instrução SET. Por exemplo:

```
SET total = 100;
```

A instrução acima é equivalente a:

```
VALUES(100) INTO total;
```

Adicionalmente, qualquer variável poder ser definida com valor NULL:

```
SET total = NULL;
```

Uma condição é atingida caso seja necessário consultar apenas a primeira linha de uma tabela.

```
SET total = (select sum(c1) from T1);
SET first_val = (select c1 from T1 fetch first 1 row only)
```

Você pode definir valor para variáveis a partir de propriedades externas da base de dados.

```
SET sch = CURRENT SCHEMA;
```

### 14.3 Cursores

Um cursor é um conjunto de resultados que possui o retorno de uma instrução SELECT. A sintaxe para declarar, abrir, navegar e fechar um cursor é:

```
DECLARE <cursor name> CURSOR [WITH RETURN <return target>]
    <SELECT statement>;
OPEN <cursor name>;
FETCH <cursor name> INTO <variables>;
CLOSE <cursor name>;
```

Quando um cursor é declarado, a cláusula WITH RETURN pode ser utilizada com os seguintes valores:

- CLIENT: o conjunto de resultados será retornado para a aplicação cliente
- CALLER: o conjunto de resultados é retornado para o cliente ou a procedimento armazenado que realizou a chamada

Abaixo segue um exemplo de um procedimento armazenado que declara um cursor:

```
CREATE PROCEDURE set()
DYNAMIC RESULT SETS 1
LANGUAGE SQL
BEGIN
DECLARE cur CURSOR WITH RETURN TO CLIENT
FOR SELECT name, dept, job
FROM staff
WHERE salary > 20000;
OPEN cur;
END
```

### 14.4 Controle de fluxo

Como em outras linguagens de programação, o SQL PL possui um conjunto de instruções que podem ser utilizadas para controlar a lógica do fluxo. Abaixo listamos algumas das instruções de controle de fluxo suportadas:

```
CASE (seleciona um caminho para execução (consulta simples))
IF
FOR (executa o corpo do bloco para cada linha de uma tabela)
WHILE
ITERATE (força a próxima iteração. Similar ao CONTINUE do C)
LEAVE (abandona um bloco ou loop . "Goto estruturado")
LOOP (loop infinito)
```

```

REPEAT
GOTO
RETURN
CALL (chamada de procedimento)

```

## 14.5 Executando procedimentos armazenados

Os exemplos abaixo mostram como executar procedimentos armazenados utilizando diferentes linguagens de programação.

### Exemplo da execução de um procedimento armazenado a partir de uma aplicação CLI/ODBC

```

SQLCHAR *stmt = (SQLCHAR *)
"CALL MEDIAN_RESULT_SET( ? )" ;
SQLDOUBLE sal = 20000.0; /* Bound to parameter marker in
stmt */
SQLINTEGER salind = 0; /* Indicator variable for sal */

sqlrc = SQLPrepare(hstmt, stmt, SQL_NTS);
sqlrc = SQLBindParameter(hstmt, 1, SQL_PARAM_OUTPUT,
SQL_C_DOUBLE, SQL_DOUBLE, 0, 0, &sal, 0, &salind);
SQLExecute(hstmt);

if (salind == SQL_NULL_DATA)
printf("Median Salary = NULL\n");
else
printf("Median Salary = %.2f\n\n", sal );

/* Get first result set */
sqlrc = StmtResultPrint(hstmt);
/* Check for another result set */
sqlrc = SQLMoreResults(hstmt);
if (sqlrc == SQL_SUCCESS) {
/* There is another result set */
sqlrc = StmtResultPrint(hstmt);
}

```

Para maiores detalhes, veja o arquivo de exemplo do DB2:  
[sqllib/samples/sqlproc/resultset.c](#)

**Exemplo da execução de um procedimento armazenado a partir de uma aplicação VB.NET**

```

Try
    ` Create a DB2Command to run the stored procedure
    Dim procName As String = "TRUNC_DEMO"
    Dim cmd As DB2Command = conn.CreateCommand()
    Dim parm As DB2Parameter

    cmd.CommandType = CommandType.StoredProcedure
    cmd.CommandText = procName

    ` Register the output parameters for the DB2Command
    parm = cmd.Parameters.Add("v_lastname",
DB2Type.VarChar)
    parm.Direction = ParameterDirection.Output
    parm = cmd.Parameters.Add("v_msg",
DB2Type.VarChar)
    parm.Direction = ParameterDirection.Output

    ` Call the stored procedure
    Dim reader As DB2DataReader = cmd.ExecuteReader

Catch myException As DB2Exception
    DB2ExceptionHandler(myException)
Catch
    UnhandledExceptionHandler()
End Try

```

**Exemplo de chamada a um procedimento armazenado a partir de uma aplicação Java**

```

try
{
    // Connect to sample database
    String url = "jdbc:db2:sample";
    con = DriverManager.getConnection(url);

    CallableStatement cs = con.prepareCall("CALL
trunc_demo(?, ?)");

    // register the output parameters
    callStmt.registerOutParameter(1, Types.VARCHAR);
    callStmt.registerOutParameter(2, Types.VARCHAR);

    cs.execute();
    con.close();
}
catch (Exception e)
{
    /* exception handling logic goes here */
}

```

## 14.6 Manipuladores de erros e condições

No DB2, as palavras-chave `SQLCODE` e `SQLSTATE` são utilizadas para determinar o sucesso ou a falha na execução de uma instrução SQL. Estas palavras-chave devem ser explicitamente declaradas no escopo do procedimento, como segue:

```
DECLARE SQLSTATE CHAR(5);  
DECLARE SQLCODE INT;
```

O DB2 definirá os valores das palavras-chave automaticamente, após cada operação SQL.

Para a palavra-chave `SQLCODE`, os valores definidos são:

- = 0, sucesso.
- > 0, sucesso com avisos
- < 0, erro
- = 100, nenhum dado foi encontrado. (exemplo: instrução `FETCH` não retornou dados)

Para a palavra-chave `SQLSTATE`, os valores definidos são:

- sucesso: `SQLSTATE '00000'`
- não encontrado: `SQLSTATE '02000'`
- aviso: `SQLSTATE '01XXX'`
- exceção: todos os outros valores

O `SQLCODE` é específico do SGBD e mais detalhado do que `SQLSTATE`. O `SQLSTATE` é padrão de acordo com o SGBD. Vários `SQLCODE` podem fazer referência a um `SQLSTATE`. O `SQLCODE` e o `SQLSTATE` serão discutidos em maiores detalhes no capítulo 18; Troubleshooting.

Uma condição pode ser atingida por qualquer instrução SQL e fará referência a um `SQLSTATE`. Por exemplo, uma condição específica como `SQLSTATE '01004'` é atingida quando um valor é truncado durante uma operação SQL. Ao invés de utilizar o `SQLSTATE '01004'` para testar esta condição, nomes podem ser atribuídos. Neste exemplo, o nome `'trunc'` pode ser atribuído para a condição `SQLSTATE '01004'`, como mostrado abaixo.

```
DECLARE trunc CONDITION FOR SQLSTATE '01004'
```

Outras condições pré-definidas são:

- `SQLWARNING`
- `SQLEXCEPTION`
- `NOT FOUND`

### Manipulação de condições

Para manipular uma condição, você pode criar um manipulador de condições, que pode especificar:

- quais condições ele manipula;

- onde finalizar a execução (baseado no tipo do manipulador: CONTINUE, EXIT ou UNDO)
- as ações executadas para manipular a condição. As ações podem ser qualquer instrução, incluindo estruturas de controle.

Caso uma SQLEXCEPTION seja atingida e nenhum manipulador exista, o procedimento será finalizado e retornará um erro ao cliente.

### Tipos de manipuladores

Existem três tipos de manipuladores:

**CONTINUE** – Este manipulador é utilizado para indicar que após uma exceção ser atingida, e uma condição for manipulada, o fluxo continuará para a próxima instrução, após a instrução que atingiu a condição.

**CONTINUE** – Este manipulador é utilizado para indicar que, após uma exceção ser atingida, e a manipulação acontecer, o fluxo irá para o fim do procedimento.

**UNDO** – Este manipulador indica que após uma exceção ser atingida, e a condição ser manipulada, o fluxo irá para o final do procedimento, e retornará todas as instruções executadas.

A figura 14.8 ilustra os diferentes manipuladores de condição e os seus comportamentos.

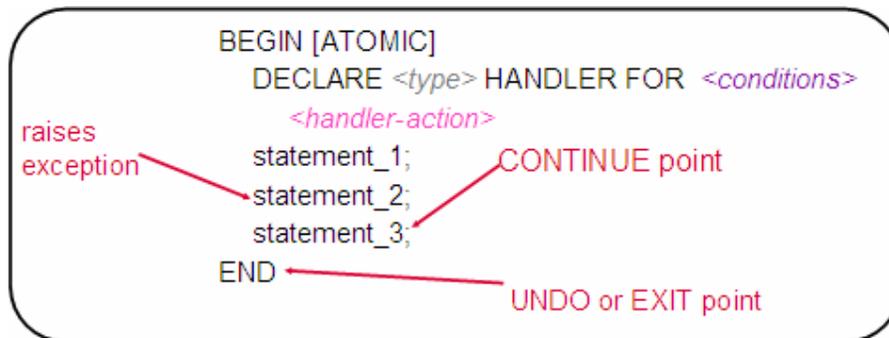


Figura 14.8 – Tipos de manipuladores de condição

## 14.7 SQL Dinâmico

No SQL dinâmico, ao contrário do SQL estático, a instrução SQL não é conhecida em tempo de execução. Por exemplo, se `col1` e `tablename` são variáveis na instrução, então temos um SQL dinâmico.

```
'SELECT ' || col1 || ' FROM ' || tablename;
```

O SQL dinâmico é recomendado em instruções DDL, para evitar problemas de dependência e pacotes. Também são recomendados para implementar recursividade.

SQL Dinâmicos podem ser executados através de duas maneiras:

- Utilizando a instrução EXECUTE IMMEDIATE – ideal para uma instrução SQL simples.
- Utilizando a instrução PREPARE com a instrução EXECUTE – ideal para a execução de múltiplas instruções SQL.

O código abaixo mostra um exemplo de SQL Dinâmico utilizando as duas maneiras. O exemplo considera que a tabela T2 tenha sido criada com a seguinte estrutura:

```
CREATE TABLE T2 (c1 INT, c2 INT)

CREATE PROCEDURE dyn1 (IN value1 INT, IN value2 INT)
SPECIFIC dyn1
BEGIN
DECLARE stmt varchar(255);
DECLARE st STATEMENT;

SET stmt = 'INSERT INTO T2 VALUES (?, ?)';

PREPARE st FROM stmt;

EXECUTE st USING value1, value1;
EXECUTE st USING value2, value2;

SET stmt = 'INSERT INTO T2 VALUES (9,9)';
EXECUTE IMMEDIATE stmt;
END
```



# 15

## Capítulo 15 – Inline SQL PL, Triggers, e UDFs

Neste capítulo, discutiremos SQL PL e outros objetos de aplicação como funções definidas pelo usuário (user-defined functions - UDFs) e triggers.

**Nota:**

Para mais informações sobre triggers e UDFs assista estes vídeos:  
<http://www.channeldb2.com/video/video/show?id=807741:Video:4367>  
<http://www.channeldb2.com/video/video/show?id=807741:Video:4362>

### 15.1 Inline SQL PL

No capítulo 14 discutimos como criar stored procedures (procedimentos armazenados) usando a linguagem SQL PL. A linguagem SQL PL também pode ser usada “inline” (em linha) o que significa que ela pode ser usada sozinha sem que seja armazenada em uma stored procedure. A linguagem SQL PL usada em UDF's e triggers é também chamada inline porque o código é adicionado na própria UDF/trigger, o que representa um SQL dinâmico. O SQL PL inline suporta somente alguns sub conjuntos de instruções SQL PL. A listagem abaixo mostra as palavras chaves que podem ser usadas no inline SQL PL.

```
DECLARE <variable>
SET
CASE
FOR
GET DIAGNOSTICS
GOTO
IF
RETURN
SIGNAL
WHILE
ITERATE
LEAVE
```

Aqui estão as palavras chaves que não são suportadas no inline SQL PL:

```
ALLOCATE CURSOR
ASSOCIATE LOCATORS
DECLARE <cursor>
```

```

DECLARE ...HANDLER
PREPARE
EXECUTE
EXECUTE IMMEDIATE
LOOP
REPEAT
RESIGNAL
CALL
COMMIT/ROLLBACK

```

Verifique abaixo um exemplo de um código SQL dinâmico que utiliza inline SQL PL. Se você quiser testá-lo, você pode utilizá-lo como parte de um script, mas antes tenha certeza que você criou as seguintes tabelas:

```

CREATE TABLE T1 (c1 INT)
CREATE TABLE T3 (c1 INT)

      BEGIN ATOMIC
        DECLARE cnt          INT DEFAULT 0;
        DECLARE sumevens INT DEFAULT 0;
        DECLARE err_msg VARCHAR(1000) DEFAULT '';
        WHILE (cnt < 100) DO
          IF mod(cnt,2) = 0 THEN
            SET sumevens = sumevens + cnt;
          END IF;
          SET cnt=cnt+1;
        END WHILE;
        INSERT INTO T3 values (sumevens);
        SET cnt = (SELECT 0 FROM SYSIBM.SYSDUMMY1);
        FOR cur1 AS SELECT * FROM T1 DO
          IF cur1.c1 > 100 THEN
            SET cnt = cnt + 1;
          END IF;
        END FOR;

        SET err_msg = 'Rows with values > 100 is: ' ||
char(cnt);
        SIGNAL SQLSTATE '80000' SET MESSAGE_TEXT = err_msg;
      END!

```

Se você salvar o código SQL PL acima em um script chamado "myScript.txt" você pode executá-lo da seguinte maneira:

```
db2 -td! -vf myScript.txt
```

## 15.2 Triggers

Triggers (também chamado de gatilhos em português) são objetos de banco de dados associados a uma tabela os quais definem que operações sejam executadas quando ocorre uma instrução INSERT, UPDATE, ou DELETE. Os triggers são ativados (ou disparados)

automaticamente. As operações que causam o disparo de uma trigger são chamadas instruções SQL *triggering*.

### 15.2.1 Tipos de triggers

Existem três tipos de triggers: “before” triggers, “after” triggers, and “instead of” triggers

#### Before triggers

Before triggers são disparadas antes de uma linha ser inserida, atualizada ou removida. As operações disparadas por estas triggers não envolvem o disparo de outra trigger (somente operações de INSERT, UPDATE, e DELETE são permitidas).

Um exemplo simples de uma before trigger é mostrado na Figura 15.1

```

CREATE TRIGGER default_class_end
NO CASCADE BEFORE INSERT ON cl_sched
REFERENCING NEW AS n
FOR EACH ROW
MODE DB2SQL
WHEN (n.ending IS NULL)
    SET n.ending = n.starting + 1 HOUR
  
```

define  
qualifier for  
new values

optional  
WHEN

if no value  
provided on  
insert, column is  
NULL

Figura 15.1 – Exemplo de uma before trigger

Na Figura 15.1 a trigger “default\_class\_end” será disparada antes que aconteça uma instrução SQL de INSERT na tabela cl\_sched. Esta tabela faz parte da base de dados SAMPLE, sendo assim, você pode criar e testar esta trigger se estiver conectado nesta base de dados. A variável “n”, que aparece na definição da trigger representa o novo valor a ser utilizado na instrução INSERT, ou seja, o valor a ser inserido. A trigger checará se o valor que vai ser inserido na tabela é válido. Se a coluna “ending” não tiver nenhum valor, a trigger utilizará o valor da coluna “starting” acrescido de 1 hora.

As seguintes instruções mostrarão como testar uma trigger:

```

C:\Program Files\IBM\SQLLIB\BIN>db2 insert into cl_sched
(class_code, day, starting) values ('abc',1,current time)
DB20000I The SQL command completed successfully.

C:\Program Files\IBM\SQLLIB\BIN>db2 select * from cl_sched

CLASS_CODE DAY    STARTING ENDING
  
```

```

-----
042:BF          4 12:10:00 14:00:00
553:MJA        1 10:30:00 11:00:00
543:CWM        3 09:10:00 10:30:00
778:RES        2 12:10:00 14:00:00
044:HD         3 17:12:30 18:00:00
abc            1 11:06:53 12:06:53

```

```
6 record(s) selected.
```

A trigger “validate\_sched”, abaixo, estende a funcionalidade da trigger “default\_class\_end” mostrada anteriormente. Novamente, você pode criá-la e testá-la na base de dados SAMPLE.

```

CREATE TRIGGER validate_sched
NO CASCADE BEFORE INSERT ON cl_sched
REFERENCING NEW AS n
FOR EACH ROW
MODE DB2SQL
BEGIN ATOMIC
-- supply default value for ending time if null
IF (n.ending IS NULL) THEN
    SET n.ending = n.starting + 1 HOUR;
END IF;

-- ensure that class does not end beyond 9pm
IF (n.ending > '21:00') THEN
    SIGNAL SQLSTATE '80000'
    SET MESSAGE_TEXT='class ending time is beyond 9pm';
ELSEIF (n.DAY=1 or n.DAY=7) THEN
    SIGNAL SQLSTATE '80001'
    SET MESSAGE_TEXT='class cannot be scheduled on a
weekend';
END IF;
END

```

### After triggers

After triggers são disparadas depois que algumas instruções SQL tenham sido executadas com sucesso. As operações executadas por esta trigger podem ativar outras triggers (é permitido um cascadeamento de triggers até 16 níveis). After triggers suportam operações de INSERT, UPDATE e DELETE. Veja abaixo um exemplo de uma after trigger.

```

CREATE TRIGGER audit_emp_sal
AFTER UPDATE OF salary ON employee
REFERENCING OLD AS o NEW AS n
FOR EACH ROW
MODE DB2SQL
INSERT INTO audit VALUES (
    CURRENT TIMESTAMP, ' Employee ' || o.empno || ' sal-
ary changed from ' || CHAR(o.salary) || ' to ' ||
CHAR(n.salary) || ' by ' || USER)

```

Neste exemplo, a trigger `audit_emp_sal` é usada para realizar uma auditoria na coluna "salary" da tabela "employee". Quando alguém fizer uma alteração nesta coluna, a trigger será disparada para que informações sobre a alteração de salário sejam inseridas na tabela "audit". Na linha "OLD as o NEW as n", o prefixo "o" representa o valor antigo ou o valor já existente na tabela, e o prefixo "n" indica o novo valor que está sendo usada na instrução de UPDATE. Sendo assim, "o.salary" representa o valor antigo do salário, e "n.salary" representa o valor atualizado da coluna salário.

### **"Instead of" triggers**

Instead of triggers são definidas como views (visões). A lógica definida na trigger é executada ao invés da instrução SQL. Por exemplo, se você realizar uma operação de atualização em uma visão, a trigger "Instead of" será disparada para executar a atualização das tabela que fazem parte da visão.

Triggers não podem ser criadas no IBM Data Studio. Elas podem ser criadas no Control Center ou a partir de ferramentas de linha de comando (Command Window, Command Line Processor, ou Command Editor).

## Quicklab #12 – Criando uma trigger no Control Center

### Objetivo

Triggers são objetos de banco de dados utilizadas para realizar necessidades da lógica do negócio quando uma operação de modificação de dados acontece em uma tabela. Neste Quicklab, você criará uma trigger utilizando o Control Center. Esta trigger manterá a log de alterações realizadas na tabela SALES, de acordo com necessidades de auditoria. Você armazenará o ID do usuário que realizou a alteração, assim como a data/hora que a transação foi feita.

### Procedure

1. Abra o Control Center.
2. Neste Quicklab, você precisa criar uma tabela adicional que será usada para armazenar os logs. Crie a tabela com as seguintes características:

Nome da Tabela: `saleslog`

Primeira Coluna:

Nome: `userid`

Tipo do dado: `VARCHAR(128)`

Outros atributos: `NOT NULL`

Segunda Coluna

Nome: `daytime`

Tipo do dado: `TIMESTAMP`

Outros atributos: `NOT NULL`

Dica: Crie esta tabela usando a instrução `CREATE TABLE` no Editor de Comandos; ou use o wizard *Create Table* wizard do Control Center.

3. A partir do Control Center, expanda a pasta da base de dados *EXPRESS*. Clique com o botão da direita na pasta *Triggers* e selecione a opção *Create*. A janela de diálogo *Create Trigger* será aberta.
4. Preencha as seguintes informações na janela de diálogo.

Trigger esquema (Esquema onde a trigger será criada): ID do usuário que você utilizou para logar (deve permanecer o valor padrão)

Trigger name (Nome da Trigger): `audit_sales`

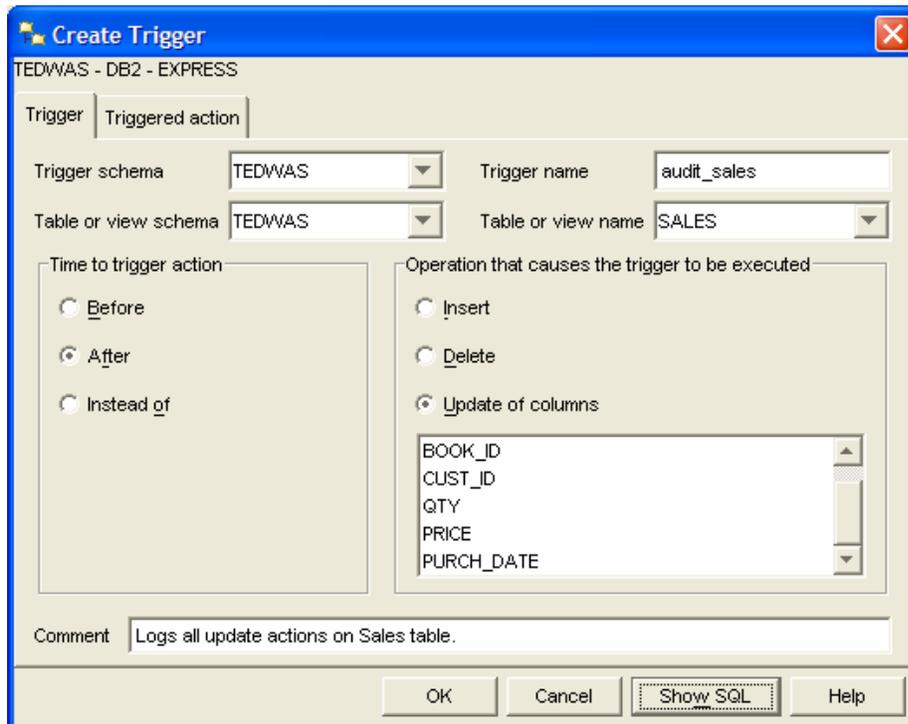
Table or view schema (Esquema da tabela ou visão): ID do usuário que você utilizou para logar (deve permanecer o valor padrão)

Table or visão name (Nome da tabela ou visão): *SALES*

Time to trigger action (Tipo da trigger): *After*

Operation that causes the trigger to be executed (operação que causa o disparo da trigger): Atualização de colunas (não especifique nenhuma coluna porque faremos com que a trigger seja disparada na atualização de qualquer coluna)

Comment (Comentários): Todas operação de atualização na tabela *Sales* serão logadas.

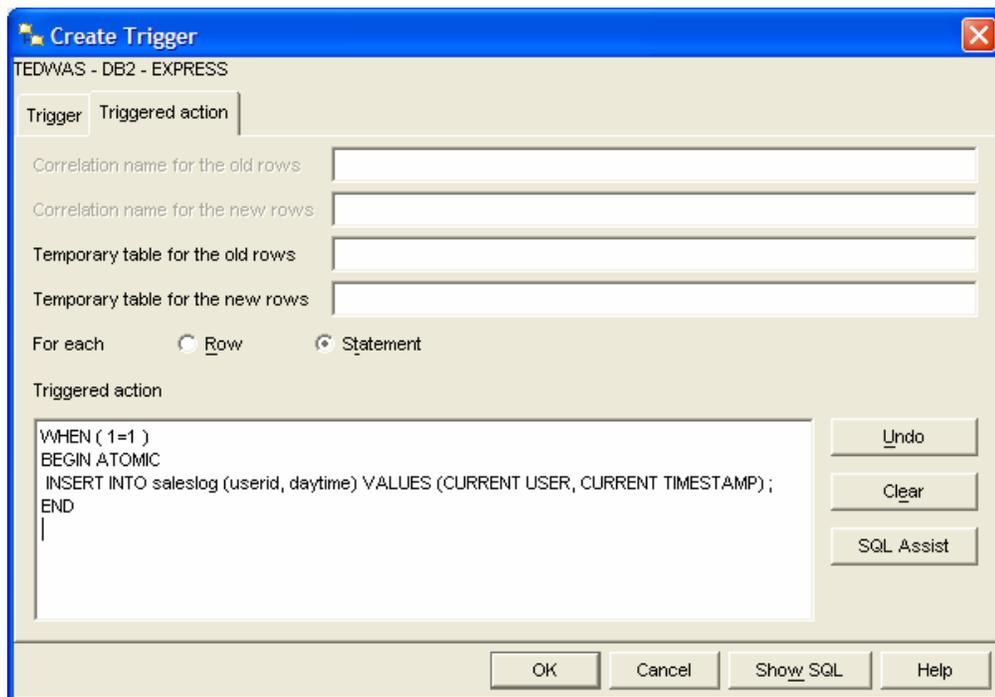


5. Na aba *Triggered action*, selecione a opção *For Each STATEMENT*. Use o seguinte código que disparará a trigger.

```
WHEN ( 1=1 )
BEGIN ATOMIC
```

```
INSERT INTO saleslog (userid, daytime) VALUES (CURRENT
USER, CURRENT TIMESTAMP);
END
```

(Nota: Uma *statement trigger* (trigger de instrução) será disparada assim que a instrução SQL for executada. Uma *row trigger* (trigger de linha) indica que a trigger será disparada quando uma instrução SQL afetar aquela linha.)



Clique no botão *OK* para criar a trigger.

6. Agora você vai conseguir visualizar a trigger na pasta *Triggers* no Control Center.
7. Consulte a tabela *saleslog* para confirmar que não existe nenhum dado nesta tabela. Delete qualquer linha existente (`DELETE FROM saleslog`).
8. Tente atualizar um registro na tabela *sales*. (Dica: use o Command Editor ou o SQL Assist Wizard).
9. Verifique o conteúdo da tabela *saleslog* novamente. Quantas linhas existem na tabela?

### 15.3 Funções definidas pelo usuário (UDFs)

Uma função definida pelo usuário (UDF) é um objeto de aplicação de banco de dados que mapeia um conjunto de entradas de dados em um conjunto de saída. Por exemplo, uma função pode ter como valor de entrada uma medida em polegares e retornar a mesma medida em centímetros.

O DB2 suporta a criação de funções usando SQL PL, C/C++, Java, CLR (Common Language Runtime), e OLE (Object Linking and Embedding). Neste livro, iremos focar nas funções SQL PL por serem mais simples, mais utilizadas e por obterem melhor performance.

Existem quatro tipos de funções: escalares, aplicadas em tabela, linha e coluna. Neste capítulo iremos focar somente nas funções escalares e aplicadas em tabela.

#### 15.3.1 Funções Escalares

Funções escalares retornam somente um valor. Funções escalares não podem incluir instruções SQL que mudem o estado da base de dados; ou seja, instruções INSERT, UPDATE, e DELETE não são permitidas. Algumas funções escalares já existentes no DB2 são: SUM(), AVG(), DIGITS(), COALESCE(), and SUBSTR().

O DB2 permite que você construa funções definidas pelo usuário que encapsulem lógicas de negócio utilizadas com frequência. Por exemplo, considere a migração de sua aplicação do Oracle para o DB2. Na sua aplicação, você necessita com frequência da função NVL() do Oracle. A função equivalente a esta no DB2 é chamada COALESCE. Ao invés de renomear todas as funções NVL da sua aplicação para COALESCE, você pode criar uma função definida pelo usuário, no DB2, e chamá-la NVL, fazendo com que ela invoque a função COALESCE conforme mostrado abaixo.

```
CREATE FUNCTION NVL (p_var1 VARCHAR(30),
                   p_var2 VARCHAR(30))
SPECIFIC nvlvarchar30
RETURNS VARCHAR(30)
RETURN COALESCE(p_var1, p_var2)
```

A função COALESCE retorna o primeiro argumento não nulo.

Abaixo segue outro exemplo de função escalar. A função é chamada “deptname” e retorna o número do departamento quando é informado o id do empregado.

```
CREATE FUNCTION deptname(p_empid VARCHAR(6))
RETURNS VARCHAR(30)
SPECIFIC deptname
BEGIN ATOMIC
  DECLARE v_department_name VARCHAR(30);
  DECLARE v_err VARCHAR(70);
  SET v_department_name = (
    SELECT d.deptname FROM department d, employee e
    WHERE e.workdept=d.deptno AND e.empno= p_empid);
  SET v_err = 'Error: employee ' || p_empid || ' was not
found';
```

```

        IF v_department_name IS NULL THEN
            SIGNAL SQLSTATE '80000' SET MESSAGE_TEXT=v_err;
        END IF;
    RETURN v_department_name;
END

```

Para testar esta função, execute o seguinte comando a partir do Command Window ou do shell do Linux/UNIX:

```
db2 "values (deptname ('000300'))"
```

### Chamando UDFs escalares

UDFs escalares podem ser chamadas em instruções escalares sempre que um valor escalar for esperado ou na cláusula VALUES. Abaixo você tem dois exemplos que mostram uma chamada a função escalar COALESCE.

```

SELECT DEPTNAME, COALESCE(MGRNO, 'ABSENT') FROM DEPARTMENT
VALUES COALESCE('A', 'B')

```

### 15.3.2 Funções aplicadas em tabela

Funções aplicadas em tabela retornam linhas da tabela. Você pode chamá-las na cláusula FROM de uma consulta. Funções aplicadas em tabela, ao contrário das funções escalares, podem mudar o estado de uma tabela; sendo assim, instruções INSERT, UPDATE, e DELETE são permitidas. Algumas funções aplicadas em tabelas já existentes no DB2 são SNAPSHOT\_DYN\_SQL() e MQREADALL(). Funções aplicadas em tabelas são parecidas com visões, porém, como permitem o uso com instruções que modificam dados (INSERT, UPDATE, and DELETE) elas são mais robustas. São tipicamente utilizadas para retornar uma tabela e manter registros de auditoria.

Abaixo segue um exemplo de uma função aplicada em tabela que enumera um conjunto de empregados de um departamento.

```

CREATE FUNCTION getEnumEmployee(p_dept VARCHAR(3))
RETURNS TABLE
    (empno CHAR(6),
     lastname VARCHAR(15),
     firstnme VARCHAR(12))
SPECIFIC getEnumEmployee
RETURN
    SELECT e.empno, e.lastname, e.firstnme
    FROM employee e
    WHERE e.workdept=p_dept

```

Para testar a função acima execute:

```
db2 "SELECT * FROM table(getEnumEmployee('D11')) AS t"
```

**Chamando UDFs aplicadas em tabela**

Uma UDF aplicada em tabela pode ser chamada a partir da cláusula FROM de uma instrução SQL. A função TABLE() deve ser aplicada e deve ser utilizado um alias. A Figura 15.2 mostra um exemplo de como chamar a função “getEnumEmployee” a qual testamos anteriormente.

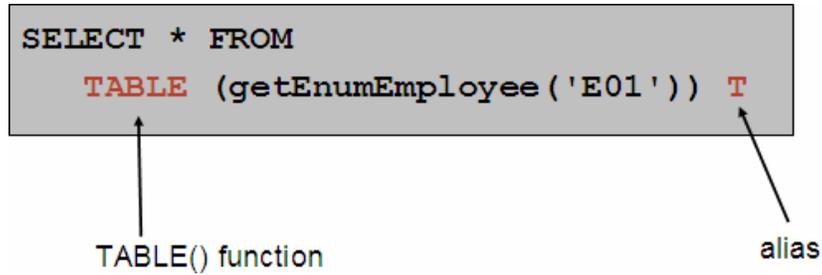


Figura 15.2 – Chamando uma função aplicada em tabela

## Quicklab #13 – Criando uma UDF usando o IBM Data Studio

### Objetivo

Neste Quicklab você criará uma função definida pelo usuário utilizando o IBM Data Studio. Este exercício lhe dará mais experiência com o Data Studio, assim como aumentará a sua familiaridade com a linguagem SQL PL em funções definidas pelo usuário.

### Procedimento

1. Abra o IBM Data Studio (Dica: acesse-o via o menu Iniciar)
2. A partir da janela Data Project Explorer, escolha o projeto que você criou no Quicklab anterior e selecione Open Project.
3. Clique com o botão da direita na pasta *User-Defined Functions*. Selecione o menu *New*, e então o menu *User-Defined Function*. Quando o Wizard User-Defined Function for aberto assegure-se que as seguintes opções estão preenchidas:

Project: Ensure the project name is the one you created earlier.

Name: FUNCTION1

Language: SQL

Clique no botão FINISH.

4. A visão *Editor* será aberta com um “esqueleto” da função. Modifique o código de acordo com os dados a seguir:

```
CREATE FUNCTION booktitle(p_bid INTEGER)
RETURNS VARCHAR(300)
-----
SQL UDF (Scalar)
-----
SPECIFIC booktitle
F1: BEGIN ATOMIC
DECLARE v_book_title VARCHAR(300);
DECLARE v_err VARCHAR(70);
SET v_book_title = (SELECT title FROM books WHERE p_bid =
book_id);
SET v_err = 'Error: The book with ID ' || CHAR(p_bid) || '
was not found.';
IF v_book_title IS NULL THEN SIGNAL SQLSTATE '80000' SET
MESSAGE_TEXT=v_err;
END IF;
RETURN v_book_title;
```

END

5. Construa a função clicando com o botão da direita na função e escolhendo *Deploy*. Se for aberta uma janela de dialogo pedindo que as alterações sejam gravadas, clique em *Yes*. Será aberto um painel onde você pode escolher as opções de entrega (Deploy Options). Deixe marcado os valores default e clique em *Finish*.
6. Execute a função clicando com o botão da direita na função e escolhendo *Run*.
7. Como a função aceita a entrada de um parâmetro, uma janela de dialogo aparecerá requisitando um valor para o parâmetro.

Entre o valor: 80002

Qual o resultado?

Tente novamente utilizando o valor: 1002

O que aconteceu agora? (Dica: Veja a seção *Messages* da visão *Output*).

8. Feche o IBM Data Studio quando você terminar.

# 16

## Capítulo 16 – DB2 pureXML

Neste capítulo, discutiremos pureXML, a nova tecnologia oferecida a partir do DB2 9 para suportar armazenamento nativo de XML. Muitos dos exemplos e conceitos discutidos neste capítulo foram extraídos do Redbook *DB2 9: pureXML overview and fast start* da IBM. Veja a seção de Recursos para maiores informações. A Figura 16.1 apresenta quais seções da Visão Geral do DB2 discutiremos neste capítulo.

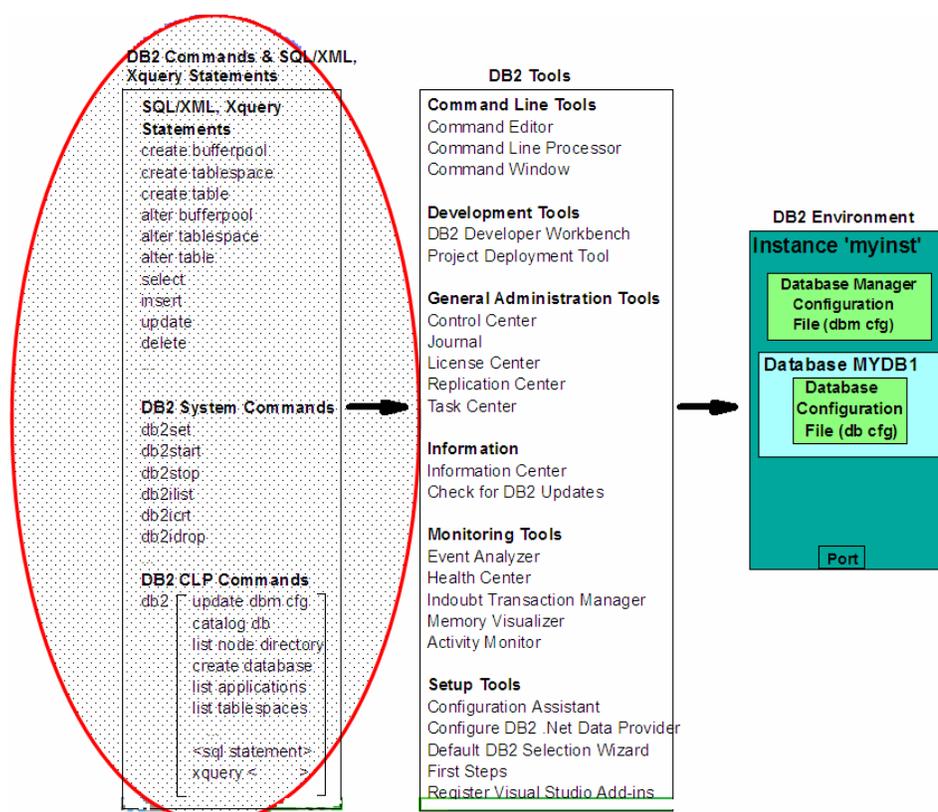


Figura 16.1 – Visão geral do DB2: comandos do DB2, SQL/XML e XQuery

**Nota:**

Para maiores informações sobre pureXML, assista ao vídeo:

<http://www.channeldb2.com/video/video/show?id=807741:Video:4382>

## 16.1 Usando XML em bases de dados

Documentos XML podem ser armazenados em arquivos de texto, repositórios XML ou bases de dados. Há duas razões principais pelas quais muitas empresas propõem armazená-los em bases de dados:

- Administrar grande volume de dados XML é um problema de responsabilidade para a base de dados. XML é um dado como outro qualquer, apenas possui um formato diferente. As mesmas razões de armazenar dados relacionais em bases de dados se aplicam ao dado XML: bases de dados oferecem busca e recuperação eficiente, suporte robusto à persistência de dados, backup e recovery, suporte a transações, desempenho e escalabilidade.
- Integração: ao armazenar dados relacionais e documentos XML juntos, pode-se integrar o novo dado XML com os dados relacionais existentes, e combinar SQL com XPath ou XQuery em uma consulta. Através da integração, bases de dados podem suportar melhor aplicações Web, SOA e web services.

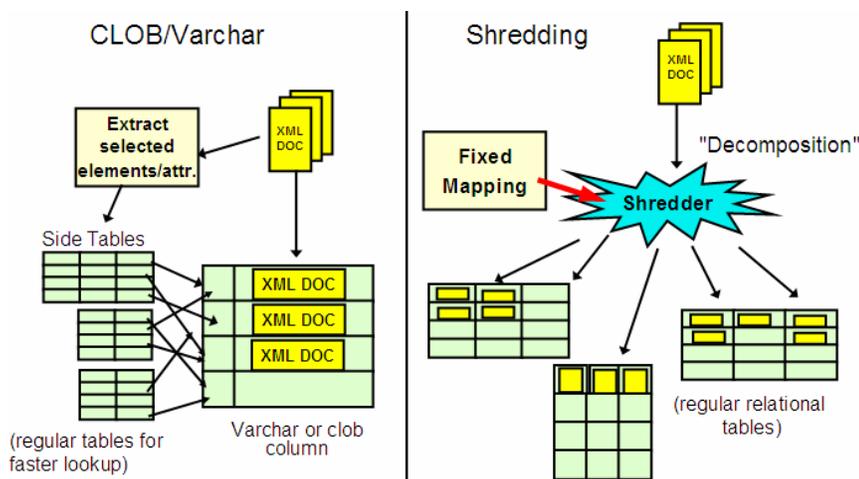
## 16.2 Bases de dados XML

Existem dois tipos de bases de dados que armazenam dados XML:

- Bases de dados XML-enabled
- Bases de dados de XML nativo

### 16.2.1 Bases de dados XML-enabled

Uma base de dados XML-enabled usa o modelo relacional como núcleo do modelo de armazenamento. Isto requer um mapeamento entre o modelo de dados XML (hierárquico) e o modelo de dados relacional, ou então armazena-se o dado XML como um character large object. Apesar de ser considerada uma tecnologia antiga, ainda está sendo utilizado por várias bases de dados. A Figura 16.2 explica com mais detalhes as duas opções para bases de dados XML-enabled.



**Figura 16.2 – Duas opções de armazenamento em bases de dados XML-enabled**

O lado esquerdo da Figura 16.2 mostra o método “CLOB e varchar” de armazenamento de documentos XML em uma base de dados. Utilizando este método, um documento XML é armazenado como uma string não analisada sintaticamente (*unparsed*) em um tipo CLOB ou varchar na base de dados. Se o documento XML é armazenado como uma string, quando você quiser recuperar parte dele, seu programa terá que recuperar a string e realizar a análise sintática para procurar o que você deseja. Este método não é muito flexível.

A outra opção para bases de dados XML-enabled é chamada fragmentação (*shredding*) ou decomposição e é ilustrada no lado direito da Figura 16.2. Usando este método, um documento XML é inteiramente fragmentado em partes pequenas que são armazenadas em tabelas. Ao utilizar este método, o modelo hierárquico de um documento XML é forçado ao modelo relacional. Isto não é bom em termos de flexibilidade: uma mudança no documento XML não é propagada facilmente para as tabelas correspondentes e pode haver a necessidade de criação de muitas outras tabelas. Este método também não é bom para o desempenho: se você precisa recuperar o documento XML de volta, você necessitará fazer uma operação SQL cara, que aumentará o custo ainda mais conforme existam mais joins entre tabelas.

### 16.2.2 Bases de dados de XML nativo

As bases de dados de XML nativo usam o modelo de dados hierárquico para armazenar e processar XML internamente. O formato de armazenamento é o mesmo que o de processamento: não há mapeamento para o modelo relacional, e documentos XML não são armazenados como imagens. Quando instruções XPath ou XQuery são utilizadas, elas são processadas nativamente, ao invés de serem convertidas para SQL. Por isso que estas bases de dados são conhecidas como base de dados de XML “nativo”. O DB2 a partir da versão 9 é atualmente o único servidor de dados que provê este suporte.

### 16.3 XML no DB2

A Figura 16.3 abaixo representa como os dados relacionais e hierárquico (documentos XML) são armazenados no DB2 a partir da versão 9. Na figura, assuma que a tabela `dept` é definida da seguinte maneira:

```
CREATE TABLE dept (deptID CHAR(8),..., deptdoc XML);
```

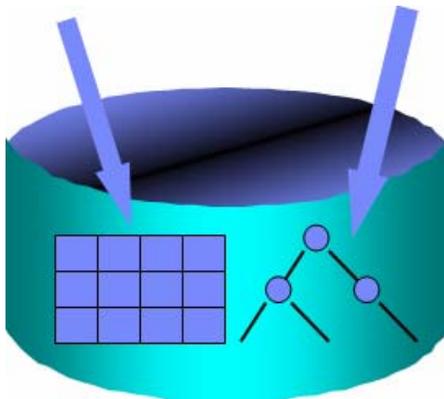


Figura 16.3 – XML no DB2

Note que a definição da tabela utiliza um novo tipo de dados, XML, para a coluna `deptdoc`. A seta à esquerda na figura indica a coluna relacional `deptID` armazenada no formato relacional (tabelas), enquanto a coluna `deptdoc` é armazenada em formato hierárquico. A Figura 16.4 ilustra que no DB2 a partir da versão 9, existem agora quatro maneiras de acessar dados:

- Utilizando SQL para acessar dados relacionais
- Utilizando SQL com extensões XML para acessar dados XML
- Utilizando XQuery para acessar dados XML
- Utilizando XQuery para acessar dados relacionais

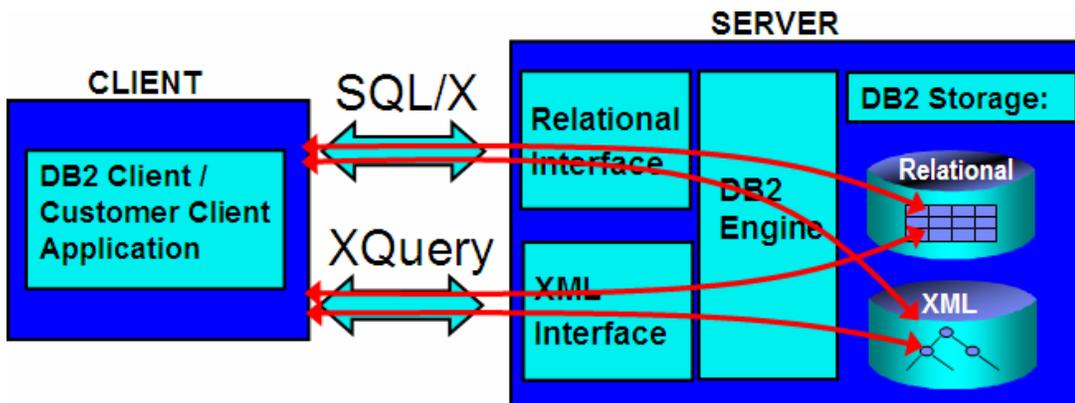


Figura 16.4 – Quatro maneiras de acessar dados no DB2

Portanto, enquanto uma pessoa pode ver o DB2 como um SGBD relacional de qualidade internacional que também suporta XML, outra pessoa veria o DB2 como um repositório XML de qualidade internacional que também suporta SQL.

Note que a IBM usa o termo pureXML em vez de “XML nativo” para descrever esta tecnologia. Outros SGBDs ainda usam tecnologias de CLOB/varchar ou fragmentação para armazenar documentos XML, mas chamam estas tecnologias antigas de “XML nativo”. Para evitar confusão, a IBM decidiu usar o novo termo pureXML e registrar este nome para que nenhuma outra empresa possa usar este mesmo termo para denotar alguma tecnologia diferente. No DB2 9.1, o suporte a pureXML foi oferecido apenas quando a base de dados é criada como uma base Unicode. Esta restrição não existe mais no DB2 9.5, onde você pode utilizar pureXML em uma base de dados unicode ou não.

### 16.3.1 Vantagens da tecnologia pureXML

Muitas vantagens são oferecidas através do uso da tecnologia pureXML.

1. Você pode impulsionar seu investimento relacional consistentemente, dado que 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 16.5 ilustra um script PHP escrito com e sem o uso de pureXML. Utilizando pureXML (retângulo verde) as linhas de código são reduzidas. Isto não significa apenas que o código é menos complexo, como também que o desempenho geral é melhor, pois existem menos linhas de código para serem analisadas e mantidas.

```

<?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 16.5 – Complexidade de código com e sem pureXML

- Mudanças no seu esquema são mais fáceis usando XML e a tecnologia pureXML. A Figura 16.6 ilustra um exemplo desse aumento de flexibilidade. Nessa figura, assuma que você tinha uma base de dados com as tabelas Employee e Department. Tipicamente em uma base de dados não-XML, se seu gerente pedisse que fosse armazenado não somente um número de telefone por empregado (o número do telefone de casa), mas também um segundo número de telefone (o número do telefone celular), então você precisaria adicionar uma coluna extra na tabela Employee e armazenar o número do telefone celular na nova coluna. Contudo, este método seria contra as regras de normalização de bases de dados relacionais. Se você quiser preservar estas regras, você deveria criar uma nova tabela Phone e mover toda a informação de telefone para esta tabela. Depois, você poderia adicionar os números de telefone celular. Criar uma tabela "Phone" é custoso, porque todo o código SQL de suas aplicações deverá ser mudado para apontar para a nova tabela.

Ao invés disso, no lado esquerdo da figura, mostramos como isso seria feito usando XML. Se a empregada "Christine" também possui um número de telefone celular, uma nova tag poderia ser adicionada para colocar esta informação. Se o empregado "Michael" não possui um número de telefone celular, apenas deixamos como está.

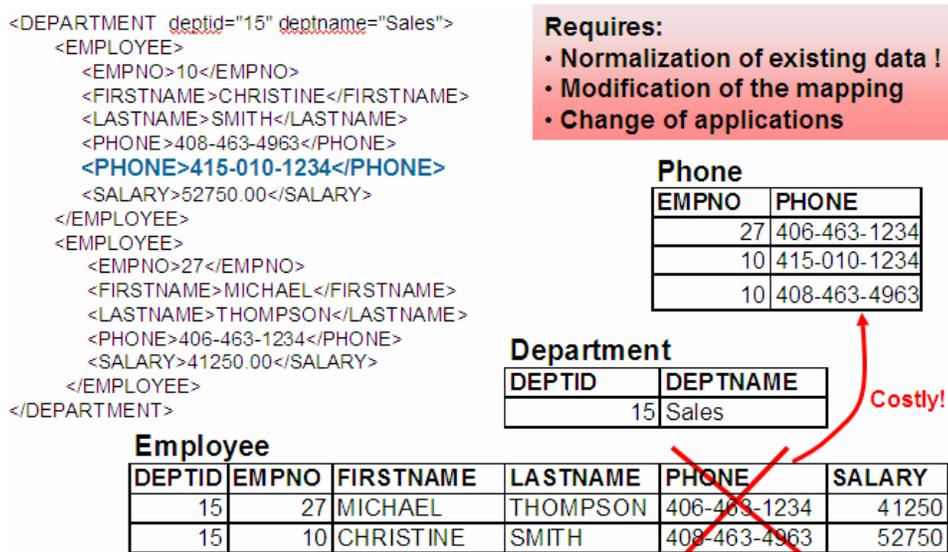


Figura 16.6 – Aumento de flexibilidade de dados usando XML

- Você pode aumentar o desempenho de sua aplicação XML. Testes realizados utilizando a tecnologia pureXML mostraram grande melhoria de desempenho para várias aplicações XML. A Figura 16.7 mostra o resultado dos testes para uma companhia que utilizava tecnologias antigas e mudou para pureXML. A coluna do

meio mostra os resultados utilizando o método antigo de trabalhar com XML e a terceira coluna mostra os resultados utilizando pureXML.

Tarefa	Outra base de dados relacional	DB2 com pureXML
Desenvolvimento de busca e recuperação de processos de negócios	CLOB: 8 h Fragmentação: 2 h	30 min
Linhas relativas a código de E/S	100	35 (redução de 65%)
Adicionar um campo ao esquema	1 semana	5 min
Consultas	24 - 36 h	20 segundos - 10 min

Figura 16.7 – Aumento de desempenho utilizando a tecnologia pureXML

### 16.3.2 Básico sobre XPath

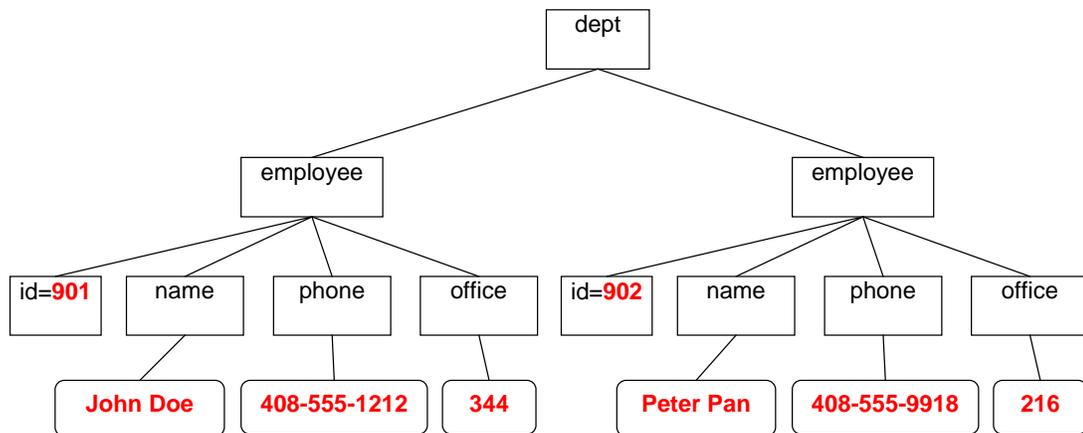
XPath é uma linguagem que pode ser utilizada para consultar documentos XML. A Figura 16.8 mostra um documento XML, e a Figura 16.9 ilustra o mesmo documento representado em um formato “parsed-hierarchical” (também chamado formato nó ou folha). Utilizaremos o formato “parsed-hierarchical” para explicar como funciona XPath.

```

<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>

```

Figura 16.8 – Um documento XML



**Figura 16.9 – Representação “parsed-hierarchical” do documento XML da Fig. 16.8**

Uma maneira rápida de aprender XPath é comparar ao comando `change directory` (`cd`) do MS-DOS ou Linux/UNIX. Usando o comando `cd` você navega por uma árvore de diretórios como se segue:

```
cd /directory1/directory2/...
```

De forma similar, no XPath usa-se barras para ir de um elemento para outro dentro do documento XML. Por exemplo, usando o documento da Figura 16.9, em XPath você poderia recuperar o nome de todos os empregados usando esta consulta:

```
/dept/employee/name
```

**Expressões XPath**

Expressões XPath usam caminhos totalmente qualificados para especificar elementos e/ou atributos. O sinal “@” é utilizado para especificar um atributo. Para recuperar apenas um valor (nó texto) de um elemento, utilize a função “text()”. A tabela 16.1 mostra consultas XPath e os resultados correspondentes utilizando o documento XML da Figura 16.9.

XPath	Resultado
/dept/@bldg	101
/dept/employee/@id	901 902
/dept/employee/name	<name>Peter Pan</name> <name>John Doe</name>
/dept/employee/name/text()	Peter Pan John Doe

**Tabela 16.1 – Exemplos de expressões XPath****Caracteres curinga do XPath**

Existem dois caracteres curinga principais no XPath:

- “\*” corresponde a qualquer nome de tag
- “//” é o curinga para “descendente-ou-própria” tag

A Tabela 16.2 provê mais exemplos utilizando o documento XML da Figura 16.9

XPath	Resultado
/dept/employee/*/text()	John Doe 408 555 1212 344 Peter Pan 408 555 9918 216
/dept*/@id	901 902
//name/text()	Peter Pan John Doe
/dept//phone	<phone>408 555 1212</phone> <phone>408 555 9918</phone>

**Tabela 16.2 – Exemplos com caracteres curinga do XPath**

**Predicados XPath**

Predicados são delimitados por colchetes [ ]. Como uma analogia, você pode imaginá-los como o equivalente à cláusula WHERE em SQL. Por exemplo, [*@id="902"*] pode ser lido como: “WHERE o atributo id é igual a 902”. Podem haver muitos predicados em uma expressão XPath. Para especificar um predicado posicional, utilize [n], que significa o n-ésimo filho a ser selecionado. Por exemplo, *employee[2]* significa que o segundo empregado deve ser selecionado. A Tabela 16.3 provê mais exemplos.

XPath	Resultado
/dept/employee[ <i>@id="902"</i> ]/name	<name>Peter Pan</name>
/dept[ <i>@bldg="101"</i> ]/employee[ <i>office &gt;"300"</i> ]/name	<name>John Doe</name>
//employee[ <i>office="344" OR office="216"</i> ]/@id	901 902
/dept/employee[2]/@id	902

**Tabela 16.3 – Exemplos de predicado XPath**

**XPath: o eixo pai**

Similar ao MS-DOS ou Linux/UNIX, você pode utilizar “.” (ponto) para indicar na expressão que você se refere ao contexto atual, e “..” (ponto ponto) para referenciar ao contexto pai. A Tabela 16.4 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 16.4 – O eixo pai no XPath

### 16.3.3 XQuery definida

XQuery é uma linguagem de consulta criada para XML que suporta expressões para navegar na estrutura hierárquica do XML. Na verdade, XPath é um subconjunto de XQuery; portanto, tudo que foi apresentado anteriormente sobre XPath também se aplica a XQuery. XQuery suporta tanto dados tipados quanto não-tipados. XQuery não possui valores nulos porque documentos XML omitem dados desconhecidos ou que estão faltando. XQuery retorna seqüências de dados XML. É importante salientar que expressões XQuery e XPath são case sensitive.

XQuery suporta a expressão FLWOR. Se fizermos analogia com SQL, é equivalente à expressão SELECT-FROM-WHERE. A próxima seção descreve FLWOR com mais detalhes.

### XQuery: expressão FLWOR

FLWOR significa:

- FOR: faz iteração através de uma seqüência, vincula uma variável a itens
- LET: vincula uma variável a uma seqüência
- WHERE: elimina itens da iteração
- ORDER: reordena itens da iteração
- RETURN: constrói resultados da consulta

É uma expressão que permite manipulação de documentos XML, permitindo que outra expressão seja retornada. Por exemplo, assumo que você possui uma tabela com a definição:

```
CREATE TABLE dept(deptID CHAR(8),deptdoc XML);
```

E o seguinte documento XML 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>

```

Então a seguinte instrução XQuery que utiliza a expressão FLWOR pode 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>

```

E irá retornar o seguinte:

```

<EmpList bldg="101">
  <name>
    John Doe
  </name>
  <name>
    Peter Pan
  </name>
</EmpList>

```

#### 16.3.4 Inserindo documentos XML

Pode-se inserir documentos XML em uma base de dados DB2 utilizando a instrução INSERT de SQL, ou o utilitário IMPORT. XQuery não pode ser usada para este propósito pois ela ainda não foi definida no padrão.

Vamos examinar o seguinte script, que pode ser executado a partir do DB2 Command Window ou shell do Linux utilizando a seguinte instrução:

```
db2 -tvf table_creation.txt
```

##### 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)
;

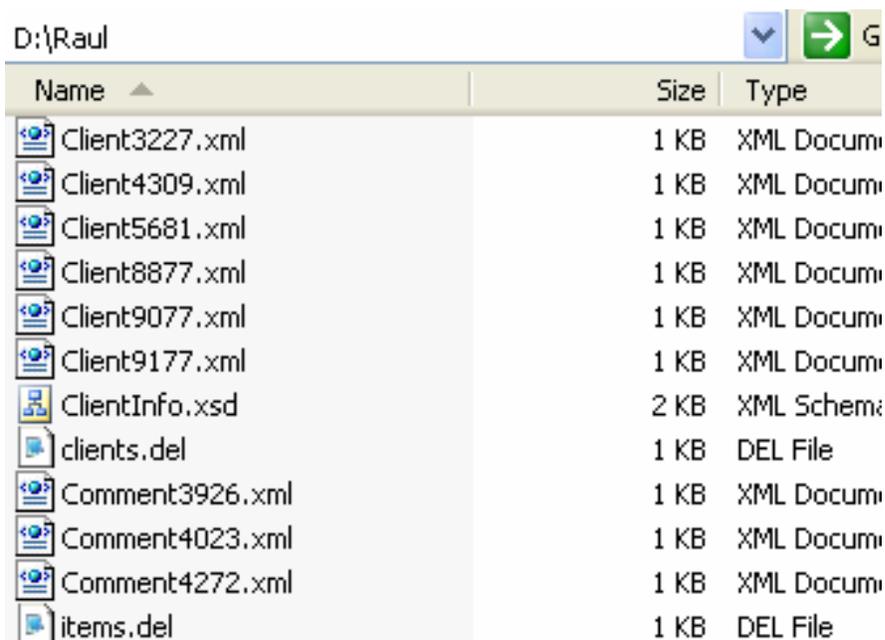
-- (8)
IMPORT FROM "D:\Raul\items.del" of del xml from "D:\Raul" INSERT
INTO ITEMS (ID, BRANDNAME, ITEMNAME, SKU, SRP, COMMENTS)
;
```

Note que este arquivo de scripts e arquivos relacionados estão contidos no arquivo compactado **expressc\_book\_quicklabs.zip** que acompanha este livro. A seguir, descrevemos cada linha do script.

1. Exclui a base de dados "mydb". Isto é normalmente feito em arquivos de script para limpar dados já existentes. Se "mydb" não existia anteriormente, você irá receber uma mensagem de erro, mas isto está OK.

2. Cria a base de dados "mydb" utilizando o codeset UTF-8. Uma base de dados UNICODE é requerida para suportar pureXML, então este passo é necessário para que a base seja unicode.
3. Conecta-se à base de dados "mydb" recentemente criada. Isto é necessário para criar objetos dentro da base de dados.
4. Cria a tabela "items". Note que a última coluna da tabela (coluna "comments") é definida como uma coluna XML usando o novo tipo de dados XML.
5. A tabela "clients" é criada. Note que a última coluna da tabela (coluna "contact") também é definida com o novo tipo de dados XML.
6. Utilizando esta instrução SQL INSERT, você pode inserir um documento XML em uma coluna XML. Na instrução INSERT você passa o documento XML como uma string delimitada por aspas simples.
7. Utilizando um comando IMPORT, você pode inserir ou importar vários documentos XML além de dados relacionais dentro da base de dados. Em (7) você está importando os dados do arquivo clients.del (um arquivo ascii delimitado), e você pode indicar onde o dado XML referenciado pelo arquivo clients.del está localizado (neste exemplo, em D:\Raul).

Iremos observar mais detalhadamente o arquivo clients.del, mas primeiramente, vamos ver o conteúdo do diretório D:\Raul (Figura 16.10).



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 16.10 – Conteúdo do diretório D:\Raul com documentos XML

Este é o conteúdo do arquivo clients.del:

**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' />
    
```

No arquivo clients.del, “XDS FIL=” é usado para apontar para um documento XML específico.

A Figura 16.11 mostra o Control Center após a execução do script acima.

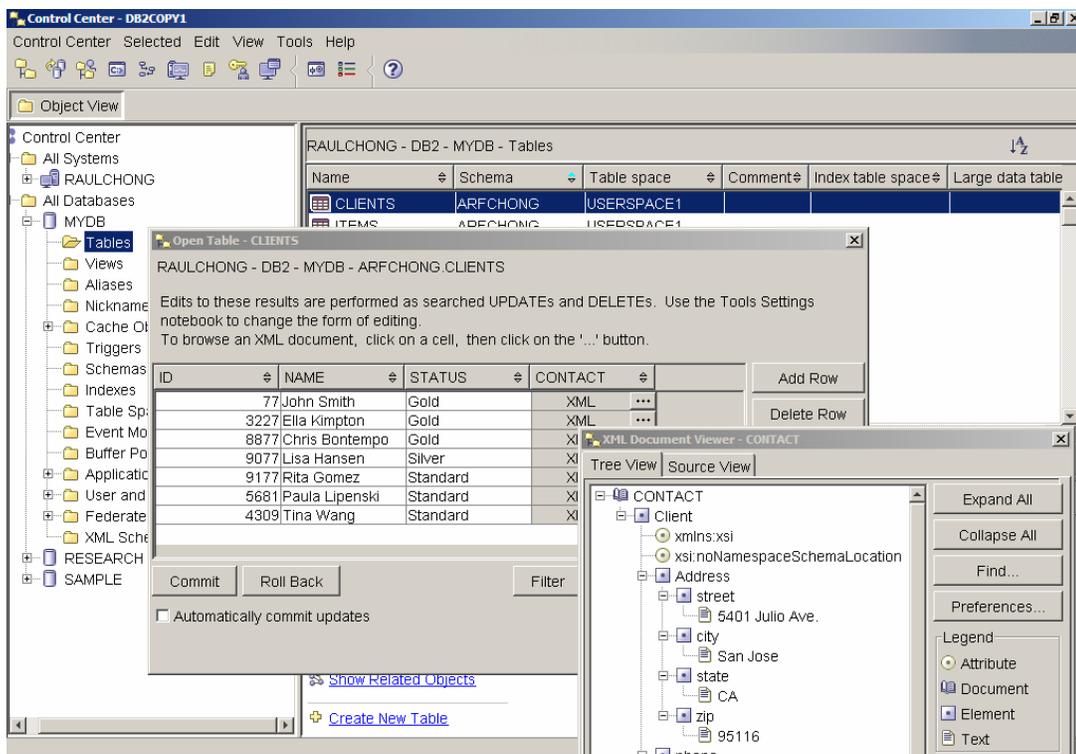


Figura 16.11 – Control Center após a execução de table\_creation.txt

Note que na figura é mostrado o conteúdo da tabela CLIENTS. A última coluna “Contact” é uma coluna XML. Quando se clica no botão com três pontos, outra janela é aberta, mostrando o conteúdo do documento XML, conforme mostrado no canto inferior direito da Figura 16.11.

### 16.3.5 Consultando dados XML

Existem duas maneiras de consultar dados XML no DB2:

- Usando SQL com extensões XML (SQL/XML)
- Usando XQuery

Em ambos os casos, o DB2 segue padrões XML internacionais.

### Consultando dados XML com SQL/XML

Instruções SQL simples permitem que você trabalhe com linhas e colunas. Uma instrução SQL pode ser utilizada para trabalhar com documentos XML completos; contudo, não irá auxiliar na tentativa de recuperar apenas parte do documento. Neste caso, você precisará utilizar SQL com extensões XML (SQL/XML).

A Tabela 16.5 descreve algumas funções SQL/XML disponíveis no padrão SQL 2006.

Nome da função	Descrição
XMLPARSE	Analisa caracteres ou dados BLOB, produzindo um valor XML
XMLSERIALIZE	Converte um valor XML para caracter ou BLOB
XMLVALIDATE	Valida valor XML sobre um XML esquema e faz anotações de tipo no valor XML
XMLEXISTS	Determina se uma XQuery retorna resultado (i.e. uma seqüência de um ou mais itens)
XMLQUERY	Executa uma XQuery e retorna a seqüência resultante
XMLTABLE	Executa uma XQuery e retorna a seqüência resultante como uma tabela relacional (se possível)
XMLCAST	Converte a partir de um tipo XML ou para um tipo XML

**Tabela 16.5 – Funções SQL/XML**

Os seguintes exemplos podem ser testados utilizando a base de dados “mydb” criada anteriormente.

#### Exemplo 1

Este é um exemplo de uma consulta modelo. Imagine que você precisa localizar o nome de todos os clientes que vivem em uma região com um zip code específico. A tabela “clients” armazena endereços de clientes, incluindo zip codes, em uma coluna XML. Utilizando XMLEXISTS, você pode buscar pelo zip code na coluna XML e restringir o resultado desejado.

```
SELECT name FROM clients
WHERE xmlexists(
  '$c/Client/Address[zip="95116"]'
  passing clients.contact as "c"
)
```

A primeira linha é uma cláusula SQL que especifica que você quer recuperar informação da coluna “name” da tabela “clients”.

A cláusula WHERE invoca a função XMLEXISTS, especificando a expressão XPath que leva o DB2 a navegar ao elemento “zip” e verificar o valor de 95116.

A cláusula “\$c/Client/Address” indica o caminho dentro da hierarquia do documento XML onde o DB2 pode localizar o elemento “zip”. O sinal de cifrão (\$) é utilizado para especificar uma variável; portanto “c” é uma variável. Esta variável é definida na linha: `passing clients.contact as "c"`, onde “clients” é o nome da tabela e “contact” é o nome da coluna que possui um tipo de dados XML. Em outras palavras, estamos passando um documento XML para a variável “c”.

O DB2 inspeciona o dado XML contido na coluna “contact”, navega a partir do nó raiz “Client” até o nó “Address” e depois para o nó “zip” e finalmente determina se o cliente mora no zip code desejado. A função XMLEXISTS avalia em “verdadeiro” e DB2 retorna o nome do cliente associado a tal linha.

No DB2 9.5, a consulta acima pode ser simplificada da seguinte forma:

```
SELECT name FROM clients
WHERE xmlexists(
    '$CONTACT/Client/Address[zip="95116"]'
)
```

Uma variável com o mesmo nome que a 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 CONTACT do XML.

#### Exemplo 2

Vamos considerar o problema de como criar um relatório listando o endereço de e-mail dos clientes classificados como “Gold”. A seguinte consulta pode ser realizada:

```
SELECT xmlquery('$c/Client/email' passing contact as "c")
FROM clients
WHERE status = 'Gold'
```

A primeira linha indica que queremos que seja retornado o endereço de e-mail, o qual é um elemento do documento XML (e não uma coluna relacional). Como no exemplo anterior, “\$c” é uma variável que contém o documento XML. Neste exemplo, utilizamos a função XMLQUERY que pode ser utilizada após um SELECT, enquanto a função XMLEXISTS pode ser utilizada após uma cláusula WHERE.

#### Exemplo 3

Podem existir situações em que você prefira apresentar dados XML como tabelas. Isto é possível com a função XMLTABLE, como mostrado no exemplo 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
```

A primeira linha especifica a coluna que deve ser incluída no resultado. Colunas que possuem a variável “t” como prefixo são baseadas em valores de elementos XML.

A segunda linha invoca a função XMLTABLE para especificar a coluna XML que contém o dado alvo (“i.comments”) e o caminho dentro da coluna de documento XML onde os elementos de interesse estão localizados.

A cláusula “columns”, nas linhas 4 a 6, identifica os elementos XML específicos que serão mapeados para colunas de saída no resultado especificado na linha 1. Parte desse mapeamento envolve especificar os tipos de dados que os valores de elementos XML serão convertidos. Neste exemplo, todos os dados XML são convertidos para tipos de dados tradicionais do SQL.

#### Exemplo 4

Agora, vamos explorar um exemplo simples no qual uma expressão XQuery FLWOR é inserida em uma função XMLQUERY SQL/XML.

```
SELECT name, xmlquery(
  'for $e in $c/Client/email[1] return $e'
  passing contact as "c"
)
FROM clients
WHERE status = 'Gold'
```

A primeira linha especifica que o nome dos clientes e a saída da função XMLQUERY serão incluídas no resultado. A segunda linha indica que o primeiro sub-elemento “email” do elemento “Client” deve ser retornado. A terceira linha identifica a fonte do dado XML (coluna “contact”). A quarta linha informa que esta coluna é referente à tabela “clients”; e a quinta linha indica que apenas os clientes “Gold” são de interesse.

#### Exemplo 5

Este exemplo demonstra novamente a função XMLQUERY, que possui uma expressão XQuery FLWOR, mas note que desta vez não estamos retornando somente 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'
```

A cláusula de retorno de uma XQuery permite transformar a saída XML conforme desejado. O uso da função text() na primeira linha indica que apenas a representação textual do primeiro endereço de e-mail é de interesse. A segunda linha especifica que esta informação deve ser cercada por tags de parágrafo do HTML.

#### Exemplo 6

O seguinte exemplo utiliza a função XMLELEMENT para criar uma série de elementos, que contêm sub-elementos para os valores de ID, brand name e stock keeping unit (SKU) obtidos das colunas correspondentes da tabela “items”. Basicamente, você pode usar a função XMLELEMENT quando você deseja converter dado relacional para dado XML.

```
SELECT
  xmlelement (name "item", itemname),
  xmlelement (name "id", id),
  xmlelement (name "brand", brandname),
  xmlelement (name "sku", sku)
FROM items
WHERE srp < 100
```

A consulta acima irá gerar a seguinte saída:

```
<item>
  <id>4272</id>
  <brand>Classy</brand>
  <sku>981140</sku>
</item>
...
<item>
  <id>1193</id>
  <brand>Natural</brand>
  <sku>557813</sku>
</item>
```

## Consultando dado XML com XQuery

Na seção anterior, vimos como consultar dados XML utilizando SQL com extensões XML. SQL foi sempre o método de consulta primário e XPath foi embutido no SQL. Nesta seção, discutiremos como consultar dados XML utilizando SQL embutido em XQuery (usando a função “db2-fn:sqlquery”). Quando utilizarmos XQuery, iremos invocar algumas funções e também utilizar a expressão FLWOR.

### Exemplo 1

Uma XQuery simples para retornar dados de contato de cliente.

```
xquery db2-fn:xmlcolumn('CLIENTS.CONTACT')
```

Sempre inicie qualquer expressão XQuery com o comando “xquery” para que o DB2 saiba que ele deve usar um analisador XQuery, caso contrário o DB2 irá assumir que você está tentando executar uma expressão SQL. A função **db2-fn:xmlcolumn** retorna documentos XML da coluna especificada como parâmetro. É equivalente à seguinte instrução SQL, que recupera o conteúdo de todas as colunas.

```
SELECT contact FROM clients
```

### Exemplo 2

Neste exemplo, usaremos a expressão FLWOR para recuperar dados de fax de clientes.

```
xquery
  for $y in db2-fn:xmlcolumn('CLIENTS.CONTACT')/Client/fax
  return $y
```

A primeira linha invoca o analisador XQuery. A segunda linha instrui o DB2 a interar os sub-elementos fax contidos na coluna CLIENTS.CONTACT. Cada elemento fax é atribuído à variável \$y. A terceira linha indica que, para cada iteração, o valor de “\$y” é retornado.

A saída desta consulta é similar à mostrada abaixo (pode ser incluído o namespace por padrão, mas não mostramos isso abaixo, caso contrário a saída seria mais difícil de ser lida, pois seria separada em várias linhas):

```
<fax>4081112222</fax>
<fax>5559998888</fax>
```

### Exemplo 3

Este exemplo consulta dados XML e retorna o resultado como HTML.

```
xquery
  <ul> {
    for $y in db2-fn:xmlcolumn('CLIENTS.CONTACT')/Client/Address
    order by $y/zip
    return <li>{$y}</li>
  }
</ul>
```

O HTML modelo retornado é similar ao mostrado abaixo:

```
<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>
```

### Exemplo 4

O seguinte exemplo mostra como embutir SQL em XQuery utilizando a função db2-fn:sqlquery. A função db2-fn:sqlquery executa uma consulta SQL e retorna apenas os

dados XML selecionados. A consulta SQL passada para db2-fn:sqlquery deve retornar apenas dados XML, que serão futuramente processados por XQuery.

```
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>
  )
```

No exemplo, a consulta SQL filtra linhas baseadas na condição que a coluna “srp” tenha um valor maior que 100. A partir das linhas filtradas, a coluna “comments”, que é uma coluna XML, será utilizada. Em seguida, XQuery (ou XPath) é aplicada para navegar aos sub-elementos de “comments”.

**Nota:** O DB2 não é case sensitive e trata todos os nomes de tabelas e colunas em maiúsculas; porém o XQuery é case sensitive. As funções de recurso acima são funções de interface XQuery, logo todos os nomes de tabelas e colunas devem ser passados para estas funções em maiúsculas. Passar o nome de objetos em letras minúsculas pode resultar em um erro de nome de objeto indefinido.

### 16.3.6 Joins utilizando SQL/XML

Esta seção descreve como realizar operações JOIN entre duas colunas XML de tabelas diferentes ou entre uma coluna XML e uma coluna relacional. Suponha que você tenha criado duas tabelas com estas instruções:

```
CREATE TABLE dept (unitID CHAR(8), deptdoc XML)

CREATE TABLE unit (unitID CHAR(8) primary key not null,
                   name CHAR(20),
                   manager VARCHAR(20),
                   ...
                  )
```

Você pode fazer uma operação JOIN de duas maneiras:

#### Método 1:

```
SELECT u.unitID
  FROM dept d, unit u
 WHERE XMLEXISTS (
   '$e//employee[name = $m]'
   passing d.deptdoc as "e", u.manager as "m")
```

A linha 3 desta instrução mostra que a operação JOIN ocorre entre o elemento “name”, que é um sub-elemento da coluna XML “deptdoc” na tabela “dept”, e a coluna relacional “manager” da tabela “unit”.

#### Método 2:

```
SELECT u.unitID
FROM dept d, unit u
WHERE u.manager = XMLCAST(
  XMLQUERY('$e//employee/name '
    passing d.deptdoc as "e")
  AS char(20))
```

Neste método alternativo, a coluna relacional está do lado esquerdo do JOIN. Se a coluna relacional está do lado esquerdo ou em equação, o índice relacional pode ser usado em vez de um índice XML.

### 16.3.7 Joins utilizando XQuery

Suponha que as seguintes tabelas foram criadas:

```
CREATE TABLE dept(unitID CHAR(8), deptdoc XML)
CREATE TABLE project(projectDoc XML)
```

Se utilizarmos SQL/XML, a consulta seria:

```
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")
```

A consulta equivalente utilizando XQuery seria:

```
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
```

Este segundo método é mais fácil de interpretar – a variável “\$dept” guarda o documento XML da coluna XML “deptdoc” da tabela “dept”. A variável “\$proj” guarda o documento XML da coluna XML “projectdoc” da tabela “project”. A linha 4 realiza a operação JOIN entre um atributo do primeiro documento XML e um elemento do segundo documento XML.

### 16.3.8 Operações de atualização e exclusão

Operações de atualização e exclusão em dados XML podem ser realizadas de duas maneiras:

- ▶ Utilizando as instruções SQL UPDATE e DELETE
- ▶ Utilizando a expressão TRANSFORM

Da primeira maneira, utilizando instruções SQL UPDATE e DELETE, a atualização ou exclusão ocorre no nível do documento, isto é, o documento XML por completo é substituído por um atualizado. Por exemplo, se no exemplo abaixo desejamos alterar o elemento <state>, todo o documento XML 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
```

Na segunda alternativa, você pode executar atualizações em sub-documentos utilizando a expressão TRANSFORM, que é bem mais eficiente. Ela permite que você substitua, adicione, exclua ou renomeie nós em um documento XML. Você também pode mudar o valor de um nó sem substituir o próprio nó, tipicamente para mudar um elemento ou valor de um atributo – que é um tipo de atualização muito comum. Isto é novo no DB2 9.5.

A expressão TRANSFORM é parte da linguagem XQuery, e você pode utilizá-la onde normalmente você usa XQuery, por exemplo em uma expressão FLWOR ou na função XMLQUERY de uma instrução SQL/XML. O uso mais comum de uma instrução SQL UPDATE é para modificar o documento XML de uma coluna XML.

Abaixo se encontra 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-----|
```

---

A cláusula `copy` é utilizada para atribuir a uma variável os documentos XML que se deseja processar. Na cláusula `modify`, você pode invocar uma expressão `insert`, `delete`, `rename` ou `replace`. Estas expressões permitem que atualizações possam ser realizadas em um documento XML. Por exemplo, se você deseja adicionar novos nós ao documento, você usaria a expressão `insert`, para excluir nós de um documento XML, use a expressão `delete`, para renomear um elemento ou atributo em um documento XML, use a expressão `rename`, e para substituir um nó existente por um novo nó ou uma seqüência de nós, use a expressão `replace`. O valor substituído de uma expressão somente pode ser usado para mudar o valor de um elemento ou atributo. A cláusula `return` retorna o resultado da expressão `TRANSFORM`.

Este é um exemplo de instrução `UPDATE` utilizando 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
```

---

No exemplo acima, as linhas (1), (2), e (9) são parte da sintaxe do SQL `UPDATE`. Na linha (2), a função `XMLQUERY` é invocada, chamando a expressão `TRANSFORM` na linha (4). Esta expressão vai da linha (4) até a linha (8), e é usada para inserir um novo nó contendo o elemento `email2` dentro do documento XML. Note que não há suporte para atualização de um documento XML através de uma visão.

Excluir documentos XML inteiros de tabelas é tão simples quanto utilizar uma instrução `SELECT` em SQL/XML. Utilize a instrução SQL `DELETE` e especifique o predicado `WHERE` caso necessário.

### 16.3.9 Índice em XML

Em um documento XML, índices podem ser criados para elementos, atributos, ou para valores (nós textuais). Alguns exemplos são apresentados a seguir. Assuma que a tabela abaixo foi criada.

```
CREATE TABLE customer(info XML)
```

E suponha que este é um dos documentos XML armazenados:

```
<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>

```

1) Esta instrução cria um índice no atributo “Cid”

```

CREATE UNIQUE INDEX idx1 ON customer(info)
  GENERATE KEY USING
  xmlpattern '/customerinfo/@Cid'
  AS sql DOUBLE

```

2) Esta instrução cria um índice no elemento “name”

```

CREATE INDEX idx2 ON customer(info)
  GENERATE KEY USING
  xmlpattern '/customerinfo/name'
  AS sql VARCHAR(40)

```

3) Esta instrução cria um índice em todos os elementos “name”

```

CREATE INDEX idx3 ON customer(info)
  GENERATE KEY USING
  xmlpattern '//name'
  AS sql VARCHAR(40);

```

4) Esta instrução cria um índice em todos os nós textuais (todos os valores). Isto não é recomendado, pois pode ser muito caro para manter os índices para operações de atualização, exclusão e inserção e o índice seria muito extenso.

```

CREATE INDEX idx4 ON customer(info)
  GENERATE KEY USING
  xmlpattern '//text()'
  AS sql VARCHAR(40);

```

---

## QuickLab #14 - SQL/XML e XQuery

### Objetivo

Você viu vários exemplos da sintaxe de SQL/XML e XQuery e também conheceu o DB2 Command Editor e o IBM Data Studio. Neste quicklab, você irá testar seus conhecimentos sobre SQL/XML e XQuery e ganhar experiência com estas ferramentas. Iremos utilizar a base de dados "mydb" criada usando o arquivo de script `table_creation.txt` explicado anteriormente neste capítulo.

### Procedimento

1. Crie a base de dados "mydb" e carregue-a com os dados XML, como discutido anteriormente no capítulo.
2. Utilizando o Command Editor ou o IBM Data Studio:
  - a) Recupere todos os documentos XML da coluna comments da tabela ITEMS de duas maneiras, mas usando somente XQuery.
  - b) Por que a execução do comando SQL abaixo não resulta na mesma saída?  

```
SELECT comments FROM items
```
  - c) Recupere o ID e o BRANDNAME para os registros cujo documento XML possua um elemento ResponseRequested com o valor "No".

---

### SOLUÇÕES:

2a)

```
xquery db2-fn:xmlcolumn('ITEMS.COMMENTS')
xquery db2-fn:sqlquery("select comments from
items")
```

2b)

A saída é diferente porque SQL retorna valores NULL quando um valor não está presente, enquanto XQuery não retorna nada.

2c)

```
SELECT id, brandname FROM items WHERE
XMLEXISTS('$c/Comments/Comment[ResponseRequested="No"]'
passing ITEMS.COMMENTS as "c
```

# 17

## Capítulo 17 – Desenvolvendo com Java, PHP, e Ruby

Este capítulo discute o básico sobre o desenvolvimento de aplicações em Java, PHP e Ruby on Rails utilizando um servidor DB2. O propósito deste capítulo não é ensinar estas linguagens, mas prover informação pertinente para usá-las com o DB2.

**Nota:**

Para maiores informações sobre este tópico, assista ao vídeo:

<http://www.channeldb2.com/video/video/show?id=807741:Video:4402>

### 17.1 Desenvolvimento de aplicações em Java

O driver IBM DB2 para JDBC (também conhecido como driver JCC) tem sido otimizado para todos os servidores DB2 em todas as plataformas. O arquivo jar db2jcc.jar (com.ibm.db2.jcc) inclui os drivers tipo 2 e tipo 4. O arquivo db2jcc.jar é incluído em qualquer cliente DB2 ou pode ser obtido a partir do website do DB2 Express-C (ibm.com/db2/express), com o nome “IBM DB2 Driver for JDBC and SQLJ”.

#### 17.1.1 Driver JDBC tipo 2

O driver JDBC tipo 2 têm como pré-requisito que um cliente DB2 esteja instalado onde a aplicação JDBC está rodando. A Figura 17.1 ilustra uma aplicação JDBC usando o driver tipo 2.

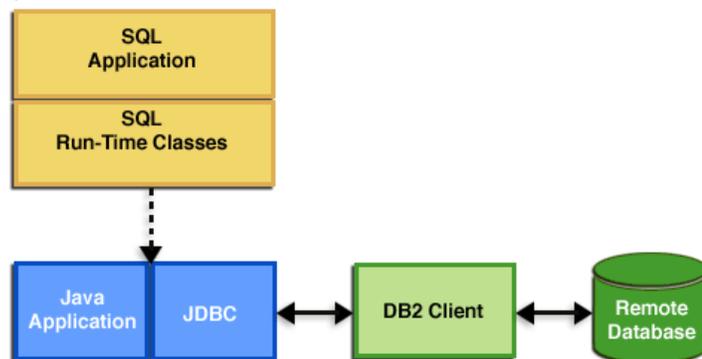


Figura 17.1 – O driver JDBC tipo 2

A Figura 17.2 provê uma listagem que mostra como estabelecer conexão utilizando o driver JDBC tipo 2. Note que a URL não inclui informações sobre hostname nem porta porque ela foi obtida do cliente DB2.

```

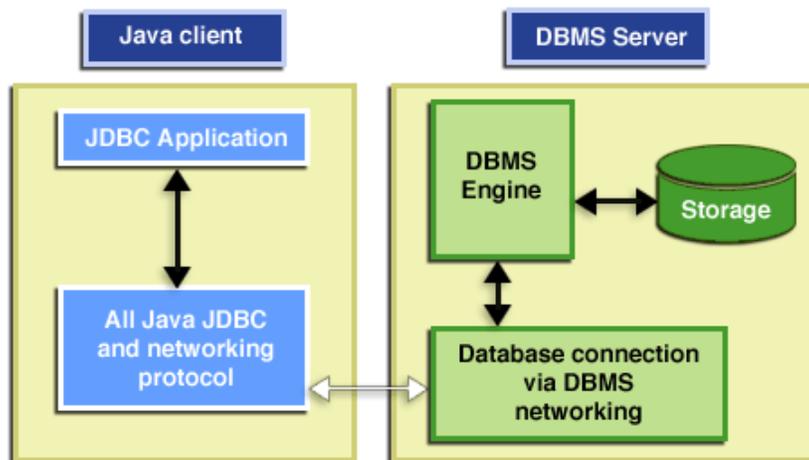
...
public static final String DB_URL = "jdbc:db2:sample";
Properties connectProperties = new Properties();
connectProperties.put("user", "db2admin");
connectProperties.put("password", "ibmdb2");
Connection connection = null
try
{
    Class.forName("com.ibm.db2.jcc.DB2Driver").newInstance();
    connection = DriverManager.getConnection(url, connectProperties)
}
catch (Exception e)
throw e;
}
...

```

**Figura 17.2 – Estabelecendo uma conexão usando o driver JDBC tipo 2**

#### 17.1.2 Driver JDBC tipo 4

O driver JDBC tipo 4 não requer um cliente DB2 para conectar a um servidor DB2. A Figura 17.3 ilustra uma aplicação JDBC usando o driver tipo 4.



**Figura 17.3 – O driver JDBC tipo 4**

A Figura 17.4 provê uma listagem mostrando como estabelecer uma conexão usando o driver JDBC tipo 4. Note que a URL não inclui informações sobre o hostname nem sobre a porta.

```
...
public static final String DB_URL = "jdbc:db2://server1:50000/sample";
Properties connectProperties = new Properties();
connectProperties.put("user", "db2admin");
connectProperties.put("password", "ibmdb2");
Connection connection = null
try
{
    Class.forName("com.ibm.db2.jcc.DB2Driver").newInstance();
    connection = DriverManager.getConnection(url,connectProperties)
}
catch (Exception e)
    throw e;
}
...
```

Figura 17.4 – Estabelecendo uma conexão usando o driver JDBC tipo 4

## 17.2 Desenvolvimento de aplicações em PHP

PHP (PHP Hypertext Preprocessor) é uma linguagem de scripts open source e independente de plataformas, projetada para desenvolvimento de aplicações Web. É uma das linguagens mais utilizadas no mundo atualmente. A popularidade do PHP é baseada nas seguintes características da linguagem:

- Desenvolvimento em ciclos rápidos e iterativos, com uma curva de aprendizado pequena.
- Robusta, com alto desempenho e escalável.
- Estável e segura.
- Uma alternativa para J2EE™ e .NET na Web.
- Facilmente integrada em ambientes e sistemas heterogêneos.
- Provada através de amplo posicionamento estratégico.
- Bem estabelecida na comunidade.

PHP é parte da pilha LAMP (Linux, servidor HTTP Apache, MySQL, **PHP** / Perl / Python), a qual é uma pilha de tecnologias open source na web, geralmente disponível em ISPs por taxas mensais razoáveis.

### 17.2.1 Opções de conexão do DB2 para PHP

A IBM suporta acesso de aplicações PHP a uma base de dados DB2 através de duas extensões.

#### **ibm\_db2:**

A extensão `ibm_db2` oferece uma API procedural para criar, ler, atualizar e escrever operações de base de dados com inclusão de acesso extensivo aos meta-dados da base de dados. Ela pode ser compilada para funcionar tanto com o PHP 4 quanto o PHP 5. A extensão está disponível no repositório PECL sob a licença Apache 2.0. Ela foi

desenvolvida e é suportada pela IBM. Possui suporte completo a características de stored procedures e LOBs e é rápida, pois foi otimizada para o DB2.

#### **PDO\_ODBC:**

O PDO\_ODBC é um driver para a extensão PHP Data Objects (PDO) e oferece acesso para bases de dados DB2 através de uma interface padrão orientada a objetos para base de dados introduzida no PHP 5.1. Pode ser compilada diretamente sobre bibliotecas DB2. Provê uma interface padrão de acesso a dados para o PHP. É rápida, leve e orientada a objetos. A extensão PDO\_ODBC utiliza bibliotecas do DB2 para acesso nativo, e foi construída no PHP 5.1. Para maiores informações, visite estes sites:

- <http://pecl.php.net/package/pdo>
- [http://pecl.php.net/package/PDO\\_ODBC](http://pecl.php.net/package/PDO_ODBC)

### **Conectando a uma base de dados não catalogada no DB2**

A Listagem 17.1 mostra como conectar uma base de dados do DB2 utilizando uma das duas extensões descritas anteriormente.

```
$host = 'localhost';
$port = 50000;
$DSN = "DRIVER={IBM DB2 ODBC DRIVER}; PORT=$port;
        HOSTNAME=$host; DATABASE=$database; PROTOCOL=TCPIP;
        USER=$user; PWD=$password";

-- If using the ibm_db2 extension --
$suconn = db2_connect($DSN, null, null);

-- If using the PDO_ODBC extension --
try {
    $suconn = new PDO("odbc:$DSN", null, null);
}
catch (PDOException $e) { print $e->errormsg(); }
```

#### **Listagem 17.1 – Conectando a uma base de dados não catalogada no DB2**

A Listagem 17.2 provê um exemplo de uma aplicação PHP simples utilizando a extensão `ibm_db2`.

```
<?php
$sql = "SELECT name, breed FROM ANIMALS WHERE weight < ?";
$conn = db2_connect($database, $user, $password);
$stmt = db2_prepare($conn, $sql);
$res = db2_execute($stmt, array(10));
while ($row = db2_fetch_assoc($stmt)) {
    print "{$row['NAME']} is a {$row['BREED']}. \n";
}
?>
```

#### **Listagem 17.2 – Uma aplicação PHP simples usando a extensão `ibm_db2`**

### Configurando o PHP para ibm\_db2

No Linux ou no UNIX, você pode precisar modificar o arquivo php.ini, como segue:

```
extension=ibm_db2.so  
ibm_db2.instance_name=<instance name>
```

No Windows, modifique o arquivo php.ini como mostrado abaixo:

```
extension=php_ibm_db2.dll
```

Alternativamente, você pode fazer o download e instalar a suite **Zend Core for IBM**, descrita na próxima seção, não precisando realizar estas configurações.

#### 17.2.2 Zend Core for IBM

O Zend Core é um ambiente de desenvolvimento e produção consistente e surpreendente para aplicações Web críticas para negócios. Possui confiabilidade, produtividade e flexibilidade necessárias para executar aplicações PHP. Pode ser baixado sem custos pelo site <http://ibm.com/software/data/info/ZendCore>.

O Zend Core for IBM instala o DB2 e clientes IDS, um servidor HTTP Apache opcional, PHP 5 e extensões PHP populares, incluindo ibm\_db2 e PDO\_INFORMIX. Zend Core for IBM pode instalar opcionalmente o servidor DB2 Express-C, o servidor IBM Cloudscape™, o manual completo de PHP, e aplicações modelo para o DB2. Possui um ambiente PHP fácil de usar e configurar, como mostrado nas Figuras 17.5, 17.6 e 17.7.

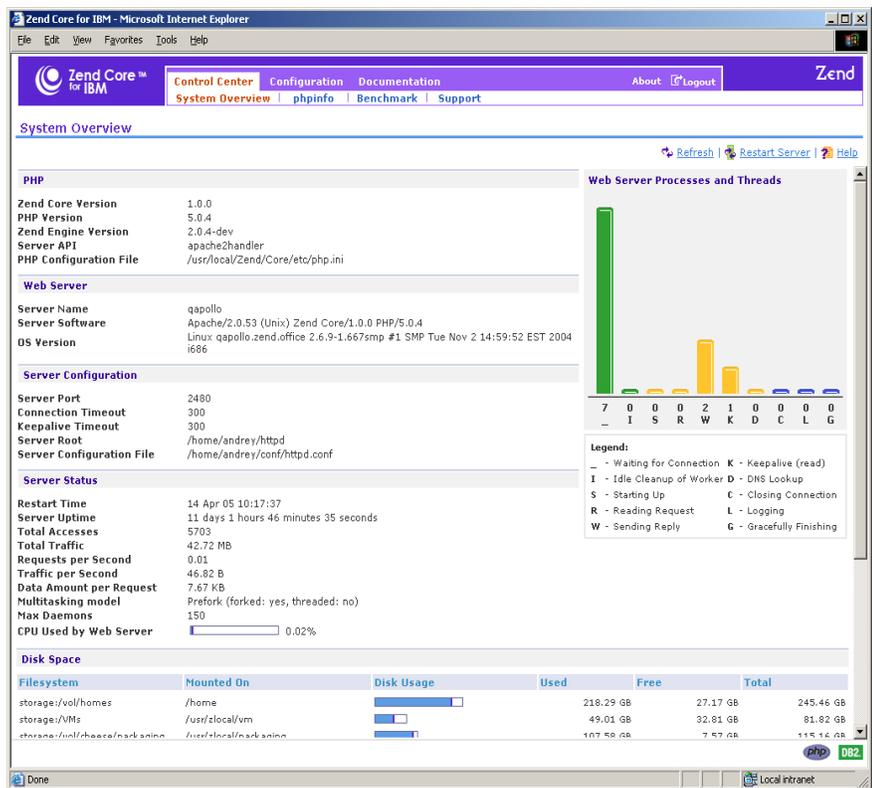


Figura 17.5 – Gerenciamento e interface de controle do Zend Core

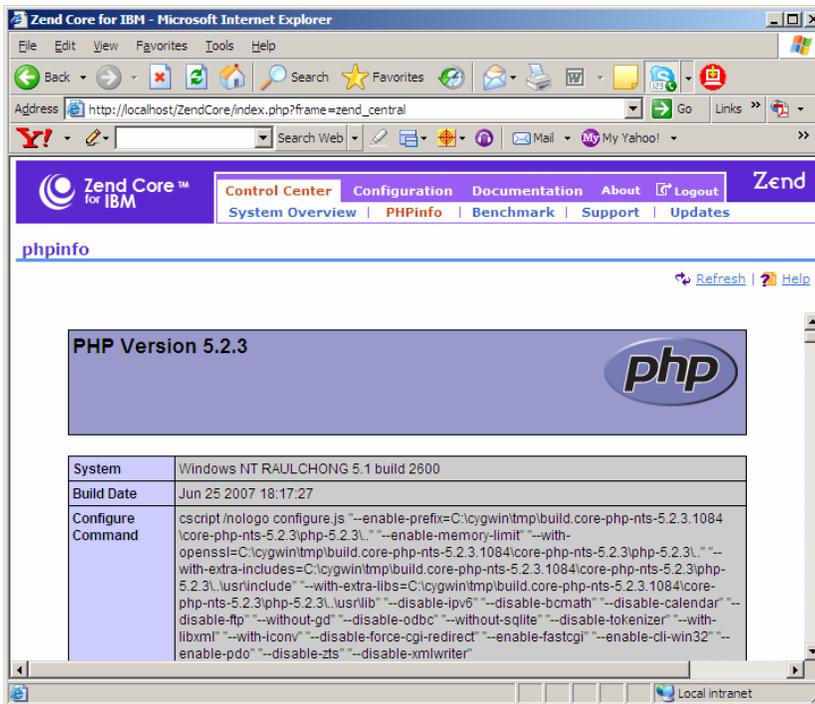


Figura 17.6 – Interface de configuração de PHP do Zend Core

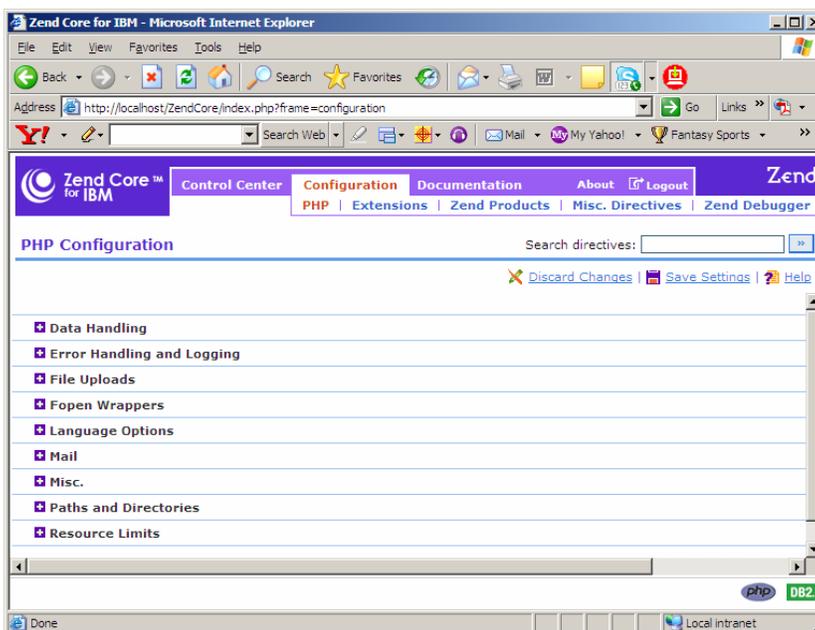


Figura 17.7 – Interface de configuração do PHP do Zend Core

### 17.3 Desenvolvimento de aplicações em Ruby on Rails

Ruby é uma linguagem de scripts orientada a objetos, dinâmica e multi-plataforma. Ela facilita o rápido desenvolvimento e inclui uma biblioteca rica. Ruby é uma linguagem simples e divertida inventada por Yukihiro Matsumoto ("Matz") em 1995.

Rails é um framework completo para aplicações web escritas em Ruby que utilizam bases de dados. Ele implementa a arquitetura model-view-control (MVC). É incrivelmente produtivo e fácil de usar. Rails é um dos frameworks mais rápidos emergentes desde 2004 e foi inventado por David Heinemeier Hansson.

#### 17.3.1 Startup Toolkit for DB2 on Rails

A IBM reconhece a importância de Ruby on Rails na comunidade de desenvolvimento; portanto, criou o pacote chamado **Startup Toolkit for DB2 on Rails**. É um instalador integrado que cria um ambiente de desenvolvimento completo de DB2 Ruby on Rails. Pode ser baixado e utilizado sem custos do site IBM alphaWorks: <http://www.alphaworks.ibm.com/tech/db2onrails>.

O Startup Toolkit for DB2 on Rails:

- ▶ Inclui um instalador integrado.
- ▶ Ajuda você a instalar e configurar facilmente Ruby on Rails.
- ▶ Instala o DB2 Express-C 9 e suas ferramentas.
- ▶ Inclui um driver DB2 Ruby desenvolvido pela IBM e um adaptador DB2 Rails.
- ▶ Inclui vários demos e tutoriais.

# A

## Apêndice A – Resolvendo Problemas

Este apêndice mostra como resolver problemas que podem ocorrer quando você estiver trabalhando com o DB2. A Figura A.1 mostra um resumo das ações que podem ser realizadas em caso de problemas.

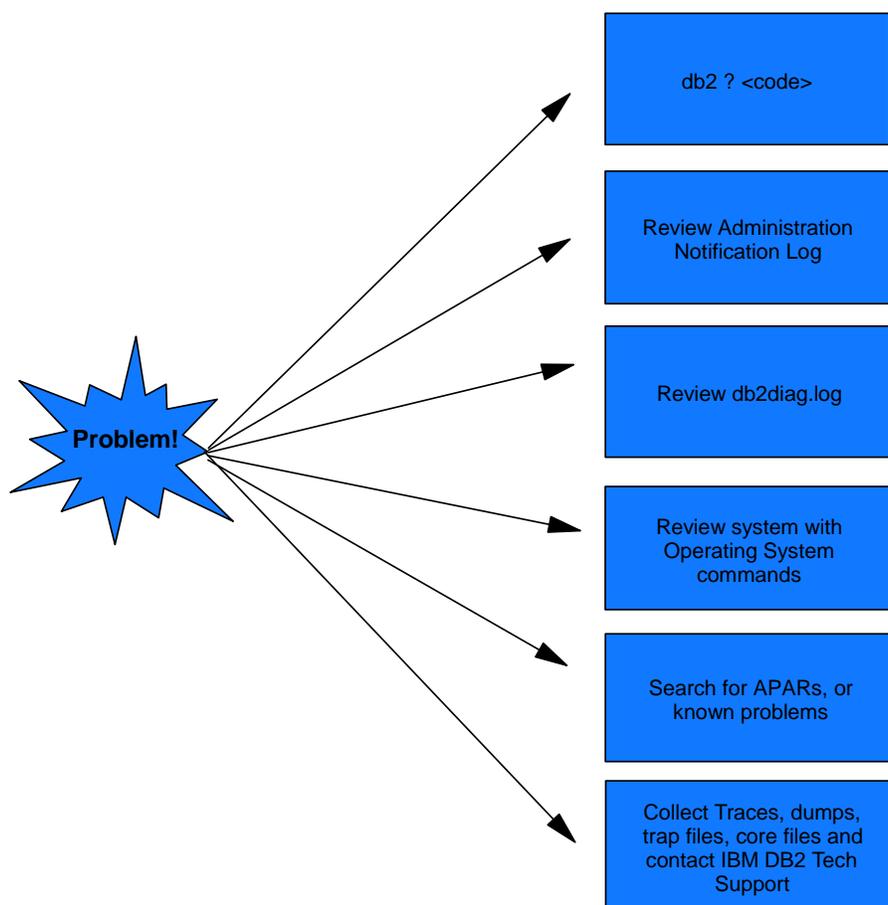


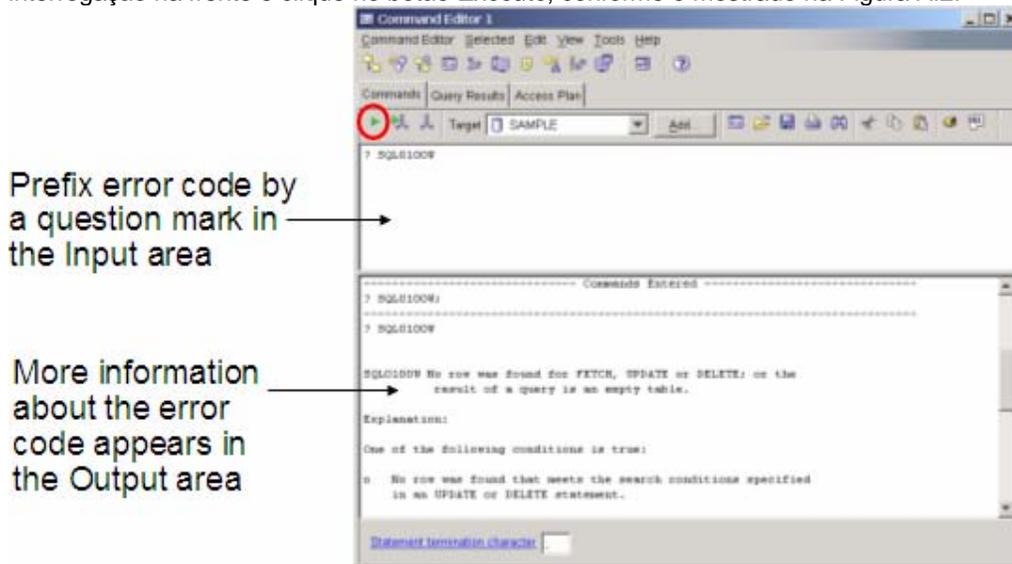
Figura A.1 – Como resolver um problema (Resumo)

**Nota:**

Para obter mais informações sobre como resolver problemas no DB2 assista ao vídeo: <http://www.channeldb2.com/video/video/show?id=807741:Video:4462>

**A.1 Encontrando mais informações sobre códigos de erro**

Para obter mais informações sobre os códigos de erro recebidos, digite, na área de entrada de dado do Command Editor, o prefixo do código de erro com um ponto de interrogação na frente e clique no botão *Execute*, conforme é mostrado na Figura A.2.



**Figura A.2 – Encontrando mais informações sobre códigos de erro**

O caractere ponto de interrogação (?) chama a ajuda do DB2. Abaixo seguem vários exemplos de como chamar a ajuda se você receber, por exemplo, o código de erro 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**

Um SQLCODE é um código recebido após a execução de uma instrução SQL. O significado dos valores está resumido abaixo:

SQLCODE = 0; o comando foi executado com sucesso  
SQLCODE > 0; o comando foi executado com sucesso com uma mensagem de aviso  
SQLCODE < 0; o comando não foi executado com sucesso e retornou uma mensagem de erro

O SQLSTATE é um conjunto de cinco caracteres de acordo com os padrões ISO/ANSI SQL92. Os primeiros dois caracteres são chamados de códigos de classe:

O código de classe 00 significa que o comando foi executado com sucesso.  
O código de classe 01 indica um aviso.  
O código de classe 02 significa que foi executada uma condição que não existe.  
Todas as outras classes são consideradas erros.

### A.3 DB2 – Log de Administração e Notificação

A log de administração e notificação do DB2 fornece informações sobre erros no momento que os mesmos acontecem. Nas plataformas Linux/UNIX, A log de administração e notificação do DB2 é um arquivo de texto nomeado como <nome da instancia>.nfy (por exemplo “db2inst.nfy”). No Windows, todas as logs de administração e notificação do DB2 são escritas no Evento de Logs do Windows.

O parâmetro de configuração do DB2 `notifylevel` permite que os administradores especifiquem o nível de informação a ser gravado:

- 0 -- Nenhuma mensagem de notificação é capturada (não recomendado)
- 1 -- Armazena erros fatais ou irrecuperáveis
- 2 -- Grava informações que requerem ações imediatas
- 3 -- Armazena informações importantes, que não requerem ações imediatas (padrão)
- 4 -- Salva mensagens informativas

### A.4 db2diag.log

O db2diag.log provê informações mais detalhadas do que a log de administração e notificação do DB2. É mais utilizado por técnicos de suporte do DB2 e DBA's experientes. Informações contidas no db2diag.log incluem:

- O local do código DB2 que esta reportando o erro
- Identificadores da aplicação que permitem que você compare as mensagens da aplicação com o db2diag.logs de servidores e clientes.
- Mensagens de diagnostico (que começam com “DIA”) que explicam a razão do erro.
- Dados de suporte, como a estrutura de dados SQLCA e ponteiros para a localização de arquivos de dump.

No Windows, o db2diag.log está localizado no seguinte diretório (padrão):

```
C:\Program Files\IBM\sqlllib\\db2diag.log
```

No Linux/UNIX, o db2diag.log está localizado no seguinte diretório (padrão):

```
/home/<instance_owner>/sqlllib/db2dump/db2diag.log
```

A quantidade de texto diagnóstico é determinada pelo parâmetro de configuração dbm cfg chamado DIAGLEVEL, onde o intervalo de 0 a 4 indica que 0 é a menor quantidade e 4 a maior. O valor padrão é 3.

## A.5 CLI traces

Em aplicações CLI e Java, você pode habilitar a facilidade de trace CLI para facilitar a correção dos erros. Isto pode ser configurado através de alterações no arquivo db2cli.ini do servidor que roda a aplicação. Entradas típicas do db2cli.ini são mostradas abaixo:

```
[common]
trace=0
tracerefreshinterval=300
tracepathname=/path/to/writeable/directory
traceflush=1
```

O nível de trace (db2trc) também está disponível porem é mais utilizados pelo suporte técnico do Db2.

## A.6 DB2 – Defeitos e Correções

Você pode encontrar um problema que causa um defeito no DB2. A IBM lança regularmente pacotes de correções (fix packs) chamados (APARs). A documentação dos pacotes de correção contém a lista de defeitos corrigidos. Quando novos produtos são desenvolvidos a IBM sempre recomenda a utilização do ultimo pacote de correções. Para verificar a versão do DB2 e o pacote de correções que esta sendo utilizado: a partir do Control Center, selecione a opção **About** no menu **Help**; a partir da janela de comandos, digite "db2level". Lembre-se que o suporte oficial da IBM só é oferecido, no DB2 Express-C, quando você compra os 12 meses de suporte oficial.

---

## Recursos

### Web sites:

1. DB2 Express-C web site:  
[www.ibm.com/db2/express](http://www.ibm.com/db2/express)  
Use este site para fazer o download de imagens de servidores DB2 Express-C, de clientes DB2, manuais, para acessar o blog, para assinar a lista de distribuição, etc.
2. DB2 Express forum:  
[www.ibm.com/developerworks/forums/dw\\_forum.jsp?forum=805&cat=19](http://www.ibm.com/developerworks/forums/dw_forum.jsp?forum=805&cat=19)  
Use o fórum para postar perguntas técnicas.
3. DB2 Information Center  
<http://publib.boulder.ibm.com/infocenter/db2luw/v9/index.jsp>  
Através do centro de informações você tem acesso aos manuais on-line.
4. developerWorks  
<http://www-128.ibm.com/developerworks/db2>  
Este site é um excelente recurso para desenvolvedores e DBA's e contém muitos artigos, tutorias e outros recursos grátis.
5. alphaWorks  
<http://www.alphaworks.ibm.com/>  
Este site prove um acesso direto às novas tecnologias IBM.
6. planetDB2  
[www.planetDB2.com](http://www.planetDB2.com)  
Este blog contém contribuições de usuários, desenvolvedores e DBA's DB2.
7. DB2 Technical Support  
Se você comprar a licença de suporte de 12 meses do DB2 Express-C, você pode fazer o download dos pacotes de correção através deste site:  
[http://www.ibm.com/software/data/db2/support/db2\\_9/](http://www.ibm.com/software/data/db2/support/db2_9/)
8. ChannelDB2  
O ChannelDB2 é uma rede social da comunidade DB2. Este canal contém vídeos DB2, demonstrações, apresentações, blogs, discussões e recursos para o DB2 em diversas plataformas (Linux, UNIX, Windows, System z to z/OS and System i to i5/OS).  
<http://www.ChannelDB2.com/>

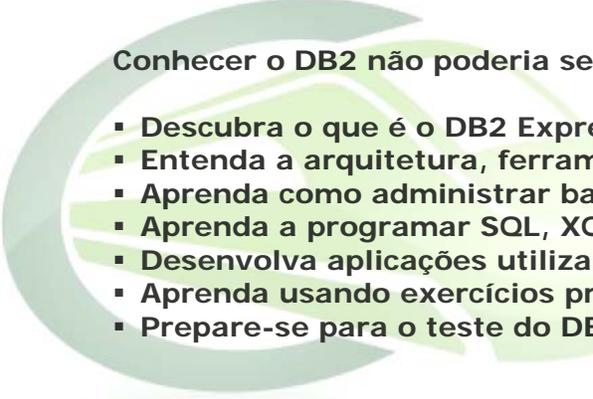
## Livros

1. Free Redbook: DB2 Express-C: The Developer Handbook for XML, PHP, C/C++, Java, and .NET  
Whei-Jen Chen, John Chun, Naomi Ngan, Rakesh Ranjan, Manoj K. Sardana,  
August 2006 - SG24-7301-00  
<http://www.redbooks.ibm.com/abstracts/sq247301.html?Open>
2. Understanding DB2 – Learning Visually with Examples V9.5  
Raul F. Chong, et all. January 2008  
ISBN-10: 0131580183
3. DB2 9: pureXML overview and fast start by Cynthia M. Saracco, Don Chamberlin,  
Rav Ahuja June 2006 SG24-7298  
<http://www.redbooks.ibm.com/abstracts/sq247298.html?Open>
4. DB2® SQL PL: Essential Guide for DB2® UDB on Linux™, UNIX®, Windows™,  
i5/OS™, and z/OS®, 2nd Edition  
Zamil Janmohamed, Clara Liu, Drew Bradstock, Raul Chong, Michael Gao, Fraser  
McArthur, Paul Yip  
ISBN: 0-13-100772-6
5. Free Redbook: DB2 pureXML Guide  
Whei-Jen Chen, Art Sammartino, Dobromir Goutev, Felicity Hendricks, Ippei Komi,  
Ming-Pang Wei, Rav Ahuja, Matthias Nicola. August 2007  
<http://www.redbooks.ibm.com/abstracts/sq247315.html?Open>
6. Information on Demand - Introduction to DB2 9 New Features  
Paul Zikopoulos, George Baklarz, Chris Eaton, Leon Katsnelson  
ISBN-10: 0071487832  
ISBN-13: 978-0071487832
7. Redbook: Developing PHP Applications for IBM Data Servers.  
Whei-Jen Chen, Holger Kirstein, Daniel Krook, Kiran H Nair, Piotr Pietrzak  
May 2006 - SG24-7218-00  
<http://www.redbooks.ibm.com/abstracts/sq247218.html?Open>

## E-mails de Contato

General DB2 Express-C mailbox: [db2x@ca.ibm.com](mailto:db2x@ca.ibm.com)

General DB2 on Campus program mailbox: [db2univ@ca.ibm.com](mailto:db2univ@ca.ibm.com)



**Conhecer o DB2 não poderia ser mais fácil. Neste livro:**

- Descubra o que é o DB2 Express-C
- Entenda a arquitetura, ferramentas e segurança do DB2
- Aprenda como administrar bases de dados DB2
- Aprenda a programar SQL, XQuery e stored procedures
- Desenvolva aplicações utilizando o banco de dados o DB2
- Aprenda usando exercícios práticos
- Prepare-se para o teste do DB2 on Campus

A rápida adoção de XML para integração de aplicações, Web 2.0 e SOA estão conduzindo à necessidade de ter servidores de dados híbridos. O DB2 Express-C da IBM é um servidor de banco de dados híbrido sem custo e sem limites, capaz de gerenciar dados XML e dados relacionais tradicionais com facilidade. Sem custo quer dizer que o DB2 Express-C é livre para ser baixado, livre para construir suas aplicações, livre para fazer a entrega em produção e livre para redistribuir com sua solução. E o DB2 não impõe qualquer limite ao tamanho da base de dados, número de bases de dados ou quantidade de usuários.

O DB2 Express-C roda em sistemas Windows e Linux e provê drivers de aplicação para uma variedade de linguagens de programação, incluindo C/C++, Java, .NET, PHP, Perl, e Ruby. Está disponível também uma inscrição opcional a baixo custo e suporte com capacidades adicionais. Se você precisar de mais escalabilidade ou funcionalidades mais avançadas, você pode transportar suas aplicações construídas em DB2 Express-C para outras edições, como DB2 Enterprise.

Esta edição free do DB2 é ideal para desenvolvedores, consultores, ISVs (*Independent Software Vendors*), DBAs, estudantes, ou qualquer pessoa que pretende desenvolver, testar, fazer a entrega, ou distribuir suas aplicações que utilizam bancos de dados. Junte-se à crescente comunidade de usuários do DB2 Express-C hoje e faça um test-drive com o DB2 Express-C. Comece descobrindo como você pode criar aplicações da próxima geração e entregar soluções inovadoras.