

Sistemas Gereciadores de Banco de Dados

Prof. Marcos Alexandruk
alexandruk@uninove.br
www.unilivros.com.br



Aula 1

Informações Importantes



- EMENTA:
- Sistemas gerenciadores de banco de dados relacionais;
- Configurações do ambiente de trabalho;
- Diferenças entre as diversas plataformas;
- Mecanismos de back-up;
- Importação e exportação;
- Criação de tabelas e consultas

Informações Importantes



- OBJETIVO:
- Implementar um banco de dados em um SGBD, elaborando, montando, executando e avaliando instruções de definição, consulta e manipulação de dados.

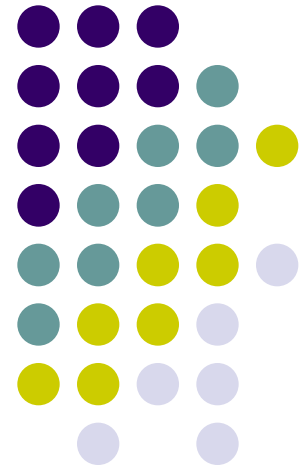
Informações Importantes



- BIBLIOGRAFIA:
- COSTA, Rogério L. de C. **SQL Guia Prático**. 2ª ed. Rio de Janeiro: Brasport, 2006.
- SILBERSCHATZ, A.; KORTH, H.; SUDARSHAN, S. **Sistema de Banco de Dados**. 5ª ed. Rio de Janeiro: Campus, 2006.
- SOARES, Walace. **MySQL conceitos e aplicações**. São Paulo: Érica, 2004.

Sistemas Gereciadores de Banco de Dados

Conceitos e Características Gerais



Conceitos Gerais e Características



SGBD (Sistema de Gerenciamento de Banco de Dados):

coleção de dados inter-relacionados

+

conjunto de programas para acessar e manipular esses dados

Silberschatz p.4

Conceitos Gerais e Características



O principal objetivo de um SGBD é fornecer um ambiente que seja tanto **conveniente** como **eficiente** para recuperação e armazenamento de informações.

Conceitos Gerais e Características



Gerenciamento de dados envolve:

- Definir estruturas de armazenamento
- Fornecer mecanismos para a manipulação de informações

Conceitos Gerais e Características



- O SGBD precisa garantir a **segurança** apesar de falhas de sistema ou tentativas de acesso não autorizado.

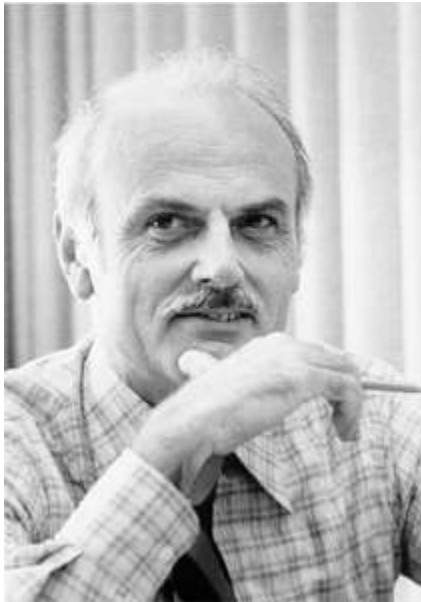
Conceitos Gerais e Características



“Embora as interfaces de usuário ocultem os detalhes de acesso a um banco de dados, e a maioria das pessoas nem mesmo tenha consciência de estar lidando com um banco de dados, acessar banco de dados é uma parte essencial da vida de quase todo mundo hoje.”

Silberschatz p. 2

As doze regras de Codd



- Doze regras estabelecidas por **Edgard F. Codd**, em 1985, por meio das quais podemos determinar o quanto um banco de dados é relacional ou não.



As doze regras de Codd

- 1. Regra das informações em tabelas:
 - As informações a serem apresentadas no banco de dados devem ser apresentadas como relações (tabelas formadas por linhas e colunas) e o vínculo de dados entre as tabelas deve ser estabelecido por meio de valores de campos comuns.

Isto se aplica tanto aos dados quanto aos metadados (descrições dos objetos do banco de dados).

As doze regras de Codd



- 2. Regra de acesso garantido:
 - Para que o usuário possa acessar as informações contidas no banco de dados, o método de referência deve ser o nome da tabela, o valor da chave primária e o nome do campo.

A ordem de apresentação dos dados não tem importância no contexto.

As doze regras de Codd



- 3. Regra de tratamento sistemático de valores nulos:
 - O SGBD deve ter capacidade de tratar valores que não são fornecidos pelos usuários de maneira que permita a distinção dos valores reais.

Por exemplo, em um campo de armazenamento de dados numéricos, podemos ter valores válidos, o valor zero e valores nulos.



As doze regras de Codd

- 4. Regra do catálogo relacional ativo:
 - Toda a estrutura do banco de dados (tabelas, campos, índices, etc.) deve estar disponível em tabelas (catálogo).

Essas tabelas são manipuladas pelo próprio sistema, quando o usuário efetua alterações na estrutura do banco de dados.



As doze regras de Codd

- 5. Regra da atualização de alto nível:
 - O usuário deve ter capacidade de manipular as informações do banco de dados em grupos de registros, ou seja, ser capaz de inserir, alterar e excluir vários registros ao mesmo tempo.

As doze regras de Codd



- 6. Regra da sublinguagem de dados abrangente:
 - Pelo menos uma linguagem deve ser suportada para que o usuário possa manipular a estrutura do banco de dados (exemplo: criação e alteração de tabelas), assim como extrair, inserir, atualizar ou excluir dados, definir restrições de acesso e controle de transações (COMMIT/ROLLBACK).

Deve ser possível também a manipulação de dados por meio de programas aplicativos.

As doze regras de Codd



- 7. Regra da independência física:
 - Quando for necessária alguma modificação na forma como os dados são armazenados fisicamente, nenhuma alteração deve ser necessária nas aplicações que fazem uso do banco de dados.

Devem também permanecer inalterados os mecanismos de consulta e manipulação de dados utilizados pelos usuários finais.

As doze regras de Codd



- 8. Regra da independência lógica:
 - Qualquer alteração efetuada na estrutura do banco de dados, como inclusão e exclusão de campos de uma tabela ou alteração no relacionamento entre tabelas não deve afetar o aplicativo que o usa. O aplicativo deve manipular visões das tabelas.

Visões são uma espécie de tabela virtual, que agrupa dados de uma ou mais tabelas físicas e apresenta ao usuário os dados.

As doze regras de Codd



- 9. Regra da atualização de visões:
 - Visto que as visões dos dados são teoricamente suscetíveis a atualizações, então um aplicativo que faz uso desses dados deve ser capaz de efetuar alterações, exclusões e inclusões neles

As atualizações devem ser repassadas automaticamente às tabelas originais.

As doze regras de Codd



- 10. Regra da independência de integridade:
 - As várias formas de integridade do banco de dados (integridade de entidade, referencial, restrição e obrigatoriedade de valores, etc.) precisam ser estabelecidas dentro do catálogo do sistema ou dicionário de dados, e ser totalmente independente da lógica dos aplicativos.

As doze regras de Codd



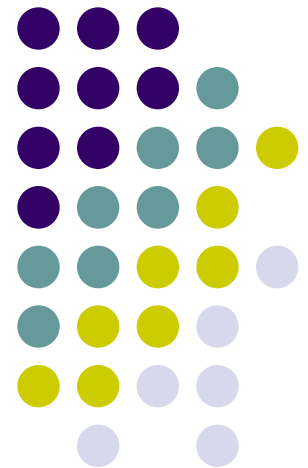
- 11. Regra da independência de distribuição:
 - Sistemas de banco de dados podem estar distribuídos em diversas plataformas, interligados em rede e podem inclusive estar fisicamente distantes entre si. Essa capacidade de distribuição não pode afetar a funcionalidade do sistema e dos aplicativos que fazem uso do banco de dados.

As doze regras de Codd



- **12. Regra não subversiva:**
 - O sistema deve ser capaz de impedir que qualquer usuário ou programador de passar por cima de todos os mecanismos de segurança, regras de integridade do banco de dados e restrições, utilizando algum recurso ou linguagem de baixo nível que eventualmente possam ser oferecidas pelo próprio sistema.

Teste





As doze regras de Codd

- 1. Regra das informações em tabelas
- 2. Regra de acesso garantido
- 3. Regra de tratamento sistemático de valores nulos
- 4. Regra do catálogo relacional ativo
- 5. Regra da atualização de alto nível
- 6. Regra da sub linguagem de dados abrangente
- 7. Regra da independência física
- 8. Regra da independência lógica
- 9. Regra da atualização de visões
- 10. Regra da independência de integridade
- 11. Regra da independência de distribuição
- 12. Regra não subversiva

As doze regras de Codd



- Associe cada uma das regras a seguir conforme o seu número segundo a relação do slide anterior



As doze regras de Codd

- Toda a estrutura do banco de dados (tabelas, campos, índices, etc.) deve estar disponível em tabelas (catálogo).

Essas tabelas são manipuladas pelo próprio sistema, quando o usuário efetua alterações na estrutura do banco de dados.



As doze regras de Codd

- Visto que as visões dos dados são teoricamente suscetíveis a atualizações, então um aplicativo que faz uso desses dados deve ser capaz de efetuar alterações, exclusões e inclusões neles

As atualizações devem ser repassadas automaticamente às tabelas originais.

As doze regras de Codd



- Sistemas de banco de dados podem estar distribuídos em diversas plataformas, interligados em rede e podem inclusive estar fisicamente distantes entre si. Essa capacidade de distribuição não pode afetar a funcionalidade do sistema e dos aplicativos que fazem uso do banco de dados.

As doze regras de Codd



- Quando for necessária alguma modificação na forma como os dados são armazenados fisicamente, nenhuma alteração deve ser necessária nas aplicações que fazem uso do banco de dados.

Devem também permanecer inalterados os mecanismos de consulta e manipulação de dados utilizados pelos usuários finais.



As doze regras de Codd

- As informações a serem apresentadas no banco de dados devem ser apresentadas como relações (tabelas formadas por linhas e colunas) e o vínculo de dados entre as tabelas deve ser estabelecido por meio de valores de campos comuns.

Isto se aplica tanto aos dados quanto aos metadados (descrições dos objetos do banco de dados).

As doze regras de Codd



- O usuário deve ter capacidade de manipular as informações do banco de dados em grupos de registros, ou seja, ser capaz de inserir, alterar e excluir vários registros ao mesmo tempo.



As doze regras de Codd

- Para que o usuário possa acessar as informações contidas no banco de dados, o método de referência deve ser o nome da tabela, o valor da chave primária e o nome do campo.

A ordem de apresentação dos dados não tem importância no contexto.

As doze regras de Codd



- Qualquer alteração efetuada na estrutura do banco de dados, como inclusão e exclusão de campos de uma tabela ou alteração no relacionamento entre tabelas não deve afetar o aplicativo que o usa. O aplicativo deve manipular visões das tabelas.

Visões são uma espécie de tabela virtual, que agrupa dados de uma ou mais tabelas físicas e apresenta ao usuário os dados.

As doze regras de Codd



- O sistema deve ser capaz de impedir que qualquer usuário ou programador de passar por cima de todos os mecanismos de segurança, regras de integridade do banco de dados e restrições, utilizando algum recurso ou linguagem de baixo nível que eventualmente possam ser oferecidas pelo próprio sistema.

As doze regras de Codd



- As várias formas de integridade do banco de dados (integridade de entidade, referencial, restrição e obrigatoriedade de valores, etc.) precisam ser estabelecidas dentro do catálogo do sistema ou dicionário de dados, e ser totalmente independente da lógica dos aplicativos.

As doze regras de Codd



- O SGBD deve ter capacidade de tratar valores que não são fornecidos pelos usuários de maneira que permita a distinção dos valores reais.

Por exemplo, em um campo de armazenamento de dados numéricos, podemos ter valores válidos, o valor zero e valores nulos.

As doze regras de Codd



- Pelo menos uma linguagem deve ser suportada para que o usuário possa manipular a estrutura do banco de dados (exemplo: criação e alteração de tabelas), assim como extrair, inserir, atualizar ou excluir dados, definir restrições de acesso e controle de transações (COMMIT/ROLLBACK).

Deve ser possível também a manipulação de dados por meio de programas aplicativos.



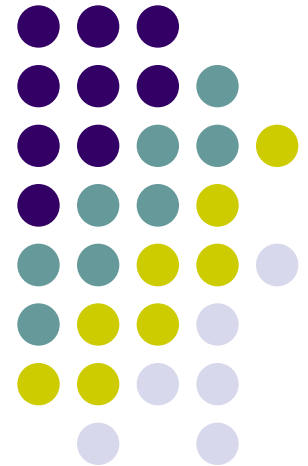
Resposta

- A sequência correta é:

4, 9, 11, 7, 1, 5, 2, 8, 12, 10, 3, 6

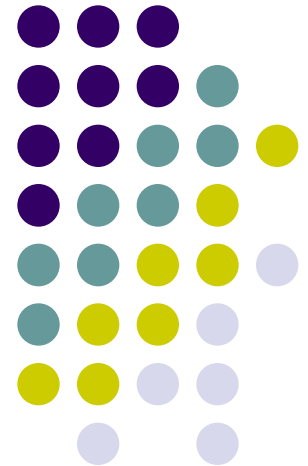
Sistemas Gereciadores de Banco de Dados

Prof. Marcos Alexandruk
alexandruk@uninove.br
www.unilivros.com.br



Aula 2

Classificação de Banco de Dados



Cenário atual



- Hoje encontramos uma grande variedade de SGDBs (Sistemas Gerenciadores de Banco de Dados) para as mais diversas plataformas de hardware e sistemas operacionais.

Classificação quanto ao modelo de dados



- **Bancos de Dados Hierárquicos:**

- IMS - Information Management System: desenvolvido pela IBM e pela Rockwell no fim da década de 1960 para ambientes de grande porte (OS/VS1, OS/VS2, MVS, MVS/XA e ESA)
- Utiliza a organização de endereços físicos do disco na sua estrutura
- Baseado em dois conceitos fundamentais: registros e relacionamentos pai-filho
- Um registro "pai" pode se corresponder com vários registros do lado "filho"



Classificação quanto ao modelo de dados

● Bancos de Dados em Rede:

- Definidos pelo DBTG (Data Base Task Group) do comitê do CODASYL (Conference on Data Systems Language) a partir de 1971
- Permitem que um mesmo registro participe de vários relacionamentos devido à eliminação da hierarquia
- Os comandos de manipulação de registros devem ser incorporados a uma linguagem hospedeira (COBOL, a mais comum, Pascal e FORTRAN)
- Estruturas fundamentais: registros (records) e conjuntos (sets)
- Registros contêm dados relacionados e são agrupados em tipos de registros que armazenam os mesmos tipos de informações



Classificação quanto ao modelo de dados

- **Bancos de Dados Relacionais:**

- A maioria dos SGBDs atualmente em uso se enquadra no tipo relacional
- Princípios básicos formulados por Edgard F. Codd em 1968 baseados na teoria dos conjuntos e da álgebra relacional
- Em 1985, Codd propôs um conjunto de doze regras para que um banco de dados fosse considerado como relacional
- Organiza os dados em tabelas (relações) formadas por linhas e colunas
- Tabelas são similares a conjuntos de elementos: relacionam as informações de um mesmo assunto de um modo organizado



Classificação quanto ao modelo de dados

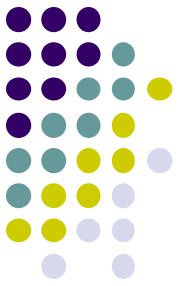
● Bancos de Dados Orientados a Objetos:

- Surgiram em meados de 1980 para armazenamento de dados complexos, não adequados aos sistemas relacionais: GIS (Geographical Information System) e CAD/CAM/CAE
- O modelo é caracterizado pela definição de objetos com suas propriedades e operações
- O OMDG (Object Database Management Group) definiu um padrão de estrutura para bancos de dados orientados a objetos
- O grupo propôs um padrão conhecido como ODMG-93, atualmente revisado e denominado ODMG 2.0
- Linguagens: ODL (Object Definition Language) e OQL (Object Query Language)



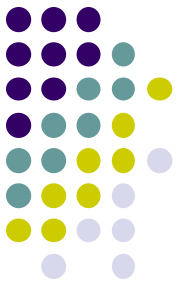
Classificação quanto ao número de usuários

- **Bancos de Dados Monousuários:**
 - Permitem que apenas um usuário por vez acesse o banco de dados
 - Antigos (1980-1990) e direcionados a uso pessoal: dBASE III, dBASE IV, FoxBase, FoxPro



Classificação quanto ao número de usuários

- **Bancos de Dados Multiusuários:**
 - Suporta o acesso de vários usuários ao mesmo tempo
 - A maioria dos bancos de dados atuais oferece suporte a multiusuários



Classificação quanto a localização

- **Bancos de Dados Centralizados::**
 - Localizados em uma única máquina denominada Servidor de Banco de Dados
 - Embora centralizados, podem oferecer suporte a acesso concorrente de vários usuários



Classificação quanto a localização

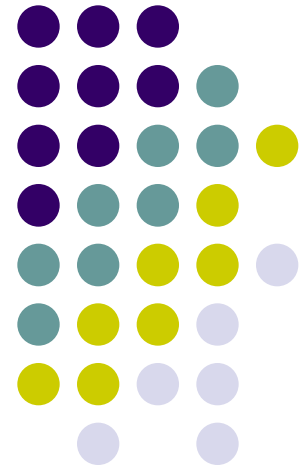
- **Bancos de Dados Distribuídos:**

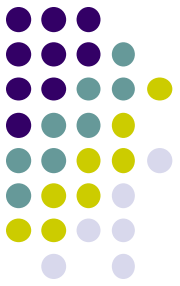
- O sistema gerenciador e o banco de dados estão localizados em diferentes máquinas interligadas em redes (LANS ou WANS)
- Independentemente de serem centralizados ou distribuídos os SGBDs atualmente trabalham dentro da arquitetura cliente-servidor



- **Bancos de Dados Heterogêneos:**
 - Tendência que vem crescendo muito atualmente: distribuir na arquitetura de SGBDs vários bancos de dados de fornecedores diferentes

Resumo

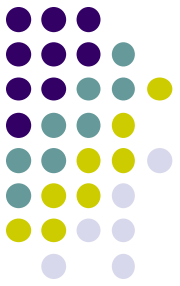




- Quais os tipos de classificações dos bancos de dados?

Podem ser classificados quanto a(o):

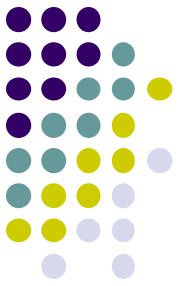
- modelo de dados
- número de usuários
- localização



- Como são classificados os bancos de dados quanto aos MODELOS DE DADOS?
 - hierárquicos
 - em rede
 - relacionais
 - orientados a objetos



- Como são classificados os bancos de dados quanto ao NÚMERO DE USUÁRIOS?
 - monousuários
 - multiusuários



- Como são classificados os bancos de dados quanto a sua LOCALIZAÇÃO?
 - centralizados
 - distribuídos

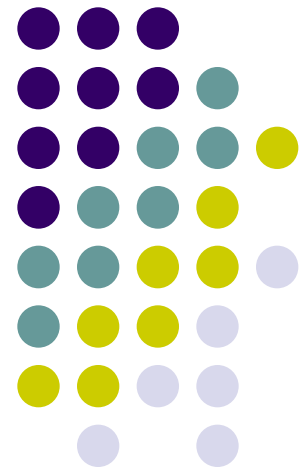
Sistemas Gereciadores de Banco de Dados

Prof. Marcos Alexandruk
alexandruk@uninove.br
www.unilivros.com.br



Aula 3

Principais SGBDs





dBase

- Lançado em 1984 pela Ashton-Tate e adquirido posteriormente pela Borland
- Linguagem de programação relativamente fácil de aprender tornava possível criar aplicações inteiras.
- Versões para DOS e Windows
- Sistema de gerenciamento de arquivos planos (flat files)



Paradox

- Lançado em 1985 pela Ansa Software, adquirida em 1987 pela Borland
- Ambiente integrado de desenvolvimento para criação de aplicativos
- Versões para DOS e Windows
- Utiliza QBE (Query by Example) para consultas



DataFlex

- Produzido pela empresa Data Access Corporation, fundada em 1976
- Versões para UNIX, VAX/VMS, Netware, CP/M, DOS, OS/2 e LINUX
- Ambiente de desenvolvimento VDF Visual DataFlex disponível para Windows (acessa bases SQL Server, Oracle, DB2, etc.)
- Para conhecer mais:
<http://www.dataaccess.com.br/dataflex.asp>



FoxBase/FoxPro

- Lançado em dezembro de 1984
- Concorrente do dBase (arquivos-fonte compatíveis)
- Disponível originalmente para ambiente DOS foi portado para o Windows
- A versão 9 será a última, conforme informação da Microsoft



Access

- Lançado em 1992 pela Microsoft
- Ambiente integrado com interface intuitiva que permite a criação e gerenciamento do banco de dados e o desenvolvimento de aplicações e relatórios
- Utiliza a linguagem de programação VBA Visual Basic for Applications
- Incluído em algumas versões do MS Office



Oracle Database

- 1977: Larry Ellison, Bob Miner e Ed Oates fundam a SDL (Software Level Laboratories)
- 1978: O nome da empresa é mudado para RSI (Rational Software Inc.)
- 1979: A RSI lança o primeiro produto comercial de banco de dados relacional utilizando a linguagem SQL
- 1983: Lançado o Oracle 3, o primeiro SGBD a rodar em mainframes e em minicomputadores
- 1998: A Oracle oferece suporte ao Linux
- 1999: Lançado o Oracle 8i
- Versão atual: 11g



InterBase

- Lançado em 1984 pela Groton Database Systems (Interbase, a partir de 1986, a empresa passou a ser totalmente controlada, em 1991, pela Borland)
- A versão 6.0 deu origem ao FireBird (open source)



SQL Server

- Lançado pela Microsoft em 1988
- Inicialmente era uma versão especial do Sybase (parceria com a Microsoft, encerrada em 1994)
- SQL Server 2005: grande integração com a plataforma .NET
- Versão atual: SQL Server 2008



Sybase

- Fundada por Mark Hoffman e Bob Epstein em 1984, em Berkeley, na Califórnia
- 1995: PowerBuilder – ferramenta de desenvolvimento cliente/servidor
- 2002: PowerDesigner – solução de modelagem e projeto em uma única ferramenta
- Para saber mais: www.sybase.com.br

MySQL



- 1995: David Axmark, Allan Larsson e Michael Widenius iniciam o desenvolvimento do MySQL pela empresa MySQL AB
- Licença: GNU-GPL (General Public License)
- Versões para Windows, LINUX, UNIX, FreeBSD e Mac OS X
- Muito utilizado em soluções para Web (provedores de hospedagem)

PostgreSQL



- Origem: Projeto POSTGRE, Universidade Berkeley, Califórnia. Equipe orientada pelo Prof. Michael Stonebraker
- 1987: primeira versão de demonstração
- 1988: primeira versão estável
- 1991: Código adquirido pela Illustra, a qual se fundiu com a Informix (de Stonebraker), hoje pertencente à IBM
- Licença: BSD (Berkeley Software Distribution)
- Versões para LINUX, UNIX, Mac OS X e Windows

Informix



- Projetado por Roger Sippl no final dos anos 70
- A Informix foi fundada em 1980 e tornou-se pública em 1986
- Na década de 90 foi o segundo banco mais popular depois do Oracle
- Em 2001 a IBM, por sugestão do Wal-Mart (O maior usuário do Informix), adquiriu a Informix
- Em meados de 2005, a IBM lançou a versão 10 do Informix IDS

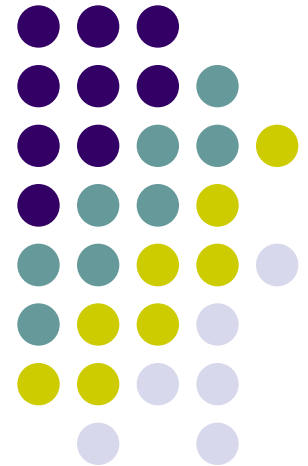


DB2

- Projeto começou no início dos anos 70 (Edgar Frank Codd – IBM)
- A princípio o produto foi chamado System R
- Lançado em 1983 com base no SQL/DS (para mainframe)
- A partir da década de 90 inclui versões para Windows, LINUX e PDAs
- 2006: Lançamento do DB2 9 Express
- DB2 9 é o primeiro SGBD que, segundo a IBM armazena XML nativo

TESTE

Informe o nome do SGBD
a partir das informações a seguir:





1

- Lançado em 1985 pela Ansa Software, adquirida em 1987 pela Borland
- Ambiente integrado de desenvolvimento para criação de aplicativos
- Versões para DOS e Windows
- Utiliza QBE (Query by Example) para consultas

2



- Lançado em dezembro de 1984
- Concorrente do dBase (arquivos-fonte compatíveis)
- Disponível originalmente para ambiente DOS foi portado para o Windows
- A versão 9 será a última, conforme informação da Microsoft

3



- Lançado em 1992 pela Microsoft
- Ambiente integrado com interface intuitiva que permite a criação e gerenciamento do banco de dados e o desenvolvimento de aplicações e relatórios
- Utiliza a linguagem de programação VBA Visual Basic for Applications
- Incluído em algumas versões do MS Office

4



- 1977: Larry Ellison, Bob Miner e Ed Oates fundam a SDL (Software Level Laboratories)
- 1978: O nome da empresa é mudado para RSI (Rational Software Inc.)
- 1979: A RSI lança o primeiro produto comercial de banco de dados relacional utilizando a linguagem SQL
- 1983: Primeiro SGBD a rodar em mainframes e em minicomputadores
- 1998: Suporte ao Linux
- 1999: Lançada a versão 8i
- Versão atual: 11g

5



- Produzido pela empresa Data Access Corporation, fundada em 1976
- Versões para UNIX, VAX/VMS, Netware, CP/M, DOS, OS/2 e LINUX
- Ambiente de desenvolvimento VDF Visual DataFlex disponível para Windows (acessa bases SQL Server, Oracle, DB2, etc.)

6



- Lançado em 1984 pela Ashton-Tate e adquirido posteriormente pela Borland
- Linguagem de programação relativamente fácil de aprender tornava possível criar aplicações inteiras.
- Versões para DOS e Windows
- Sistema de gerenciamento de arquivos planos (flat files)

7



- Lançado pela Microsoft em 1988
- Inicialmente era uma versão especial do Sybase (parceria com a Microsoft, encerrada em 1994)
- Grande integração com a plataforma .NET

8



- Fundada por Mark Hoffman e Bob Epstein em 1984, em Berkeley, na Califórnia
- 1995: PowerBuilder – ferramenta de desenvolvimento cliente/servidor
- 2002: PowerDesigner – solução de modelagem e projeto em uma única ferramenta

9



- 1995: David Axmark, Allan Larsson e Michael Widenius iniciam o desenvolvimento
- Licença: GNU-GPL (General Public License)
- Versões para Windows, LINUX, UNIX, FreeBSD e Mac OS X
- Muito utilizado em soluções para Web (provedores de hospedagem)

10



- Origem: Universidade Berkeley, Califórnia. Equipe orientada pelo Prof. Michael Stonebraker
- 1987: primeira versão de demonstração
- 1988: primeira versão estável
- 1991: Código adquirido pela Illustra, a qual se fundiu com a Informix (de Stonebraker), hoje pertencente à IBM
- Licença: BSD (Berkeley Software Distribution)
- Versões para LINUX, UNIX, Mac OS X e Windows

11



- Projetado por Roger Sippl no final dos anos 70
- Na década de 90 foi o segundo banco mais popular depois do Oracle
- Adquirido em 2001 pela IBM, por sugestão do Wal-Mart (O maior usuário deste SGBD)
- Em meados de 2005, a IBM lançou a versão 10

12



- Projeto começou no início dos anos 70 (Edgar Frank Codd – IBM)
- A princípio o produto foi chamado System R
- Lançado em 1983 com base no SQL/DS (para mainframe)
- A partir da década de 90 inclui versões para Windows, LINUX e PDAs
- 2006: Lançamento da versão 9 Express
- É o primeiro SGBD que, segundo a IBM armazena XML nativo

13



- Lançado em 1984 pela Groton Database Systems
- A empresa passou a ser totalmente controlada, em 1991, pela Borland
- A versão 6.0 deu origem ao FireBird (open source)

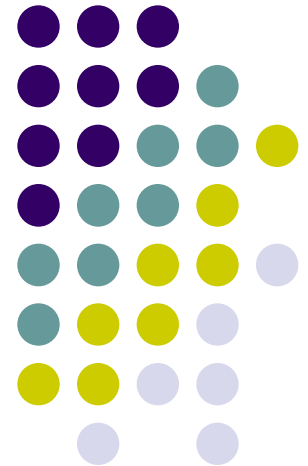
Respostas



1. Paradox
2. FoxBase/FoxPro
3. Access
4. Oracle Database
5. DataFlex
6. dBase
7. MS SQL Server
8. Sybase
9. MySQL
10. Postgre
11. Informix
12. DB2
13. Interbase

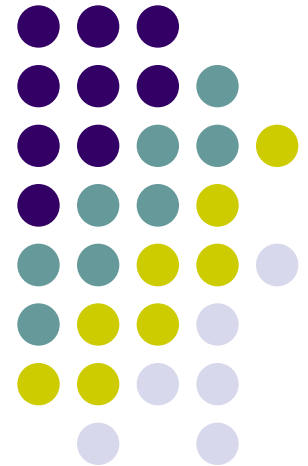
Sistemas Gereciadores de Banco de Dados

Prof. Marcos Alexandruk
alexandruk@uninove.br
www.unilivros.com.br



SGBDOOs

Sistemas Gerenciadores de Banco
de Dados Orientados a Objetos



SGBDOO



- Os SGBDs baseados nos modelos relacional, em rede ou hierárquico apresentam deficiências quando é preciso desenvolver aplicações para engenharia (CAE/CAD/CAM), para Sistemas de Informações Geográficas, simulações científicas ou médicas, etc.
- Essas aplicações fazem uso de estruturas de dados complexas (vídeos, imagens, áudio, textos formatados, etc.)
- Para atender a esta demanda surgiram os SGBDOOs (Sistemas Gerenciares de Banco de Dados Orientados a Objetos).

SGBDOO



- Outro fator que impulsionou o desenvolvimento de SGBDOOs foi a crescente popularidade de linguagens orientadas a objetos:
 - Smalltalk
 - C++
 - Java

SGBDOO



- Vários protótipos foram desenvolvidos, alguns inclusive foram disponibilizados comercialmente:
 - GemStone (www.gemstone.com)
 - Objectivity (www.objectivity.com)
 - ObjectStore (<http://web.progress.com/pt-br/objectstore/index.html>)
 - FastObjects (http://www.versant.com/en_US/products/fastobjects)
 - Versant (http://www.versant.com/en_US/products/objectdatabase)
 - OpenODB (Hewlett-Packard)

SGBDOO



- Para propor uma linguagem padrão para os bancos de dados orientados a objetos foi formado o ODMG (Object Database Management Group)
- O grupo propôs um padrão conhecido como ODMG-93, atualmente denominado ODMG 2.0
- O padrão foi adotado mundialmente pelos fornecedores e usuários de SGBDOOs
- O ODMG é responsável também pela definição de um padrão de linguagem para o modelo orientado a objetos.

SGBDOO



- O ODMG é responsável também pela definição de um padrão de linguagem para o modelo orientado a objeto:
 - ODL (Object Definion Language)
 - OQL (Object Query Language)
- Nesse padrão foi estabelecido que o banco de dados orientado a objeto deve possuir um vínculo com alguma linguagem hospedeira orientada a objeto (Smalltalk, C++, Java, etc.)
- O padrão define também os tipos de dados e métodos a serem suportados pelo sistema

SGBDOO



- Uma das principais características dos sistemas de banco de dados orientados a objetos é que o desenvolvedor pode especificar não apenas a estrutura de dados de objetos, mas também funções que desempenham operações nesses objetos, comumente chamados de **métodos**.

SGBDOO



- Cada objeto armazenado no banco de dados possui uma referência única, gerado pelo sistema quando ele é adicionado, denominada OID (Object Identifier). Normalmente utilizam-se números inteiros grandes como OIDs.
- Este identificador não é visível ao usuário e é responsável pela correspondência entre um objeto do "mundo real" e um objeto do banco de dados.
- O identificador não se repete entre objetos diferentes e quando um objeto é excluído o seu OID não é reutilizado em um novo objeto criado no banco de dados, nem pode ser alterado pelo sistema.

SGBDOO



- O **estado** de um objeto (seu valor corrente) é determinado a partir de outros objetos ou valores utilizando-se **construtores de tipos**.
- Há seis construtores básicos:
 - **atom** (valores atômicos)
 - **tuple** (tupla/registro de tabela)
 - **set** (conjunto de valores)
 - **list** (lista ordenada)
 - **bag** e **array** (matriz de dados)



SGBDOO

- Os construtores de tipos são utilizados na definição das estruturas de bancos de dados.
- Exemplo:

```
define type Cliente:  
    tuple (CodigoCliente integer;  
           NomeCliente string;  
           Telefone string;  
           Email string);
```

SGBDOO



ENCAPSULAMENTO/OCULTAÇÃO:

- Uma aplicação nunca acessa ou modifica diretamente os valores de um objeto.
- Essas operações somente são efetuadas por meio da chamada dos métodos desse objeto.
- Os métodos são invocados por meio do envio de mensagens ao objeto.
- Exemplo:

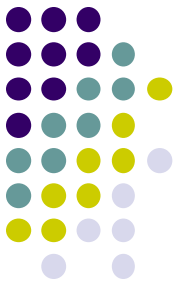
```
objNovoCliente = Cliente.novo_cliente("Fulano");
```

SGBDOO



- Apesar das suas qualidades, um banco de dados orientado a objetos normalmente apresenta problemas relacionados ao desempenho e à escalabilidade.
- Não são também adequados na manipulação de dados convencionais, como os existentes em bases relacionais.

SGBDOO – DB4OBJECTS



DOWNLOAD NOW

ABOUT

DEVELOPERS

CUSTOMERS

Product Test Drive

Forums

Downloads


Projects

Documentation

Download db4o

Download db4o's current production version, full-featured and unrestricted!

Production Release (registration optional)

 db4o 8.0 for Java
(zip, 40.9 mb)

 db4o 8.0 for .NET 3.5
(msi, 23.7 mb)


 db4o 8.0 for .NET 4.0
(msi, 23 mb)

☐ Do NOT keep me updated with a
monthly db4o product newsletter.

Email Address

 **Download**

Development Release (registration required)

 You need to register to download the
development release of db4o. Note
that this version is always a Beta
release, ideal for testing new features
but not yet suitable for production
environments.

New to db4o?

 **Join Now**

Already registered?

 **Download**

**Free General
Public License**
db4o is
licensed under
the [GPL](#) by default.



The GPL license is ideal if
you plan to use db4o in
house or you plan to
develop and distribute your
own derivative work as free
software under the [GPL](#), as
well.

**Commercial
License**
A [commercial
license](#) is




required if you want to
embed db4o in a
commercial non-GPL
product. Commercial
licensees gain access to
premium services and
support.

For more guidance on the
best licensing option,
please download this [free
whitepaper](#) about
db4objects' dual licensing
model.

SGBDOO – CACHÉ





Visite o Website Corporativo
**Advanced software technologies
for breakthrough applications**

[Produtos](#) | [Seu Segmento](#) | [Estudos de Caso](#) | [Parceiros](#) | [Suporte e Educação](#) | [Sobre a InterSystems](#) | [Vídeos](#)

- 
- ▶ Guia de Tecnologia Caché
 - ▶ Vantagens e Benefícios
 - ▶ Demonstrações On Line
 - ▶ Estudos de Caso
 - ▶ Brochuras
 - ▶ White Papers & Relatórios de Analistas
 - ▶ Espaço dos Desenvolvedores Caché
 - ▶ Fórum de Discussões
 - ▶ Caché e-Learning
 - ▶ Download Caché

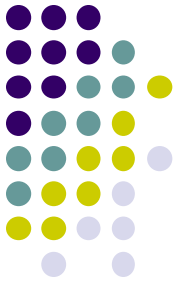


Banco de dados mais rápido do mundo

Caché é um banco de dados pós-relacional que oferece exclusivamente três opções integradas de acessos: um banco de dados robusto orientado a objetos, alta performance SQL, e um rico acesso multidimensional. O Caché permite desenvolvimento Web rápido, extraordinária velocidade no processamento de transações, escalabilidade máxima, e consultas em tempo-real contra dados transacionais - com o mínimo de requerimento de manutenção.

O Caché está disponível para Windows, Open VMS, Linux e a maioria das plataformas UNIX e é entregue em sistemas que vão de dois a mais de 10,000 de usuários simultâneos.

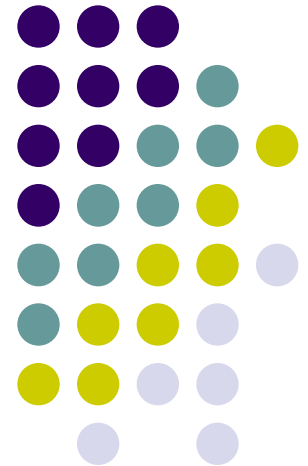
SGBDOO: Questionário



1. ?
2. ?
3. ?
4. ?
5. ?
6. ?
7. ?
8. ?
9. ?
10. ?
11. ?
12. ?
13. ?
14. ?
15. ?

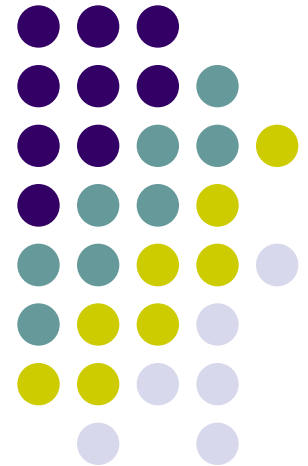
Sistemas Gereciadores de Banco de Dados

Prof. Marcos Alexandruk
alexandruk@uninove.br
www.unilivros.com.br



SGBDORs

Sistemas Gerenciadores de Banco
de Dados Objeto-Relacionais



SISTEMAS HÍBRIDOS (OBJETO-RELACIONAL)



- Fornecedores de bancos de dados relacionais adicionaram a seus produtos capacidade de incorporar objetos mais complexos (imagem, som e vídeo) além de recursos de orientação a objetos.
- No entanto, isso não os torna sistemas puramente orientados a objetos, apesar de sua denominação ORDMS – *Object-Relational Database Management System* (Sistema de Gerenciamento de Banco de Dados Objeto-Relacional).

SISTEMAS HÍBRIDOS (OBJETO-RELACIONAL)



- Esses sistemas na realidade implementam uma camada de abstração de dados em cima de métodos relacionais, o que torna possível a manipulação de dados mais complexos.
- Seguem, portanto, as especificações da SQL3 que fornecem capacidades estendidas e de objetos adicionadas ao padrão SQL.

SISTEMAS HÍBRIDOS (OBJETO-RELACIONAL)



- Alguns Sistemas de Gerenciamento de Banco de Dados Objeto-Relacionais:
 - Informix Universal Server
 - IBM DB2 Universal DB
 - Oracle Database 10g
- O **Informix Universal Server**, hoje pertencente à IBM, combina as tecnologias de banco de dados relacionais e banco de dados orientado a objetos que já existiam em dois produtos independentes: o **Informix Dynamic Server** e o **Illustra**.

IMPLEMENTAÇÃO OBJETO-RELACIONAL NO ORACLE



- O Oracle 10g incorporou ao banco de dados relacional a tecnologia orientada a objetos, tornando-se assim um SGBD Objeto-Relacional.
- Ele nem pode ser considerado puramente orientado a objetos, nem tampouco apenas relacional.
- Todas as características relacionais permanecem, ou seja, as tabelas continuam a existir, porém elas possuem alguns recursos adicionais.

IMPLEMENTAÇÃO OBJETO-RELACIONAL NO ORACLE



- Anteriormente, as tabelas apenas podiam conter valores **atômicos** em seus atributos, agora pode-se definir novos tipos de dados e usá-los para receber valores complexos.
- O Oracle, a partir da versão 9i, permite que os usuários criem outros tipos de objetos, de tabelas, referências para objetos, entre outros.
- *Observe a seguir alguns recursos interessantes oferecidos nas últimas versões do Oracle:*

IMPLEMENTAÇÃO OBJETO-RELACIONAL NO ORACLE



● TIPO OBJETO

- Podemos criar tipos de dados adicionais e depois fazer referência a eles dentro de outros objetos.
- Os tipos criados são gravados no esquema armazenado no banco de dados. Outras declarações que acessam o banco de dados podem fazer uso das definições desses tipos.

```
CREATE TYPE T_PESSOA AS OBJECT (  
  CODIGO_PESSOA NUMBER(5) ,  
  NOME_PESSOA VARCHAR2(50) ,  
  ENDERECO VARCHAR2(50)) NOT FINAL;
```

- Por padrão, os tipos de objeto são **FINAL**. Para permitir subtipos, deve ser obrigatoriamente adicionada a expressão **NOT FINAL** à declaração do tipo do objeto.

IMPLEMENTAÇÃO OBJETO-RELACIONAL NO ORACLE



● HERANÇA

- Um dos recursos mais importantes da orientação a objetos é a herança.
- A definição do tipo **T_PESSOA** no exemplo anterior pode funcionar como uma super-classe. A partir dela podemos definir outros dois tipos, **T_FISICA** e **T_JURIDICA**:

```
CREATE TYPE T_FISICA UNDER T_PESSOA (  
  CPF CHAR(11) ,  
  SEXO CHAR(1)) ;
```

```
CREATE TYPE T_JURIDICA UNDER T_PESSOA (  
  CNPJ CHAR(14) ,  
  INSC_ESTADUAL VARCHAR(30)) ;
```

IMPLEMENTAÇÃO OBJETO-RELACIONAL NO ORACLE



- **HERANÇA**

- Criação das tabelas PESSOA_FISICA e PESSOA_JURIDICA:

```
CREATE TABLE PESSOA_FISICA OF T_FISICA;  
CREATE TABLE PESSOA JURIDICA OF T_JURIDICA;
```

- Inserção de dados na tabela PESSOA_FISICA:

```
INSERT INTO PESSOA FÍSICA VALUES  
(1, 'FULANO', 'RUA X, 10', '11122233399', 'M');
```

- Seleção de dados na tabela PESSOA_FISICA:

```
SELECT * FROM PESSOA_FISICA;
```


IMPLEMENTAÇÃO OBJETO-RELACIONAL NO ORACLE



- **TABELAS ANINHADAS (NESTED TABLES)**

- Nested tables são tabelas cujo tipo de dado é domínio de outra tabela:

```
CREATE TYPE T_ENDERECO AS OBJECT (  
  RUA VARCHAR2(50) ,  
  NUMERO INTEGER ,  
  CIDADE VARCHAR2(50) ,  
  UF CHAR(2) ,  
  CEP CHAR(9) ) ;
```

```
CREATE TYPE ENDERECOS AS TABLE OF T_ENDERECO ;
```

IMPLEMENTAÇÃO OBJETO-RELACIONAL NO ORACLE



- **ARRAY**

- O exemplo a seguir cria um tipo chamado TELEFONES que permite inserir até cinco telefones diferentes:

```
CREATE TYPE TELEFONES AS VARRAY (5) OF VARCHAR2(10) ;
```

IMPLEMENTAÇÃO OBJETO-RELACIONAL NO ORACLE



- **CRIAÇÃO DE TABELAS COM OS TIPOS DEFINIDOS:**
 - O exemplo a seguir cria uma tabela chamada CLIENTE que utiliza os tipos definidos anteriormente:

```
CREATE TABLE CLIENTE (  
  CODIGO_CLIENTE NUMBER(5) ,  
  NOME_CLIENTE VARCHAR2(50) ,  
  TELEFONE_CLIENTE TELEFONES ,  
  ENDERECO_CLIENTE ENDERECOS)  
NESTED TABLE ENDERECO_CLIENTE STORE AS ENDERECOS_TAB;
```

IMPLEMENTAÇÃO OBJETO-RELACIONAL NO ORACLE



- **INSERIR DADOS NA TABELA:**

```
INSERT INTO CLIENTE VALUES (  
1, 'FULANO',  
TELEFONES ('11111111', '22222222'),  
ENDERECOS (  
T_ENDERECO ('RUA X', 10, 'SÃO PAULO', 'SP', '10000-000'),  
T_ENDERECO ('RUA Y', 20, 'JUNDIAÍ', 'SP', '20000-000')));
```

- **SELECIONAR DADOS DA TABELA:**

```
SELECT * FROM CLIENTE;
```

```
SELECT C.CODIGO_CLIENTE, C.NOME_CLIENTE, E.RUA  
FROM CLIENTE C, TABLE(C.ENDERECO_CLIENTE) E;
```

SGBDOR: Questionário



1. O que são os SGBDORs (Sistemas de Gerenciamento de Banco de Dados Objeto-Relacionais)?
2. Que especificação seguem os SGBDORs?
3. Cite o nome de três produtos que podem ser caracterizados como SGBDORs.
4. Que comando deve ser utilizado para criar um tipo objeto?
5. Para permitir subtipos, que expressão deve ser adicionada à declaração do tipo do objeto?
6. Que comando deve ser utilizado para criar um subtipo? (Exemplo: T_DOIS herda de T_UM)
7. O que são tabelas aninhadas (nested tables)?
8. Que comando deve ser utilizado para criar um array?

EXERCÍCIOS



1. CRIAR UM TIPO DENOMINADO T_NOTA COM A SEGUINTE ESTRUTURA:

```
COD_DISC NUMBER(5)
ANO       NUMBER(4)
SEMESTRE  NUMBER(1)
NOTA      NUMBER(3,1)
```

```
CREATE OR REPLACE TYPE T_NOTA AS OBJECT (
COD_DISC NUMBER(5) ,
ANO NUMBER(4) ,
SEMESTRE NUMBER(1) ,
NOTA NUMBER(3,1)) ;
/
```

2. CRIAR UM TIPO TABELA DENOMINADO TB_NOTA PARA O TIPO_NOTA

```
CREATE TYPE TB_NOTA AS TABLE OF TIPO_NOTA;
/
```

EXERCÍCIOS



3. CRIAR UMA TABELA DENOMINADA TB_ALUNO COM A SEGUINTE ESTRUTURA:

RA VARCHAR2(10)

NOME VARCHAR2(50)

NOTAS TIPO_NOTA)

NESTED TABLE NOTAS STORE AS TB_NOTAS_ALUNOS;

CREATE TABLE TB_ALUNO (

RA VARCHAR2(10),

NOME VARCHAR2(50),

NOTAS TIPO_NOTA)

NESTED TABLE NOTAS STORE AS TB_NOTAS_ALUNOS;

EXERCÍCIOS



4. INSERIR NA TABELA TB_ALUNO OS SEGUINTE REGISTROS:

RA='111222333'

NOME='ANTONIO'

COD_DISC=1001; ANO=2010; SEMESTRE=1; NOTA=3.5

COD_DISC=1001; ANO=2010; SEMESTRE=2; NOTA=5.0

COD_DISC=1001; ANO=2011; SEMESTRE=1; NOTA=7.5

```
INSERT INTO TB_ALUNO VALUES ('111222333', 'ANTONIO',  
TB_NOTA(  
TIPO_NOTA(1001,2010,1,3.5),  
TIPO_NOTA(1001,2010,2,5.0),  
TIPO_NOTA(1001,2011,1,7.5)  
));
```

5. APRESENTAR OS DADOS DO(S) ALUNO(S) E SUAS RESPECTIVAS NOTAS:

```
SELECT A.RA, A.NOME, N.COD_DISC, N.ANO, N.SEMESTRE, N.NOTA  
FROM TB_ALUNO A, TABLE (A.NOTAS) N;
```

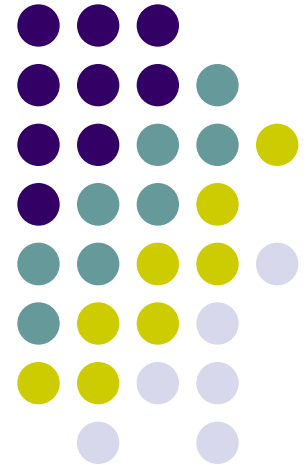

Sistemas Gereciadores de Banco de Dados

Prof. Marcos Alexandruk
alexandruk@uninove.br
www.unilivros.com.br

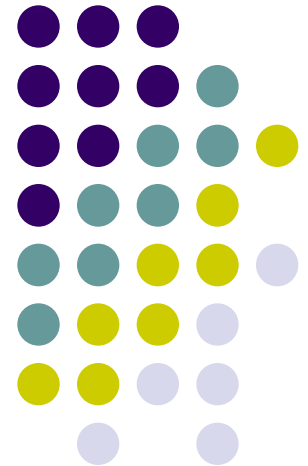


SGBDs

- ❖ IBM DB2
- ❖ MySQL
- ❖ PostgreSQL
- ❖ MS SQL Server
- ❖ Oracle Database



IBM DB2



DB2: Introdução



- O **DB2 Express-C** faz parte da família de produtos IBM DB2 para gerenciamento de dados relacionais e no formato XML.
- A letra **C** no nome DB2 Express-C significa **Comunidade**.
 - Os membros da comunidade incluem:
 - Desenvolvedores de aplicação que precisam de padrões abertos de software de banco de dados;
 - 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 pequenas ou de porte médio que precisam da segurança de um servidor de banco de dados para suas aplicações e operações;
 - Entusiastas da tecnologia de ultima geração que querem utilizar um banco de dados fácil para construir aplicações para a Web 2.0;
 - Estudantes, professores, e outros usuários acadêmicos que querem um servidor de banco de dados versátil para lecionar, realizar projetos e pesquisa.



DB2: Introdução

- O **DB2 Express-C** utiliza a mesma funcionalidade principal e o **mesmo código fonte** de outras edições do DB2 para Linux, UNIX, e Windows.
- Pode ser utilizado em sistemas **32-bit** ou **64-bit** em sistemas operacionais **Linux** ou **Windows**.
- Inclui o padrão **pureXML** sem nenhum custo. O padrão pureXML é uma tecnologia única do DB2 para armazenar e processar documentos XML nativo.



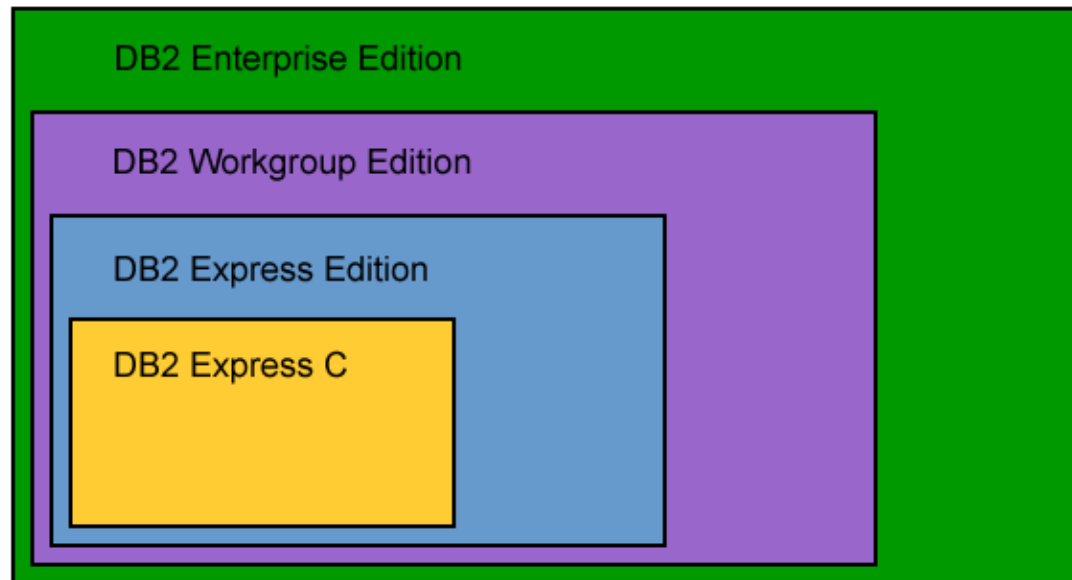
DB2: Suporte técnico

- Dúvidas técnicas podem ser postadas no fórum do DB2 Express-C. Este fórum é monitorado pelo time do DB2 Express-C e a comunidade de voluntários que participam do fórum poderá responder as 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, através de uma assinatura anual (US\$ 2.995 por servidor por ano nos Estados Unidos) pode-se utilizar duas características adicionais: HADR (cluster para Alta Disponibilidade e Desastre e Recuperação), e replicação de SQL (para replicar dados entre outros servidores DB2).

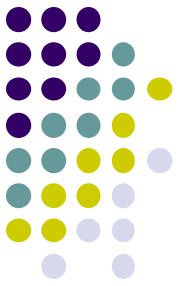


DB2: Edições

- 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. Qualquer aplicação desenvolvida em uma edição poderá ser executada, sem nenhuma modificação, em outras edições.
- A figura abaixo ilustra a diferença entre as edições do DB2.



Edições dos servidores IBM DB2



DB2: Softwares gratuitos

- 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**: Ferramenta baseada no Eclipse na qual pode ser realizada a modelagem, desenvolvimento, distribuição e gerenciamento de dados. O Data Studio substitui o DB2 Developer Workbench 9.1.
 - **DB2 Net Search Extender**: Permite executar consultas rápidas e detalhadas em arquivos de texto, incluindo qualquer documento XML armazenado nativamente no DB2.
 - **DB2 Spatial Extender**: Amplia a função do DB2 com um conjunto de técnicas avançadas de tipos de dados geográficos que representam desenhos geométricos como pontos, linhas, e polígonos.



DB2: Softwares pagos

- Os seguintes produtos pagos que estão relacionados aos DB2:
 - **DB2 Connect:** Permite que o DB2 no Linux, UNIX ou Windows se conecte ao DB2 nas plataformas z/OS ou DB2 i5/OS. 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).
 - **WebSphere Federation Server:** Utilizado em bancos de dados federados para realizar consultas que utilizem objetos de diferentes bancos de dados relacionais. Por exemplo:

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

- **WebSphere Replication Server:** Permite a replicação SQL de registros de base de dados não IBM. Inclui a característica chamada Que-Replication ou replicação de dados utilizando filas de mensagens.



DB2: Instalação

- O DB2 Express-C está disponível para Linux, Solaris (x64), e Windows 2003, 2000, XP, Vista e Seven.
- A arquitetura de processador disponível é 32-bit, 64-bit e PowerPC (Linux).
- Para executar o DB2 em outras plataformas (como UNIX), deve-se comprar uma das outras edições do DB2.
- 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.



DB2: Instalação

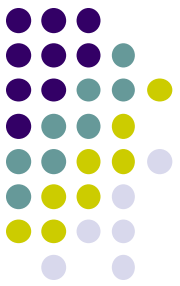
- Para instalar o DB2 Express-C no **Linux** ou no **Windows**, você deve usar o sistema operacional de acordo com as seguintes regras.
- No **Linux**, você deve ser o usuário root (ou super usuário) para instalar o DB2 Express-C. Você também pode instalar o DB2 Express-C como um usuário não root; porém, 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 instancia 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.



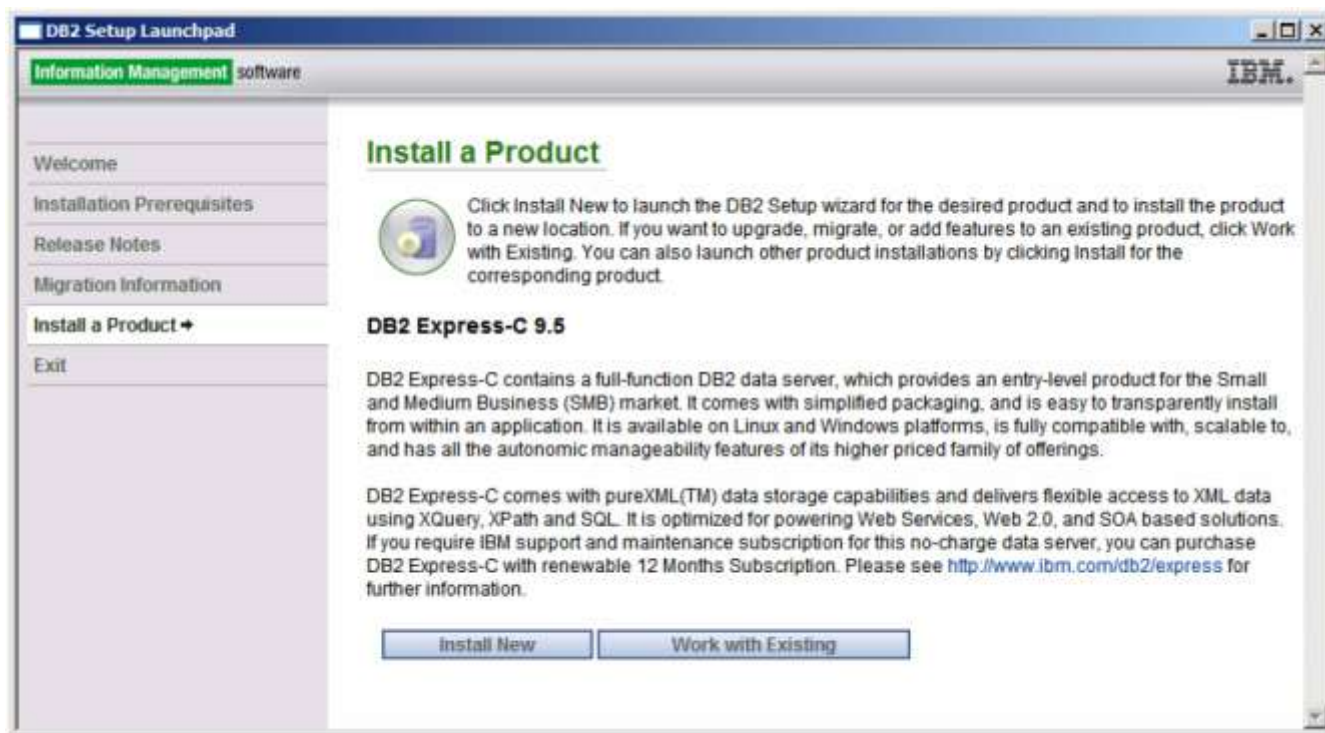
DB2: Instalação

- 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.

DB2: Instalação

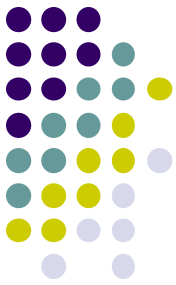


- A figura abaixo mostra a tela “DB2 Setup Launchpad”. Clique em “Install a Product” e escolha “Install New”.
- Depois de aceitar a licença, é aconselhado escolher a instalação típica (Typical).



Instalação do DB2 Express C

DB2: Instalação



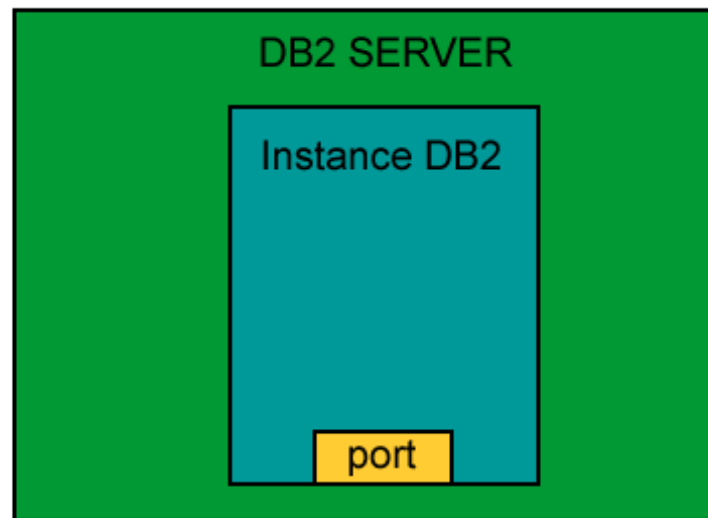
- Na tela mostrada abaixo, deve-se entrar com o nome do usuário que será utilizado para trabalhar com as instâncias e outros serviços. Este usuário deve fazer parte do grupo de Administradores Locais do Windows. Se o ID do usuário não existir, ele será criado como Administrador Local. Pode-se deixar o campo “domain” em branco se o ID do usuário não pertencer a nenhum domínio. O ID padrão do usuário criado no Windows é **db2admin** (db2inst1 no Linux).

Informações do usuário na configuração de uma instância padrão do DB2



DB2: Instances

- Como parte da instalação no Windows, uma instância padrão, chamada **DB2** (**db2inst1** no Linux) é criada.
- Uma instância é simplesmente um ambiente independente onde aplicações podem ser executadas e bases de dados podem ser criadas.

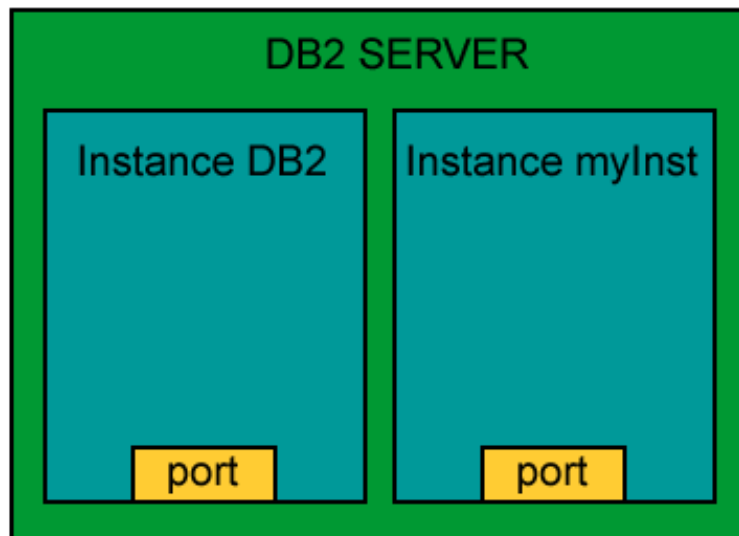


Instância padrão criada pelo DB2



DB2: Instances

- Você pode criar várias instâncias em um servidor, e usá-las para diferentes propósitos. (uma instância pode ser usada para armazenar bases de dados de ambiente de produção, outra para bases de dados do ambiente de testes, etc.)
- Todas estas instâncias são independentes (as operações executadas em uma instância não afetam as outras instâncias).



Servidor DB2 com duas instâncias

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.



DB2: Instances

- Para criar uma nova instância use o comando:
`db2icrt myinst`
- Se você utilizar o [Command Window](#), você pode tornar qualquer instância do DB2 ativa usando o seguinte comando:
`set db2instance = myinst`
- 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. Portanto, para trocar de instância basta simplesmente trocar de usuário (com o comando `su`).

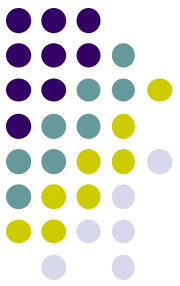


DB2: Instances

- A tabela abaixo 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 todas as instâncias do sistema
db2 get instance	Lista as instâncias ativas

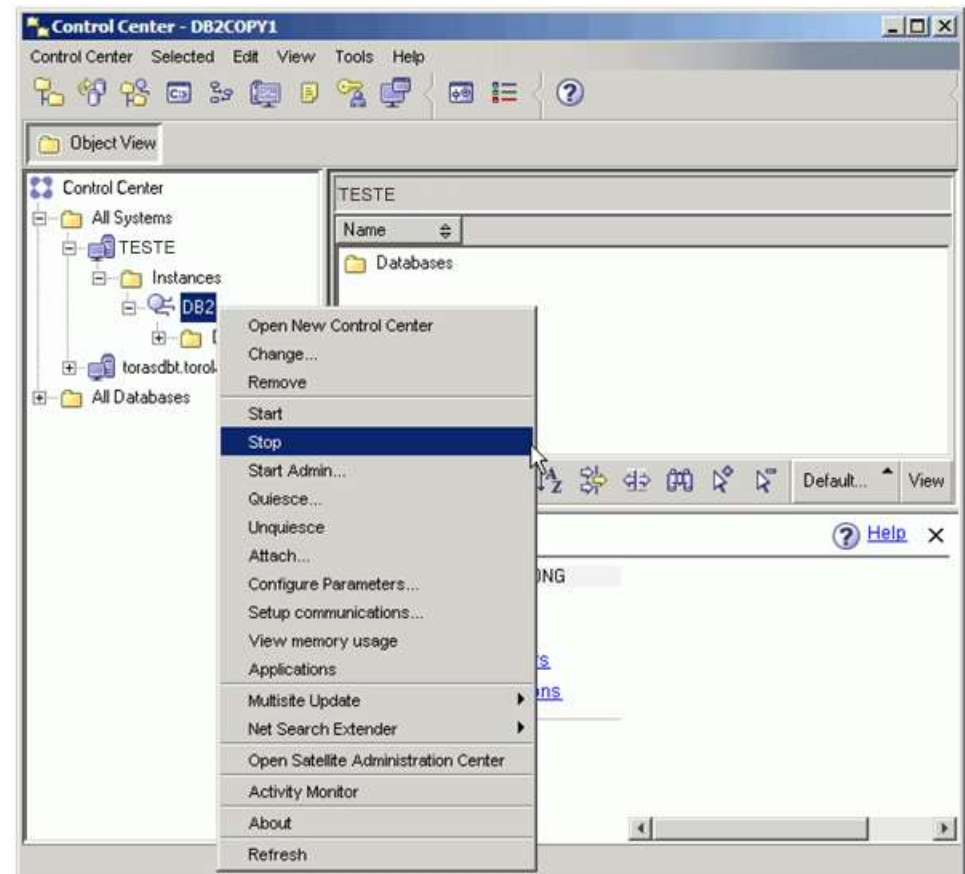
DB2: Instances



- Os comandos podem também ser executados através do Control Center.

Exemplo: expandir o diretório **Instances**, clicar com o botão direito na instância desejada e escolher a opção **Stop** é equivalente a executar o comando **db2stop** no DB2

Command Window.



Control Center: comandos de instância



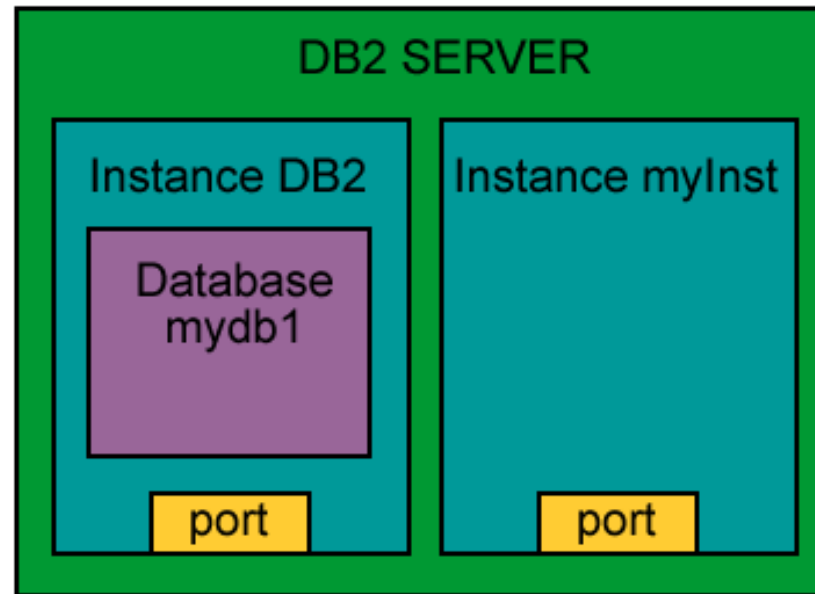
DB2: Databases

- Para criar uma base de dados na instância ativa, execute o comando:
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.



DB2: Databases

- A figura abaixo mostra uma representação de uma base de dados **mydb1** criada dentro da instância **DB2**:



Base de dados mydb1 criada na instância DB2



DB2: Databases

- A tabela abaixo mostra alguns comandos úteis para se trabalhar em nível de instância:

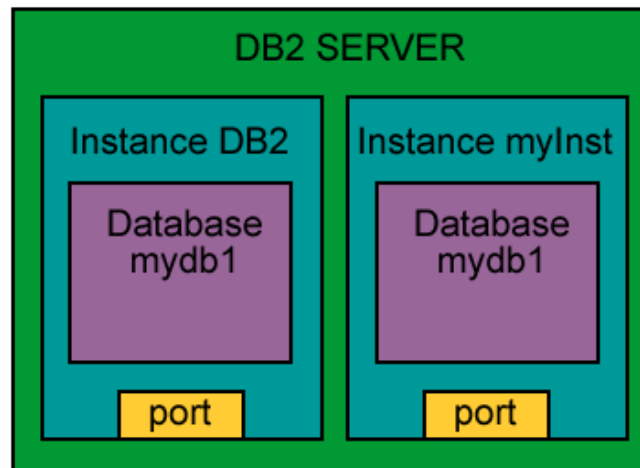
COMANDO	DESCRIÇÃO
db2 create database	Cria uma nova base de dados
db2 drop database	Exclui uma base de dados
db2 connect to <database_name>	Conecta a uma base de dados
db2 create table <table_name>	Cria uma nova tabela
db2 create view <view_name>	Cria uma nova visão
db2 create index <index_name>	Cria um novo índice



DB2: Databases

- 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
```

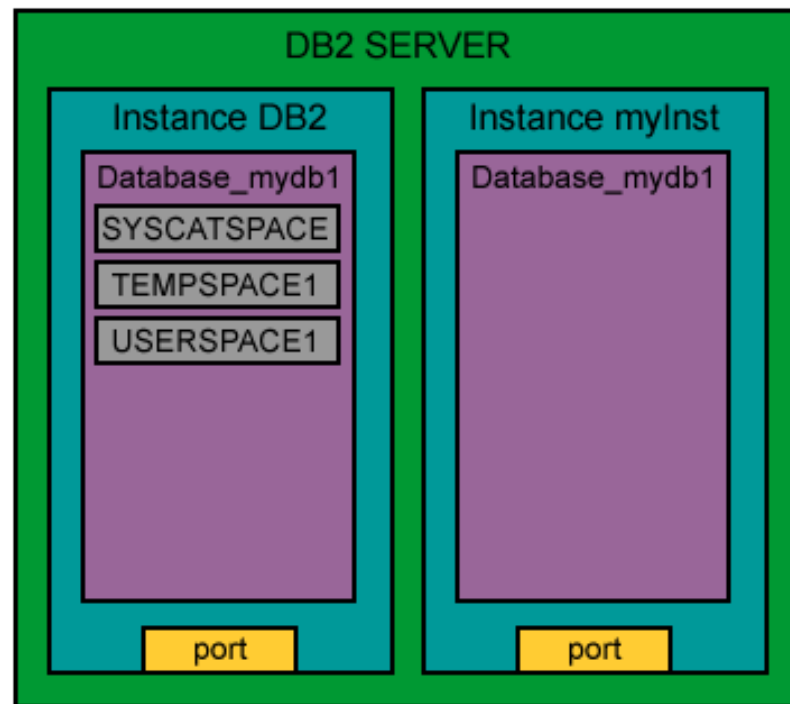


Base de dados mydb1 criada na instância myInst



DB2: Databases

- Quando uma base de dados é criada, vários objetos são criados por padrão: tablespaces, tabelas, um buffer pool e arquivos de log. A figura abaixo mostra três tablespaces criadas por padrão.



Tablespaces criadas por padrão quando uma base de dados é criada



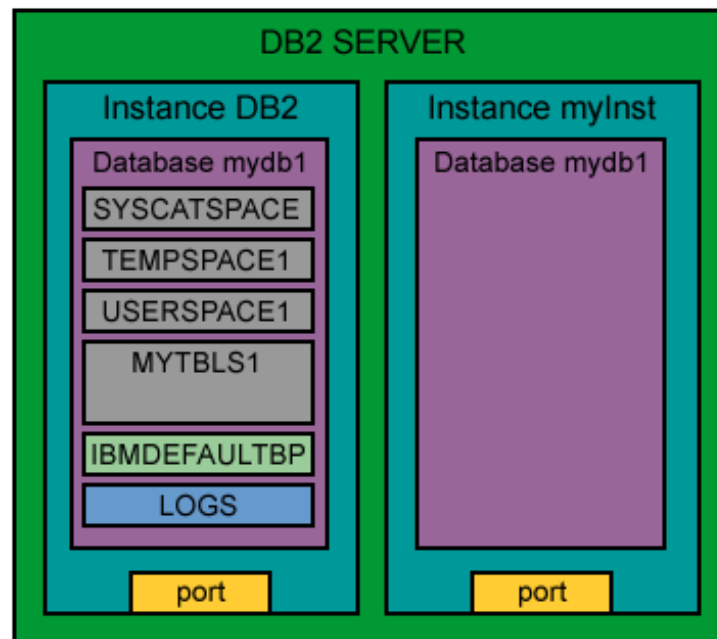
DB2: Tablespaces

- 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.



DB2: Tablespaces

- Você também pode criar suas próprias tablespaces usando o comando `CREATE TABLESPACE`. A figura abaixo mostra a tablespace **MYTBLS1** criada dentro da base de dados **mydb1** na instância **DB2**. São mostrados também dois outros objetos criados por padrão: um buffer pool chamado **IBMDEFAULTBP** e os **arquivos de log**.



Buffer pool e logs criados por padrão



DB2: Tablespaces

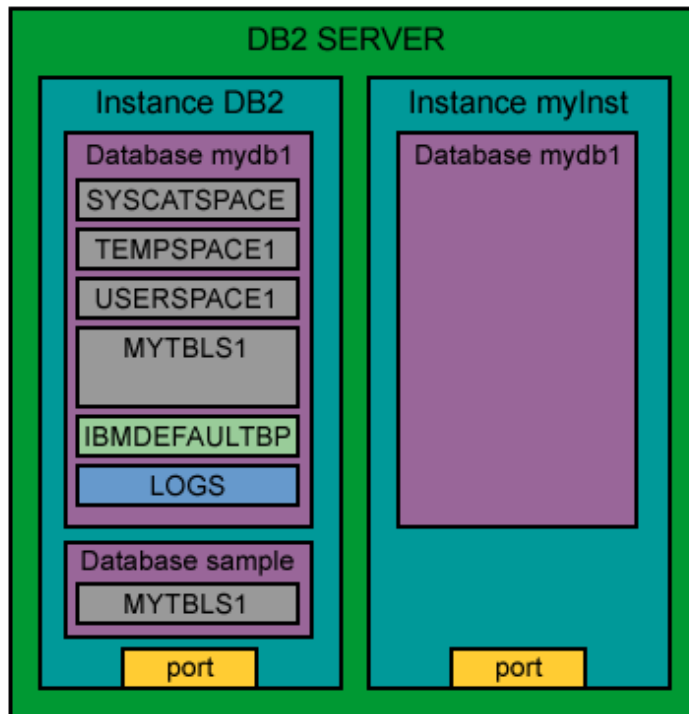
- Quando se cria uma tablespace, deve-se especificar quais discos e memória (buffer pool) devem ser usados.
- Um **buffer pool** é basicamente um cache usado pela base de dados. Você pode criar um ou mais buffer pools, mas sempre deve haver um buffer pool com o tamanho de página¹ correspondente ao tamanho de página dos tablespaces existentes.
- 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.

¹ *Página é a unidade mínima de armazenamento no DB2. Os tamanhos de página permitidos são: 4K, 8K, 16K e 32K.*

DB2: Tablespaces



- 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 abaixo mostra a tablespace **MYTBLS1** em ambas as bases de dados **mydb1** e **sample**, dentro da instância **DB2**.



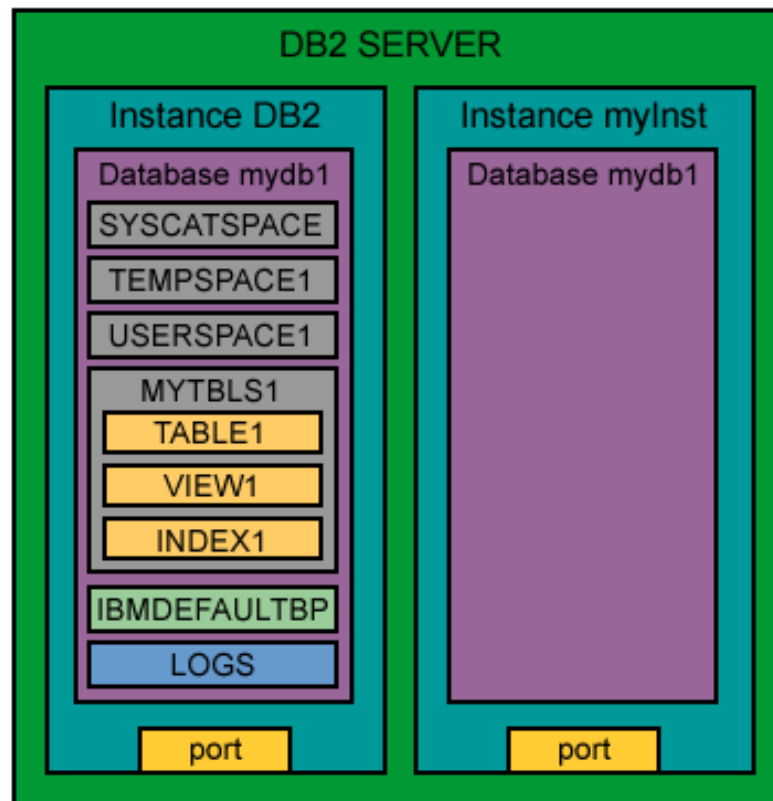
*Outros objetos padrão para a base de dados **sample** não são mostrados devido a restrições de espaço na figura.*

Tablespaces com mesmo nome em diferentes databases



DB2: Tablespaces

- Uma vez que o tablespace foi criado, pode-se criar objetos dentro do tablespace, como tabelas, visões e índices.



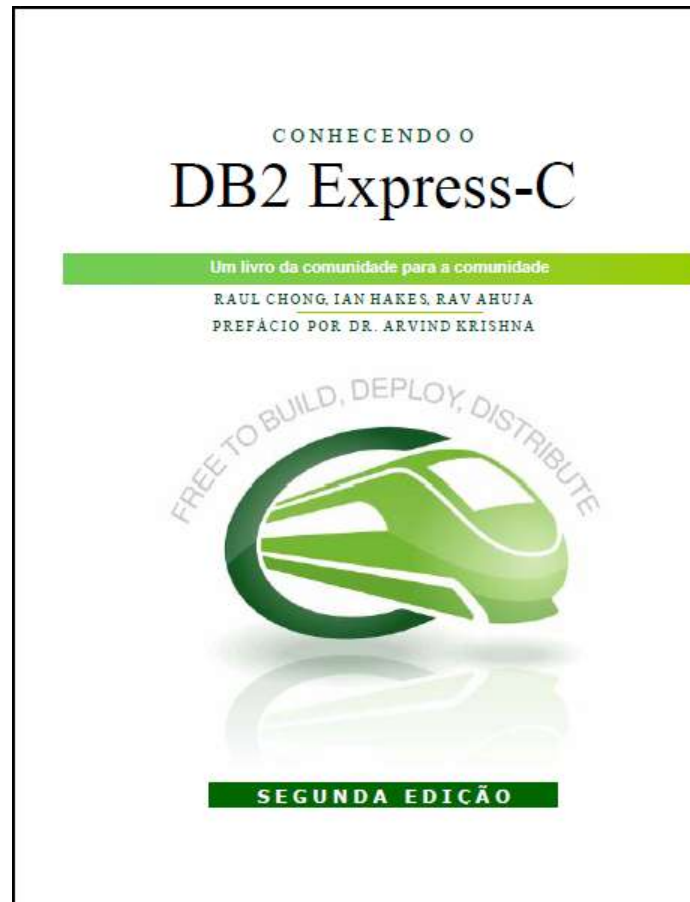
Tabelas, visões e índices criados em MYTBLS1

DB2: Livro



Livro gratuito “Conhecendo o DB2 Express C” (267 páginas) disponível em:

ftp://ftp.software.ibm.com/software/data/db2/express-c/wiki/Getting_Started_with_DB2_Express-C_9.5_Portuguese_Brazil.pdf



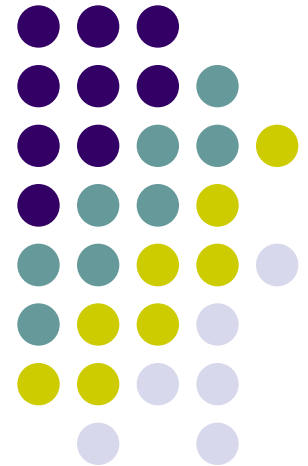
DB2: Questionário



1. Quem compõe a comunidade que fornece suporte ao DB2 Express C?
2. O DB2 Express C é compatível com quais plataformas? (Hardware e Software)
3. O que é o padrão pureXML?
4. Além das imagens do DB2 Express-C quais outros softwares úteis relacionados podem ser baixados gratuitamente?
5. Descreva as características de três softwares pagos relacionados ao DB2.
6. Quais são as limitações em termos de CPU e memória do DB2 Express C respectivamente na versão gratuita e na versão paga?
7. Que privilégios devem possuir os usuários para instalar o DB2 Express C no Windows e no Linux?
8. Que instâncias são criadas como padrão no Windows e no Linux?
9. O que são instâncias? É possível criar mais de uma em um servidor DB2 Express C? Em caso afirmativo qual seria o propósito de criar mais de uma instância no servidor?
10. Que comando deve ser utilizado no DB2 Express C para criar uma nova instância denominada TESTINST?
11. É possível criar mais de uma base de dados em uma instância? Em caso afirmativo, é possível que a nova base de dados tenha um nome idêntico a outra criada em uma instância diferente?
12. Que comando deve ser utilizado para criar uma base de dados denominada TESTDB?
13. Quando uma base de dados é criada, que outros objetos são criados por padrão?
14. O que contém basicamente a tablespace SYSCATSPACE?
15. Explique as funções do buffer pool e dos arquivos de log.

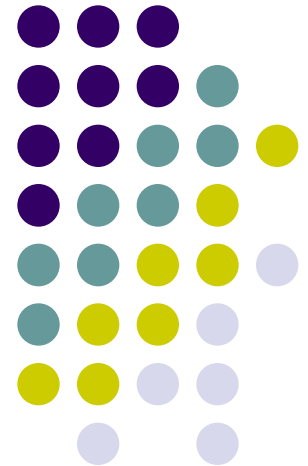
Sistemas Gereciadores de Banco de Dados

Prof. Marcos Alexandruk
alexandruk@uninove.br
www.unilivros.com.br



Aula 7

MySQL



MySQL: História



- 1979: David Axmark, Allan Larsson (suecos) e Michael Widenius (finlandês) da TcX desenvolvem uma ferramenta (não SQL) denominada Unireg (escrita originalmente em BASIC e reescrita em C).
- 1994: Michael Widenius inicia o desenvolvimento de um gerenciador de banco de dados de código aberto com base nos programas Unireg e mSQL.
- 1995: Os três iniciam o desenvolvimento do MySQL pela empresa MySQL AB.
- 1996: Lançamento da versão 3.11.1 do MySQL.

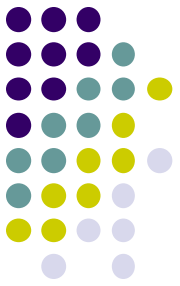


MySQL: História

- 16/01/2008 – A MySQL é adquirida pela Sun Microsystems por aproximadamente US\$ 1 bilhão.
- 20/04/2009 – A Oracle adquire a Sun Microsystems por aproximadamente US\$ 7,4 bilhões.
- 14/12/2009 – A Oracle assume o compromisso de manter o MySQL sob a licença GPL.¹

- ¹ Documento disponível em:

<http://www.marketwire.com/press-release/Oracle-Makes-Commitments-to-Customers-Developers-and-Users-of-MySQL-NASDAQ-ORCL-1090000.htm>



MySQL: Empresas usuárias

- Motorola
- NASA
- Silicon Graphics
- Texas Instruments
- Yahoo
- Wikipedia

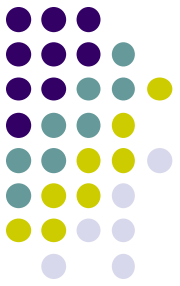


MySQL: Licença de uso

GNU-GPL (General Public License): <http://www.gnu.org>

- Permite utilizar o software para qualquer propósito
- Permite a distribuição livre do software
- Permite que o funcionamento do software seja estudado a partir do código-fonte
- Permite que o código-fonte seja alterado desde que o novo código-fonte continue sendo livre conforme a mesma licença

Licença comercial: <http://www.mysql/oem/licensing.html>



MySQL: Compatibilidade

Sistemas Operacionais:

- Windows (2000, 2003, XP)
- Linux (Fedora, Debian, SuSE, RedHat)
- Unix (Solaris, HP-UX, AIX, SCO)
- FreeBSD
- Mac OS X Server

MySQL: Padrão SQL



- Padrão SQL-92
- Padrão SQL-99 e SQL 2003 (parcialmente)

MySQL: Última versão



- MySQL Community Server 5.5.10
- A MySQL disponibiliza também a seguinte "ferramenta gráfica" (GUI Tool) para administração do banco de dados:
- MySQL Workbench 5.2.33

Dados atualizados em março/2011



MySQL: Características

- **PORTABILIDADE:** Desenvolvido em C/C++ torna-se portátil entre diferentes plataformas e compiladores.
- **MULTITHREADS:** Aumenta significativamente a velocidade de processamento e facilita a integração com hardware com mais de uma CPU.
- **ARMAZENAMENTO:** Disponibiliza vários tipos de tabelas para priorizar a velocidade ou o volume de dados, entre outras características.
- **VELOCIDADE:** As tabelas tipo ISAM (MyISAM na versão 5) utilizam cachês em consultas e indexação BTREE para tabelas tipo HEAP proporcionando maior velocidade de acesso.



Thread

- Thread é a forma de um processo dividir a si mesmo em duas ou mais tarefas que podem ser executadas simultaneamente. O suporte à thread é fornecido pelo próprio sistema operacional (SO), no caso da Kernel-Level Thread (KLT), ou implementada através de uma biblioteca de uma determinada linguagem, no caso de uma User-Level Thread (ULT).
- Sistemas que suportam apenas uma única tarefa são chamados de **monothread** e os que suportam múltiplas tarefas simultâneas são chamados de **multithread**.



MySQL: Características

- **FULL TEXT SEARCH:** Visa atender as necessidades de desempenho em consultas de grandes quantidades de texto. O MySQL permite gerenciar os índices com base em campos do tipo texto (para bancos de pequeno ou médio porte) ou em uma tabela própria de índices (para bancos de grande porte).
- **STORED PROCEDURES:** Blocos de código armazenados no servidor e que podem ser invocados a partir de outras aplicações.
- **TRIGGERS:** Blocos de código armazenados no servidor são invocados automaticamente a partir de certos eventos.



MySQL: Características

- **CURSORES:** Permitem a navegação em conjuntos de resultados através de laços de repetição possibilitando realizar operações e transações à parte para cada linha de uma tabela.
- **VISÕES:** Consultas pré-programadas à partir de determinadas colunas de uma ou mais tabelas.
- **TRANSAÇÕES DISTRIBUÍDAS XA:** Uma espécie de extensão da ACID (Atomicidade, Consistência, Isolamento e Durabilidade), fornece a possibilidade de gerenciamento de transações realizadas com a união de vários bancos de dados.

ACID



- **Atomicidade:** a transação deverá ser executada por completo ou abortada (“tudo ou nada”)
- **Consistência:** cada campo deverá ser preenchido com valores válidos dentro do seu domínio e conforme os relacionamentos entre tabelas (as transações não podem quebrar as regras do Banco de Dados)
- **Isolamento:** nos casos de transações concorrentes uma somente poderá ser executada após a conclusão da outra (duas transações só podem ser simultâneas se elas não alterarem os mesmos dados)
- **Durabilidade:** os efeitos de uma transação em caso de sucesso (commit) são permanentes (em caso de erro todos os campos terão seus dados restaurados com seus valores originais)



MySQL: Características

- **INTEGRIDADE REFERENCIAL:** Relacionamentos entre diferentes tabelas são gerenciados pelo banco de dados na inclusão, alteração ou exclusão de dados.
- **REPLICAÇÃO:** Torna possível configurar clones ou réplicas de servidores que mantêm as informações sincronizadas com um servidor principal aumentando a disponibilidade.
- **CLUSTERIZAÇÃO:** Baseada na integração e sincronismo de dois ou mais servidores para dividir a demanda e aumentar a disponibilidade. Este recurso permite que caso o servidor primário fique indisponível, a carga gerada pelas consultas seja balanceada entre os outros servidores restantes.



Métodos de armazenamento

- **MyISAM**

- Método padrão do MySQL
- Muito rápido
- Não apresenta restrições de tipos de dados
- Permite o uso de todos os recursos do MySQL, exceto suporte a transações
- Único mecanismo do MySQL que suporta buscas do tipo FullText Searches
- Nível de bloqueio: tabelas



Métodos de armazenamento

- **InnoDB**

- Recomendado para bancos de dados grandes e complexos
- Oferece suporte a transações ACID (Atomicidade, Consistência, Isolamento e Durabilidade)
- Armazenamento em disco e memória dos dados e índices: processamento mais veloz
- Nível de bloqueio: linhas. Portanto, aumenta a disponibilidade: apenas os registros envolvidos em uma transação são bloqueados (não a tabela toda como no MyISAM)

Métodos de armazenamento



- **Memory (HEAP)**

- O armazenamento dos dados é realizado na memória RAM
- Velocidade de processamento muito rápida: não há busca em disco
- Dados são perdidos quando o servidor é deslido ou reinicializado
- Indicado em aplicações cujos dados devem ser armazenados apenas temporariamente (ex: sessões)
- Nível de bloqueio: tabelas
- Não suporta dados do tipo BLOB e TEXT
- Não oferece suporte a transações e índices



Métodos de armazenamento

- **BerkeleyDB**

- Baseia-se na geração de código hash (identificador único), armazenado com cada registro (funcionando como uma espécie de “chave primária”)
- Eficiente nas recuperações de dados baseadas em chaves única (menos eficiente nas recuperações de múltiplos registros sequenciais)
- Suporta todos os tipos de dados
- Nível de bloqueio: página (blocos de 8.192 bytes)



Métodos de armazenamento

- **ARCHIVE**

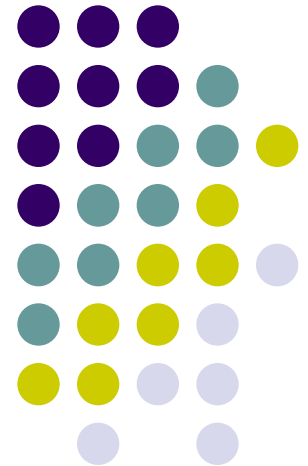
- Dados são gravados em arquivos-texto no formato de tabelas sequenciais (ordem de gravação)
- Utilizado em aplicações de log e outras que podem gerar grande volume de dados (podendo ser armazenadas posteriormente em unidades de backup mensalmente ou em outros períodos)
- Mecanismo de funcionamento limitado: suporta apenas INSERT e SELECT
- Não dá suporte a índices: cada SELECT percorre a tabela inteira para garantir que todos os resultados foram obtidos
- Não oferece suporte a transações e índices

Métodos de armazenamento



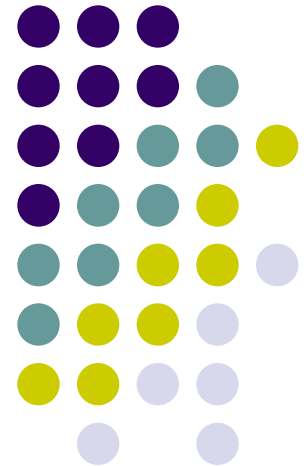
- **CSV (Command Separated Values)**
 - Similar ao método ARCHIVE, armazena os dados em arquivos texto. Os valores são separados por vírgula ou outro caractere definido previamente.
 - O padrão CSV permite que os dados tornem-se portáteis para outras aplicações (ex: planilhas) de forma simples e rápida
 - Não oferece suporte a transações e índices
 - Não recomendado o uso de tipos BLOB e TEXT, pois podem comprometer a portabilidade

Instalando o MySQL

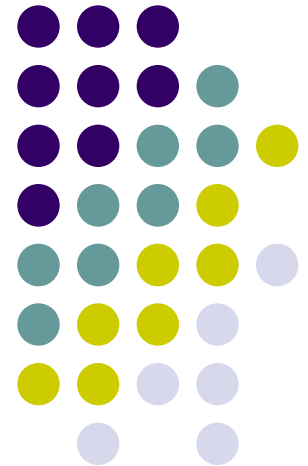


Instalando o MySQL

[vídeo]



Utilizando o MySQL





Acessando o MySQL

A screenshot of a Windows command prompt window titled "Prompt de comando - mysql -u root -p". The window has a blue title bar and standard Windows window controls (minimize, maximize, close). The command prompt shows the following text:

```
C:\>mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5
Server version: 5.1.45-community MySQL Community Server (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql>
```




Criando um novo Banco de Dados

```
C:\ Prompt de comando - mysql -u root -p

C:\>mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5
Server version: 5.1.45-community MySQL Community Server (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE TESTE;
Query OK, 1 row affected (0.01 sec)

mysql> _
```



Apresentando os Bancos de Dados

```
C:\ Prompt de comando - mysql -u root -p

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| test |
| teste |
+-----+
4 rows in set (0.03 sec)

mysql> _
```



Selecionando o novo Banco de Dados

```
C:\ Prompt de comando - mysql -u root -p

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| test |
| teste |
+-----+
4 rows in set (0.03 sec)

mysql> USE TESTE;
Database changed
mysql> _
```



Eliminando um Banco de Dados

```
C:\ Prompt de comando - mysql -u root -p

mysql>
mysql> DROP DATABASE TESTE;
Query OK, 0 rows affected (0.02 sec)

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| test |
+-----+
3 rows in set (0.00 sec)

mysql> _
```



Selecionando um Banco de Dados

```
C:\ Prompt de comando - mysql -u root -p
mysql>
mysql> USE TEST;
Database changed
mysql>
mysql> SHOW TABLES;
Empty set (0.00 sec)

mysql> _
```



Criando uma Tabela

```
C:\ Prompt de comando - mysql -u root -p

mysql> CREATE TABLE CLIENTE (
  -> ID INT AUTO_INCREMENT,
  -> NOME VARCHAR (50),
  -> COMENTARIO TEXT,
  -> PRIMARY KEY (ID),
  -> INDEX (NOME),
  -> FULLTEXT (COMENTARIO))
  -> ENGINE = MyISAM;
Query OK, 0 rows affected (0.06 sec)

mysql> SHOW TABLES;
+-----+
| Tables_in_test |
+-----+
| cliente        |
+-----+
1 row in set (0.00 sec)

mysql>
```



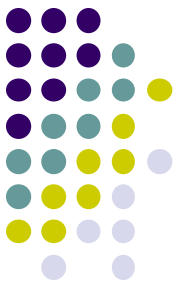
Apresentando estrutura de uma Tabela

```
C:\ Prompt de comando - mysql -u root -p

mysql> DESCRIBE CLIENTE;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key  | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| ID         | int(11)       | NO   | PRI  | NULL    | auto_increment |
| NOME       | varchar(50)   | YES  | MUL  | NULL    |                |
| COMENTARIO | text          | YES  | MUL  | NULL    |                |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> DESC CLIENTE;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key  | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| ID         | int(11)       | NO   | PRI  | NULL    | auto_increment |
| NOME       | varchar(50)   | YES  | MUL  | NULL    |                |
| COMENTARIO | text          | YES  | MUL  | NULL    |                |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)

mysql>
```



Inserindo dados em uma Tabela

```
C:\ Prompt de comando - mysql -u root -p

mysql> INSERT INTO CLIENTE (NOME,COMENTARIO) VALUES ('Fulano','Texto 1');
Query OK, 1 row affected (0.003 sec)

mysql> SELECT * FROM CLIENTE;
+-----+-----+-----+
| ID | NOME   | COMENTARIO |
+-----+-----+-----+
| 1  | Fulano | Texto 1    |
+-----+-----+-----+
1 row in set (0.000 sec)

mysql> INSERT INTO CLIENTE (NOME,COMENTARIO) VALUES ('Beltrano','Texto 2');
Query OK, 1 row affected (0.000 sec)

mysql> SELECT * FROM CLIENTE;
+-----+-----+-----+
| ID | NOME   | COMENTARIO |
+-----+-----+-----+
| 1  | Fulano | Texto 1    |
| 2  | Beltrano | Texto 2    |
+-----+-----+-----+
2 rows in set (0.000 sec)

mysql>
```




Eliminando uma Tabela

```
C:\ Prompt de comando - mysql -u root -p

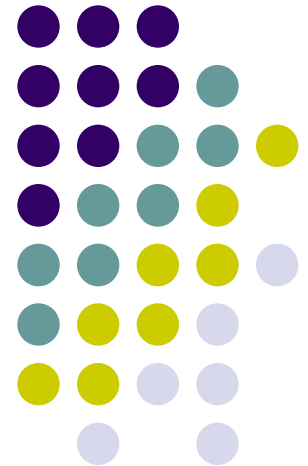
mysql> DROP TABLE CLIENTES;
Query OK, 0 rows affected (0.00 sec)

mysql> SHOW TABLES;
Empty set (0.00 sec)

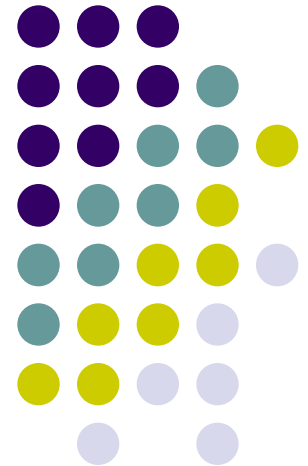
mysql> _
```

Sistemas Gereciadores de Banco de Dados

Prof. Marcos Alexandruk
alexandruk@uninove.br
www.unilivros.com.br



PostgreSQL





O que é o PostgreSQL?

- O PostgreSQL é um SGBD (Sistema Gerenciador de Banco de Dados) objeto-relacional de código aberto.

Ele é considerado objeto-relacional por implementar, além das características de um SGBD relacional, algumas características de orientação a objetos, como herança e tipos personalizados.

- A equipe de desenvolvimento do PostgreSQL sempre teve uma grande preocupação em manter a compatibilidade com os padrões SQL92/SQL99.)



Breve histórico

- O PostgreSQL é resultado de uma ampla evolução que se iniciou com o projeto Ingres, desenvolvido na Universidade de Berkeley, Califórnia.
- O líder do projeto, Michael Stonebraker, um dos pioneiros dos bancos de dados relacionais, deixou a universidade em 1982 para comercializar o Ingres, porém retornou a ela logo em seguida.
- Após seu retorno a Berkeley, em 1985, Stonebraker começou um projeto pós-Ingres com o objetivo de resolver problemas com o modelo de banco de dados relacional.

O principal problema era a incapacidade do modelo relacional compreender “tipos” (atualmente, chamados de objetos), ou seja, combinações de dados simples que formam uma única unidade.



Breve histórico

- O POSTGRES foi originalmente patrocinado pelo DARPA (Defense Advanced Research Projects Agency), ARO (Army Research Office), NSF (National Science Foundation) e ESL Inc.
- O projeto POSTGRES iniciou em 1986, já em 1987 tornou-se operacional.
- Em 1989 foi lançada a primeira versão para o público externo.
- Em 1991 foi lançada a versão 3, com melhorias no executor de consultas e algumas partes do código foram re-escritas.
- As versões sub-sequentes, até o Postgres95, foram focadas em confiabilidade e portabilidade.



Breve histórico

- O Postgres95 teve mudanças radicais em relação ao projeto original. O seu código foi totalmente revisado, o tamanho dos fontes foi reduzido em 25%. A performance foi consideravelmente melhorada e vários recursos foram adicionados.
- Em 1996 o nome Postgres95 tornou-se inadequado, o projeto foi rebatizado "PostgreSQL", para enfatizar a relação do POSTGRES original com a linguagem SQL.
- A numeração da versão voltou a seguir o padrão anterior ao Postgres95 (considerada a 5.0), e a primeira versão do PostgreSQL foi a 6.0.
- Enquanto a ênfase do Postgres95 tinha sido a correção de falhas e otimização do código, o desenvolvimento das primeiras versões do PostgreSQL foi orientada à melhoria de recursos e implementação de novos recursos, sempre seguindo os padrões de SQL anteriormente estabelecidos.



Breve histórico

- A versão 8.0 foi lançada em janeiro de 2005 e entre outras novidades, foi a primeira a ter suporte nativo para Microsoft Windows (anteriormente o PostgreSQL só rodava de forma nativa em sistemas Unix e, em sistemas Windows - através da biblioteca Cygwin).
- Dentre as muitas novidades da versão 8.x, pode-se destacar o suporte a tablespaces, savepoints, roles e commit em duas fases.
- Em 20 de setembro de 2010 foi lançada a versão (estável) mais recente: 9.0.



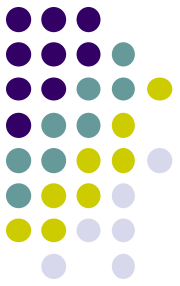
O PostgreSQL hoje

- A equipe do projeto cresceu e se espalhou pelo mundo.
- O **Grupo Global de Desenvolvimento do PostgreSQL** tem membros nos Estados Unidos, Canadá, Japão, Rússia e vários outros países.
- Esse grupo é formado essencialmente por empresas especializadas em PostgreSQL, empresas usuárias do sistema, além dos pesquisadores acadêmicos e programadores independentes.
- Além da programação, essa comunidade é responsável pela documentação, tradução, criação de ferramentas de modelagem e gerenciamento, e elaboração de extensões e acessórios.



O PostgreSQL hoje

- Pela riqueza de recursos e conformidade com os padrões, ele é um SGBD muito adequado para o estudo universitário do modelo relacional, além de ser uma ótima opção para empresas implementarem soluções de alta confiabilidade sem altos custos de licenciamento.
- É um programa distribuído sob a licença **BSD (Berkeley Software Distribution)**, o que torna o seu código fonte disponível e o seu uso livre para aplicações comerciais ou não.
- O PostgreSQL foi implementado em diversos ambientes de produção no mundo, entre eles, um bom exemplo do seu potencial é o banco de dados que armazena os registros de domínio **.org** e **.info**, mantidos pela empresa **Afilias**, sediada em Dublin, Irlanda.



Plataformas suportadas pelo PostgreSQL

- Unix
- Solaris, AIX, HP-UX, etc.
- Linux
- FreeBSD
- Windows
 - 2000, XP, Vista, Seven, etc.
- MacOS X



Recursos recentes

- **Sub-consultas**

- Uma subconsulta é uma instrução SELECT aninhada dentro de uma instrução SELECT, INSERT, DELETE ou UPDATE ou dentro de uma outra subconsulta.

EXEMPLO:

```
SELECT NRPEDIDO FROM PEDIDO WHERE CODCLIENTE =  
(SELECT CODCLIENTE FROM CLIENTE WHERE NOMECLIENTE = 'ALFA');
```



Recursos recentes

- **Controle de concorrência multi-versão (MVCC)**
 - Ao contrário de outros SGBDs que utilizam LOCKs para controle de concorrência, o PostgreSQL mantém a consistência dos dados usando um modelo multiversão. Neste modelo, cada transação terá sua versão do banco de dados, estando protegidas de acessar dados inconsistentes que poderiam ser gerados por outras transações. Portanto, o MVCC oferece o isolamento de transações, além de garantir que leituras nunca aguardarão escritas e vice-versa.)



Recursos recentes

- **Integridade Referencial**

- A integridade referencial garante a não corrupção dos dados, de modo a não existir um registro "filho" sem um registro "pai".



Recursos recentes

- **Funções armazenadas (Stored Procedures)**
 - As funções podem ser escritas em várias linguagens (PL/PgSQL, Perl, Python, Ruby, e outras)



Recursos recentes

- **Gatilhos (Triggers)**

- Trigger é um recurso de programação executado sempre que o evento associado ocorrer.
- É muito utilizado para ajudar a manter a consistência dos dados ou para propagar alterações em um determinado dado de uma tabela para outras.



Recursos recentes

- **Esquemas (Schemas)**

- Schema é um espaço lógico (namespace) dentro do banco de dados para armazenar objetos: tabelas, funções, etc.
- Este conceito é semelhante ao cross-database, a diferença é que o cross-database relaciona objetos de banco de dados distintos, já o Schema relaciona objetos que estão no mesmo banco de dados, mas em estruturas lógicas (namespaces) distintas.



Recursos recentes

- **Áreas de armazenamento (Tablespaces)**
 - Tablespace designa uma sub-divisão lógica de um banco de dados utilizada para agrupar estruturas lógicas relacionadas. As tablespaces apenas especificam a localização de armazenamento do banco de dados. Os dados são armazenadas fisicamente em datafiles.



Recursos recentes

- **Pontos de salvamento (Savepoints)**
 - Savepoints são pontos dentro de uma transação que indicam que os comandos posteriores podem sofrer rollback, enquanto os comandos anteriores são mantidos no banco de dados mesmo que a transação tenha sido abortada. Uma transação pode ter mais de um Savepoint que é indicado por um nome fornecido pelo programador.

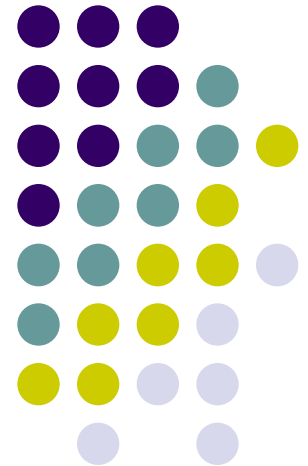


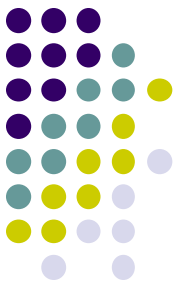
Recursos recentes

- **Commit em duas fases**

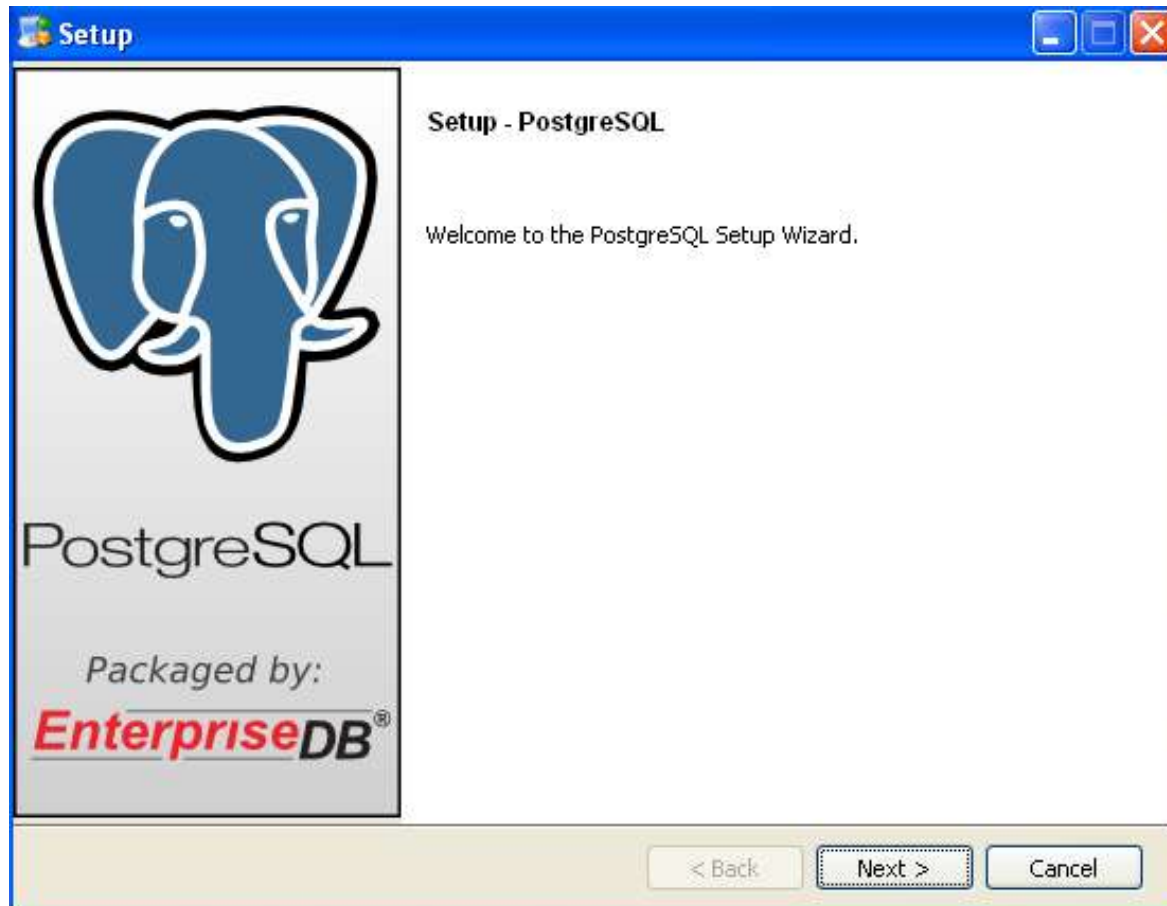
- Commit em duas fases refere-se a uma transação que pode utilizar dois ou mais bancos de dados (multi-database), que podem estar localizados em servidores diferentes.
- Durante uma transação em bancos com essa característica garante-se que o commit seja realizado em todos os bancos participantes ou em nenhum, ou seja, ou grava tudo ou não grava nada.
- Por exemplo, se sua aplicação atualiza dados em dois bancos de dados e você faz um commit, o recurso de commit em duas fases previne situações como a de um dos bancos ficar indisponível e suas mudanças serem atualizadas somente em um dos bancos envolvidos.

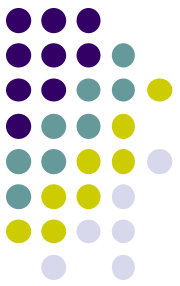
PostgreSQL



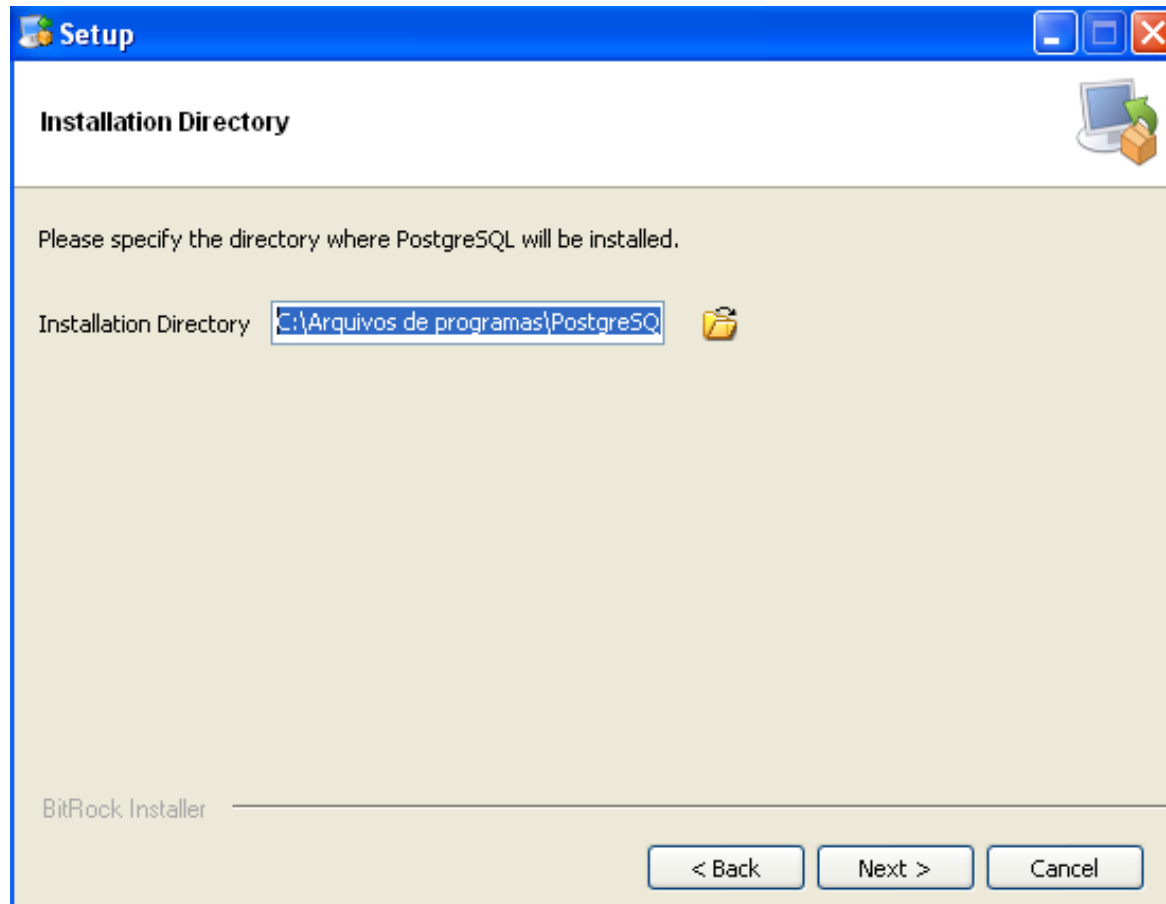


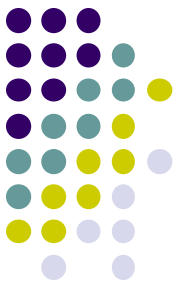
PostgreSQL



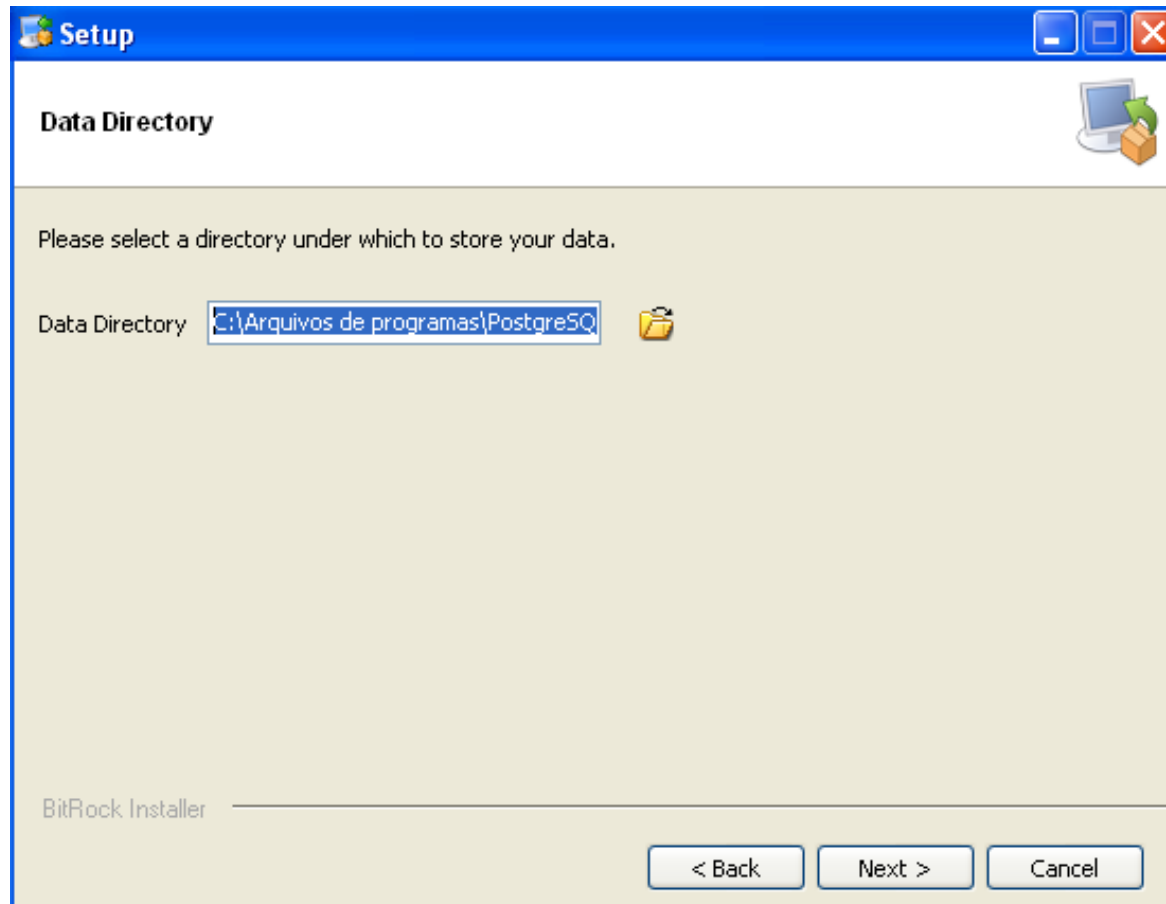


PostgreSQL





PostgreSQL



PostgreSQL



Setup

Password

Please provide a password for the database superuser (postgres) and service account (postgres). If the service account already exists in Windows, you must enter the current password for the account. If the account does not exist, it will be created when you click 'Next'.

Password

Retype password

BitRock Installer

< Back Next > Cancel



PostgreSQL

The screenshot shows a Windows-style setup window titled "Setup" with a blue title bar. The window has standard minimize, maximize, and close buttons in the top right corner. The main content area has a light beige background. At the top left of the content area, the word "Port" is displayed in bold. To its right is a small icon of a computer monitor with a green arrow pointing to it. Below this, the text "Please select the port number the server should listen on." is centered. A text input field is labeled "Port" and contains the number "5432". At the bottom left, the text "BitRock Installer" is visible. At the bottom right, there are three buttons: "< Back", "Next >", and "Cancel".

Setup

Port

Please select the port number the server should listen on.

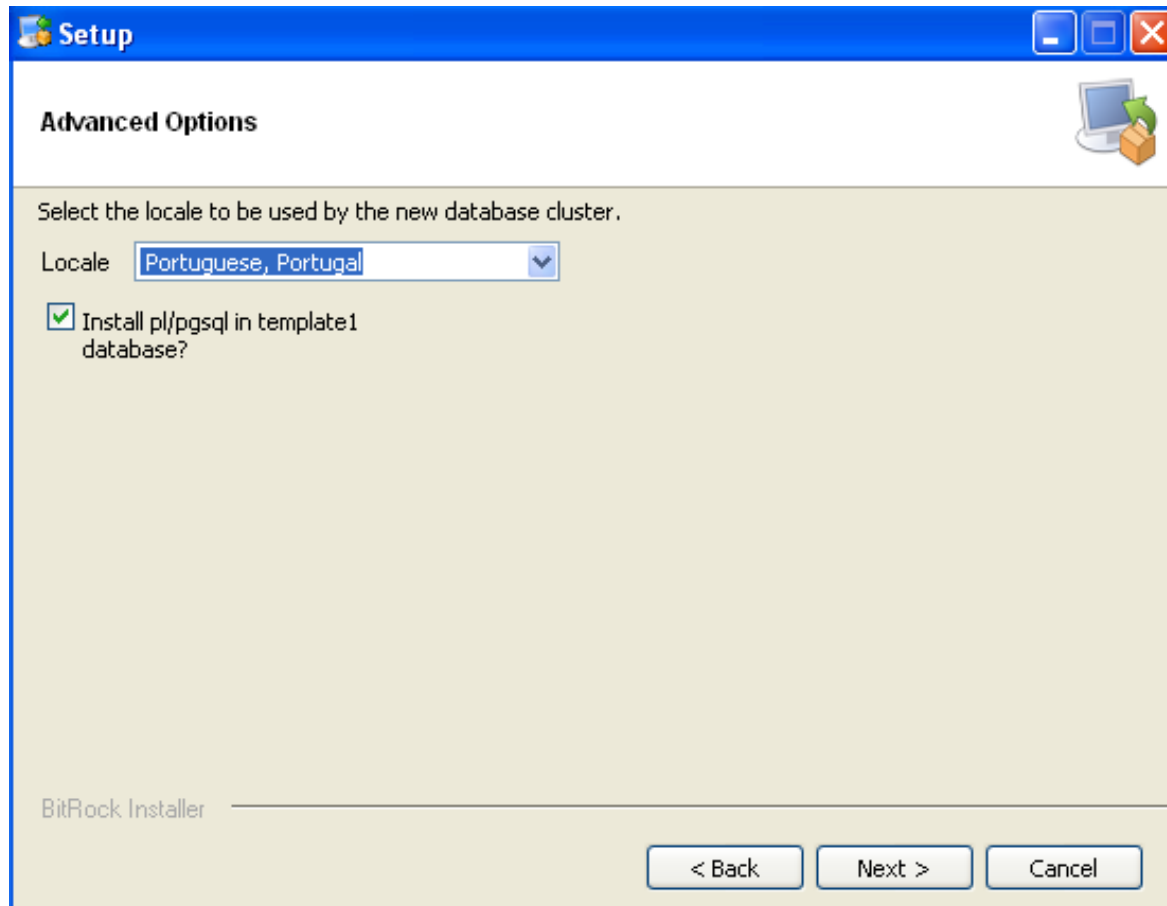
Port

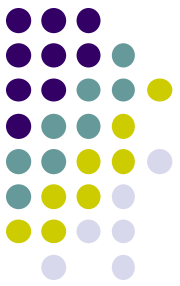
BitRock Installer

< Back Next > Cancel

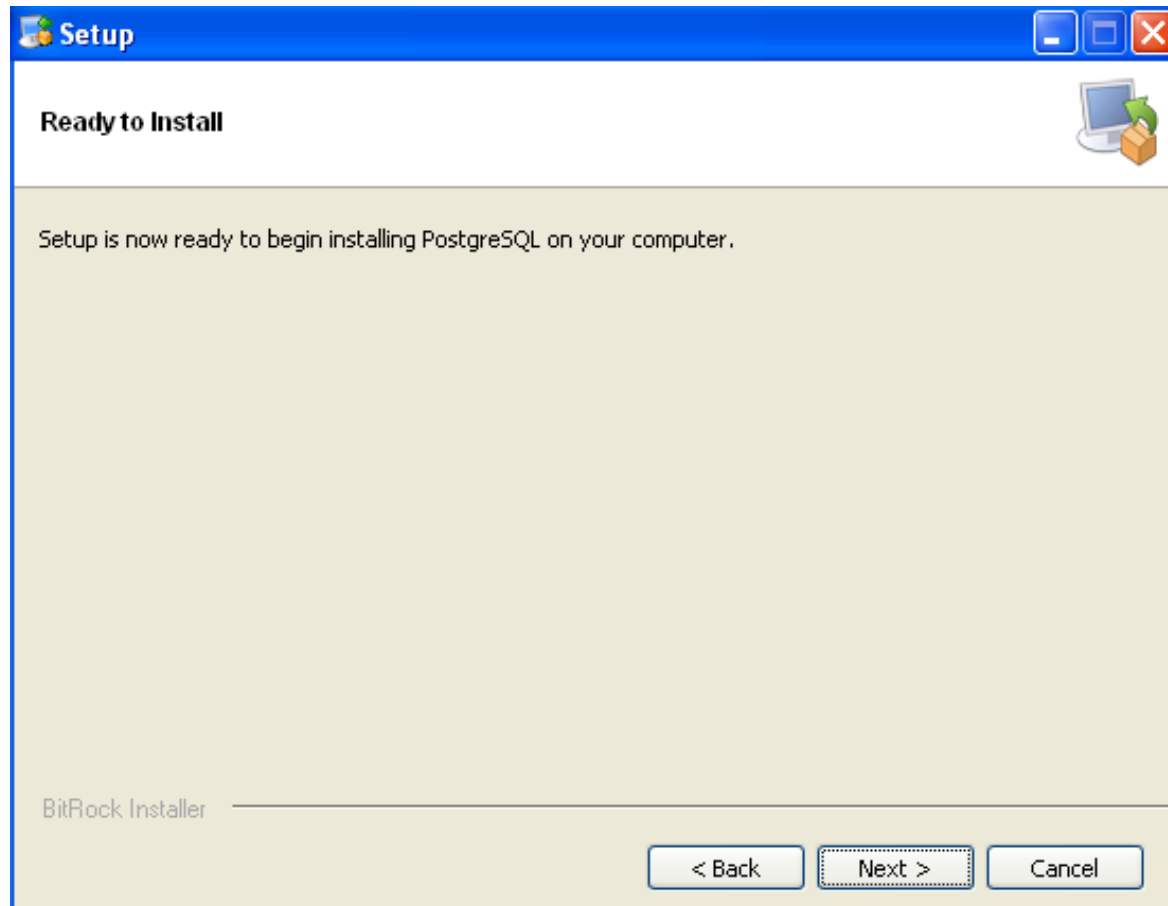


PostgreSQL



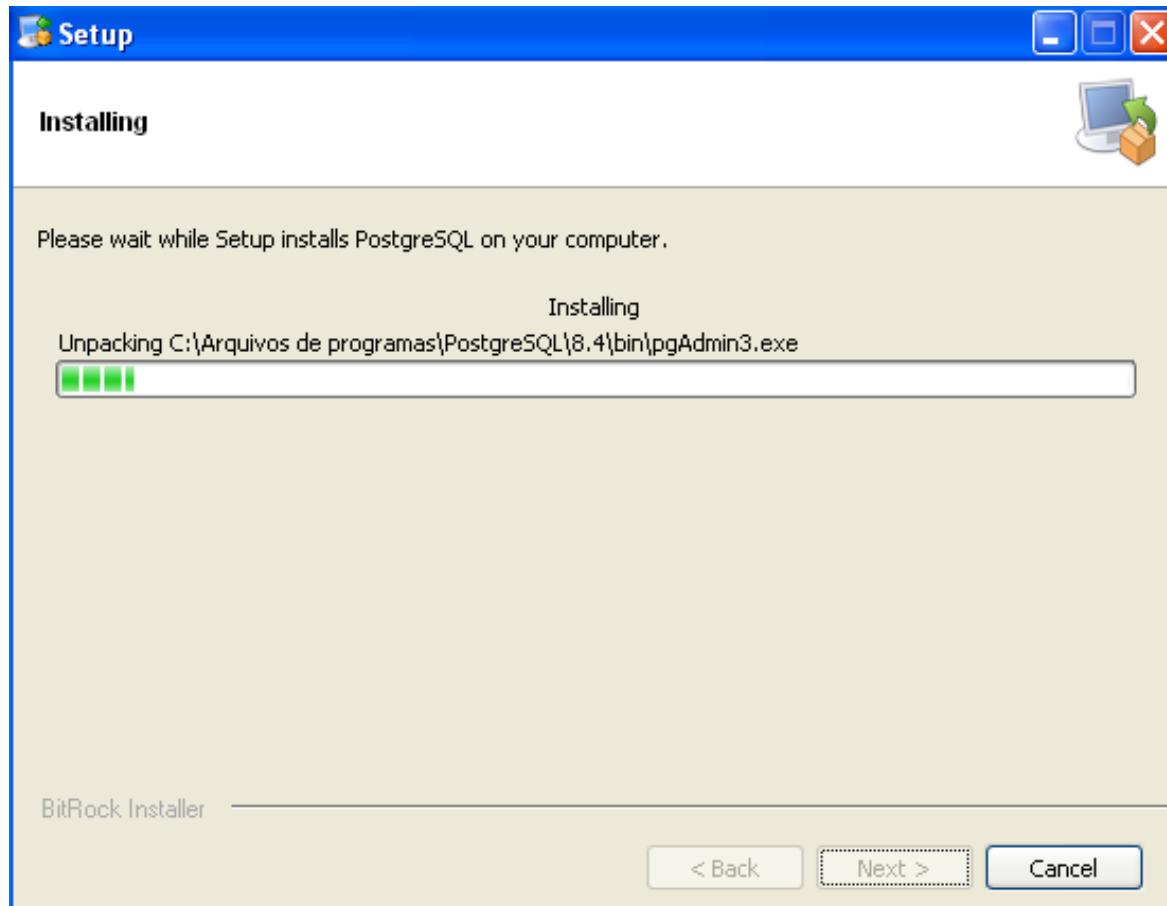


PostgreSQL

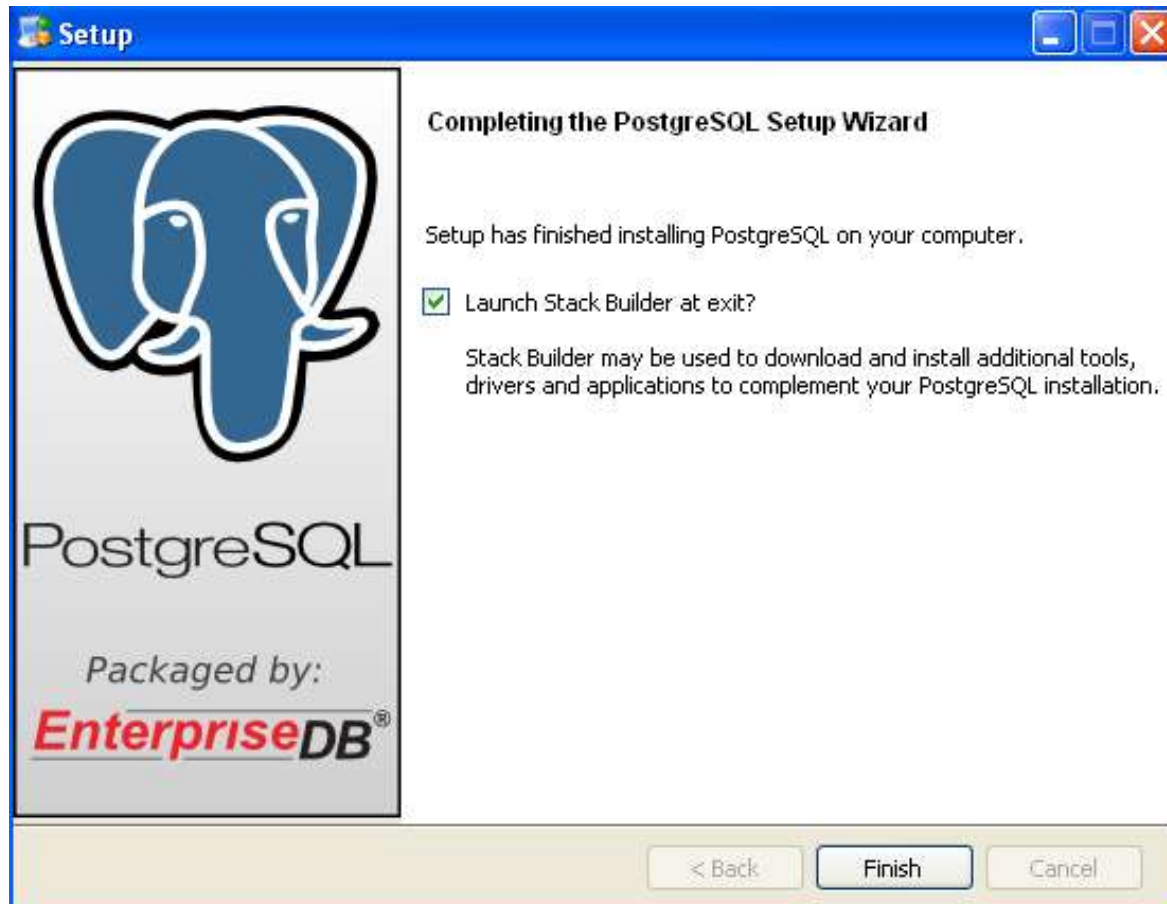


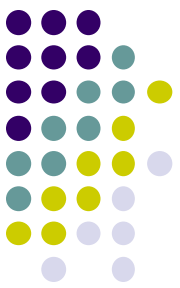


PostgreSQL

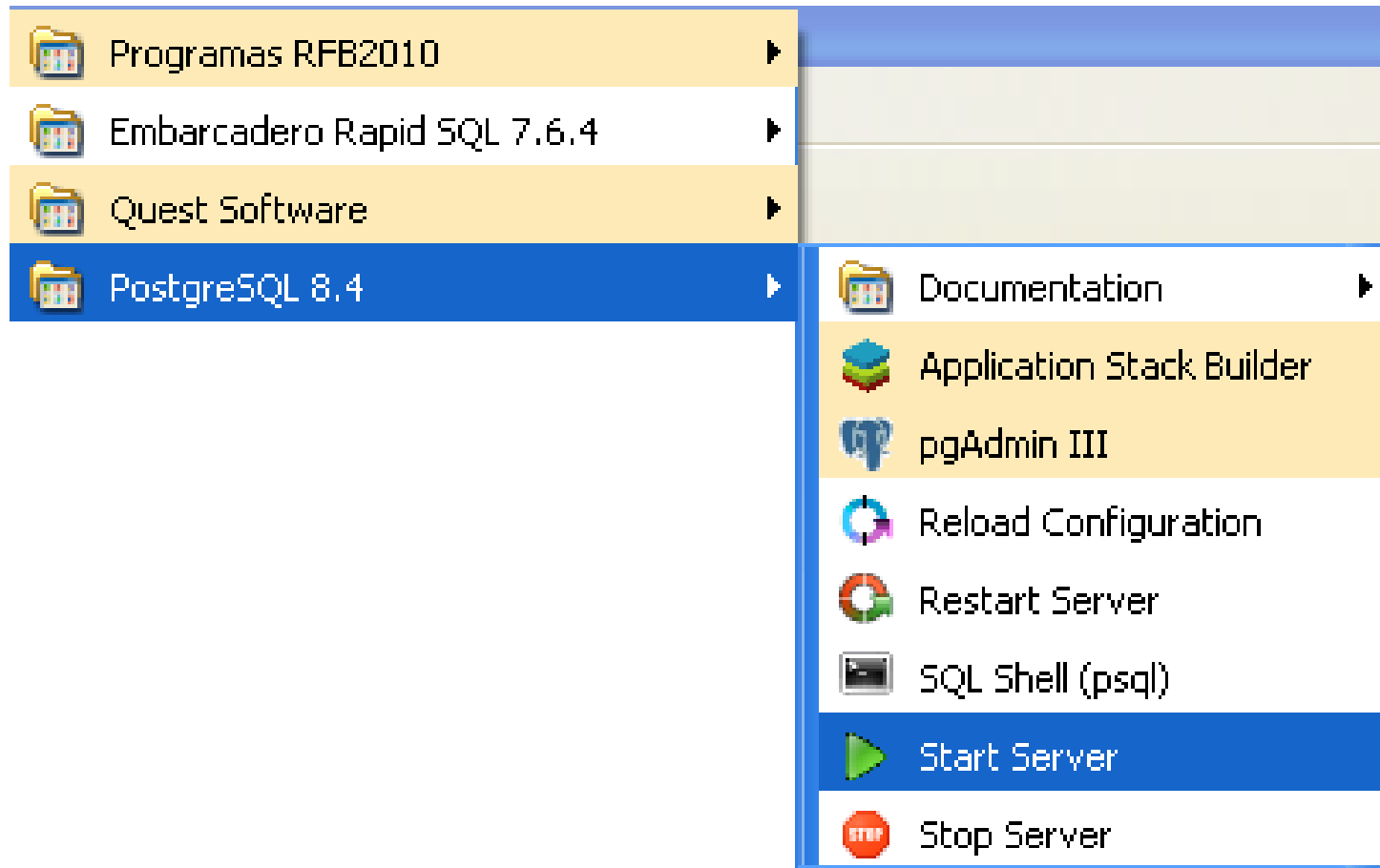


PostgreSQL





PostgreSQL





PostgreSQL

A screenshot of a terminal window titled "SQL Shell (psql)". The window has a blue title bar with standard window controls (minimize, maximize, close). The terminal content shows the following prompts and input:

```
Server [localhost]:  
Database [postgres]:  
Port [5432]:  
Username [postgres]:  
Senha para usuário postgres: _
```

The cursor is positioned at the end of the password prompt. The terminal background is black, and the text is white.

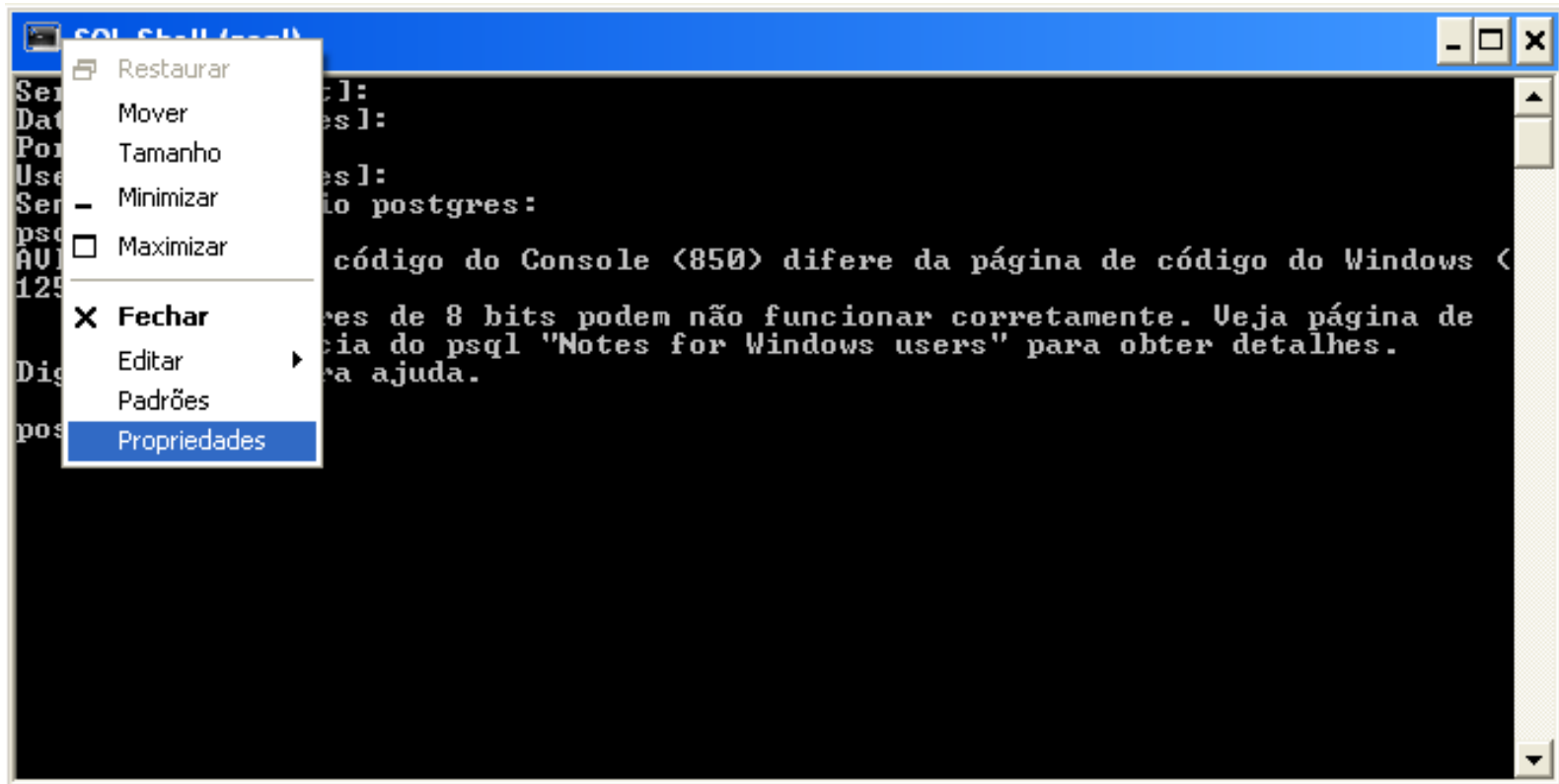


PostgreSQL

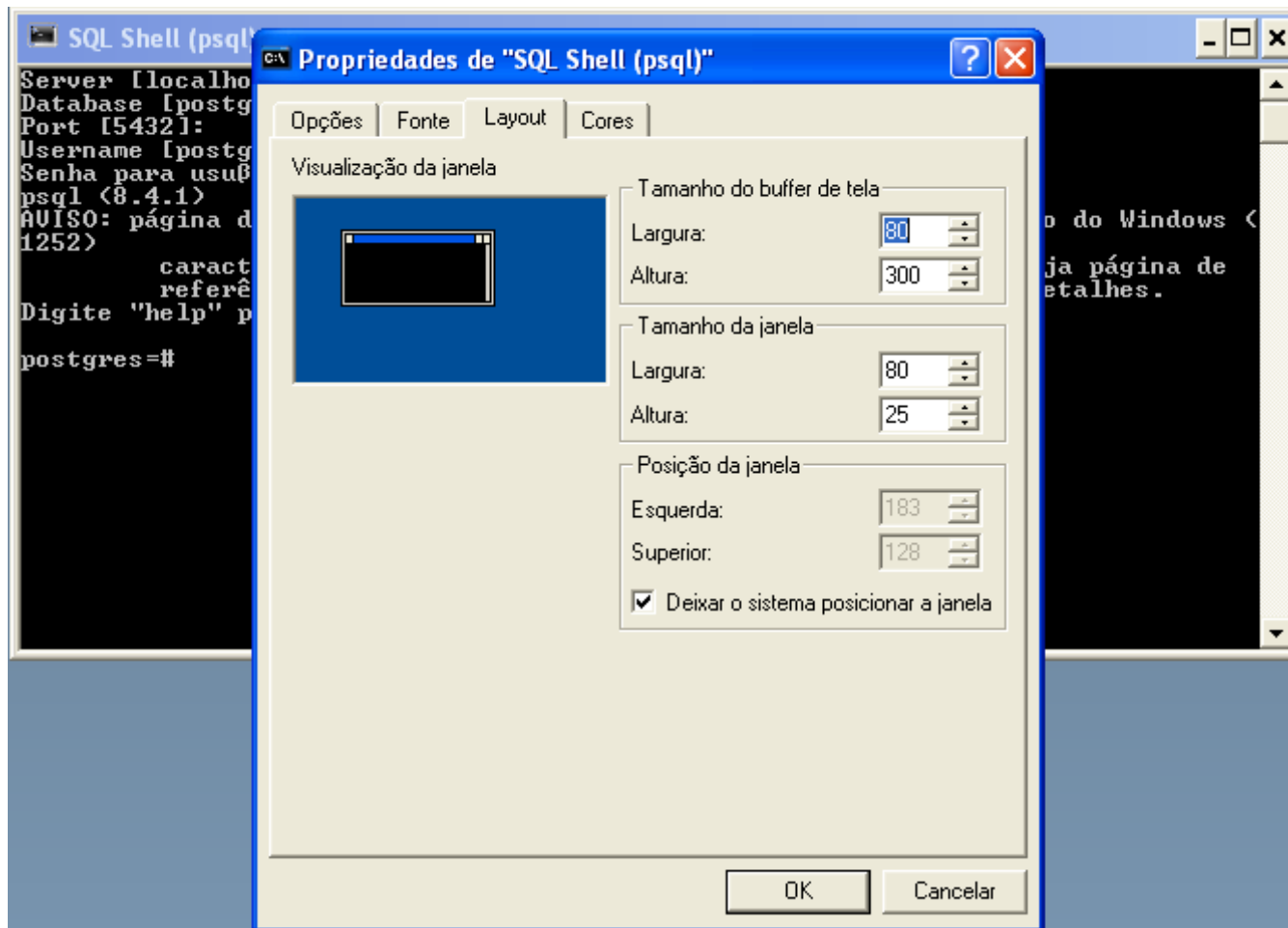
A screenshot of a Windows command prompt window titled "SQL Shell (psql)". The window has a blue title bar with standard minimize, maximize, and close buttons. The main area is black with white text. The text shows the process of connecting to a PostgreSQL database. It starts with prompts for Server, Database, Port, and Username, all of which are filled in with their default values. Then, it asks for the password for the 'postgres' user. After the password is entered, it shows the psql version (8.4.1) and a warning about the console code page (850) differing from the Windows code page (1252), suggesting that 8-bit characters might not work correctly and pointing to the "Notes for Windows users" page for more details. Finally, it prompts the user to type "help" for assistance, and the prompt changes to "postgres=#".

```
SQL Shell (psql)
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Senha para usuário postgres:
psql (8.4.1)
AVISO: página de código do Console (850) difere da página de código do Windows (
1252)
      caracteres de 8 bits podem não funcionar corretamente. Veja página de
      referência do psql "Notes for Windows users" para obter detalhes.
Digite "help" para ajuda.
postgres=#
```

PostgreSQL



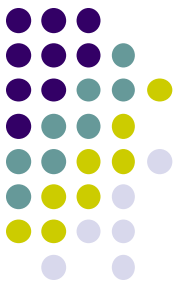
PostgreSQL



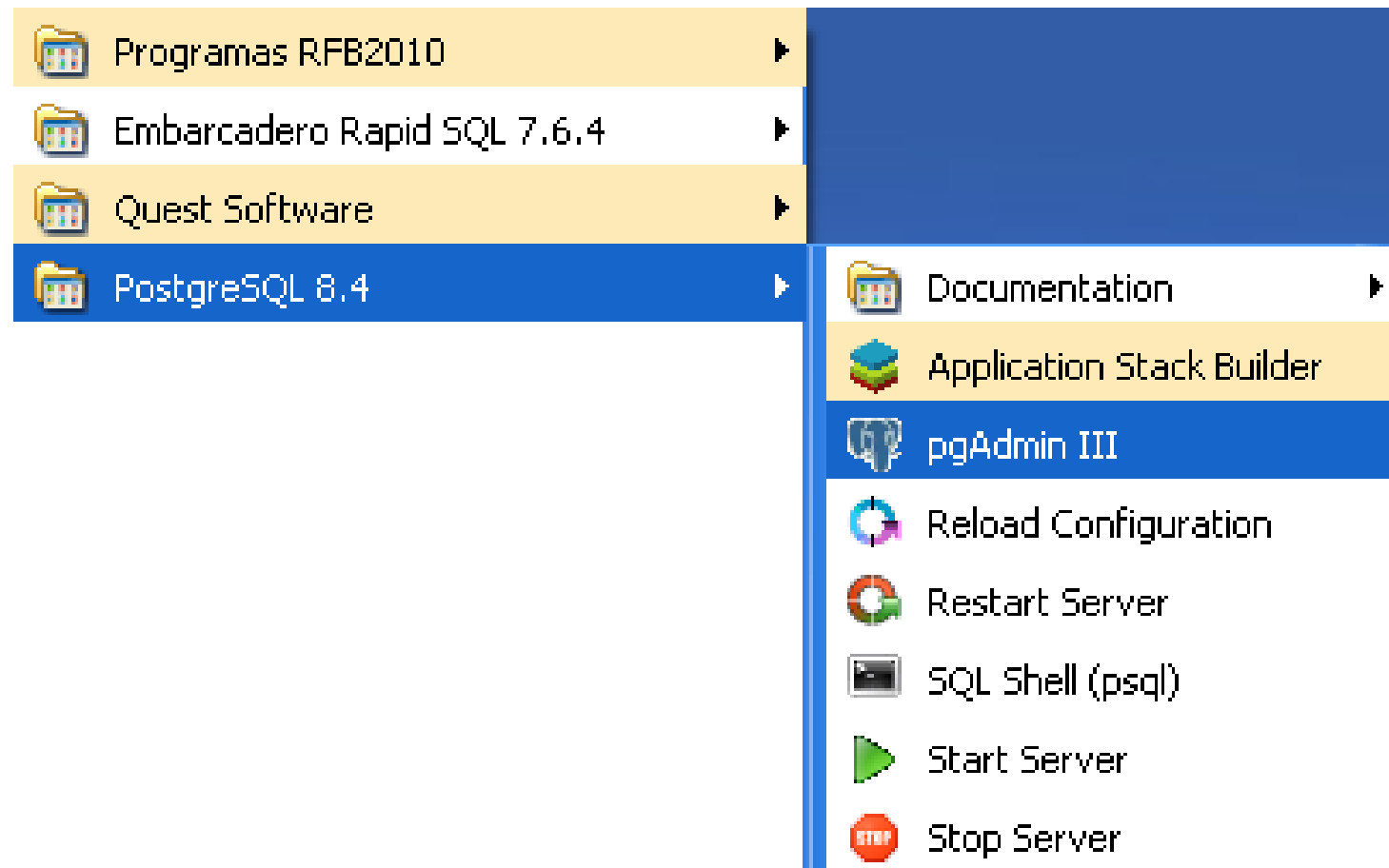


PostgreSQL

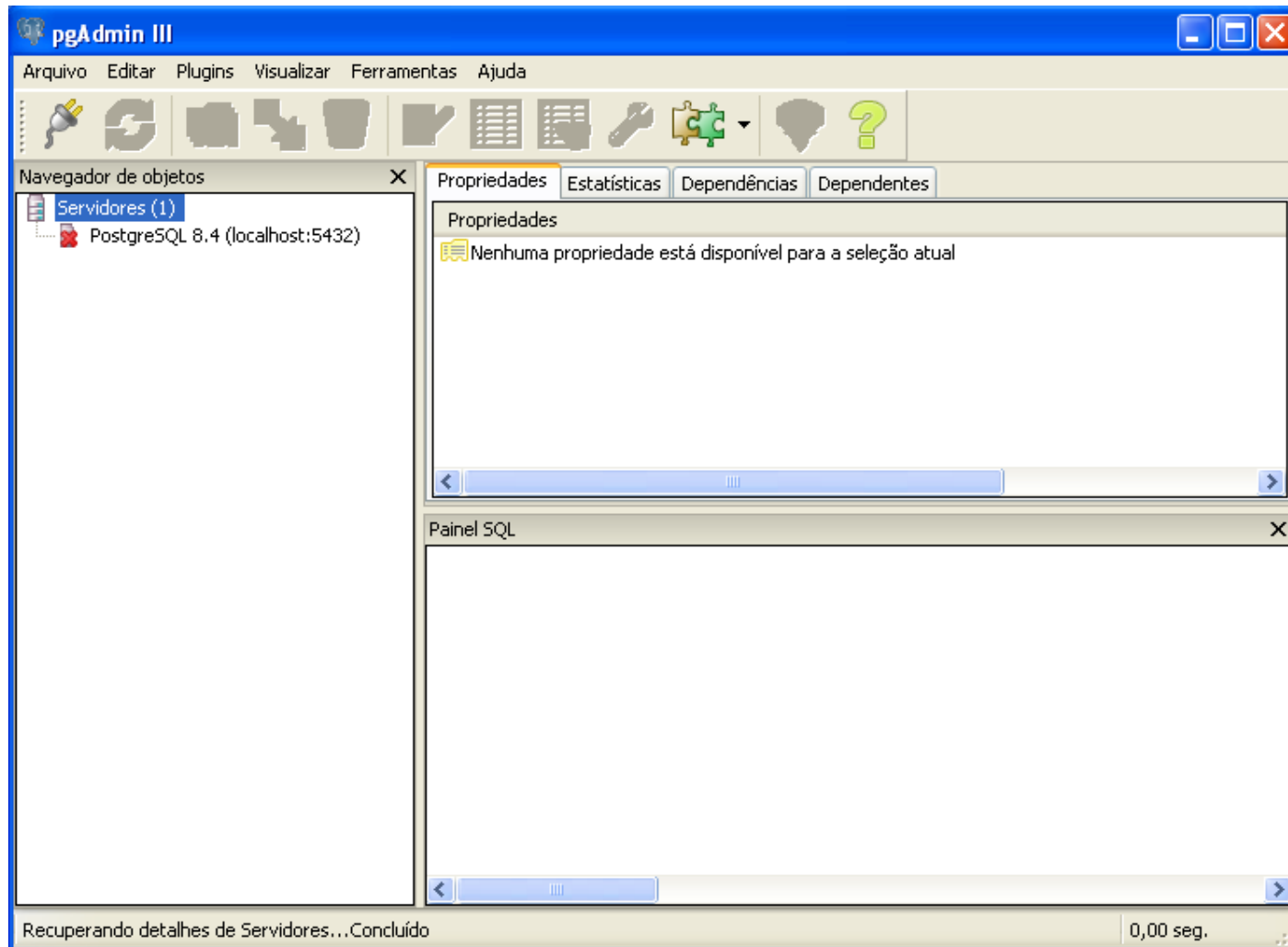
```
SQL Shell (psql)
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Senha para usuário postgres:
psql (8.4.1)
Aviso: página de código do Console (850) difere da página de código do Windows (
1252)
      caracteres de 8 bits podem não funcionar corretamente. Veja página de
referência do psql "Notes for Windows users" para obter detalhes.
Digite "help" para ajuda.
postgres=# _
```



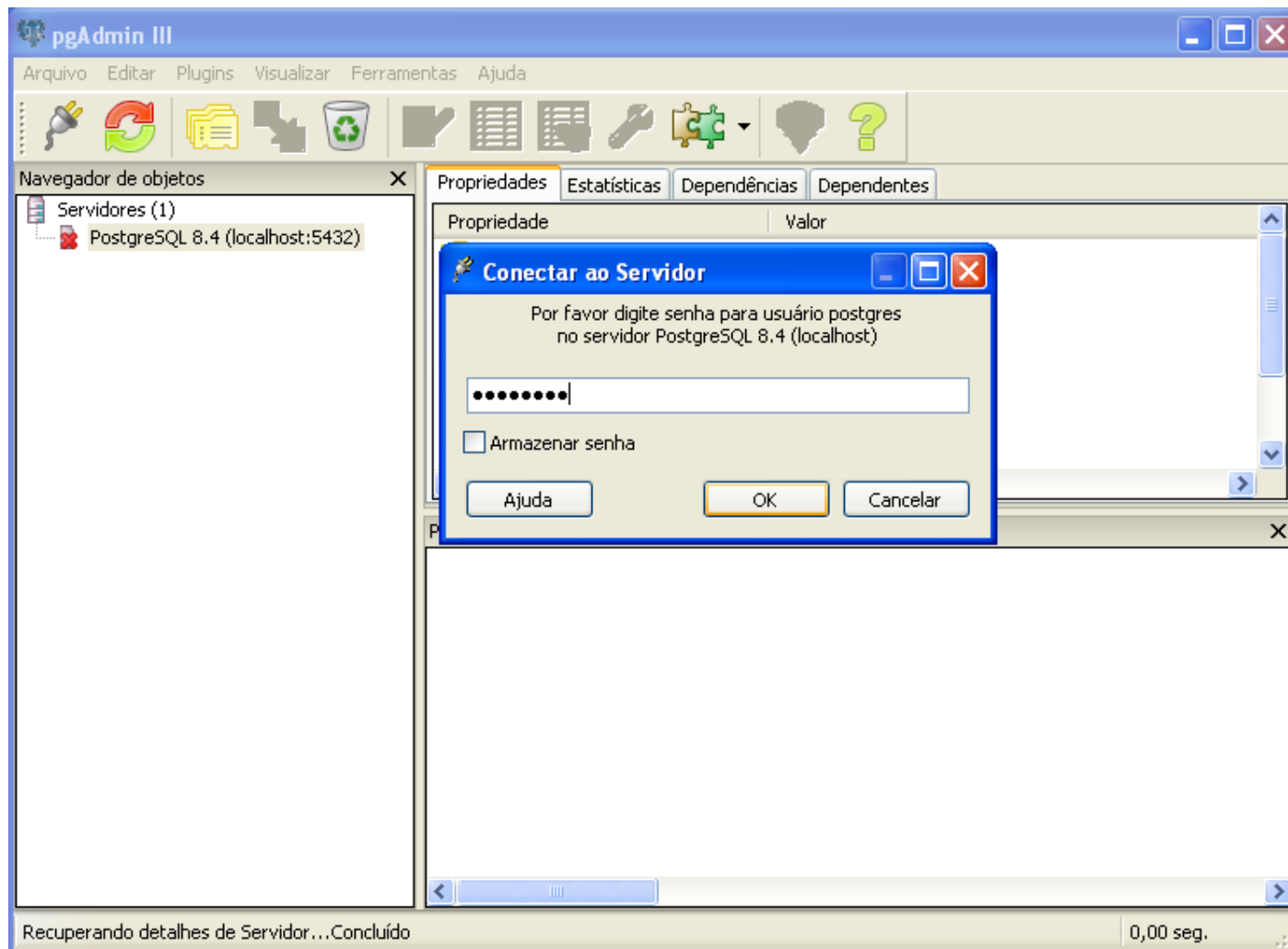
PostgreSQL



PostgreSQL



PostgreSQL



PostgreSQL



pgAdmin III

Arquivo Editar Plugins Visualizar Ferramentas Ajuda

Navegador de objetos

- Servidores (1)
 - PostgreSQL 8.4 (localhost:5432)
 - Bancos de Dados (1)
 - postgres
 - Tablespaces (2)
 - pg_default
 - pg_global
 - Roles do Grupo (0)
 - Login Roles (1)
 - postgres

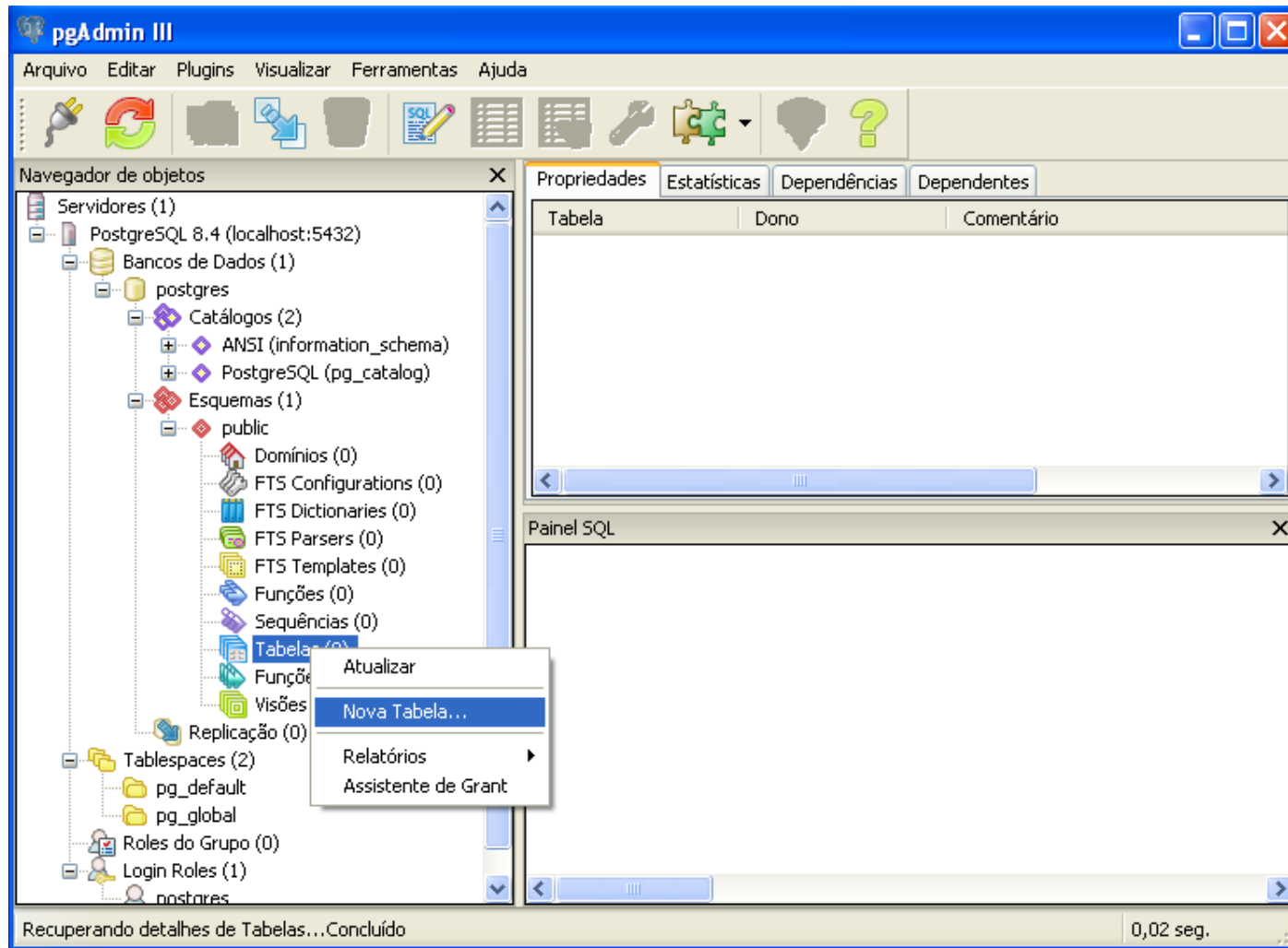
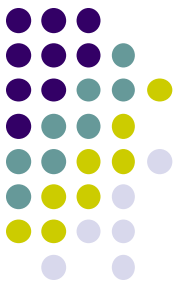
Propriedades Estatísticas Dependências Dependentes

Propriedade	Valor
Descrição	PostgreSQL 8.4
Nome da Máquina	localhost
Porta	5432
Serviço	postgresql-8.4
Manutenção do banco de dados	postgres
Nome de usuário	postgres
Armazenar senha?	Não
Restaurar ambiente?	Não

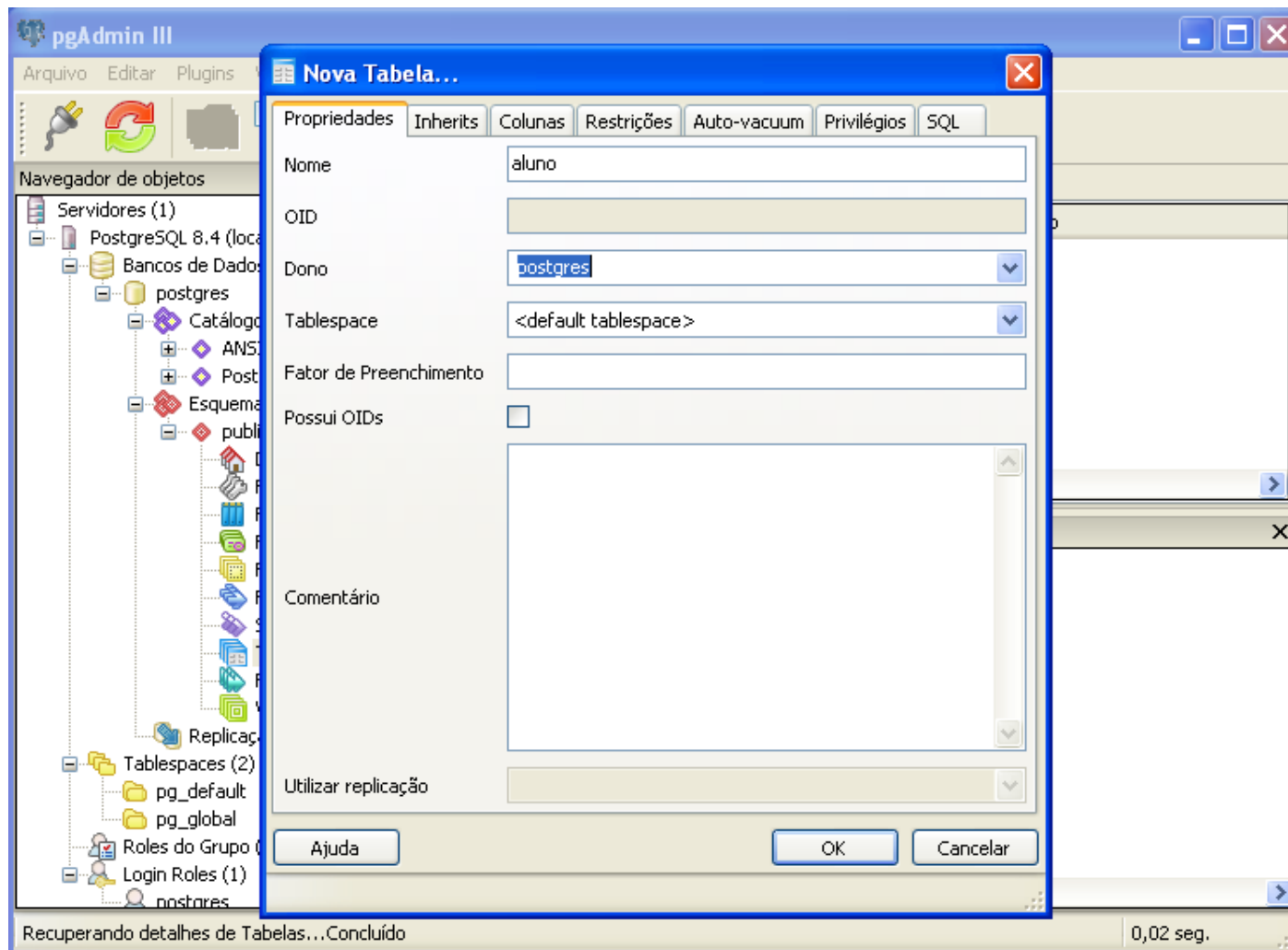
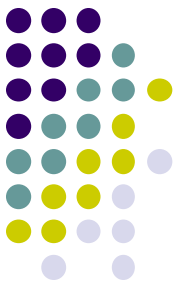
Painel SQL

Estabelecendo conexão...Concluído 0,03 seg.

PostgreSQL

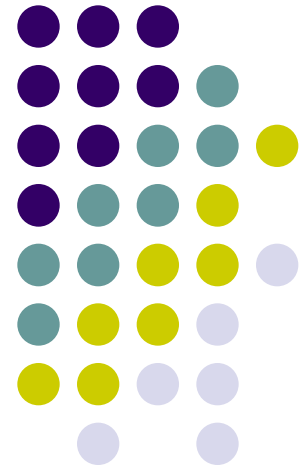


PostgreSQL



Sistemas Gereciadores de Banco de Dados

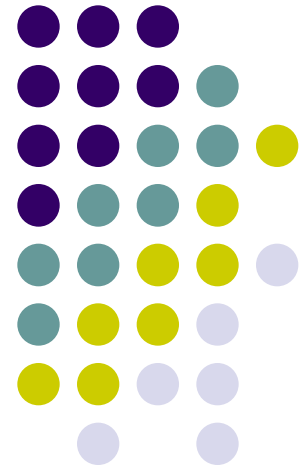
Prof. Marcos Alexandruk
alexandruk@uninove.br
www.unilivros.com.br



Microsoft SQL Server

Versão 2008 R2

Novos Recursos



SQL SERVER 2008 R2



- **Gerenciamento Baseado em Diretivas**

- Permite o gerenciamento eficiente de várias instâncias do SQL Server a partir de um único local.
- As diretivas podem avaliar a conformidade de servidores com um conjunto de condições pré-definidas e evitar que alterações indesejáveis sejam feitas nos servidores.

SQL SERVER 2008 R2



- **Coleta de Dados de Desempenho**

- O Coletor de Dados colhe, armazena e exibe dados de desempenho automaticamente.
- Coleta dados de utilização de disco, atividade de servidor e estatísticas de consulta, carrega em um data warehouse de gerenciamento e os dados de desempenho podem ser analisados no SQL Server Management Studio ou usando ferramentas de terceiros.

SQL SERVER 2008 R2



- **Compactação de Dados**

- A compactação de dados reduz o espaço de armazenamento necessário para tabelas e índices, o que permite um armazenamento de dados mais eficiente.
- A compactação não requer alterações em aplicações para ser habilitada.

SQL SERVER 2008 R2



- **Administrador de Recursos**

- O Administrador de Recursos permite a administradores controlar e alocar recursos de CPU e memória para aplicações de alta prioridade.
- Permite que se mantenha desempenho previsível e ajuda a evitar que este seja afetado negativamente por aplicações ou processos de uso intenso de recursos.

SQL SERVER 2008 R2



- **Criptografia Transparente de Dados**

- Permite que os dados sejam armazenados com segurança através da criptografia dos arquivos do banco de dados.
- O SQL Server executa essa criptografia e decodificação diretamente, assim o processo é inteiramente transparente para as aplicações.
- Cópias de backup de arquivos do banco de dados criptografados também são automaticamente criptografadas.

SQL SERVER 2008 R2



- **Gerenciamento de Chaves Externas**

- Permite que certificados e chaves de criptografia sejam armazenados usando módulos de segurança de hardware de terceiros projetados especificamente para esse fim.
- Armazenar as chaves separadamente dos dados permite uma arquitetura de segurança mais extensível e robusta.

SQL SERVER 2008 R2



- **Auditoria de Dados**

- Oferece uma forma simples de rastrear e registrar em log eventos relacionados aos bancos de dados e servidores.
- Pode-se auditar logons, alterações de senha, acesso e modificação de dados e muitos outros eventos.
- Os resultados de auditorias podem ser salvos em arquivos ou em logs de segurança ou aplicações do Windows para análise ou arquivamento posterior.

SQL SERVER 2008 R2



- **Inclusão a Quente de CPU e Memória**

- A inclusão a quente de CPUs, um recurso disponível na edição **64 bits** do **SQL Server Enterprise**, permite que CPUs sejam acrescentadas dinamicamente quando necessário, sem a necessidade de desligar o servidor ou limitar conexões de clientes.
- A inclusão a quente de memória permite que se acrescente memória da mesma forma.

SQL SERVER 2008 R2



- **Instalação Dinamizada**

- A instalação agora informa sobre problemas de configuração, como pré-requisitos de instalação.
- Componentes individuais do SQL Server, como Database Services, Analysis Services e Integration Services, podem ser selecionados opcionalmente para instalação.
- A configuração de suporte a cluster de failover também foi incluída na instalação.

SQL SERVER 2008 R2



- **Gerenciamento de Grupo de Servidores**

- Permite que consultas T-SQL sejam emitidas contra vários servidores a partir de um único Servidor Central de Gerenciamento, o que simplifica a administração.
- O gerenciamento é centralizado, assim os servidores não precisam ser configurados individualmente.

SQL SERVER 2008 R2



- **Supervisor de Atualização**

- Ajuda a dinamizar o processo de atualização identificando problemas antecipadamente.
- Gera um relatório que destaca quaisquer questões que possam impedir uma atualização.
- Fornece informações detalhadas que podem ser usadas para preparar atualizações.

SQL SERVER 2008 R2



- **Exibições Indexadas Alinhadas pela Partição**

- Exibições Indexadas agora podem ser criadas para seguir o esquema de particionamento da tabela a que se referem.
- Melhoram o desempenho de tabelas particionadas muito grandes, como tabelas de fatos em data warehouses.

SQL SERVER 2008 R2



- **Compactação de Backup**

- A compactação de backup elimina a necessidade de compactar o próprio banco de dados.
- Todos os tipos de backup, inclusive os de log, são suportados e dados são automaticamente descompactados na restauração.

SQL SERVER 2008 R2



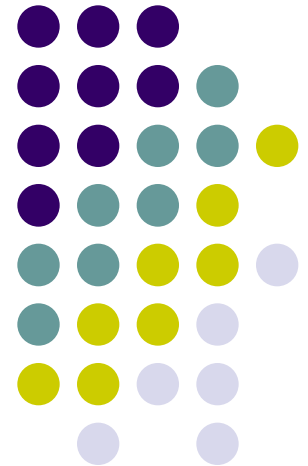
- **Eventos Estendidos**

- Fornece uma ferramenta de solução de problemas que permite tratar problemas de difícil solução com mais eficiência.
- Administradores podem investigar uso excessivo de CPU, deadlocks e tempo limite de aplicações, assim como outras questões.
- Dados de eventos estendidos podem ser correlacionados com dados de eventos do Windows para obter-se um quadro mais completo que auxilie na solução do problema.

Microsoft SQL Server

Versão 2008 R2

Requisitos de hardware e de software



SQL SERVER 2008 R2



- **Framework**

- A Instalação do SQL Server instala os seguintes componentes de software requeridos pelo produto:
 - NET Framework 3.5 SP11
 - SQL Server Native Client
 - Arquivos de suporte à instalação do SQL Server

SQL SERVER 2008 R2



- **Software**

- A Instalação do SQL Server requer o Microsoft Windows Installer 4.5 ou uma versão mais recente.
- Depois de instalar os componentes necessários, a Instalação do SQL Server verifica se o computador em que o SQL Server 2008 R2 será instalado também atende a todos os outros requisitos para uma instalação bem-sucedida.

SQL SERVER 2008 R2



- **Processador**

- Requisitos do Server 2008 R2 **Enterprise (32 bits)**:
 - Processador compatível com Pentium III ou mais rápido
 - Mínimo: 1,0 GHz
 - Recomendável: 2,0 GHz ou mais rápido

Para outras versões consultar a documentação

SQL SERVER 2008 R2



- **Memória**

- Mínimo: 1 GB
- Recomendável: 4 GB ou mais

*O SQL Server Enterprise Edition dá suporte a um máximo de **2 TB** de RAM ou o máximo do sistema operacional.*

SQL SERVER 2008 R2



- **Vídeo**

- As ferramentas gráficas do SQL Server 2008 R2 exigem resolução Super VGA ou superior: resolução mínima de 800 x 600 pixels.

SQL SERVER 2008 R2



● Virtualização

- O SQL Server 2008 R2 tem suporte para ambientes de máquina virtual em execução na função Hyper-V nas edições Windows Server 2008 SP2 Standard, Enterprise e Datacenter.
- A máquina virtual deve executar um sistema operacional com suporte para a edição específica do SQL Server 2008 R2.
- É recomendável que o SQL Server 2008 R2 seja encerrado antes de desligar a máquina virtual.

SQL SERVER 2008 R2



- **Manuais on line**

- Conjunto de documentos para auxiliar na implementação de projetos de gerenciamento de dados e de business intelligence.

- Disponível em:

<http://msdn.microsoft.com/pt-br/library/ms130214.aspx>

SQL SERVER 2008 R2



- **Curso online GRATUITO!**

- A Microsoft oferece o seguinte treinamento online gratuito (em inglês):

SQL Server 2008 R2 Business Intelligence Training Course

- Disponível em:

<http://msdn.microsoft.com/pt-br/SQL10R2BYFBI-TrainingCourse>

*O curso é composto por dezenas de **vídeo-aulas**.*

SQL SERVER 2008 R2



- **Download SQL Server 2008 R2 Express**

- O software está disponível para download em:

<http://www.microsoft.com/express/Database/>

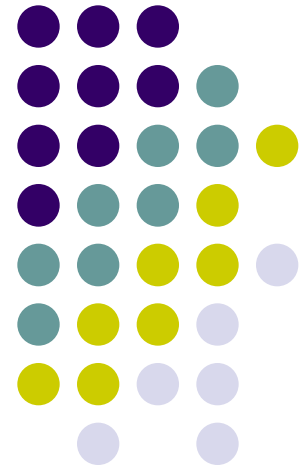
- *Versões:*

32 bits - 235 MB

64 bits - 247 MB

Sistemas Gereciadores de Banco de Dados

Prof. Marcos Alexandruk
alexandruk@uninove.br
www.unilivros.com.br



Aula

Oracle Database 11g

AULA PRÁTICA NO LABORATÓRIO

