

Delphi

SQL

Jocélio Passos
joceliodpassos@bol.com.br

SGBD e SQL

⌘ Machado, Felipe Nery Rodrigues

⌘ **Projeto de Banco de Dados :**

⌘ **uma visão prática**

⌘ Editora Érica , 1996, Capítulo 14

⌘ C:\Arquivos de programas\

⌘ Arquivos comuns\

⌘ Borland Shared\BDE\ **Localsql.hlp**

2/2/2007

2

SGBD

⌘ Sistema Gerenciador de Banco de Dados

⌘ **Coleção de dados relacionados**

⌘ Associados semanticamente

⌘ Juntamente com um conjunto de programas para acessá-los.

⌘ Objetivo : prover um ambiente que seja conveniente e eficiente para recuperar e armazenar informações.

2/2/2007

3

SGBD

⌘ Pretende Eliminar ou Reduzir :

⌘ **Redundância e inconsistência de Dados**

⌘ Dificuldade no Acesso aos Dados (para usuários autorizados)

⌘ Isolamento dos Dados (COBOL provoca isto)

⌘ Anomalias de Acesso Concorrente

⌘ Problemas de Segurança (Limitar o usuário autorizado)

⌘ Exemplos : Oracle, Informix, SyBase, SQL Server, Watcom, InterBase

⌘ O conjunto de programas, ou linguagens, podem ser utilizados para :

2/2/2007

4

SGBD

⌘ **Descrever**

⌘ **DDL – Data Definition Language**

⌘ Linguagem de Definição dos Dados

⌘ Especifica o Esquema do Banco de Dados

⌘ **CREATE TABLE UNIDADE**

⌘ (**UNNCODG** INTEGER NOT NULL ,

⌘ **UNCDESC** CHAR (20) NOT NULL,

⌘ **PRIMARY KEY** (UNNCODG));

⌘ **CREATE UNIQUE INDEX SIUNCDESC ON UNIDADE (UNCDESC);**

⌘ **ALTER TABLE UNIDADE ADD UNCTIPO** CHAR (1);

2/2/2007

5

SGBD

⌘ **Manipular**

⌘ **DML – Data Manipulation Language**

⌘ Linguagem de Manipulação dos Dados

⌘ Inserir, Alterar, Excluir

⌘ **INSERT INTO UNIDADE**

⌘ (UNNCODG, UNCDESC) VALUES (20, 'Caixa');

⌘ **UPDATE UNIDADE SET UNCDESC = 'Caixas'**
where (UNNCODG = 20);

⌘ **DELETE FROM UNIDADE WHERE (UNCDESC IS NULL);**

⌘ Segurança : Grant e Revoke

2/2/2007

6

SGBD

⌘ Interrogar

- ⌘ **QL** – Query Language
- ⌘ Linguagem de Consulta
- ⌘ Resgatar informações (inferência)
- ⌘ A QL é uma porção da DML.

2/2/2007

7

SGBD

⌘ Banco de Dados relacional

- ⌘ Possui arquitetura client/Server (responde a solicitações feitas pelos clientes através da rede)
- ⌘ Possibilita **integridade** dos dados
- ⌘ **Controla** usuários e acessos (fornece acesso seguro aos dados compartilhados)
- ⌘ **Reduz o tráfego** na rede
- ⌘ Facilita **backup e restores**.
- ⌘ Possui **SQL** embutido
- ⌘ Possui **funções administrativas**
- ⌘ Tem a capacidade de **tolerar falhas**

2/2/2007

8

SQL

⌘ **Structured Query Language**

- ⌘ Linguagem de Consulta Estruturada
- ⌘ Linguagem de grande utilização
- ⌘ Fundamentada no modelo relacional de Codd (1970)
- ⌘ A primeira versão recebeu o nome de **SEQUEL** ("Structured English Query Language")
- ⌘ Definida por D.D. Chamberlin
- ⌘ Em 1974
- ⌘ Nos laboratórios de pesquisa da IBM (Califórnia).

2/2/2007

9

SQL

- ⌘ Em 1975, foi implementado um protótipo de aplicação dessa nova linguagem.
- ⌘ Entre 1976 e 1977, o SEQUEL foi revisado e ampliado, e teve seu nome alterado para "**SQL**" por razões jurídicas.
- ⌘ Devido ao sucesso dessa nova forma de consulta e manipulação de dados, **dentro de um ambiente de banco de dados**, a utilização da SQL foi se tornando cada vez maior.
- ⌘ Tornou-se a **linguagem básica** de uma grande quantidade de SGBD's – SQL/DS e DB2 da IBM, Oracle, RDB da Digital, Sybase e Microsoft SQL Server

2/2/2007

10

SQL

- ⌘ Em **1982**, o American National Standard Institute (**ANSI**) tornou a **SQL padrão oficial** de linguagem em ambiente relacional.

2/2/2007

11

SQL - Utilizações

- ⌘ Linguagem interativa de consulta (query **adhoc**). Por meio de comandos SQL, os usuários podem montar **consultas poderosas** sem a necessidade da criação de um programa, podendo utilizar Forms ou ferramentas de montagem de relatórios;
- ⌘ **Linguagem de programação** para acesso a banco de dados – Comandos SQL embutidos em programas de aplicação que acessam os dados armazenados;

2/2/2007

12

SQL - Utilizações

- ⌘ Linguagem de **Administração de banco de dados** – O responsável pela administração do banco de dados (**DBA**- Data Base Administrator) pode utilizar **comandos SQL** para realizar suas tarefas;
- ⌘ Linguagem client/server – Os programas (cliente) dos computadores pessoais usam **comandos SQL** para se **comunicar** por meio de uma rede local, compartilhando os dados armazenados em um único local (Servidor). A arquitetura client/server **minimiza o tráfego** de dados pela rede;

2/2/2007

13

SQL - Utilizações

- ⌘ Linguagem para **banco de dados distribuído** – A SQL auxilia na **distribuição dos dados** por meio de vários nós conectados ao sistema de computação. Auxilia também na **comunicação de dados com outros sistemas**;
- ⌘ Caminho de acesso a outros bancos de dados em diferentes máquinas – A SQL auxilia na **conversão entre diferentes produtos** de banco de dados colocados em diferentes máquinas (de micro até **mainframe**).

2/2/2007

14

Banco de Dados para o Exercício

⌘ CLIENTE

⌘ CLNCODG*	N	_____
⌘ CLCDESC	A	40
⌘ CLCENDE	A	40
⌘ CDCCIDA	A	25
⌘ CLCUF	A	02
⌘ CLCCEP	A	08
⌘ CLCCGC	A	14
⌘ CLCIE	A	04

2/2/2007

15

Banco de Dados para o Exercício

⌘ VENDEDOR

⌘ VDNCODG*	N	_____
⌘ VDCDESC	A	40
⌘ VDYSALR	\$	40
⌘ VDCCOMI	A	01

⌘ PRODUTO

⌘ PDNCODG*	N	_____
⌘ PDCDESC	A	30
⌘ PDCUN	A	20
⌘ PDYVALR	\$	01

2/2/2007

16

Banco de Dados para o Exercício

⌘ PEDIDO

⌘ PENCODG*	N	_____
⌘ PENPRAZ	N	
⌘ PENCGL <input type="checkbox"/>	N	
⌘ PENCVD <input type="checkbox"/>	N	

⌘ ITEMPED

⌘ IPNCGPE_* <input type="checkbox"/>	N	_____
⌘ IPNCGPD_* <input type="checkbox"/>	N	
⌘ IPNQTD	N	

2/2/2007

17

Usando o SQL Explorer

- ⌘ Iniciar
- ⌘ Programas
- ⌘ Borland Delphi X
- ⌘ SQL Explorer



- ⌘ Crie um novo alias e aponte para o diretório contendo as tabelas do exercício

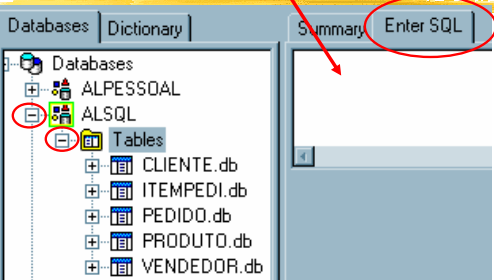
- ⌘ Clique no + e em **Enter SQL**

2/2/2007

18

Usando o SQL Explorer

☞ Coloque o Comando SQL



2/2/2007 19

Comandos SQL

☞ Extrair Dados de uma Tabela

☞ Comando **SELECT**

☞ **SELECT** <ALIAS>.<COLUNA1>,
<ALIAS>.<COLUNAN> **AS** <NOME
ALTERNATIVO> **FROM** <TABELA> **AS**
<ALIAS>

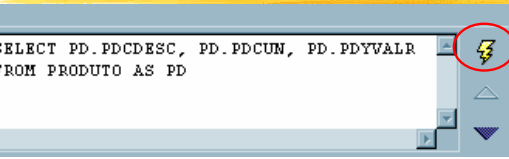
☞ **A) Selecionando colunas específicas da tabela (PROJEÇÃO)**

☞ 1) Listar todos os produtos com respectivas descrições, unidades e valores unitários.

2/2/2007 20

Comandos SQL

☞ Executando



2/2/2007 21

Comandos SQL

☞ 2) Listar da tabela CLIENTE, o CGC, o nome do cliente e seu endereço.

☞ **B) Selecionando todas as colunas da tabela**

☞ **SELECT** <ALIAS>.* **FROM** <TABELA> **AS** <ALIAS>

☞ 3) Listar todo o conteúdo da tabela vendedor.

2/2/2007 22

Comandos SQL

☞ **C) Manipulando dados numéricos.**

☞ Aplicados sobre colunas numéricas :

☞ int, smallint, float, real, money

☞ Operadores Aritméticos

- ☞ + (Adição)
- ☞ - (Subtração)
- ☞ / (Divisão)
- ☞ * (Multiplicação)
- ☞ % (Módulo)

☞ 4) Mostre o nome de cada vendedor com o seu salário dobrado. Esta coluna deve aparecer com o nome de AUMENTO.

2/2/2007 23

Comandos SQL

☞ **D) Selecionando somente algumas linhas da tabela (Tuplas)**

☞ **SELECT** <ALIAS>.<COLUNA1>,
<ALIAS>.<COLUNAN> **AS** <NOME
ALTERNATIVO> **FROM** <TABELA> **AS**
<ALIAS>

☞ **WHERE** (<CONDICAO 1>)

☞ <OPERADOR LÓGICO> (<CONDICAO N>)

2/2/2007 24

Comandos SQL

⌘ Cada condição (<CONDICAO1>) tem a seguinte sintaxe :

⌘ **WHERE** (<ALIAS>.<COLUNA1>

⌘ **<Operador Relacional> VALOR)**

⌘ Obs : quando a coluna for caracter, o **VALOR** deve estar entre aspas "".

⌘ Exemplo : **"Parafuso"**

2/2/2007

25

Comandos SQL

⌘ Operadores Relacionais

⊞ = (Igual)

⊞ <> ou != (diferente)

⊞ < (Menor)

⊞ > (Maior)

⊞ >= (Maior ou igual)

⊞ <= (Menor ou igual)

⊞ != (Não maior)

⊞ !< (Não menor)

2/2/2007

26

Comandos SQL

⌘ 5) Listar o número do pedido, o código do produto e a quantidade da tabela itens do pedido, mas apenas aqueles com quantidade igual a 35.

⌘ 6) Quais os clientes que moram em Niterói ?

2/2/2007

27

Comandos SQL

⌘ E) Operadores Lógicos

⌘ Utilizados para concatenar condições

⊞ AND (e)

⊞ OR (ou)

⊞ NOT (negação)

⌘ 7) Listar os produtos que tenham unidade igual a "Kg" e valor unitário igual a R\$ 2,00.

⌘ 8) Listar os clientes e seus respectivos endereços, que moram em "São Paulo" ou estejam na faixa de CEP entre "30077000" e "30079000".

⌘ 9) Mostre todos os pedidos que não tenham prazo de entrega igual a 15 dias.

2/2/2007

28

Comandos SQL

⌘ F) Operadores BETWEEN e NOT BETWEEN

⌘ **WHERE** (<ALIAS>.<COLUNA> **[NOT] BETWEEN** <VALOR1> AND <VALOR2>)

⌘ 10) Listar o código e a descrição dos produtos que tenham o valor unitário na faixa de R\$ 0,32 até R\$ 2,00

2/2/2007

29

Comandos SQL

⌘ G) Operadores baseados em string de caracteres : LIKE e NOT LIKE

⌘ **WHERE**

⌘ (<ALIAS>.<COLUNA> **[NOT] LIKE** <VALOR1>)

⌘ Utiliza os símbolos :

⊞ % substitui uma palavra

⊞ _ substitui um caracter

2/2/2007

30

Comandos SQL

⌘ Exemplos :

⌘ a) **LIKE "Lápis %"**

⌘ "Lápis Preto", "Lápis cera", "Lápis Borracha"

⌘ b) **LIKE "Broca N_"**

⌘ "Broca N1", "Broca N9", "Broca N3"

⌘ c) **LIKE "%ão"**

⌘ retorna qualquer nome que termine em "ão"

⌘ d) **LIKE "[CM]%"**

⌘ retorna qualquer nome que comece com "C" ou com "M"

⌘ e) **LIKE "[C-M]%"**

⌘ retorna qualquer nome que comece com "C" até "M"

2/2/2007

31

Comandos SQL

⌘ 11) Listar todos os produtos que tenham o seu nome começando por "L".

⌘ 12) Listar os vendedores que não começam com "Jo".

⌘ **H) Operadores baseados em listas :**

⌘ **IN e NOT IN**

⌘ **WHERE**

⌘ (<ALIAS>.<COLUNA> **[NOT] IN** <VALOR>)

⌘ 13) Listar os vendedores que são da faixa de comissão A e B

2/2/2007

32

Comandos SQL

⌘ **I) Operadores baseados em valores desconhecidos**

⌘ **IS NULL e IS NOT NULL**

⌘ **WHERE**

⌘ (<ALIAS>.<COLUNA> **IS [NOT] NULL**

⌘ <VALOR1>)

⌘ 14) Mostrar os clientes que não tenham inscrição estadual.

2/2/2007

33

Comandos SQL

⌘ **J) Ordenando os Dados Selecionados**

⌘ **ORDER BY** <ALIAS>.<COLUNA1>,
<ALIAS>.<COLUNAN> **[ASC|desc]**

⌘ 15) Mostrar em ordem alfabética a lista de valores e seus respectivos salários fixos.

⌘ 16) Listar os nomes, cidades e estados de todos os clientes ordenados por estado e cidade de forma descendente.

⌘ 17) Mostrar a descrição e o valor unitário de todos os produtos que tenham a unidade "Kg" em ordem de valor unitário ascendente.

2/2/2007

34

Comandos SQL

⌘ **K) Utilizando Funções de Agregação sobre conjuntos**

⌘ Máximos e Mínimos : **MAX e MIN**

⌘ 18) Mostrar o menor e o maior salário da tabela vendedor

⌘ Totalizando colunas : **SUM**

⌘ 19) Mostrar a quantidade total pedida para o produto "Arroz Integral" de código 78 na tabela item de pedido.

2/2/2007

35

Comandos SQL

⌘ Calculando Médias : **AVG**

⌘ 20) Qual a média de salários dos vendedores ?

⌘ Contando Registros : **COUNT**

⌘ 21) Quantos vendedores ganham acima de R\$ 2.500,00 de salário ?

2/2/2007

36

Comandos SQL

⌘ L) A Cláusula DISTINCT

⌘ 22) Quais as unidades de produtos, diferentes.

⌘ M) Agrupando informações selecionadas GROUP BY e HAVING

⌘ 23) Listar o número de produtos que cada pedido contém.

⌘ 24) Listar os pedidos que têm mais do que três produtos.

2/2/2007

37

Comandos SQL

⌘ N) Recuperando dados de várias tabelas

⌘ WHERE (<ALIAS>.<CHAVE PRIMÁRIA> = <ALIAS>.<CHAVE ESTRANGEIRA>)

⌘ 25) Quais os pedidos de cada cliente ?

⌘ 26) Quais clientes têm prazo de entrega superior a 15 dias e pertencem aos estados de São Paulo ("SP") ou Rio de Janeiro ("RJ") ?

2/2/2007

38

Exercício

⌘ 1) Listar o código e a descrição dos produtos que tenham o valor unitário fora da faixa de R\$ 0,32 até R\$ 2,00

⌘ 2) Mostrar o novo salário fixo dos vendedores, de faixa de comissão "C", calculado com base no reajuste de 75% acrescido de R\$ 120,00 de bonificação. Ordenar pelo nome do vendedor.

2/2/2007

39

Exercícios Propostos

⌘ 1) "A diferença entre Instância e Esquema é que instância é a coleção de dados armazenada no Banco de Dados em um determinado instante (extensão) e Esquema é o projeto geral do Banco de Dados, ou seja, sua estrutura (intensão)." Verdadeiro ou Falso ?

⌘ 2) "Antes de os SGBD surgirem, utilizávamos sistemas de arquivos que causavam replicação e isolamento dos dados. Estes arquivos eram acessados por programas criados em linguagens como COBOL, PL/1, Pascal etc." Verdadeiro ou Falso ?

2/2/2007

40

Exercícios Propostos

⌘ 3) "Surgiu o Banco de Dados hierárquico (IMS, UNIVAC 1100, CDC 6000, CYBER 70 e 170). Em seguida o Modelo em Rede (DBMS 10, IDS II, DMS II, IMAGE). Estes dois fazem parte da 1ª geração (fim dos anos 60)". Verdadeiro ou Falso ?

⌘ 4) "No fim dos anos 70, surge os Sistemas Relacionais, a 2ª geração. No modelo relacional os dados são representados em forma de tabelas. Este modelo é muito formal pois está fundamentado na Teoria dos Conjuntos (DB/2, UNIFY, PROGRESS, SYBASE, ORACLE, INFORMIX, INGRES, SQR Server)." Verdadeiro ou Falso ?

2/2/2007

41

Exercícios Propostos

⌘ 5) "Os Bancos de dados da 1ª e 2ª geração são classificados como convencionais, pois os dados são bem estruturados, os tipos de dados são simples (inteiros, reais, caracter etc), as transações são simples e curtas e o acesso se dá através de chaves. Temos aplicações como Folha de Pagamento, Controle de Estoque, contas a Pagar etc." Verdadeiro ou Falso ?

2/2/2007

42

Exercícios Propostos

- ⌘ 6) "A 3ª geração chega com os bancos de dados Orientado a Objetos, bem como com as extensões do modelo relacional. O Sistema orientado a objeto tem como características : Herança, Reutilização (que aumenta a produtividade), Aumento da consistência do resultado da análise, Melhor ligação entre usuário e analista, Mais flexível diante de alterações da realidade, Encapsulamento (a implementação não precisa ser conhecida, apenas nome do método e sua interface), Comunicação por mensagens entre os objetos, Polimorfismo (o mesmo método pode ser redefinido em diferentes objetos). Alguns exemplos são : GEMSTONE/OPAL (Acadêmico), OBJECTSTORE, ORION, ONTOS (um dos primeiros), O2 (Europeu e interface DOS), IRIS, JASMINE." Verdadeiro ou Falso ?

2/2/2007

43

Exercícios Propostos

- ⌘ 7) "Existe também o Sistema Relacional-Objeto, utilizados, por exemplo, para gerenciar acervos de vídeo, problemas de análise de séries de tempo no mercado financeiro, banco de dados científicos (Projeto Genoma), sistemas de informações geográficas (armazenar mapas/ manipulação de imagens), armazenamento de dados de projeto CAD. Exemplos : DB2/6000, Ilustra (construído do zero), CA-Ingres, Oadapter (construído para adaptar o relacional para o objeto-relacional), UniSQL, Oracle 8i." Verdadeiro ou Falso ?

2/2/2007

44

Exercícios Propostos

- ⌘ 8) "O Sistema Relacional-Objeto, suporta também novas aplicações como os sistemas de suporte à decisão, que explora as fontes de informação para melhorar a tomada de decisão por administradores. O Data Mining, por exemplo, busca informações relevantes em grandes volumes de dados (descoberta de conhecimentos). O Data Warehousing é um método que une informações espalhadas em diversas fontes de dados de uma corporação, de forma organizada, fornecendo uma visão consistente dessa corporação através das informações atuais e históricas armazenadas. Este banco de dados gerado é Data Warehouse, ou seja, um repositório de informações coletadas em diversas fontes (Banco de dados relacional, hierárquico, Web etc), armazenadas sob um esquema único, armazenados em um só local ou distribuído (Web).

2/2/2007

45

Exercícios Propostos

- ⌘ É um conjunto de dados : baseado em conjuntos, integrados, orientados por assunto, multidimensionais (cubos), consolidados (por depto, ano etc), não voláteis e ligados ao tempo (quando ?). É atualizado periodicamente. Banco de dados como o SQL Server 2000, já suportam tal tecnologia." Verdadeiro ou Falso ?
- ⌘ 9) Por que não realizar a mineração dos dados (Data mining) direto na base transacional ?

2/2/2007

46

Exercícios Propostos

- ⌘ 10) "Os bancos de dados desta 3ª geração são classificados como não convencionais, pois suportam grande volume de dados estruturados, os tipos de dados são complexos (texto, gráficos, imagens, sons), as transações são longas, o caminho de acesso aos dados não são triviais (podemos dá um quadro de um filme e chegar aos dados do diretor daquele filme). Suas aplicações são : CAD, CASE, Cartografia etc." Verdadeiro ou Falso ?

2/2/2007

47

Exercícios Propostos

- ⌘ 11) "O modelo relacional é constituído basicamente de tabelas, cada qual contendo linhas (registros, tuplas) e colunas. A localização dos registros se faz por meio de um campo-chave, ou seja, um campo (ou mais) que assume o papel de chave primária da tabela. É por intermédio dessa chave que se identifica uma, e somente uma, ocorrência do valor contido no campo." Verdadeiro ou Falso ?

2/2/2007

48

Exercícios Propostos

- ⌘ 12) “Uma das razões da popularidade dos sistemas relacionais é a sua facilidade de manipulação e entendimento.” Verdadeiro ou Falso ?
- ⌘ 13) “A linguagem SQL foi desenvolvida especialmente para o ambiente relacional, podendo ser adaptada a qualquer ambiente não relacional.” Verdadeiro ou Falso ?
- ⌘ 14) “Apesar da padronização, existem hoje vários dialetos SQL, cada um, evidentemente, tentando ser mais padronizado que o outro .” Verdadeiro ou Falso ?

2/2/2007

49

Exercícios Propostos

- ⌘ 15) “A idéia da SQL só previa seu uso de forma interativa. Após sofrer alguns acréscimos, ela passou também a ter a capacidade de ser utilizada em linguagens hospedeiras, tais como : COBOL, FORTRAN, C, etc.” Verdadeiro ou Falso ?
- ⌘ 16) Em relação a SQL, coloque Vantagem ou Desvantagem :
- ⌘ () Independência do fabricante. A SQL é oferecida em praticamente todos os SGBD's. Com isso posso mudar de SGBD sem me preocupar com o novo que vai chegar;

2/2/2007

50

Exercícios Propostos

- ⌘ () Falta de ortogonalidade nas expressões, funções embutidas, variáveis indicadoras, referência a dados correntes, constante NULL, conjuntos vazios, etc;
- ⌘ () Portabilidade entre computadores. A SQL pode ser utilizada desde um computador pessoal, passando por uma estação de trabalho, até um computador de grande porte;
- ⌘ () Definição formal da linguagem após sua criação;
- ⌘ () Redução dos custos com treinamento. As aplicações podem se movimentar de um ambiente para o outro sem que seja necessária uma reciclagem da equipe de desenvolvimento;

2/2/2007

51

Exercícios Propostos

- ⌘ () A padronização leva a uma, natural, inibição da criatividade, pois quem desenvolve aplicações fica preso a soluções padronizadas, não podendo sofrer melhorias ou alterações;
- ⌘ () Inglês estruturado de alto nível. A SQL é formada por um conjunto bem simples de sentenças em inglês, oferecendo um rápido e fácil entendimento;
- ⌘ () Discordância com as linguagens hospedeiras;
- ⌘ () Consulta interativa. A SQL provê um acesso rápido aos dados, fornecendo respostas ao usuário, a questões complexas, em minutos ou segundos;
- ⌘ () Falta de algumas funções;

2/2/2007

52

Exercícios Propostos

- ⌘ () Múltiplas visões dos dados. A SQL permite ao criador do banco de dados levar diferentes visões dos dados a diferentes usuários;
- ⌘ () Erros (valores nulos, índices únicos, cláusula FROM etc)
- ⌘ () Definição dinâmica dos dados. Por meio da SQL, podem-se alterar, expandir ou incluir, dinamicamente, as estruturas dos dados armazenados com a máxima flexibilidade;
- ⌘ () Não dá suporte a alguns aspectos do modelo relacional (atribuição de relação, join explícito, domínios);

2/2/2007

53

Respostas SQL

- ⌘ 1) **SELECT** PD.PDCDESC, PD.PDCUN, PD.PDYVALR **FROM** PRODUTO **AS** PD
- ⌘ 2) **SELECT** CL.CLCCGC, CL.CLCDISC, CL.CLCENDE **FROM** CLIENTE **AS** CL
- ⌘ 3) **SELECT** **VD.*** **FROM** VENDEDOR **AS** VD
- ⌘ 4) **SELECT** VD.VDCDESC, (**VD.VDYSALR * 2**) **AS** **AUMENTO** **FROM** VENDEDOR **AS** VD

2/2/2007

54

Respostas SQL

⌘ 5) SELECT IP.IPNCGPE, IP.IPNCGPD, IP.IPNQTD
FROM ITEMPEI AS IP
⌘ WHERE (IP.IPNQTD = 35)
⌘ 2 tuplas
⌘ 6) SELECT CL.CLCDESC FROM CLIENTE AS CL
WHERE (CL.CLCCIDA = "Niterói")
⌘ Suzana, Ana
⌘ 7) SELECT PD.PDCDESC FROM PRODUTO AS PD
WHERE (PD.PDCUN = "Kg") AND (PD.PDYVALR = 2)
⌘ Arroz Integral

2/2/2007

55

Respostas SQL

⌘ 8) SELECT CL.CLCDESC, CL.CLCENDE FROM
CLIENTE AS CL
⌘ WHERE (CL.CLCCIDA = "São Paulo") OR
⌘ ((CL.CLCCEP >= "30077000") AND
⌘ (CL.CLCCEP <= "30079000"))
⌘ 7 tuplas

2/2/2007

56

Respostas SQL

⌘ 9) SELECT PE.PENCODG FROM PEDIDO AS PE
WHERE NOT (PE.PENPRAZ = 15)
⌘ SELECT PE.PENCODG FROM PEDIDO AS PE
WHERE (PE.PENPRAZ <> 15)
⌘ 14 tuplas
⌘ 10) SELECT PD.PDNCODG, PD.PDCDESC FROM
PRODUTO AS PD
⌘ WHERE (PD.PDYVALR BETWEEN 0.32 AND 2)
⌘ 6 tuplas
⌘ WHERE (PD.PDYVALR >= .32 and PD.PDYVALR <= 2)

2/2/2007

57

Respostas SQL

⌘ 11) SELECT PD.PDNCODG, PD.PDCDESC FROM
PRODUTO AS PD
⌘ WHERE (PD.PDCDESC LIKE "L%")
⌘ Leite Crescimento
⌘ 12) SELECT VD.VDNCODG, VD.VDCDESC FROM
VENDEDOR AS VD
⌘ WHERE (VD.VDCDESC NOT LIKE "Jo%")
⌘ 4 tuplas

2/2/2007

58

Respostas SQL

⌘ 13) SELECT VD.VDCDESC FROM
VENDEDOR AS VD
⌘ WHERE (VD.VDCCOMI IN ("A", "B"))
⌘ 5 tuplas
⌘ WHERE (VD.VDCCOMI = "A" OR VD.VDCCOMI = "B")
⌘ 14) SELECT CL.* FROM CLIENTE AS CL
⌘ WHERE (CL.CLCIE IS NULL)
⌘ Jorge, Lívio
⌘ WHERE (CL.CLCIE IS NOT NULL)
⌘ 15) SELECT VD.VDCDESC, VD.VDYSALR FROM
VENDEDOR AS VD
⌘ ORDER BY VD.VDCDESC

2/2/2007

59

Respostas SQL

⌘ 16) SELECT CL.CLCUF, CL.CLCCIDA,
CL.CLCDESC FROM CLIENTE AS CL
⌘ ORDER BY CL.CLCUF DESC, CL.CLCCIDA DESC
⌘ ORDER BY CL.CLCUF, CL.CLCCIDA, CL.CLCDESC
⌘ 17) SELECT PD.PDCDESC, PD.PDYVALR FROM
PRODUTO AS PD WHERE (PD.PDCUN = "Kg")
⌘ ORDER BY PD.PDCDESC
⌘ Arroz Integral, Farinha e Feijão
⌘ ORDER BY PD.PDYVALR

2/2/2007

60

Respostas SQL

```
18) SELECT MIN (VD.VDYSALR),  
MAX (VD.VDYSALR)  
FROM VENDEDOR AS VD  
R$ 870,00 e R$ 9.500,00  
SELECT MIN (VD.VDYSALR) AS Menor_Salario,  
MAX (VD.VDYSALR) AS Maior_Salario,  
SUM (VD.VDYSALR) AS Total_Folha,  
COUNT (*) AS Total_Vendedores,  
AVG (VD.VDYSALR) AS Salario_Medio  
FROM VENDEDOR AS VD /*AVerage*/
```

2/2/2007

61

Respostas SQL

```
19) SELECT SUM (IP.IPNQTD)  
FROM ITEMPEI AS IP  
WHERE ( IP.IPNCGPD = 78)  
183  
20) SELECT AVG (VD.VDYSALR)  
FROM VENDEDOR AS VD  
R$ 3.324,44
```

2/2/2007

62

Respostas SQL

```
21) SELECT COUNT (*)  
FROM VENDEDOR AS VD  
WHERE (VD.VDYSALR > 2500 )  
5  
SELECT VD.VDNCODG, VD.VDCDESC, VD.VDYSALR  
FROM VENDEDOR AS VD  
WHERE (VD.VDYSALR > 2500)
```

2/2/2007

63

Respostas SQL

```
22) SELECT DISTINCT PD.PDCUN  
FROM PRODUTO AS PD  
10 tuplas -> 5 tuplas  
23)  
SELECT IP.IPNCGPE, COUNT (*) AS  
TOTAL_PRODUTOS FROM ITEMPEI AS IP  
SELECT IP.IPNCGPE, COUNT (*) AS  
TOTAL_PRODUTOS FROM ITEMPEI AS IP  
GROUP BY IP.IPNCGPE
```

2/2/2007

64

Respostas SQL

```
24) SELECT IP.IPNCGPE, COUNT (*) AS  
TOTAL_PRODUTOS FROM ITEMPEI AS IP  
GROUP BY IP.IPNCGPE  
HAVING COUNT (*) > 3  
2 tuplas  
HAVING TOTAL_PRODUTOS > 3
```

2/2/2007

65

Respostas SQL

```
25) SELECT CL.CLCDESC, CL.CLNCODG,  
PE.PENCGCL, PE.PENCODG  
FROM CLIENTE AS CL, PEDIDO AS PE  
WHERE (CL.CLNCODG = PE.PENCGCL)  
ORDER BY CL.CLCDESC  
ou  
ORDER BY CL.CLNCODG
```

2/2/2007

66

Respostas SQL

```
⌘ SELECT PE.PENCGCL, PE.PENCODG,  
⌘ CL.CLCDESC, CL.CLCCIDA,  
⌘ VD.VDNCODG, VD.VDCDESC, VD.VDYSALR  
⌘ FROM PEDIDO AS PE ,  
⌘ CLIENTE AS CL, VENDEDOR AS VD  
⌘ WHERE (PE.PENCGCL = CL.CLNCODG) AND  
⌘ (PE.PENCGVD = VD.VDNCODG)  
⌘ ORDER BY CL.CLCDESC
```

2/2/2007

67

Respostas SQL

```
⌘ 26) SELECT CL.CLCDESC, CL.CLNCODG,  
⌘ PE.PENPRAZ  
⌘ FROM CLIENTE AS CL, PEDIDO AS PE  
⌘ WHERE (CL.CLNCODG = PE.PENCGCL) AND  
⌘ (CL.CLCUF IN ("SP", "RJ")) AND  
⌘ (PE.PENPRAZ > 15)  
⌘ GROUP BY CL.CLCDESC, CL.CLNCODG,  
⌘ PE.PENPRAZ  
⌘ 5 tuplas
```

2/2/2007

68

Respostas Exercício SQL

```
⌘ 1) SELECT PD.PDNCODG, PD.PDCDESC  
⌘ FROM PRODUTO AS PD  
⌘ WHERE (PD.PDYVALR  
⌘ NOT BETWEEN 0.32 AND 2)  
⌘ 4 tuplas
```

2/2/2007

69

Respostas Exercício SQL

```
⌘ 2) SELECT VD.VDCDESC,  
⌘ ((VD.VDYSALR * 1.75) + 120 ) AS NOVO_SALARIO  
⌘ FROM VENDEDOR AS VD  
⌘ WHERE ( VD.VDCCOMI = "C" )  
⌘ ORDER BY VD.VDCDESC  
⌘ 4 tuplas
```

2/2/2007

70