

Sistemas de Banco de Dados Projeto, implementação e gerenciamento

Capítulo 7

Introdução à linguagem SQL

Objetivos

- **Neste capítulo, você aprenderá:**
 - Comandos e funções básicas de SQL
 - Como utilizar SQL para a administração de dados (criar tabelas, índices e visualizações)
 - De que modo usar SQL para a manipulação de dados (adicionar, modificar, excluir e recuperar dados)
 - Como utilizar SQL para consultar informações úteis em um banco de dados

Introdução à SQL

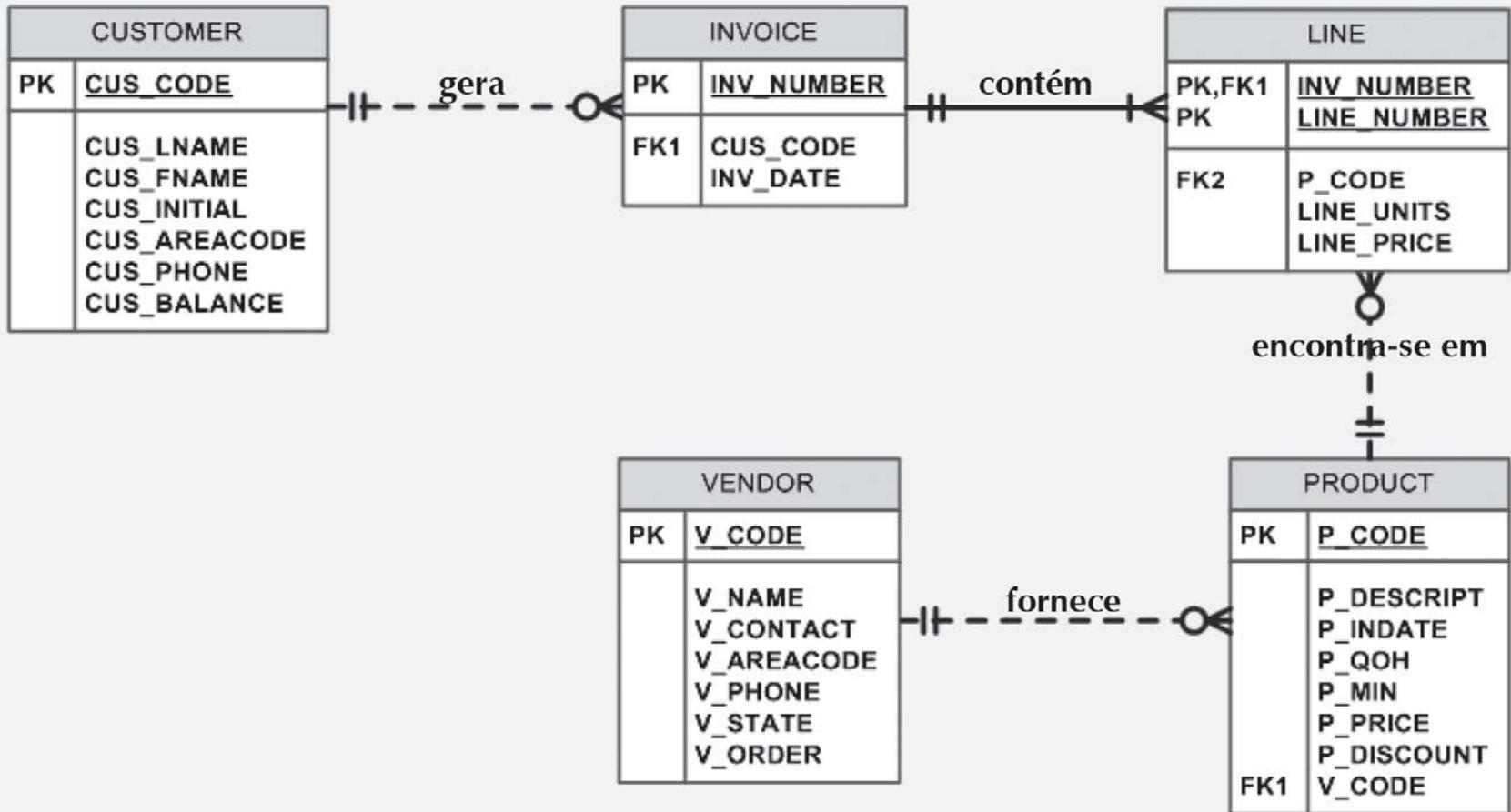
- Funções SQL se enquadram em duas amplas categorias:
 - Linguagem de definição de dados (DDL)
 - Linguagem de manipulação de dados (DML)
- Seu conjunto básico de comandos possui um vocabulário com menos de 100 palavras
- O Instituto Nacional Americano de Padrões (ANSI, American National Standards Institute) recomenda um padrão de SQL
- Há vários “dialetos” de SQL

Comandos de Definição de Dados

- Modelo simples de banco de dados
 - Um banco de dados simples, composto das seguintes tabelas, será utilizado para ilustrar os comandos de SQL neste capítulo:
 - CUSTOMER (cliente)
 - INVOICE (fatura)
 - LINE (linha)
 - PRODUCT (produto)
 - VENDOR (fornecedor)
 - Focaremos a atenção nas tabelas PRODUCT (produto) e VENDOR (fornecedor)

FIGURA 7.1

Modelo de banco de dados



Criação de Banco de Dados

- Duas tarefas devem ser realizadas:
 - Criar a estrutura do banco de dados
 - Criar as tabelas que manterão os dados do usuário final
- Primeira tarefa:
 - O SGBDR cria os arquivos físicos que conterão o banco de dados
 - A criação da estrutura do banco é um aspecto que tende a diferir significativamente de um SGBDR para outro

Esquema de Banco de Dados

- **Autenticação**

- Processo por meio do qual um SGBD garante que somente usuários registrados possam acessar o banco de dados
- Deve-se fazer logon no SGBDR utilizando uma ID de usuário e uma senha criada pelo administrador do banco de dados

- **Esquema**

- Grupo de objetos de banco de dados relacionados entre si

Tipos de Dados

- A seleção de tipos de dados costuma ser determinada pela natureza dos dados e pela utilização pretendida
- Por exemplo:
 - Number(L,D), Integer, Smallint, Decimal(L,D)
 - Char(L), Varchar(L), Varchar2(L)
 - Date, Time, Timestamp
 - Entre outros

Criação de Estruturas de Tabelas

- A maioria dos programadores utiliza uma linha por definição de coluna (atributo)
- São utilizados espaços para alinhar as características e restrições dos atributos
- Tanto os nomes de tabelas como de atributos aparecem totalmente em letras maiúsculas
- Especificações NOT NULL
- Especificação UNIQUE

Criação de Estruturas de Tabelas (cont.)

- Os atributos de chave primária contêm tanto uma especificação NOT NULL como uma UNIQUE
- O SGBDR aplicará automaticamente integridade referencial às chaves estrangeiras
- A sequência de comandos termina com ponto e vírgula

Restrições de SQL

- Restrição NOT NULL
 - Garante que uma coluna não aceite nulos.
- Restrição UNIQUE
 - Garante que todos os valores de uma coluna sejam exclusivos
- Restrição DEFAULT
 - Atribui um valor a um atributo quando uma nova linha é adicionada à tabela
- Restrição CHECK
 - Valida dados quando é inserido um valor de atributo

Índices em SQL

- Quando se declara uma chave primária, o SGBD cria automaticamente um índice exclusivo
- É comum precisarmos de índices adicionais
- Utilizando o comando **CREATE INDEX**, é possível criar índices com base em qualquer atributo selecionado
- Índices compostos
 - Índice baseado em dois ou mais atributos
 - Utilizados para evitar duplicações de dados

Comandos de Manipulação de Dados

- INSERT
- SELECT
- COMMIT
- UPDATE
- ROLLBACK
- DELETE

Inserção de Linhas na Tabela

- INSERT
 - Usado para inserir dados em uma tabela
 - Sintaxe básica
 - INSERT INTO *nome da tabela* VALUES (*valor1, valor2, ... , valoem*)

Inserção de Linhas na Tabela (cont.)

- Nas linhas de entrada de dados citadas, observe que:
 - O conteúdo da linha é inserido entre parênteses
 - Os valores de caracteres (string) e datas devem ser inseridos entre apóstrofos (' ')
 - As entradas numéricas *não* são cercadas por apóstrofos
 - As entradas de atributos são separadas por vírgulas
 - É necessário um valor para cada coluna da tabela
- Use NULL para valores desconhecidos

Salvando Alterações na Tabela

- Quaisquer alterações feitas no conteúdo de uma tabela não são salvas no disco até que
 - O usuário feche o banco de dados
 - O programa seja fechado
 - O comando COMMIT seja aplicado
- Sintaxe:
 - COMMIT [WORK];
- Salva permanentemente *todas* as alterações feitas a qualquer tabela do banco de dados

Listagem de Linhas da Tabela

- **SELECT**
 - É utilizado para listar o conteúdo de uma tabela
 - Sintaxe:
 - *SELECT lista de colunas*
 - *FROM nome da tabela*
- *A lista de colunas* representa um ou mais atributos separados por vírgulas
- Pode-se utilizar * (asterisco) como caractere coringa para listar todos os atributos

Atualização de Linhas da Tabela

- UPDATE
 - É utilizado para modificar os dados de uma tabela
 - Sintaxe:

```
UPDATE nome da tabela
SET nome da coluna = expressão [, nome da
coluna = expressão]
[WHERE lista de condições ];
```
- Se mais de um atributo deve ser atualizado na linha, separe as correções com vírgulas

Restauração de Conteúdo da Tabela

- ROLLBACK
 - Desfaz quaisquer alterações desde o último comando COMMIT
 - Retorna os dados para os valores existentes antes de as alterações serem feitas
- Sintaxe:
 - ROLLBACK;
- COMMIT e ROLLBACK funcionam apenas com comandos de manipulação de dados utilizados para adicionar, modificar e excluir linhas de tabela

Exclusão de Linhas da Tabela

- DELETE
 - Exclui uma linha da tabela
 - Sintaxe:

```
DELETE FROM nome da tabela  
[WHERE lista de condições];
```
- A condição WHERE é opcional
- Se a condição WHERE não for especificada, *todas* as linhas da tabela especificada serão excluídas

Inserção de Linhas na Tabela com uma Subconsulta de SELEÇÃO

- INSERT
 - Como adicionar várias linhas a uma tabela utilizando outra tabela como fonte dos dados
 - INSERT utiliza uma subconsulta SELECT
 - **Subconsulta (consulta integrada ou consulta interna)** é aquela incorporada no interior de outra consulta
 - Primeiro, ela é sempre executada pelo SGBDR
 - Sintaxe:

```
INSERT INTO nome da tabela
SELECT lista de colunas
FROM nome da tabela;
```

Consultas de SELEÇÃO

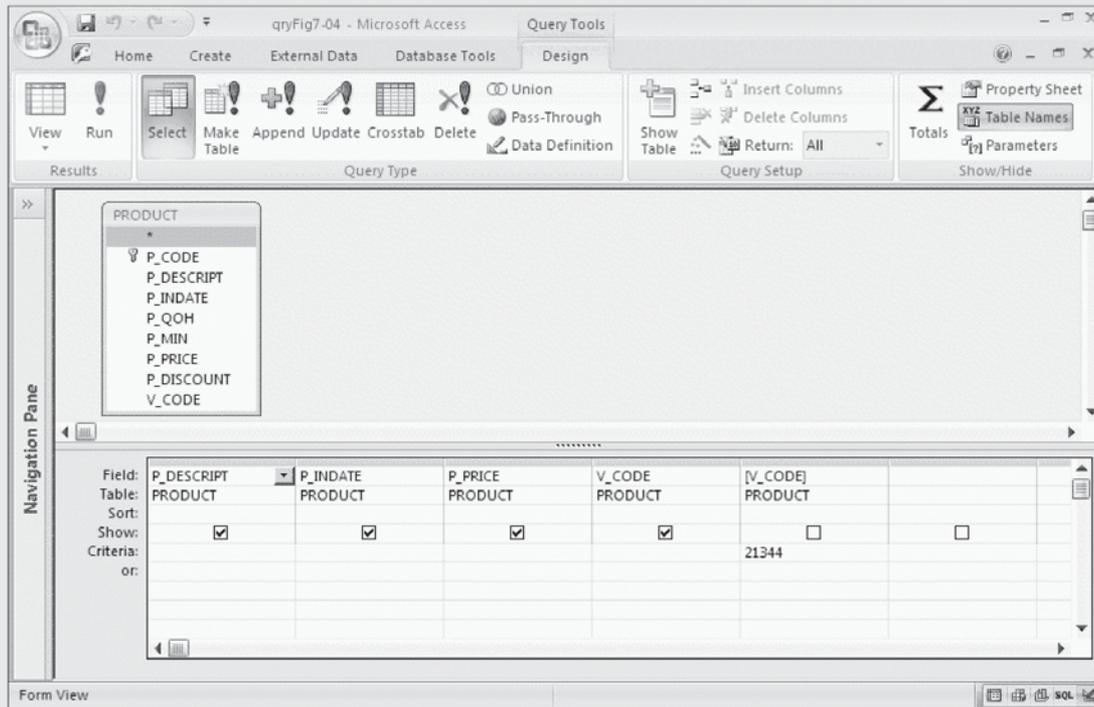
- Como refinar o comando de seleção (SELECT), adicionando restrições aos critérios de busca usando:
 - Restrições condicionais
 - Operadores aritméticos
 - Operadores lógicos
 - Operadores especiais

Seleção de Linhas com Condições Restritas

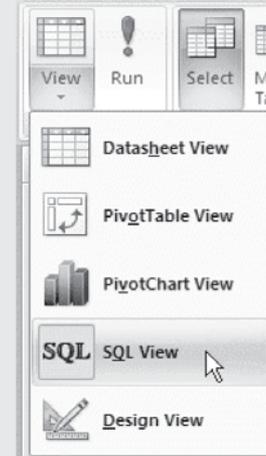
- É possível selecionar o conteúdo de uma tabela parcial colocando restrições para as linhas a serem incluídas no resultado
 - Isso é feito com a utilização da cláusula **WHERE** para adicionar restrições condicionais ao comando **SELECT**.
- Sintaxe:
SELECT lista de colunas
FROM lista de tabelas
[WHERE lista de condições];

FIGURA 7.5

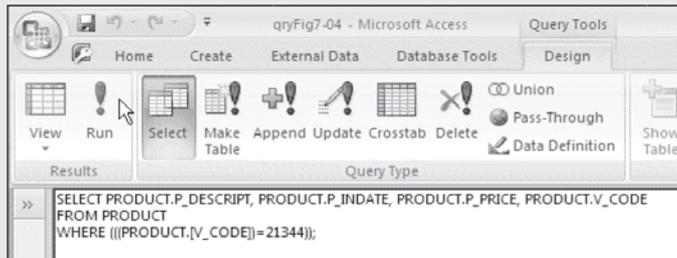
Microsoft Access QBE e sua SQL



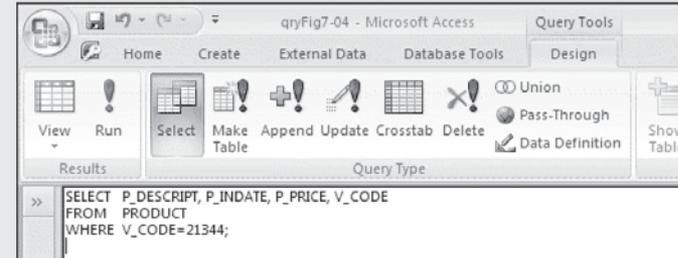
Opções de consulta



SQL gerada pelo Microsoft Access



SQL inserida pelo usuário



Operadores Aritméticos: Regra da Precedência

- Efetuar operações entre parênteses
- Efetuar operações de potenciação
- Efetuar multiplicações e divisões
- Efetuar somas e subtrações

FIGURA 7.7

Atributos selecionados da tabela **PRODUCT** com uma restrição de **P_PRICE**

P_DESCRIPT	P_QOH	P_MIN	P_PRICE
Claw hammer	23	10	9.95
Rat-tail file, 1/8-in. fine	43	20	4.99
PVC pipe, 3.5-in., 8-ft	188	75	5.87
1.25-in. metal screw, 25	172	75	6.99
2.5-in. wd. screw, 50	237	100	8.45

Operadores Lógicos: AND, OR e NOT

- A busca de dados normalmente envolve diversas condições
- Os operadores lógicos são AND, OR e NOT
- Observe a utilização de parênteses para combinar restrições lógicas
 - O local onde os parênteses devem ser colocados depende de como se deseja que as restrições lógicas sejam executadas
 - As condições listadas entre parênteses são sempre executadas primeiro

Operadores Lógicos: AND, OR e NOT (cont)

- **Álgebra booleana:** um campo de especialidade em matemática dedicado à utilização dos operadores lógicos OR e AND
- O operador lógico **NOT** é utilizado para negar o resultado de uma expressão condicional

Operadores Especiais

- **BETWEEN** – Utilizado para verificar se o valor de um atributo está dentro de uma faixa
- **IS NULL** – Utilizado para verificar se o valor de um atributo é nulo
- **LIKE** – Utilizado para verificar se o valor de um atributo coincide com um determinado padrão de caractere
- **IN** – Utilizado para verificar se o valor de um atributo coincide com qualquer valor de uma lista
- **EXISTS** – Utilizado para verificar se uma subconsulta retorna alguma linha

Comandos Avançados de Definição de Dados

- Todas as alterações na estrutura da tabela são feitas utilizando o comando `ALTER TABLE`
- Três opções:
 - `ADD` para adicionar uma coluna
 - `MODIFY` para alterar as características de uma coluna
 - `DROP` para excluir uma coluna de uma tabela
- Também pode ser utilizado para:
 - Adicionar restrições de tabelas
 - Remover uma restrição de coluna ou tabela

Alteração do Tipo dos Dados da Coluna

- O comando ALTER pode ser usado para alterar o tipo de dado
- Alguns SGBDRs, como o Oracle, não permitem fazer alterações de tipos de dados a menos que a coluna alterada esteja vazia

Alteração da Característica dos Dados da Coluna

- O comando ALTER pode ser usado para fazer alterações nas características dos dados
- Essas alterações são permitidas desde que não se altere o *tipo* de dados

Adicionando e Excluindo Colunas

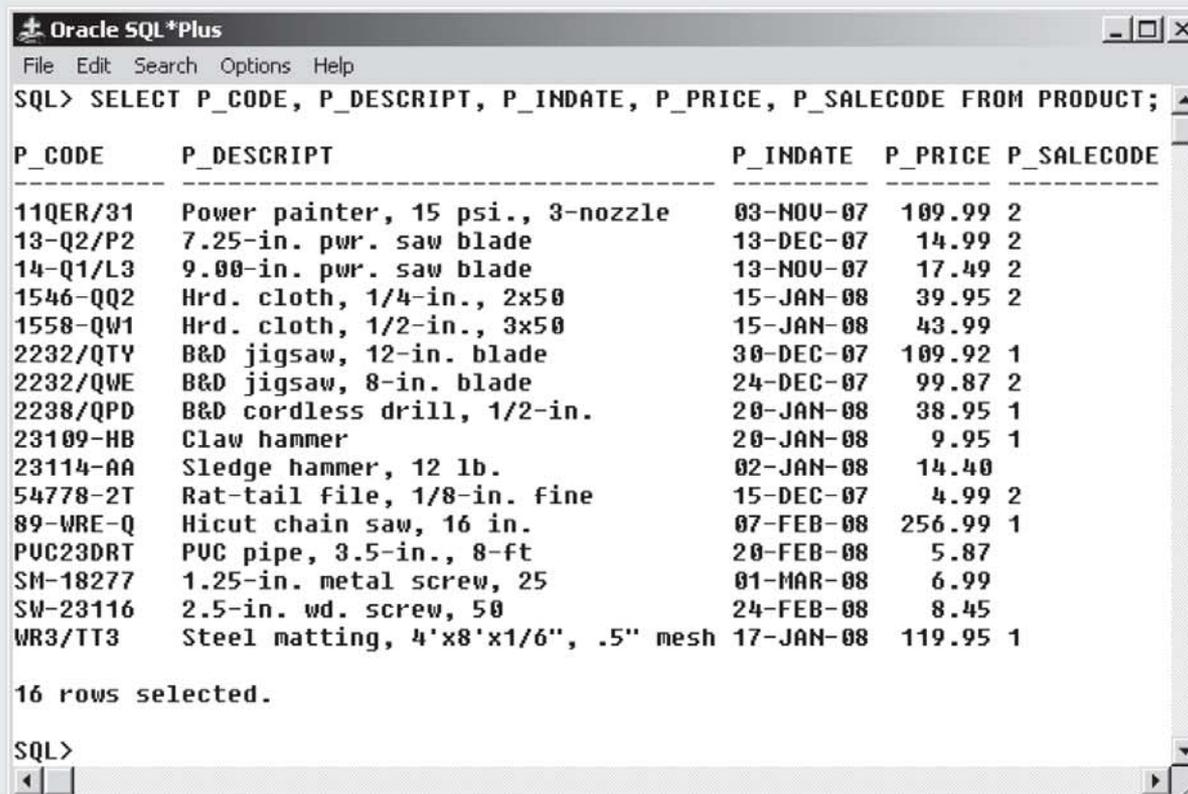
- O comando ALTER pode ser usado para adicionar colunas
 - Tenha cuidado para não incluir a cláusula NOT NULL na nova coluna
- O comando ALTER pode ser usado para excluir colunas
 - Alguns SGBDRs impõem restrições para a exclusão de atributos

Atualizações de Dados Avançadas

- Para fazer entradas de dados em colunas de uma linha existente, a SQL fornece o comando UPDATE
- Se for possível estabelecer um relacionamento entre as entradas e as colunas existentes, ele pode ser utilizado para atribuir os valores a suas posições adequadas
- Os operadores aritméticos são especialmente úteis em atualizações de dados
- Se estiver utilizando Oracle, emita um comando ROLLBACK para desfazer as alterações realizadas pelos dois últimos comandos UPDATE

**FIGURA
7.15**

Efeito cumulativo das várias atualizações na tabela PRODUCT (Oracle)



```
Oracle SQL*Plus
File Edit Search Options Help
SQL> SELECT P_CODE, P_DESCRIPT, P_INDATE, P_PRICE, P_SALECODE FROM PRODUCT;
```

P_CODE	P_DESCRIPT	P_INDATE	P_PRICE	P_SALECODE
11QER/31	Power painter, 15 psi., 3-nozzle	03-NOV-07	109.99	2
13-Q2/P2	7.25-in. pwr. saw blade	13-DEC-07	14.99	2
14-Q1/L3	9.00-in. pwr. saw blade	13-NOV-07	17.49	2
1546-QQ2	Hrd. cloth, 1/4-in., 2x50	15-JAN-08	39.95	2
1558-QW1	Hrd. cloth, 1/2-in., 3x50	15-JAN-08	43.99	
2232/QTY	B&D jigsaw, 12-in. blade	30-DEC-07	109.92	1
2232/QWE	B&D jigsaw, 8-in. blade	24-DEC-07	99.87	2
2238/QPD	B&D cordless drill, 1/2-in.	20-JAN-08	38.95	1
23109-HB	Claw hammer	20-JAN-08	9.95	1
23114-AA	Sledge hammer, 12 lb.	02-JAN-08	14.40	
54778-2T	Rat-tail file, 1/8-in. fine	15-DEC-07	4.99	2
89-WRE-Q	Hicut chain saw, 16 in.	07-FEB-08	256.99	1
PUC23DRT	PUC pipe, 3.5-in., 8-ft	20-FEB-08	5.87	
SM-18277	1.25-in. metal screw, 25	01-MAR-08	6.99	
SW-23116	2.5-in. wd. screw, 50	24-FEB-08	8.45	
WR3/TT3	Steel matting, 4'x8'x1/6", .5" mesh	17-JAN-08	119.95	1

16 rows selected.

```
SQL>
```

Copiando Partes de Tabelas

- SQL permite copiar o conteúdo das colunas de tabela selecionados para que não seja necessário reinserir os dados manualmente nas tabelas recém-criadas
- Deve-se, primeiro, criar a estrutura da tabela
- Em seguida, é necessário adicionar as linhas à nova tabela

FIGURA 7.16

Atributos da tabela PART copiados da tabela PRODUCT

PART_CODE	PART_DESCRIPTOR	PART_PRICE	V_CODE
11QER/31	Power painter, 15 psi., 3-nozzle	109.99	25595
13-Q2/P2	7.25-in. pwr. saw blade	14.99	21344
14-Q1/L3	9.00-in. pwr. saw blade	17.49	21344
1546-QQ2	Hrd. cloth, 1/4-in., 2x50	39.95	23119
1558-QW1	Hrd. cloth, 1/2-in., 3x50	43.99	23119
2232/QTY	B&D jigsaw, 12-in. blade	109.92	24288
2232/QWE	B&D jigsaw, 8-in. blade	99.87	24288
2238/QPD	B&D cordless drill, 1/2-in.	38.95	25595
23109-HB	Claw hammer	9.95	21225
23114-AA	Sledge hammer, 12 lb.	14.4	
54778-2T	Rat-tail file, 1/8-in. fine	4.99	21344
89-WRE-Q	Hicut chain saw, 16 in.	256.99	24288
PVC23DRT	PVC pipe, 3.5-in., 8-ft	5.87	
SM-18277	1.25-in. metal screw, 25	6.99	21225
SW-23116	2.5-in. wd. screw, 50	8.45	21231
WR3/TT3	Steel matting, 4'x8'x1/8", .5" mesh	119.95	25595

Adicionando Designações de Chaves Primárias e Estrangeiras

- Nenhuma regra de integridade de entidades (chave primária) e integridade referencial (chave estrangeira) deve ser aplicada automaticamente à nova tabela

Excluindo uma Tabela do Banco de Dados

- DROP
 - Exclui uma tabela do banco de dados
- É possível excluir uma tabela apenas se ela não for “um” lado de um relacionamento
 - Caso se tente excluir uma tabela desse tipo, o SGBDR gerará uma mensagem de erro indicando a ocorrência de uma violação de integridade de chave estrangeira

Consultas de SELEÇÃO Avançadas

- Os operadores lógicos apresentados anteriormente funcionam de modo muito similar ao do ambiente de consulta
- A SQL fornece funções úteis que:
 - Contam
 - Encontram valores mínimos e máximos
 - Calculam médias, etc
- SQL permite que o usuário limite as consultas apenas às entradas que não estejam duplicadas ou cujas duplicações possam ser agrupadas

Ordenando uma Lista

- A cláusula ORDER BY é especialmente útil quando a ordem da listagem é importante
- Sintaxe:
 SELECT *lista de colunas*
 FROM *lista de tabelas*
 [WHERE *lista de condições*]
 [ORDER BY *lista de colunas* [ASC | DESC]];
- Ordem-padrão é crescente

Listando Valores Únicos

- A cláusula de SQL DISTINCT produz uma lista que contém apenas os valores diferentes uns dos outros
- Exemplo:

```
SELECT DISTINCT V_CODE  
FROM PRODUCT;
```
- O Access posiciona o V_CODE nulo no topo da lista, ao passo que o Oracle o coloca no fim
 - O posicionamento de nulos não afeta o conteúdo da lista

Funções de Agregação

- A função COUNT é utilizada para contar o número de valores não nulos de um atributo
 - Utiliza um parâmetro entre parênteses, geralmente um nome de coluna
- As funções MAX e MIN ajudam a encontrar respostas para problemas como:
 - O maior (máximo) preço na tabela PRODUCT
 - O menor (mínimo) preço na tabela PRODUCT
- A função SUM calcula a soma total de um atributo especificado
- O formato da função AVG é similar ao da MIN e MAX

Agrupando Dados

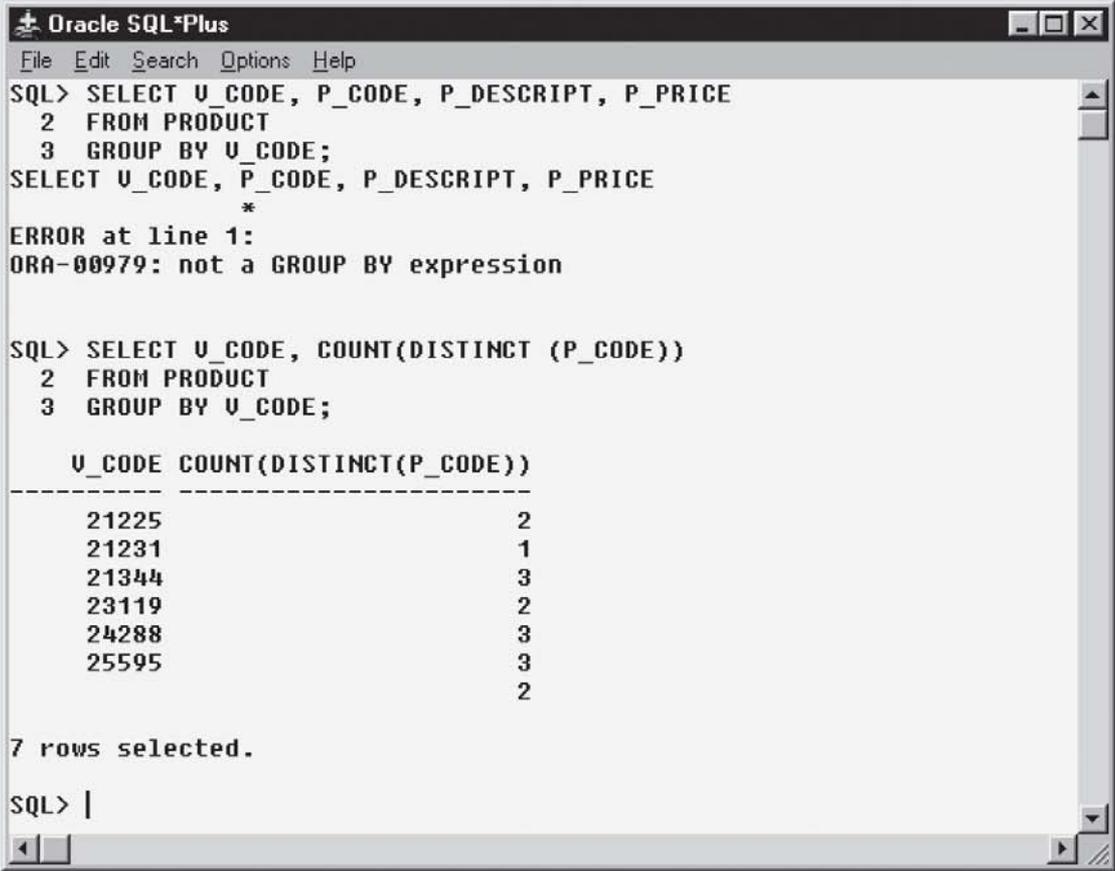
- As distribuições de frequência podem ser criadas de modo rápido e fácil utilizando a cláusula GROUP BY no comando SELECT

- Sintaxe:

```
SELECT      lista de colunas
FROM        lista de tabelas
[WHERE      lista de condições ]
[GROUP BY   lista de colunas ]
[HAVING     lista de condições ]
[ORDER BY   lista de colunas [ASC | DESC] ] ;
```

**FIGURA
7.26**

Utilização incorreta e correta da cláusula GROUP BY



```
Oracle SQL*Plus
File Edit Search Options Help
SQL> SELECT U_CODE, P_CODE, P_DESCRIPT, P_PRICE
 2 FROM PRODUCT
 3 GROUP BY U_CODE;
SELECT U_CODE, P_CODE, P_DESCRIPT, P_PRICE
      *
ERROR at line 1:
ORA-00979: not a GROUP BY expression

SQL> SELECT U_CODE, COUNT(DISTINCT (P_CODE))
 2 FROM PRODUCT
 3 GROUP BY U_CODE;

  U_CODE  COUNT(DISTINCT(P_CODE))
-----
 21225                2
 21231                1
 21344                3
 23119                2
 24288                3
 25595                3
                2

7 rows selected.

SQL> |
```

Tabelas Virtuais: Criando uma Visualização

- **Visualização** é uma tabela virtual baseada em uma consulta `SELECT`
- É possível criar uma visualização utilizando o comando `CREATE VIEW`

Tabelas Virtuais: Criando uma Visualização (cont.)

- A visualização relacional possui várias características especiais:
 - É possível utilizar o nome de uma visualização em qualquer posição de comandos de SQL em que deva ser inserido o nome de uma tabela
 - As visualizações são atualizadas dinamicamente
 - As visualizações fornecem um nível de segurança no banco de dados, pois podem restringir o acesso dos usuários apenas às colunas e linhas especificadas
 - As visualizações também podem ser utilizadas como base para relatórios

Junção de Tabelas de Bancos de Dados

- A capacidade de combinar tabelas em atributos comuns talvez seja a distinção mais importante entre o banco de dados relacional e os outros bancos
- Executa-se uma junção quando os dados são recuperados de mais de uma tabela ao mesmo tempo
 - A condição de junção costuma ser composta por uma comparação de igualdade entre a chave estrangeira e a chave primária das tabelas relacionada
- Para juntar tabelas, basta listá-las na cláusula FROM do comando SELECT
 - O SGBD criará o produto cartesiano de todas as tabelas dessa cláusula

Junção de Tabelas com Alias

- O *alias* pode ser utilizado para identificar a tabela-fonte da qual os dados foram pegos
- Os *alias* P e V serão utilizados para identificar as tabelas PRODUCT e VENDOR na próxima sequência de comandos
- Qualquer nome de tabela válido pode ser utilizado como *alias*

Junções Recursivas

Junções Externas

- O *alias* é especialmente útil quando uma tabela deve ser juntada a ela mesma em uma **consulta recursiva**
 - Deve-se também utilizar *alias* para diferenciar a própria tabela
- Dois tipos de junções externas
 - À esquerda
 - À direita

Resumo

- Os comandos de SQL podem ser divididos em duas categorias gerais:
 - De linguagem de definição de dados (DDL)
 - De linguagem de manipulação de dados (DML)
- Os tipos de dados do padrão ANSI são suportados por todos os fornecedores de SGBDRs em diferentes modos
- Os comandos básicos de definição de dados permitem a criação de tabelas, índices e visualizações

Resumo (cont.)

- Os comandos de DML permitem adicionar, modificar e excluir linhas das tabelas
- Os comandos básicos de DML são:
 - SELECT, INSERT, UPDATE, DELETE, COMMIT e ROLLBACK
- SELECT é o principal comando de recuperação de dados em SQL

Resumo (cont.)

- A cláusula `WHERE` pode ser utilizada com os comandos `SELECT`, `UPDATE` e `DELETE` para restringir as linhas afetadas pelo comando DDL
- Funções agregadas (`COUNT`, `MIN`, `MAX` e `AVG`)
 - São funções especiais que executam cálculos aritméticos em um conjunto de linhas
- A cláusula `ORDER BY`
 - Serve para classificar o resultado de um comando `SELECT`
 - Ordem crescente ou decrescente

Resumo (cont.)

- Também é possível juntar o resultado de várias tabelas com o comando **SELECT**
 - Ocorre sempre que forem especificadas duas ou mais tabelas na cláusula **FROM**
 - Se não for especificada uma condição de junção, o SGBD executará automaticamente um produto cartesiano
- A junção natural utiliza a condição de junção para obter a correspondência apenas nas linhas com valores iguais nas colunas especificadas
- Também é possível fazer uma junção externa à esquerda ou à direita para selecionar as linhas que não tenham valores correspondentes na outra tabela relacionada