

Sistemas de Banco de Dados Projeto, implementação e gerenciamento

Capítulo 8 SQL avançada

Objetivos

- **Neste capítulo, você aprenderá:**
 - Sobre os operadores de conjuntos relacionais UNION, UNION ALL, INTERSECT e MINUS
 - Como utilizar a sintaxe avançada do operador de SQL JOIN
 - Sobre os diferentes tipos de subconsultas e consultas correlacionadas
 - Como utilizar as funções de SQL para manipular datas, strings e outros dados

Objetivos (cont.)

- Como criar e utilizar visualizações atualizáveis
- Como criar e utilizar triggers e procedimentos armazenados
- Como criar SQL incorporada

Operadores do Conjunto Relacional

- UNION
- INTERSECT
- MINUS
- Funcionam adequadamente somente se as
- relações forem **compatíveis para união**
 - Os nomes dos atributos de relações devem ser os mesmos e seus tipos de dados devem ser semelhantes

UNION

- Combina linhas de duas ou mais consultas sem incluir linhas duplicadas

– Exemplo:

```
SELECT      CUS_LNAME, CUS_FNAME,
            CUS_INITIAL, CUS_AREACODE,
FROM        CUSTOMER
UNION
SELECT      CUS_LNAME, CUS_FNAME,
            CUS_INITIAL, CUS_AREACODE,
FROM        CUSTOMER_2
```

- Pode ser utilizado para unir mais do que apenas duas consultas

UNION ALL

- Pode ser utilizada para produzir uma relação que conserve as linhas duplicadas

– Exemplo de consulta:

```
SELECT      CUS_LNAME, CUS_FNAME,  
            CUS_INITIAL, CUS_AREACODE,  
FROM        CUSTOMER
```

UNION ALL

```
SELECT      CUS_LNAME, CUS_FNAME,  
            CUS_INITIAL, CUS_AREACODE,  
FROM        CUSTOMER_2;
```

- Pode ser utilizado para unir mais do que apenas duas consultas

Intersect

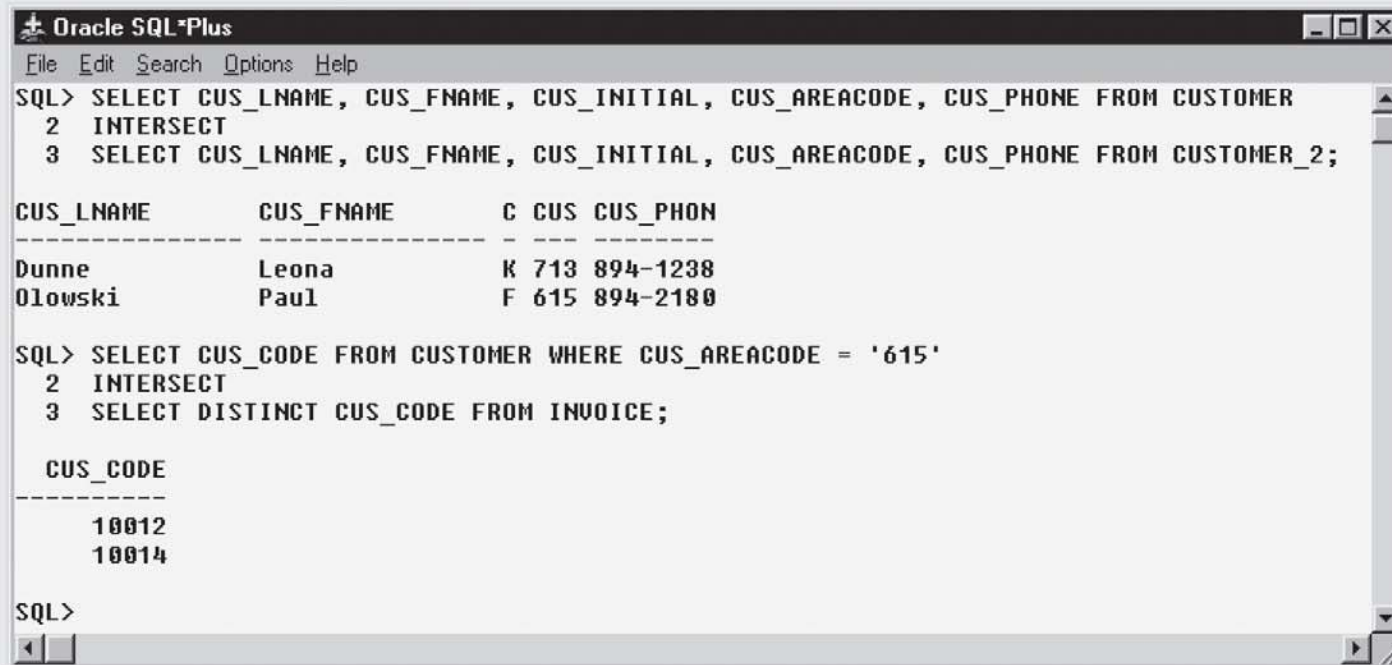
- Pode ser utilizado para combinar linhas de duas consultas, retornando apenas as que aparecem em ambos os conjuntos
- A sintaxe do comando INTERSECT é:

– Exemplo de consulta:

```
SELECT      CUS_LNAME, CUS_FNAME,
            CUS_INITIAL, CUS_AREACODE,
FROM        CUSTOMER
INTERSECT
SELECT      CUS_LNAME, CUS_FNAME,
            CUS_INITIAL, CUS_AREACODE,
FROM        CUSTOMER_2
```

**FIGURA
8.3**

Resultados da consulta INTERSECT



```
Oracle SQL*Plus
File Edit Search Options Help
SQL> SELECT CUS_LNAME, CUS_FNAME, CUS_INITIAL, CUS_AREACODE, CUS_PHONE FROM CUSTOMER
  2 INTERSECT
  3 SELECT CUS_LNAME, CUS_FNAME, CUS_INITIAL, CUS_AREACODE, CUS_PHONE FROM CUSTOMER_2;

CUS_LNAME      CUS_FNAME      C CUS CUS_PHON
-----
Dunne          Leona          K 713 894-1238
Olowski        Paul           F 615 894-2180

SQL> SELECT CUS_CODE FROM CUSTOMER WHERE CUS_AREACODE = '615'
  2 INTERSECT
  3 SELECT DISTINCT CUS_CODE FROM INVOICE;

CUS_CODE
-----
10012
10014

SQL>
```


MINUS

- Combina linhas de duas consultas e retorna apenas as que aparecem no primeiro conjunto, mas não no segundo
- A sintaxe do comando MINUS é:

– Exemplo:

```
SELECT      CUS_LNAME, CUS_FNAME,
            CUS_INITIAL, CUS_AREACODE,
FROM        CUSTOMER
MINUS
SELECT      CUS_LNAME, CUS_FNAME,
            CUS_INITIAL, CUS_AREACODE,
FROM        CUSTOMER_2
```

Alternativas de Sintaxe

- É possível utilizar as subconsultas IN e NOT IN no lugar de INTERSECT

- Exemplo:

```
SELECT      CUS_CODE FROM CUSTOMER
WHERE       CUS_AREACODE = '615' AND
           CUS_CODE IN (SELECT
DISTINCT CUS_CODE
FROM
INVOICE);
```

Operadores de Junção de SQL

- A operação de junção relacional funde linhas de duas tabelas e retorna as linhas com uma das seguintes condições:
 - Tenham valores comuns em colunas comuns
 - Junção natural
 - Atendam a determinada condição de junção
 - Igualdade ou desigualdade
 - Tenham valores comuns em colunas comuns ou não tenham valores correspondentes
 - Junção externa
- **Junção interna:** apenas as linhas que atendam a determinados critérios são selecionadas

Junção Cruzada

- executa um produto relacional de duas tabelas
 - Também conhecido como produto cartesiano
- Sintaxe:
 - `SELECT lista de colunas FROM tabela1 CROSS JOIN tabela2`
- Também é possível executar uma junção cruzada que produza apenas os atributos especificados

Junção Natural

- Retorna todas as linhas com valores correspondentes nas colunas correspondentes e elimina colunas duplicadas
- É utilizado quando as tabelas compartilham um ou mais atributos comuns com nomes comuns
- Sintaxe:
 - `SELECT lista de colunas FROM tabela1 NATURAL JOIN tabela2`

Cláusula de Junção USING

- Retorna apenas as linhas com valores correspondentes na coluna indicada nessa cláusula
- Sintaxe:

```
SELECT lista de colunas FROM tabela1 JOIN tabela2 USING (coluna comum)
```
- O operando JOIN USING não exige qualificadores de tabelas
 - O Oracle retornará um erro se for especificado o nome da tabela na cláusula USING.

Cláusula JOIN ON

- Usada quando as tabelas não têm nomes de atributos em comum
- Retornará apenas as linhas que atendam à condição de junção indicada
 - Normalmente incluirá uma expressão de comparação de igualdade de duas colunas
- Sintaxe: `SELECT lista de colunas tabela1 JOIN tabela2 ON condição de junção`

Junções Externas

- Retorna não apenas as linhas que atendam à condição de junção
- Também retorna as linhas sem valores correspondentes
- Três tipos
 - Esquerda
 - Direita
 - Completa
- As designações à esquerda e à direita referem-se à ordem em que as tabelas são processadas

Junções Externas (cont.)

- A junção externa à esquerda
 - Retorna não apenas as linhas que atendam à condição de junção
 - Também retorna as linhas da tabela do lado esquerdo sem valores correspondentes na tabela do lado direito
 - Sintaxe: `SELECT lista de colunas FROM tabela1 LEFT [OUTER] JOIN tabela2 ON condição de junção`
- A junção externa à direita
 - Retorna não apenas as linhas que atendam à condição de junção
 - Também retorna as linhas da tabela do lado direito sem valores correspondentes na tabela do lado esquerdo

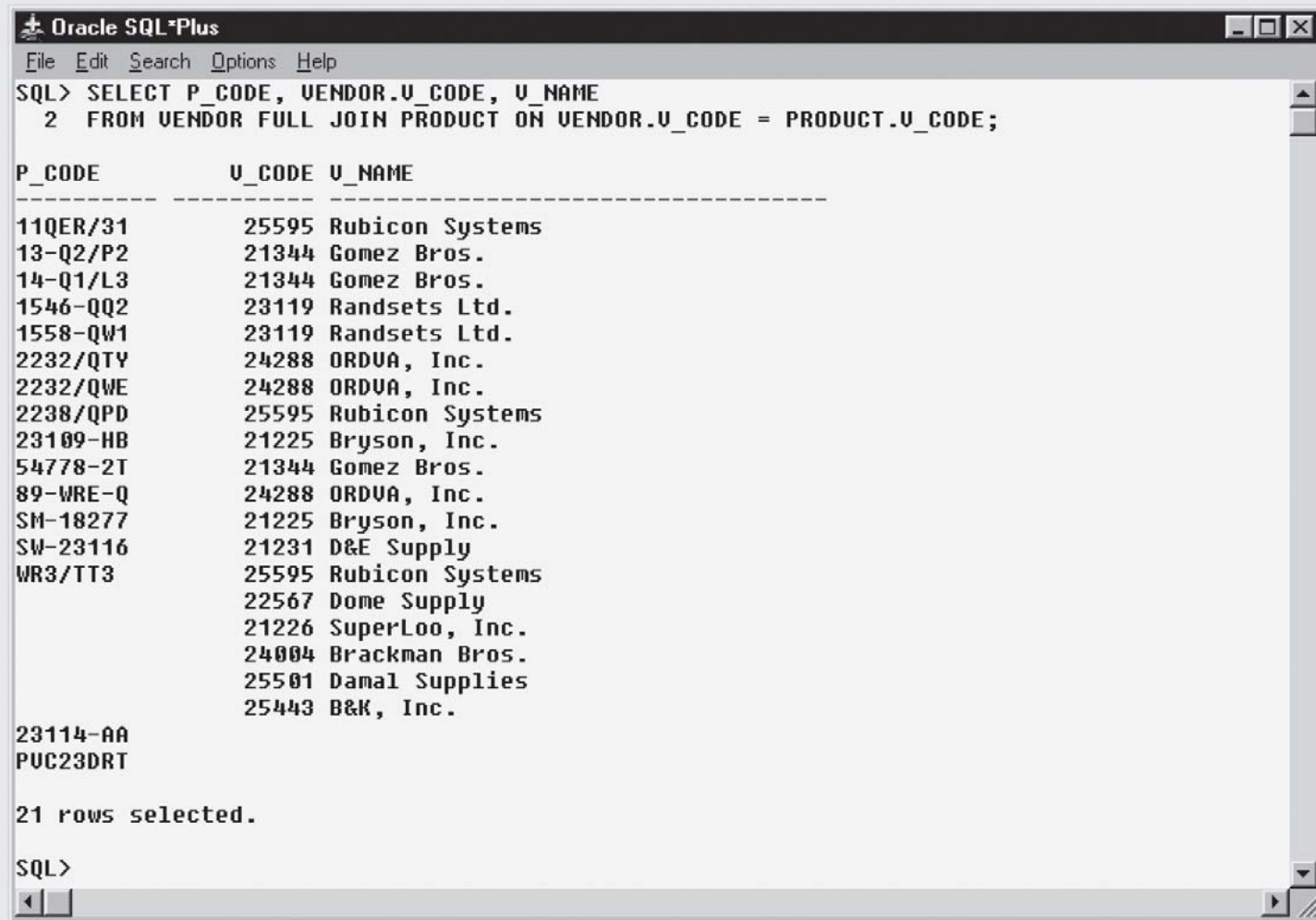
Junções Externas (cont.)

- A junção externa completa
 - Retorna não apenas as linhas que atendam à condição de junção
 - Também retorna todas as colunas sem valores correspondentes nas tabelas de ambos os lados
 - Sintaxe:

```
SELECT      lista de colunas
FROM        tabela1 FULL [OUTER] JOIN tabela2
           ON condição de junção
```

**FIGURA
8.12**

Resultados de FULL JOIN



The screenshot shows an Oracle SQL*Plus window with a menu bar (File, Edit, Search, Options, Help) and a command prompt. The user has entered a SQL query to perform a FULL JOIN between the VENDOR and PRODUCT tables on the U_CODE column. The output displays 21 rows of data with columns P_CODE, U_CODE, and U_NAME. The results are as follows:

P_CODE	U_CODE	U_NAME
11QER/31	25595	Rubicon Systems
13-Q2/P2	21344	Gomez Bros.
14-Q1/L3	21344	Gomez Bros.
1546-QQ2	23119	Randssets Ltd.
1558-QW1	23119	Randssets Ltd.
2232/QTY	24288	ORDVA, Inc.
2232/QWE	24288	ORDVA, Inc.
2238/QPD	25595	Rubicon Systems
23109-HB	21225	Bryson, Inc.
54778-2T	21344	Gomez Bros.
89-WRE-Q	24288	ORDVA, Inc.
SM-18277	21225	Bryson, Inc.
SW-23116	21231	D&E Supply
WR3/TT3	25595	Rubicon Systems
	22567	Dome Supply
	21226	SuperLoo, Inc.
	24004	Brackman Bros.
	25501	Damal Supplies
	25443	B&K, Inc.
23114-AA		
PUC23DRT		

21 rows selected.

SQL>

Subconsultas e Consultas Correlacionadas

- Frequentemente é necessário processar dados com base em *outros* dados processados
 - A subconsulta é uma consulta no interior de outra consulta
 - A subconsulta normalmente é expressa entre parênteses
 - A primeira consulta no comando de SQL é conhecida como consulta externa
 - A consulta no interior do comando de SQL é conhecida como consulta interna
 - A consulta interna é executada primeiro.
 - A saída de uma consulta interna é utilizada como entrada da consulta externa
 - O comando SQL inteiro, às vezes, é chamado de consulta integrada

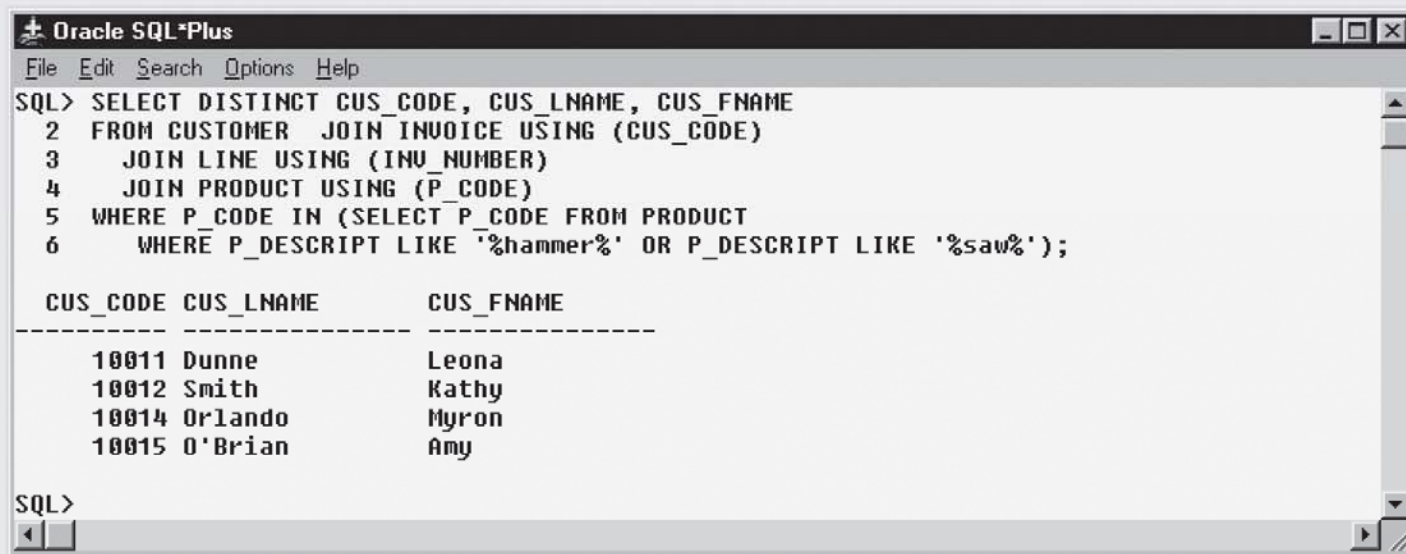
Subconsultas WHERE

- O tipo mais comum de subconsulta utiliza uma subconsulta SELECT interna ao lado direito da expressão de comparação WHERE
 - Exige uma subconsulta que retorne um único valor
- O valor gerado pela subconsulta deve apresentar um tipo de dado “comparável”
- Podem ser utilizadas em conjunto com junções

Subconsultas IN

- Usada quando se deseja comparar um único atributo a uma lista de valores

FIGURA 8.14 Exemplo de subconsulta IN



```
Oracle SQL*Plus
File Edit Search Options Help
SQL> SELECT DISTINCT CUS_CODE, CUS_LNAME, CUS_FNAME
2 FROM CUSTOMER JOIN INVOICE USING (CUS_CODE)
3 JOIN LINE USING (INV_NUMBER)
4 JOIN PRODUCT USING (P_CODE)
5 WHERE P_CODE IN (SELECT P_CODE FROM PRODUCT
6 WHERE P_DESCRIPT LIKE '%hammer%' OR P_DESCRIPT LIKE '%saw%');

CUS_CODE CUS_LNAME CUS_FNAME
-----
10011 Dunne Leona
10012 Smith Kathy
10014 Orlando Myron
10015 O'Brian Amy

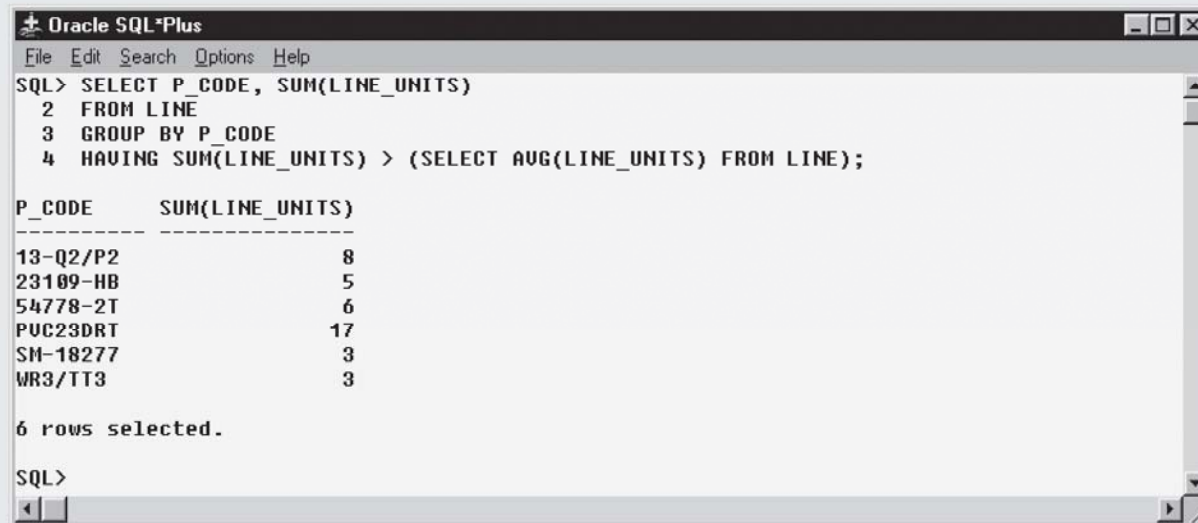
SQL>
```

Subconsultas HAVING

- É utilizada para restringir o resultado de uma consulta GROUP BY, aplicando um critério condicional às linhas agrupadas

FIGURA
8.15

Exemplo de subconsulta HAVING



```
Oracle SQL*Plus
File Edit Search Options Help
SQL> SELECT P_CODE, SUM(LINE_UNITS)
2 FROM LINE
3 GROUP BY P_CODE
4 HAVING SUM(LINE_UNITS) > (SELECT AVG(LINE_UNITS) FROM LINE);

P_CODE      SUM(LINE_UNITS)
-----
13-Q2/P2          8
23109-HB          5
54778-2T          6
PUC23DRT         17
SM-18277          3
WR3/TT3           3

6 rows selected.

SQL>
```

Operadores de Subconsulta em Várias Linhas: AN Y e ALL

- Permite comparar um único valor a uma lista de valores retornada pela primeira subconsulta, utilizando um operador de comparação diferente do igual
- “Maior que ALL” é equivalente a “maior que o maior custo de produto da lista”
- “Menor que ALL” é equivalente a “menor que o menor custo de produto da lista”
- Pode-se utilizar a igualdade com ANY, o que equivaleria ao operador IN

Subconsultas FROM

- Especifica as tabelas a partir das quais os dados serão obtidos
- É possível utilizar uma subconsulta SELECT na cláusula FROM
 - É possível utilizar um nome de visualização em qualquer posição de comando em que deve haver uma tabela

Subconsultas de Listas de Atributos

- O comando SELECT utiliza a lista de atributos para indicar quais colunas devem ser projetadas no conjunto resultante
 - Colunas podem ser atributos de tabelas de base ou atributos computados ou o resultado de uma função agregada
- lista de atributos também pode incluir uma expressão de subconsulta
 - Deve retornar um único valor
- O *alias* de coluna não pode ser utilizado em computações na lista de atributos, quando ele é definido na mesma lista

Subconsultas Correlacionadas

- É executada uma vez para cada linha na pesquisa externa
- A consulta é chamada *correlacionada*, pois a consulta interna é *relacionada* à externa pelo fato de a interna referenciar uma coluna da subconsulta externa
- As subconsultas correlacionadas também podem ser utilizadas com o operador especial **EXISTS**

Funções da SQL

- A geração de informações a partir dos dados costuma exigir muitas manipulações
- As funções sempre utilizam um valor numérico, de data ou de caracter
- O valor pode fazer parte do próprio comando ou por ser um atributo localizado em uma tabela
- A função pode aparecer em qualquer posição de um comando de SQL

Funções de Data e Hora

- Todos os SGBDs do padrão SQL dão suporte a funções de data e hora
- Todas as funções de data tomam um parâmetro e retornam um valor
- Os tipos de dados de data/hora são implementados de modo diferente por fornecedores diferentes de SGBDs
- O padrão SQL ANSI define os tipos de dados de data, mas não diz como esses tipos de dados devem ser armazenados

Funções Numéricas

- As funções numéricas podem ser agrupadas de vários modos diferentes
 - Algébrico, trigonométrico, logarítmico, etc
- Não confunda as funções agregadas de SQL, com as funções numéricas desta seção
- As funções numéricas tomam um parâmetro numérico e retornam um valor

TABELA 8.5 Funções numéricas selecionadas

FUNÇÃO	EXEMPLO(S)
<p>ABS</p> <p>Retorna o valor absoluto de um número</p> <p>Sintaxe:</p> <p>ABS(valor numérico)</p>	<p>Em Oracle, utilize:</p> <pre>SELECT 1.95, -1.93, ABS(1.95), ABS(-1.93) FROM DUAL;</pre> <p>Em MS Access/SQL Server, utilize:</p> <pre>SELECT 1.95, -1.93, ABS(1.95), ABS(-1.93);</pre>
<p>ROUND</p> <p>Arredonda um valor por determinada precisão (número de dígitos)</p> <p>Sintaxe:</p> <p>ROUND(valor numérico, p)</p> <p>p = precisão</p>	<p>Lista os preços de produtos arredondados para uma ou nenhuma casa decimal:</p> <pre>SELECT P_CODE, P_PRICE, ROUND(P_PRICE,1) AS PRICE1, ROUND(P_PRICE,0) AS PRICE0 FROM PRODUCT;</pre>
<p>CEIL/CEILING/FLOOR</p> <p>Retornam, respectivamente, o menor inteiro que seja maior ou igual a um número, ou retorna o maior inteiro que seja menor ou igual a um número.</p> <p>Sintaxe:</p> <p>CEIL(valor numérico) Oracle</p> <p>CEILING(valor numérico) SQL Server</p> <p>FLOOR(valor numérico)</p>	<p>Lista o preço dos produtos, o menor inteiro que seja maior ou igual e o maior inteiro que seja menor ou igual a esse preço.</p> <p>Em Oracle, utilize:</p> <pre>SELECT P_PRICE, CEIL(P_PRICE), FLOOR(P_PRICE) FROM PRODUCT;</pre> <p>Em SQL Server, utilize:</p> <pre>SELECT P_PRICE, CEILING(P_PRICE), FLOOR(P_PRICE) FROM PRODUCT;</pre> <p>O MS Access não dá suporte a essas funções.</p>

Funções de String

- As manipulações de strings estão entre as funções mais utilizadas em programação
- Exemplos de manipulação:
 - Concatenar adequadamente as strings de caracteres
 - Imprimir nomes em maiúscula
 - Saber o tamanho de determinado atributo

Funções de Conversão

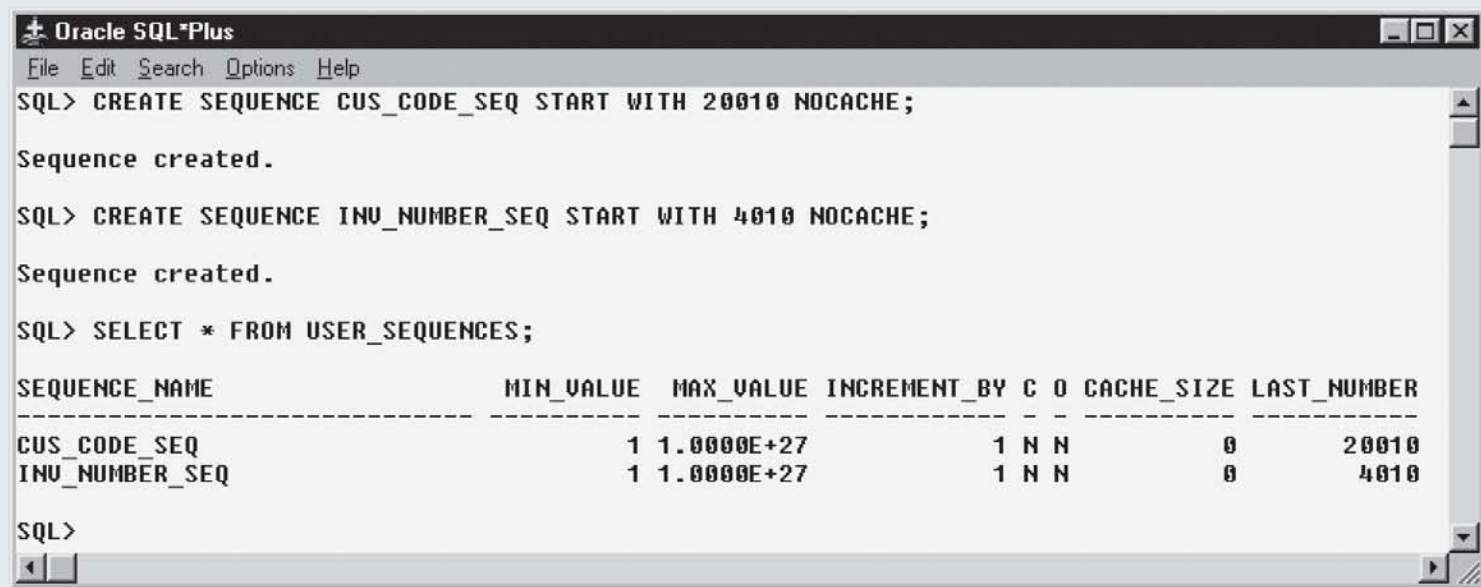
- Permitem tomar um valor de determinado tipo de dado e convertê-lo para o valor equivalente de outro tipo
- Funções básicas de conversão de SQL para Oracle:
 - TO_CHAR toma um valor de dados e retorna uma string de caracteres
 - TO_DATE toma uma string de caracteres representando uma data e retorna uma data real em formato Oracle
- O SQL Server utiliza as funções CAST e CONVERT para converter um tipo de dados em out

Sequências Oracle

- AutoNumber do MS Access pode ser utilizado para definir uma coluna de tabela preenchida automaticamente com valores numéricos exclusivos
- Sequências do Oracle
 - São um objeto independente no banco de dados
 - Possuem um nome e podem ser utilizadas em qualquer posição de comando
 - Não são atreladas a uma tabela ou coluna
 - Geram um valor numérico automático que pode ser atribuído a qualquer coluna de qualquer tabela
 - O atributo de tabela ao qual é atribuído um valor com base em sequência pode ser editado e modificado
 - É possível criar e excluir uma sequência a qualquer momento

**FIGURA
8.22**

Sequência de Oracle



```
Oracle SQL*Plus
File Edit Search Options Help
SQL> CREATE SEQUENCE CUS_CODE_SEQ START WITH 20010 NOCACHE;

Sequence created.

SQL> CREATE SEQUENCE INU_NUMBER_SEQ START WITH 4010 NOCACHE;

Sequence created.

SQL> SELECT * FROM USER_SEQUENCES;

SEQUENCE_NAME          MIN_VALUE  MAX_VALUE  INCREMENT_BY  CYCLE  CACHE_SIZE  LAST_NUMBER
-----
CUS_CODE_SEQ            1 1.0000E+27      1  N  N           0      20010
INU_NUMBER_SEQ          1 1.0000E+27      1  N  N           0      4010

SQL>
```

Vizualizações Atualizáveis

- **Rotina de atualização de batch** coloca todas as transações em um lote (batch) para atualizar uma tabela mestre em uma única operação
- **Visualização atualizável** é aquela que pode ser usada para atualizar atributos nas tabelas de base utilizadas na visualização
- Nem todas as visualizações são atualizáveis
 - Não é possível utilizar expressões GROUP BY e funções agregadas
 - Não é possível utilizar operadores de conjunto
 - A maioria das restrições baseia-se na utilização de JOINS

SQL Procedural

- A SQL não dá suporte à execução *condicional*
- Isolar o código fundamental
 - Todos os aplicativos chamem esse código compartilhado
 - Melhor controle de manutenção e lógica
- O **módulo armazenamento persistente (MAP)** é um bloco de código contendo comandos-padrão e extensões procedurais de SQL
 - Armazenado e executado no servidor de SGBD

SQL Procedural (cont.)

- A **SQL procedural (PL/SQL)** é uma linguagem que possibilita a utilização e o armazenamento de código procedural e comandos de SQL no banco de dados e a fusão de SQL com estruturas de programação tradicional
- Os usuários finais podem utilizar PL/SQL para criar:
 - Blocos de PL/SQL anônimos
 - Triggers (cobertos na Seção “Triggers”)
 - Procedimentos armazenados
 - Funções de PL/SQL

Triggers

- Código de SQL procedural chamado *automaticamente* pelo SGBDR quando da ocorrência de determinado evento de manipulação de dados
- Uma definição de trigger contém as seguintes partes:
 - O momento do trigger: BEFORE ou AFTER
 - O evento do trigger: INSERT, UPDATE ou DELETE
 - O nível do trigger:
 - **Trigger de nível de comando**
 - **Trigger no nível de linha**
 - A ação do trigger
- DROP TRIGGER *nome do trigger*

Procedimentos Armazenados

- Coleção denominada de comandos procedurais e de SQL
- Vantagens:
 - Reduzem significativamente o tráfego de rede e aumentam o desempenho
 - Não há transmissão de comandos de SQL individual pela rede
 - Ajudam a reduzir a duplicação do código, que é isolado e compartilhado
 - Minimiza a chance de erros e o custo de desenvolvimento e manutenção de aplicações

Processamento de PL/SQL com Cursores

- **Cursor** é uma estrutura especial utilizada em SQL procedural para manter as linhas de dados provenientes de uma consulta de SQL
- **Cursor implícito** é criado automaticamente em SQL procedural quando a sentença de SQL retorna apenas um valor
- **Cursor explícito** é criado para manter o produto de um comando de SQL que tenha permissão para retornar duas ou mais linhas

Funções Armazenadas de PL/SQL

- Um grupo denominado de comandos procedurais e de SQL que retorna um valor
- Sintaxe:

```
CREATE FUNCTION nome da função  
(argumento IN tipo de dados, ... ) RETURN tipo  
de dados [IS] BEGIN  
    comandos de PL/SQL;  
    ...  
    RETURN (valor ou expressão);  
END;
```

SQL Embutida

- A SQL embutida ainda é a linguagem mais comum para manter recursos procedurais em aplicações com base em SGBDs
- No entanto, misturar a SQL com linguagens procedurais exige a compreensão de algumas diferenças fundamentais entre ela e essas linguagens:
 - *Correspondência incorreta durante a execução*
 - *Correspondência incorreta de processamento*
 - *Correspondência incorreta de tipos de dados*

SQL Embutida (cont.)

- O modelo de SQL incorporada define o seguinte:
 - Uma sintaxe-padrão para identificar o código de SQL incorporada na linguagem hospedeira
 - Uma sintaxe-padrão para identificar variáveis hospedeiras
 - É utilizada uma área de comunicação para trocar informações de *status* e erro entre a SQL e a linguagem hospedeira

TABELA 8.11 Variáveis de comunicação de *status* e erro de SQL

NOME DA VARIÁVEL	VALOR	EXPLICAÇÃO
SQLCODE		Comunicação de erro em estilo antigo, suportada apenas para compatibilidade com sistemas anteriores; retorna um valor inteiro (positivo ou negativo).
	0	Conclusão de comando com sucesso.
	100	Sem dados; o comando de SQL não retornou, selecionou, atualizou ou excluiu nenhuma linha.
	-999	Qualquer valor negativo indica a ocorrência de um erro.
SQLSTATE		Adicionada pelo padrão SQL-92 para fornecer códigos predefinidos de erros; estabelecida como string de caracteres (5 caracteres de comprimento).
	00000	Conclusão de comando com sucesso.
		Diversos valores no formato XXYYY, em que: XX-> representa o código de classe. YYY-> representa o código de subclasse.

SQL Embutida (cont.)

- **SQL estática**

- SQL incorporada em que o programador utilizou comandos e parâmetros de SQL predefinidos
 - Os usuários finais dos programas ficam limitados às ações especificadas na programação das aplicações
- Os comandos SQL não se alteram enquanto a aplicação estiver sendo executada

SQL Embutida (cont.)

- **SQL dinâmica**

- É um termo utilizado para descrever um ambiente em que o comando de SQL não é previamente conhecido; pelo contrário, ele é gerado durante a execução
- Um programa pode gerar o comando de SQL necessário para responder consultas *ad hoc*
- A lista de atributos e a condição não são conhecidas até que o usuário final as especifique
- Tende a ser muito mais lenta do que a SQL estática
- Exige mais recursos do computador

Resumo

- A SQL fornece operadores de conjuntos relacionais para combinar o resultado de duas consultas e gerar uma nova relação
- As operações que unem tabelas podem ser classificadas como internas ou externas
- A junção natural retorna todas as linhas com valores correspondentes nas colunas correspondentes e elimina colunas duplicadas
- As subconsultas e consultas correlacionadas são utilizadas ao processamento de dados com base em *outros* dados processados

Resumo (cont.)

- A maioria das subconsultas é executada de modo serial
- As funções de SQL são utilizadas para extrair e transformar dados
- As sequências de Oracle podem ser utilizadas para gerar valores a serem atribuídos a um registro
- A SQL procedural (PL/SQL) pode ser utilizada para criar triggers, procedimentos armazenados e funções
- O procedimento armazenado é uma coleção denominada de comandos de SQL

Resumo (cont.)

- Quando os comandos de SQL são projetados para retornar mais de um valor no interior do código de PL/SQL, é necessário um cursor
- A SQL embutida refere-se à utilização de comandos de SQL no interior de uma linguagem de programação de aplicações