

Sistemas de Informação

Design e Desenvolvimento de Banco de Dados

PROF. MILTON

Propriedades de Banco de Dados Relacional

- Um banco de dados relacional
 - Pode ser acessado e modificado executando instruções SQL (Structured Query Language)
 - Contém um conjunto de tabelas sem ponteiros físicos
 - Usa um conjunto de operadores

Propriedades de Banco de Dados Relacional

Em um banco de dados relacional, você não especifica a rota de acesso às tabelas e não precisa saber como os dados são organizados fisicamente.

Para acessar o banco de dados, execute uma instrução SQL (Structured Query Language), que é a linguagem padrão ANSI (American National Standards Institute) para a operação em bancos de dados relacionais. A linguagem contém um grande conjunto de operadores para dividir e combinar relações.

É possível modificar o banco de dados usando as instruções SQL.

Comunicando-se com um RDBMS Usando o SQL

A instrução SQL é informada

```
SELECT loc
FROM dept;
```

A instrução é enviada para o banco de dados

Banco de Dados

Os dados são exibidos

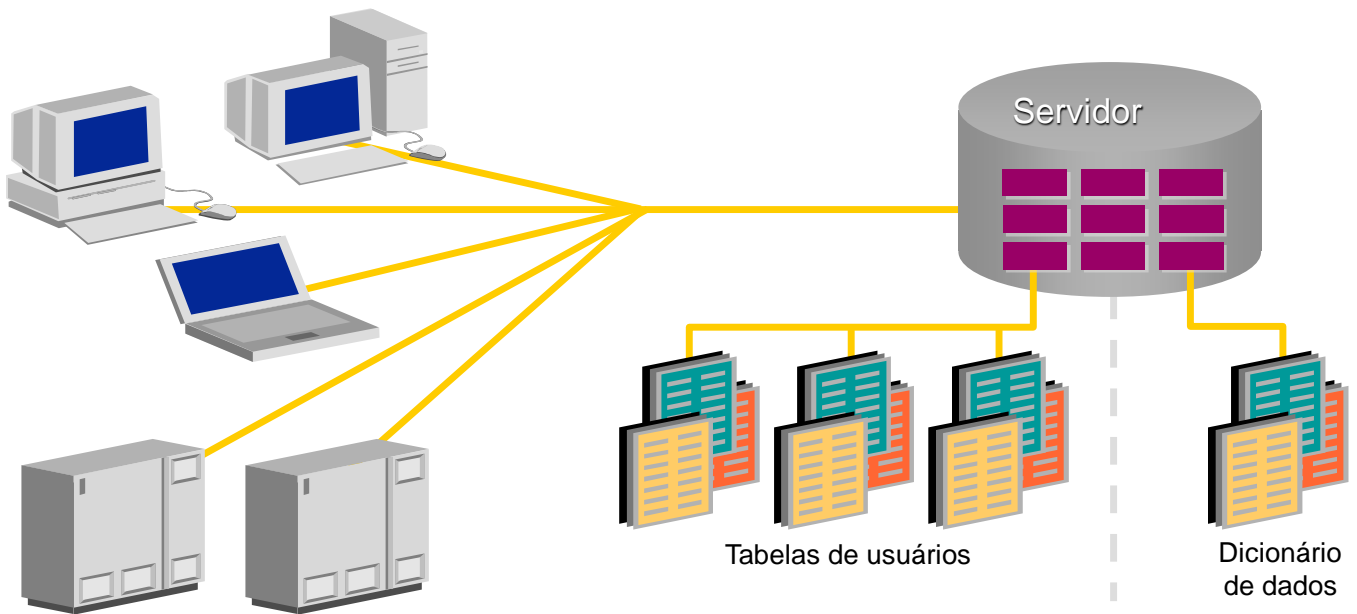
```
LOC
-----
NEW YORK
DALLAS
CHICAGO
BOSTON
```

SQL (Structured Query Language)

O **SQL** permite comunicar-se com o servidor e tem as seguintes vantagens:

- Eficiência
- Facilidade de aprendizagem e uso
- Funcionalidade completa (O SQL permite definir, recuperar e manipular dados das tabelas.)

Sistema de Gerenciamento de Banco de Dados Relacional



Sistema de Gerenciamento de Banco de Dados Relacional

A Oracle fornece um RDBMS flexível chamado Oracle. Usando os recursos desse RDBMS, você pode armazenar e gerenciar dados com todas as vantagens de uma estrutura relacional além do PL/SQL, um mecanismo com recurso para armazenar e executar unidades de programas. O servidor oferece as opções de recuperação de dados com base em técnicas de otimização. Ele inclui recursos de segurança que controlam como um banco de dados é acessado e usado. Outros recursos incluem a consistência e a proteção de dados através de mecanismos de bloqueio.

As aplicações Oracle podem ser executadas no mesmo computador que o Oracle Server. Como opção, você pode executar aplicações em um sistema local de usuário e executar o Oracle Server em outro sistema (arquitetura cliente-servidor). Nesse ambiente cliente-servidor, é possível usar uma grande variedade de recursos de computação. Por exemplo, uma aplicação de reservas de uma linha aérea baseada em forms pode ser executada no PC de um cliente, enquanto o acesso a dados de voo é gerenciado convenientemente por um Oracle Server em um computador central.

Para obter mais informações, consulte o *Oracle Server Concepts Manual*,

https://docs.oracle.com/cd/E11882_01/server.112/e40540/toc.htm.

Instruções SQL

SELECT	Recuperação de dados
INSERT UPDATE DELETE	DML (Data Manipulation Language)
CREATE ALTER DROP RENAME TRUNCATE	DDL (Data Definition Language)
COMMIT ROLLBACK SAVEPOINT	Controle de transação
GRANT REVOKE	DCL (Data Control Language)

Instruções SQL

O Oracle SQL é compatível com os padrões aceitos pela indústria. A Oracle Corporation garante a compatibilidade futura com padrões em desenvolvimento envolvendo ativamente uma equipe-chave nos comitês de padrões SQL. Os comitês aceitos pela indústria são o ANSI (American National Standards Institute) e o ISO (International Standards Organization). O ANSI e o ISO aceitaram o SQL como a linguagem padrão para os bancos de dados relacionais.

Instrução	Descrição
SELECT	Recupera dados do banco de dados
INSERT UPDATE DELETE	Informa novas linhas, altera linhas existentes e remove linhas indesejáveis de tabelas do banco de dados, respectivamente. O conjunto dessas instruções é conhecido como DML (<i>Data Manipulation Language</i>).
CREATE ALTER DROP RENAME TRUNCATE	Configura, altera e remove estruturas de dados de tabelas. O conjunto dessas instruções é conhecido como DDL (<i>Data Definition Language</i>).
COMMIT ROLLBACK SAVEPOINT	Gerencia as alterações feitas pelas instruções DML. É possível agrupar as alterações dos dados em transações lógicas.
GRANT REVOKE	Fornece ou remove direitos de acesso ao banco de dados Oracle e às estruturas contidas nele. O conjunto dessas instruções é conhecido como DCL (<i>Data Control Language</i>).

Sobre PL/SQL

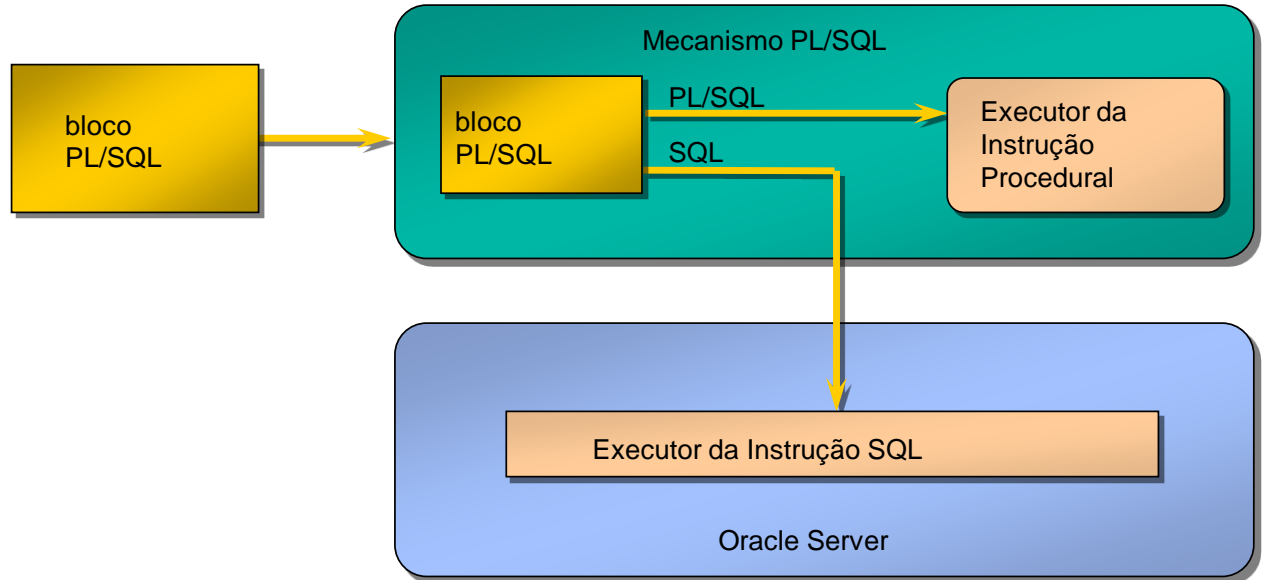
- O PL/SQL é uma extensão do SQL com recursos de design de linguagens de programação.
- As instruções de consulta e a manipulação de dados do SQL estão incluídas nas unidades procedurais de código.

Sobre PL/SQL

O **PL/SQL** (Procedural Language/SQL) é uma extensão de linguagem procedural da Oracle Corporation para SQL, a linguagem de acesso a dados padrão para bancos de dados relacionais. O PL/SQL oferece recursos de engenharia de software modernos, como, por exemplo, a encapsulação de dados, o tratamento de exceções, a ocultação de informações, a orientação de objeto e assim por diante, trazendo os recursos de programação mais modernos para o Oracle Server e o Toolset.

O PL/SQL incorpora muitos recursos avançados feitos em linguagens de programação projetadas durante as décadas de 70 e 80. Além de aceitar a manipulação de dados, ele também permite que instruções de consulta do SQL sejam incluídas em unidades procedurais de código e estruturadas em blocos, tornando o PL/SQL uma linguagem de processamento de transações poderosa. Com o PL/SQL, você pode usar as instruções SQL para refinar os dados da Oracle e as instruções de controle do PL/SQL para processar os dados.

Ambiente PL/SQL



Mecanismo PL/SQL e Oracle Server

O PL/SQL não é um produto Oracle em si; ele é uma tecnologia empregada pelo Oracle Server e por algumas ferramentas Oracle. Os blocos PL/SQL são passados e processados por um mecanismo PL/SQL, que pode residir na ferramenta ou no Oracle Server. O **mecanismo** usado depende do local no qual o PL/SQL é chamado.

O mecanismo PL/SQL no Oracle Server processa os blocos PL/SQL submetidos de um programa de saída de usuário, Pro*, SQL*Plus ou Server Manager. Ele separa as instruções SQL e as envia individualmente para o executor da instrução SQL.

Uma única transferência é necessária para enviar o bloco da aplicação para o Oracle Server, melhorando o desempenho, especialmente em uma rede cliente-servidor. Também é possível armazenar o PL/SQL no Oracle Server sob a forma de subprogramas que podem ser consultados pelas aplicações conectadas ao banco de dados.

Tabelas Usadas no Curso

EMP

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		17-NOV-81	5000		10
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7900	JAMES	CLERK	7698	03-DEC-81	950		30

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

GRADE	LOSAL	HISAL
1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000
5	3001	9999

DEPT

SALGRADE

Tabelas Usadas no Curso

- As três tabelas a seguir serão usadas neste curso:
- Tabela EMP, que fornece detalhes sobre todos os funcionários
 - Tabela DEPT, que fornece detalhes sobre todos os departamentos
 - Tabela SALGRADE, que fornece detalhes sobre salários de várias classes
- A estrutura e os dados de todas as tabelas são fornecidos no Apêndice B.

Criando Instruções SQL Básicas

Recursos das Instruções SELECT SQL

Seleção

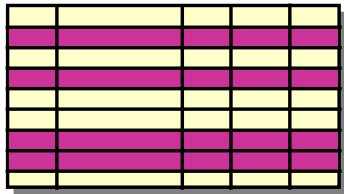


Tabela 1

Projeção

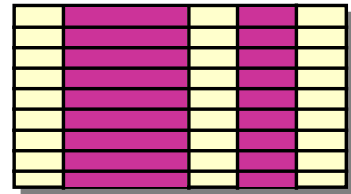


Tabela 1

Junção

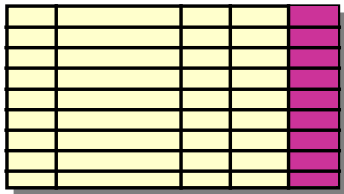


Tabela 1

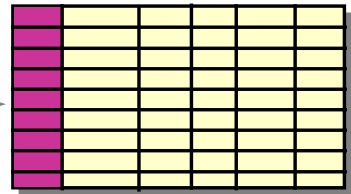


Tabela 2

Recursos das Instruções SELECT SQL

A instrução **SELECT** recupera informações do banco de dados. Usando uma instrução SELECT, você pode fazer o seguinte:

- Seleção: Você pode usar o recurso de seleção no código SQL para escolher as linhas de uma tabela que deseja ver retornadas por uma consulta. Pode usar vários critérios para restringir seletivamente as linhas que você vê.
- Projeção: Você pode usar o recurso de projeção no código SQL para escolher as colunas de uma tabela que deseja ver retornadas por uma consulta. É possível escolher mais ou menos colunas da tabela conforme sua necessidade.
- Junção: Você pode usar o recurso de junção no código SQL para reunir dados armazenados em tabelas diferentes, criando um vínculo entre eles. Você aprenderá mais sobre junções em uma lição posterior.

Instrução SELECT Básica

```
SELECT    [DISTINCT] {*, coluna [apelido],...}  
FROM      tabela;
```

- SELECT identifica *que* colunas.
- FROM identifica *qual* tabela.

Instrução SELECT Básica

Da forma mais simples, uma instrução **SELECT** deve incluir o seguinte:

Uma cláusula SELECT, que especifica as colunas a serem exibidas

Uma cláusula FROM, que especifica a tabela que contém as colunas listadas na cláusula SELECT

Na sintaxe:

SELECT	é uma lista de uma ou mais colunas
DISTINCT	suprime os itens duplicados
*	seleciona todas as colunas
coluna	seleciona a coluna nomeada
apelido	fornece cabeçalhos diferentes às colunas selecionadas
FROM tabela	especifica a tabela contendo as colunas

Observação: Em todo o curso são usados os termos palavra-chave, cláusula e instrução.

Uma *palavra-chave* refere-se a um elemento SQL individual.

Por exemplo, SELECT e FROM são palavras-chave.

Uma *cláusula* é parte de uma instrução SQL.

Por exemplo, SELECT empno, ename, ... é uma cláusula.

Uma *instrução* é uma combinação de duas ou mais cláusulas.

Por exemplo, SELECT * FROM emp é uma instrução SQL.

Criando Instruções SQL

- Instruções SQL sem distinção entre maiúsculas e minúsculas.
- Instruções SQL podem estar em uma ou mais linhas.
- Palavras-chave não podem ser abreviadas ou divididas entre as linhas.
- Normalmente, as cláusulas são colocadas em linhas separadas.
- Guias e endentações são usadas para aperfeiçoar a legibilidade.

Criando Instruções SQL

Usando as seguintes diretrizes e regras simples, você pode construir instruções válidas fáceis de ler e editar:

- As instruções SQL não fazem distinção entre maiúsculas de minúsculas, a menos que indicado. As instruções SQL podem ser digitadas em uma ou mais linhas.
- As palavras-chave não podem ser divididas entre as linhas nem abreviadas. As cláusulas são em geral colocadas em linhas separadas para melhor legibilidade e facilidade de edição.
- As guias e endentações podem ser usadas para tornar o código mais legível. Em geral, as palavras-chave são digitadas em letras maiúsculas, todas as outras palavras, como nomes de tabela e colunas são digitadas em minúsculas.
- Dentro do SQL*Plus, uma instrução SQL é digitada no prompt SQL e as linhas subsequentes são numeradas. Isso chama-se *buffer de SQL*. Somente uma instrução pode ser a atual a qualquer momento dentro do buffer.

Executando Instruções SQL

Coloque um ponto-e-vírgula (;) no final da última cláusula.
Coloque uma barra na última linha do buffer.
Coloque uma barra no prompt SQL.
Emita um comando RUN do SQL*Plus no prompt SQL.

Selecionando Todas as Colunas

```
SELECT *  
FROM dept;
```

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

Selecting All Columns, All Rows

Exiba todas as colunas de dados em uma tabela seguindo a palavra-chave SELECT com um asterisco (*). No exemplo do slide, a tabela do departamento contém três colunas: DEPTNO, DNAME e LOC. A tabela contém quatro linhas, uma para cada departamento.

É possível, também, exibir todas as colunas na tabela, listando todas elas após a palavra-chave SELECT. Por exemplo, a instrução SQL a seguir, como no exemplo do slide, exibe todas as colunas e linhas da tabela DEPT:

```
SQL> SELECT deptno, dname, loc  
2 FROM dept;
```

Selecionando Colunas Específicas

```
SELECT deptno, loc
FROM dept;
```

DEPTNO	LOC
10	NEW YORK
20	DALLAS
30	CHICAGO
40	BOSTON

Selecionando Colunas Específicas, Todas as Linhas

Você pode usar a instrução SELECT para exibir colunas específicas da tabela, ao especificar os nomes da coluna, separado por vírgulas. O exemplo do slide exibe todos os números dos departamentos e locais na tabela DEPT. Na cláusula SELECT, especifique as colunas a serem vistas, na ordem que deseja que apareçam na saída. Por exemplo, para exibir o local antes do número do departamento, use a seguinte instrução:

```
SQL> SELECT loc, deptno
2 FROM dept;
```

LOC	DEPTNO
NEW YORK	10
DALLAS	20
CHICAGO	30
BOSTON	40

Expressões Aritméticas

- Criar expressões com dados NUMBER e DATE usando operadores aritméticos.

Operador	Descrição
+	Adicionar
-	Subtrair
*	Multiplicar
/	Dividir

Expressões Aritméticas

Talvez você necessite modificar a forma de exibição dos dados, efetuar cálculos ou consultar cenários *what-if*. Isso é possível usando expressões aritméticas. Uma expressão aritmética possui nomes de coluna, valores numéricos constantes e operadores aritméticos.

Operadores Aritméticos

O slide lista os **operadores aritméticos** disponíveis no código SQL. Você pode usar operadores aritméticos em qualquer cláusula de uma instrução SQL exceto na cláusula FROM.

Usando Operadores Aritméticos

```
SELECT ename, sal, sal+300
FROM emp;
```

ENAME	SAL	SAL+300
KING	5000	5300
BLAKE	2850	3150
CLARK	2450	2750
JONES	2975	3275
MARTIN	1250	1550
ALLEN	1600	1900
...		

14 rows selected.

Usando Operadores Aritméticos

O exemplo no slide usa o operador de adição para calcular um aumento de salário de US\$300 para todos os funcionários e exibe uma nova coluna SAL+300 na saída.

Observe que a coluna SAL+300 resultante do cálculo não é uma nova coluna na tabela EMP; ela é somente para exibição. Por default, o nome de uma coluna surge do cálculo que a criou — nesse caso, sal+300.

Observação: O SQL*Plus ignora espaços em branco antes e depois do operador aritmético.

Precedência do Operador



- A multiplicação e a divisão têm prioridade sobre a adição e a subtração.
- Os operadores com a mesma prioridade são avaliados da esquerda para a direita.
- Os parênteses são usados para forçar a avaliação priorizada e para esclarecer as instruções.

Precedência do Operador

Se uma expressão aritmética tiver mais de um operador, a multiplicação e a divisão serão avaliadas primeiro. Se os operadores dentro uma expressão tiverem a mesma prioridade, então a avaliação será realizada da esquerda para a direita.

Você pode usar os parênteses a fim de forçar a avaliação da expressão entre parênteses primeiro.

Precedência do Operador

```
SELECT ename, sal, 12*sal+100
FROM emp;
```

ENAME	SAL	12*SAL+100
-----	-----	-----
KING	5000	60100
BLAKE	2850	34300
CLARK	2450	29500
JONES	2975	35800
MARTIN	1250	15100
ALLEN	1600	19300
...		

14 rows selected.

Precedência do Operador (continuação)

O exemplo no slide exibe o nome, o salário e a remuneração anual dos funcionários. Ele calcula a remuneração anual como 12 multiplicado pelo salário mensal, mais um bônus de US\$100. Observe que a multiplicação é realizada antes da adição.

Observação: Use os parênteses para reforçar a **ordem de precedência** padrão e aumentar a compreensão. Por exemplo, a expressão acima pode ser criada como $(12*sal)+100$ sem que haja alteração no resultado.

Usando Parênteses

```
SELECT ename, sal, 12*(sal+100)
FROM emp;
```

ENAME	SAL	12*(SAL+100)
-----	-----	-----
KING	5000	61200
BLAKE	2850	35400
CLARK	2450	30600
JONES	2975	36900
MARTIN	1250	16200
...		

14 rows selected.

Usando Parênteses

Você pode sobrepor as normas de precedência usando *parênteses* para especificar a ordem de execução dos operadores.

O exemplo no slide exibe o nome, o salário e a remuneração anual dos funcionários. Ele calcula a remuneração anual como o salário mensal mais um bônus mensal de US\$100, multiplicado por 12. Por causa dos parâmetros, a adição tem prioridade sobre a multiplicação.

Definindo um Valor Nulo

- Um valor nulo não está disponível, não é atribuído, é desconhecido ou não é aplicável.
- Um valor nulo não é o mesmo que um zero ou um espaço em branco.

```
SELECT ename, job, sal, comm
FROM emp;
```

ENAME	JOB	SAL	COMM
KING	PRESIDENT	5000	
BLAKE	MANAGER	2850	
...			
TURNER	SALESMAN	1500	0
...			

14 rows selected.

Valores Nulos

Se faltar o valor de dados em uma linha de uma determinada coluna, diz-se que esse valor é *nulo* ou contém nulo.

Um valor nulo não está disponível, não é atribuído, é desconhecido ou não é aplicável. Um valor nulo não é o mesmo que um zero ou um espaço. O zero é um número e o espaço é um caractere.

As colunas de qualquer tipo de dados podem conter valores nulos, a menos que tenham sido definidas como NOT NULL ou como PRIMARY KEY ao serem criadas.

Na coluna COMM da tabela EMP, note que somente o SALESMAN pode ganhar comissão. Outros funcionários não estão autorizados a ganhar comissão. Um valor nulo representa esse fato. Turner, que é um vendedor, não ganha nenhuma comissão. Observe que sua comissão é zero e não nula.

Valores Nulos nas Expressões Aritméticas

- Expressões aritméticas contendo um valor nulo avaliado como nulo.

```
SELECT  ename, 12*sal+comm  
FROM    emp  
WHERE   ename='KING';
```

ENAME	12*SAL+COMM
-----	-----
KING	

Valores Nulos (continuação)

Se qualquer valor da coluna em uma expressão aritmética for nulo, o resultado será nulo. Por exemplo, se você tentar executar uma divisão com zero, obterá um erro. No entanto, se dividir um número por nulo, o resultado será nulo ou desconhecido.

No exemplo do slide, o funcionário KING não está em SALESMAN e não receberá nenhuma comissão. Como a coluna COMM na expressão aritmética é nula, o resultado será nulo.

Para obter mais informações, consulte o *Oracle Server SQL Reference*, "Elements of SQL".

https://docs.oracle.com/cd/B19306_01/server.102/b14200/toc.htm

Definindo um Apelido de Coluna

- Renomeia um cabeçalho de coluna
- É útil para cálculos
- Segue imediatamente o nome da coluna; palavra-chave AS opcional entre o nome da coluna e o apelido
- Necessita de aspas duplas caso contenha espaços ou caracteres especiais ou faça distinção entre maiúsculas e minúsculas

Apelidos de Coluna

Ao exibir o resultado de uma consulta, o SQL*Plus normalmente usa o nome da coluna selecionada como o cabeçalho da mesma. Em muitos casos, esse cabeçalho pode não ser descritivo e, desse modo, de difícil compreensão. É possível alterar um cabeçalho de coluna usando um apelido da coluna. Especifique o **apelido** após a coluna na lista SELECT usando um espaço como um separador. Por default, os cabeçalhos de apelidos aparecem em letras maiúsculas. Se o apelido possuir espaços, caracteres especiais (tais como # ou \$) ou fizer distinção entre maiúsculas e minúsculas, coloque o apelido entre aspas duplas (" ").

Usando Apelidos de Coluna

```
SELECT ename AS name, sal salary
FROM emp;
```

NAME	SALARY

...	

```
SELECT ename "Name",
       sal*12 "Annual Salary"
FROM emp;
```

Name	Annual Salary

...	

Apelidos de Coluna (continuação)

O primeiro exemplo exibe o nome e o salário mensal de todos os funcionários. Note que a palavra-chave AS opcional foi usada antes do nome do apelido de coluna. O resultado da consulta deverá ser o mesmo caso a palavra-chave **AS** seja usada ou não. Note também que a instrução SQL possui os apelidos de coluna, nome e salário, em letra minúscula, enquanto o resultado da consulta exibe os cabeçalhos da coluna em maiúscula. Conforme mencionado no último slide, os cabeçalhos de coluna aparecem, por default, em maiúscula.

O segundo exemplo exibe o nome e o salário anual de todos os funcionários. Como o Annual Salary possui espaços, ele foi incluído entre aspas duplas. Note que o cabeçalho da coluna na saída é exatamente o mesmo do apelido da coluna.

Operador de Concatenação

- Concatena colunas ou strings de caractere a outras colunas
- É representado por duas barras verticais (| |)
- Cria uma coluna resultante que é uma expressão de caracteres

Operador de Concatenação

Você pode vincular colunas à outras colunas, expressões aritméticas ou valores constantes usando o **operador de concatenação (| |)**. As colunas em cada lado do operador são combinadas para formar uma coluna de saída única.

Usando um Operador de Concatenação

```
SELECT      ename||job AS "Employees"  
FROM        emp;
```

```
Employees  
-----  
KINGPRESIDENT  
BLAKEMANAGER  
CLARKMANAGER  
JONESMANAGER  
MARTINSALESMAN  
ALLENSALESMAN  
...  
14 rows selected.
```

Operador de Concatenação (continuação)

No exemplo, ENAME e JOB estão concatenados e recebem o apelido Employees. Note que o número e cargo do funcionário são combinados para formar uma coluna de saída única.

A palavra-chave AS antes do nome do apelido torna a cláusula SELECT mais fácil de ser lida.

Strings Literais de Caracteres

- Uma literal é um caractere, um número ou uma data incluída na lista SELECT.
- Os valores literais de caractere e data devem estar entre aspas simples.
- Cada string de caractere é gerada uma vez para cada linha retornada.

Strings Literais de Caracteres

Uma **literal** é um caractere, um número ou uma data incluída na lista SELECT que não seja um nome ou apelido de coluna. Ela é impressa para cada linha retornada. Strings literais de formato de texto livre podem ser incluídas no resultado da consulta e são tratadas da mesma forma que uma coluna na lista SELECT.

As literais de caractere e data *precisam* estar entre aspas simples (' '); as literais de número, não.

Usando Strings Literais de Caracteres

```
SELECT ename || ' is a ' || job
        AS "Employee Details"
FROM    emp;
```

```
Employee Details
-----
KING is a PRESIDENT
BLAKE is a MANAGER
CLARK is a MANAGER
JONES is a MANAGER
MARTIN is a SALESMAN
...
14 rows selected.
```

Strings Literais de Caracteres (continuação)

O exemplo do slide exibe os nomes e cargos de todos os funcionários. A coluna possui o cabeçalho Employee Details. Note os espaços entre as aspas simples na instrução SELECT. Os espaços melhoram a legibilidade da saída. No exemplo a seguir, o nome e o salário de cada funcionário estão concatenados a uma literal para dar mais sentido às linhas retornadas.

```
SQL> SELECT ename || ': ' || '1' || ' Month salary = ' || sal Monthly
2 FROM    emp;
```

```
MONTHLY
```

```
-----
KING: 1 Month salary = 5000
BLAKE: 1 Month salary = 2850
CLARK: 1 Month salary = 2450
JONES: 1 Month salary = 2975
MARTIN: 1 Month salary = 1250
ALLEN: 1 Month salary = 1600
TURNER: 1 Month salary = 1500
...
14 rows selected.
```

Linhas Duplicadas

- A exibição default das consultas é de todas as linhas, incluindo linhas duplicadas.

```
SELECT deptno  
FROM emp;
```

```
DEPTNO  
-----  
      10  
      30  
      10  
      20  
...  
14 rows selected.
```

Linhas Duplicadas

Exceto se indicado o contrário, o SQL*Plus exibe os resultados de uma consulta sem eliminar as linhas duplicadas. O exemplo do slide exibe todos os números de departamento a partir da tabela EMP. Note que os números de departamento estão repetidos.

Eliminando Linhas Duplicadas

Elimine linhas duplicadas usando a palavra-chave **DISTINCT** na cláusula **SELECT**.

```
SELECT DISTINCT deptno
FROM emp;
```

DEPTNO
10
20
30

Linhas Duplicadas (continuação)

Para eliminar linhas duplicadas de um resultado, inclua a palavra-chave **DISTINCT** na cláusula **SELECT** logo após a palavra-chave **SELECT**. No exemplo do slide, a tabela **EMP** contém, na verdade, quatorze linhas, mas há somente três números de departamento exclusivos na tabela.

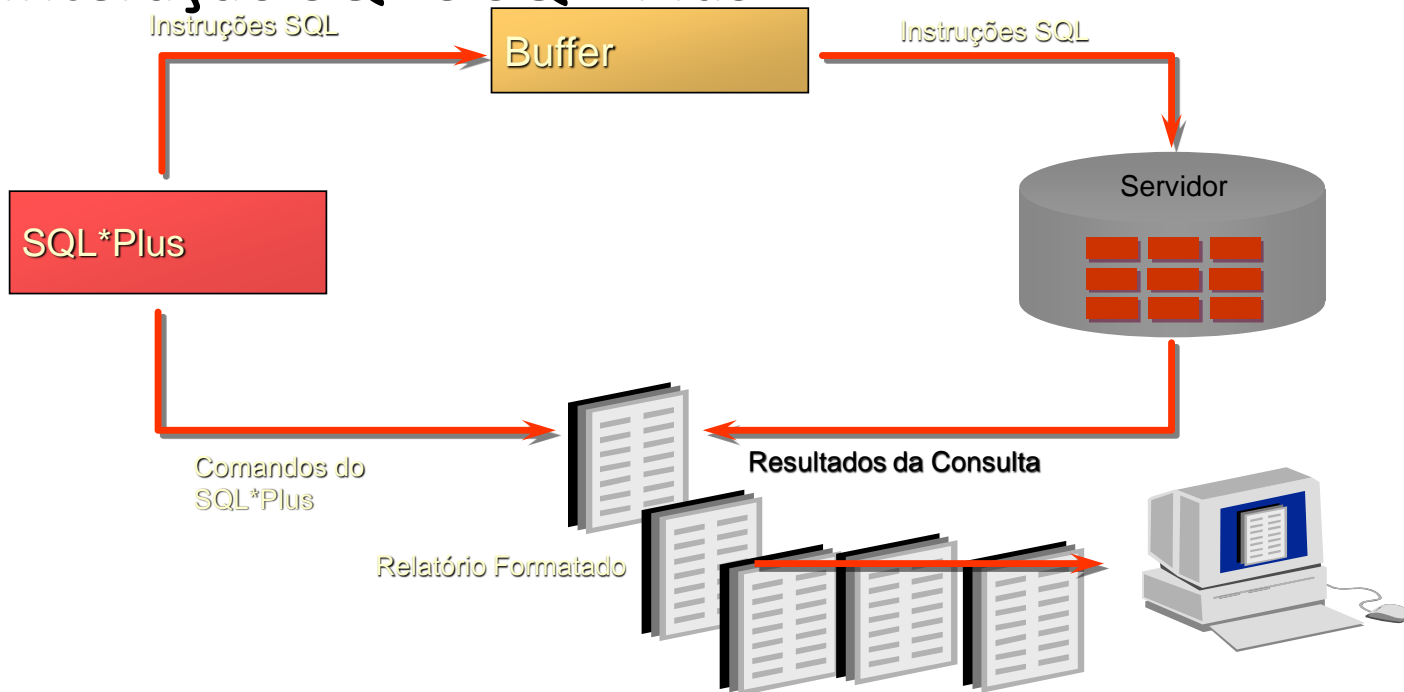
Você pode especificar várias colunas após o qualificador **DISTINCT**. O qualificador **DISTINCT** afeta todas as colunas selecionadas e o resultado representa uma combinação distinta das colunas.

```
SQL> SELECT DISTINCT deptno, job
2 FROM emp;
```

DEPTNO	JOB
10	CLERK
10	MANAGER
10	PRESIDENT
20	ANALYST
...	

9 rows selected.

Interação SQL e SQL*Plus



SQL e SQL*Plus

SQL é uma linguagem de comando para comunicação com o Oracle Server a partir de qualquer ferramenta ou aplicação. O Oracle SQL possui muitas extensões. Quando você informa uma instrução SQL, ela é armazenada em uma parte da memória chamada *buffer de SQL* e permanece lá até que você informe uma nova instrução. O **SQL*Plus** é uma ferramenta Oracle que reconhece e submete instruções SQL ao Oracle Server para execução e contém sua própria linguagem de comando.

Recursos do Código SQL

- Podem ser utilizados por uma grande faixa de usuários, incluindo aqueles com pouca ou nenhuma experiência em programação
- É uma linguagem não procedural
- Reduz o período de tempo necessário para a criação e manutenção de sistemas
- É uma linguagem similar ao inglês

Recursos do SQL*Plus

- Aceita entrada ad hoc das instruções
- Aceita entrada SQL a partir dos arquivos
- Oferece um editor de linha para modificar instruções SQL
- Controla as configurações ambientais
- Formata os resultados da consulta em um relatório básico
- Acessa bancos de dados remotos e locais

Visão Geral do SQL*Plus

- Estabelecer login no SQL*Plus.
- Descrever a estrutura de tabela.
- Editar a instrução SQL.
- Executar o código SQL a partir do SQL*Plus.
- Salvar as instruções SQL em arquivos e anexar as instruções SQL a arquivos.
- Executar arquivos salvos.
- Carregar comandos a partir de arquivo para buffer e editá-los.

SQL*Plus

SQL*Plus é um ambiente no qual você pode realizar o seguinte:

Executar instruções SQL para recuperar, modificar, adicionar e remover dados do banco de dados.

Formatar, calcular, armazenar e imprimir resultados de consulta em formulários.

Criar arquivos de script para armazenar instruções SQL para uso repetitivo no futuro.

Os comandos SQL*Plus podem ser divididos nas seguintes categorias principais:

Categoria	Objetivo
Ambiente	Afeta o comportamento geral das instruções SQL para a seção.
Formato	Formata o resultado da consulta.
Manipulação de arquivo	Salva, carrega e executa arquivos de script.
Execução	Envia instruções SQL do buffer de SQL para o Oracle8 Server.
Editar	Modifica as instruções SQL no buffer.
Interação	Permite criar e passar variáveis para instruções SQL, imprimir valores de variáveis e imprimir mensagens na tela.
Diversos	Possui diversos comandos para conectar o banco de dados, manipular o ambiente SQL*Plus e exibir definições de coluna.

Exibindo a Estrutura de Tabela

- Use o comando DESCRIBE do SQL*Plus para exibir a estrutura de uma tabela.

```
DESC[RIBE] nome da tabela
```

Exibindo a Estrutura de Tabela

No SQL*Plus, é possível exibir a estrutura de uma tabela usando o comando **DESCRIBE**. O resultado do comando é para ver os nomes da coluna e tipos de dados, assim como se uma coluna *deve* conter dados.

Na sintaxe:

nome da tabela é o nome de qualquer tabela, view ou sinônimo existente disponível para o usuário

Exibindo a Estrutura de Tabela

```
SQL> DESCRIBE dept
```

Name	Null?	Type
DEPTNO	NOT NULL	NUMBER (2)
DNAME		VARCHAR2 (14)
LOC		VARCHAR2 (13)

Exibindo a Estrutura de Tabela (continuação)

O exemplo do slide exibe as informações sobre a estrutura da tabela DEPT.

No resultado:

Null? indica se uma coluna deve conter dados; NOT NULL indica que uma coluna deve conter dados

Type exibe o **tipo de dados** de uma coluna

Os tipos de dados são descritos na tabela a seguir:

Tipo de dado	Descrição
NUMBER(<i>p,s</i>)	Valor numérico que possui um número máximo de dígitos <i>p</i> , o número de dígitos à direita do ponto decimal <i>s</i> .
VARCHAR2(<i>s</i>)	Valor de caracteres com comprimento variável do tamanho máximo <i>s</i> .
DATE	Valor de data e hora entre 1 de janeiro, 4712 A.C. e 31 de dezembro, 9999 D.C.
CHAR(<i>s</i>)	Valores de caracteres com comprimento fixo do tamanho <i>s</i> .

Sumário

```
SELECT    [DISTINCT] {*,coluna [apelido],...}  
FROM      tabela;
```

- Use o SQL*Plus como um ambiente para:
 - Executar instruções SQL
 - Editar instruções SQL

Instrução SELECT

Nesta lição, você aprendeu sobre a recuperação de dados de uma tabela de banco de dados com a instrução SELECT.

```
SELECT    [DISTINCT] {*,coluna [apelido],...}  
FROM      tabela;
```

onde:	SELECT	é uma lista de pelo menos uma coluna.
	DISTINCT	suprime as duplicatas.
	*	seleciona todas as colunas.
	coluna	seleciona a coluna nomeada.
	apelido	dá um cabeçalho diferente à coluna selecionada.
	FROM tabela	especifica a tabela contendo as colunas.

SQL*Plus

SQL*Plus é um ambiente de execução que pode ser usado para enviar instruções SQL ao servidor do banco de dados, editar e salvar as instruções SQL. As instruções podem ser executadas a partir do prompt SQL ou de um arquivo de script.

Visão Geral do Exercício

- Selecionando todos os dados a partir de tabelas diferentes
- Descrevendo a estrutura de tabelas
- Executando cálculos aritméticos e especificando nomes de coluna
- Usando o editor do SQL*Plus



Visão Geral do Exercício

Esse é o primeiro de muitos exercícios. As questões 1, 2 e 3 são teóricas. Realize o exercício devagar e com exatidão. Salve os comandos executados para referencias futuras.

Questões teóricas

Para as questões de 1, 2 e 3, marque Falso ou Verdadeiro.

Exercício 1

1. A instrução SELECT será executada corretamente?

Verdadeiro/Falso

```
SELECT  ename, job, sal Salary
FROM    emp;
```

2. A instrução SELECT será executada corretamente?

Verdadeiro/Falso

```
SELECT  *
FROM    salgrade;
```

3. Há quatro erros de codificação nesta instrução. Você pode identificá-los?

```
SELECT  empno, ename
        salary x 12 ANNUAL SALARY
FROM    emp;
```

4. Selecione todos os dados da tabela DEPT.

DEPTNO	DNAME	LOC
-----	-----	-----
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

Exercício 1 (continuação)

5. Crie uma consulta para exibir o nome (ENAME), o cargo (JOB), a data de admissão (HIREDATE) e o número do funcionário (EMPNO) para cada funcionário, com o número do funcionário aparecendo primeiro.

Exemplo da saída esperada pela consulta:

EMPNO	ENAME	JOB	HIREDATE
-----	-----	-----	-----
7839	KING	PRESIDENT	17-NOV-81
7698	BLAKE	MANAGER	01-MAY-81
7782	CLARK	MANAGER	09-JUN-81
7566	JONES	MANAGER	02-APR-81
7654	MARTIN	SALESMAN	28-SEP-81
7499	ALLEN	SALESMAN	20-FEB-81
7844	TURNER	SALESMAN	08-SEP-81
7900	JAMES	CLERK	03-DEC-81
7521	WARD	SALESMAN	22-FEB-81
7902	FORD	ANALYST	03-DEC-81
7369	SMITH	CLERK	17-DEC-80
7788	SCOTT	ANALYST	09-DEC-82
7876	ADAMS	CLERK	12-JAN-83
7934	MILLER	CLERK	23-JAN-82

14 rows selected.

Exercício 1 (continuação)

6. Crie uma consulta para exibir os cargos exclusivos (sem repetição) a partir da tabela EMP.

Exemplo da saída esperada pela consulta:

```
JOB
-----
ANALYST
CLERK
MANAGER
PRESIDENT
SALESMAN
```

7. Crie uma consulta para exibir o nome (ENAME), o cargo (JOB), a data de admissão (HIREDATE) e o número do funcionário (EMPNO) para cada funcionário, com o número do funcionário aparecendo primeiro. Nomeie os cabeçalhos das colunas como Emp #, Employee, Job, e Hire Date, respectivamente.

Exemplo da saída esperada pela consulta:

Emp #	Employee	Job	Hire Date
7839	KING	PRESIDENT	17-NOV-81
7698	BLAKE	MANAGER	01-MAY-81
7782	CLARK	MANAGER	09-JUN-81
7566	JONES	MANAGER	02-APR-81
7654	MARTIN	SALESMAN	28-SEP-81
7499	ALLEN	SALESMAN	20-FEB-81
7844	TURNER	SALESMAN	08-SEP-81
7900	JAMES	CLERK	03-DEC-81
7521	WARD	SALESMAN	22-FEB-81
7902	FORD	ANALYST	03-DEC-81
7369	SMITH	CLERK	17-DEC-80
7788	SCOTT	ANALYST	09-DEC-82
7876	ADAMS	CLERK	12-JAN-83
7934	MILLER	CLERK	23-JAN-82

14 rows selected.

Exercício 1 (continuação)

8. Exiba o nome concatenado com o cargo (job), separado por uma vírgula e espaço, e nomeie a coluna Employee and Title.

Exemplo da saída esperada pela consulta:

```
Employee and Title
-----
KING, PRESIDENT
BLAKE, MANAGER
CLARK, MANAGER
JONES, MANAGER
MARTIN, SALESMAN
ALLEN, SALESMAN
TURNER, SALESMAN
JAMES, CLERK
WARD, SALESMAN
FORD, ANALYST
SMITH, CLERK
SCOTT, ANALYST
ADAMS, CLERK
MILLER, CLERK
```

9. Crie uma consulta para exibir todos os dados a partir da tabela EMP. Separe cada coluna por uma vírgula. Nomeie a coluna como THE_OUTPUT.

Exemplo da saída esperada pela consulta:

```
THE_OUTPUT
-----
7839,KING,PRESIDENT,,17-NOV-81,5000,,10
7698,BLAKE,MANAGER,7839,01-MAY-81,2850,,30
7782,CLARK,MANAGER,7839,09-JUN-81,2450,,10
7566,JONES,MANAGER,7839,02-APR-81,2975,,20
7654,MARTIN,SALESMAN,7698,28-SEP-81,1250,1400,30
7499,ALLEN,SALESMAN,7698,20-FEB-81,1600,300,30
7844,TURNER,SALESMAN,7698,08-SEP-81,1500,0,30
7900,JAMES,CLERK,7698,03-DEC-81,950,,30
7521,WARD,SALESMAN,7698,22-FEB-81,1250,500,30
7902,FORD,ANALYST,7566,03-DEC-81,3000,,20
7369,SMITH,CLERK,7902,17-DEC-80,800,,20
7788,SCOTT,ANALYST,7566,09-DEC-82,3000,,20
7876,ADAMS,CLERK,7788,12-JAN-83,1100,,20
7934,MILLER,CLERK,7782,23-JAN-82,1300,,10
```