

1

Comando SELECT

Objetivos deste Capítulo

- Ao concluir este capítulo, você poderá:
 - Retornar informações de uma tabela usando o comando `SELECT`

1-2

Objetivos deste Capítulo

Este capítulo descreve como retornar informações de uma tabela usando o comando `SELECT`. Serão abordados os conceitos básicos desse comando.

Comando SELECT

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

- SELECT identifica as colunas que devem ser exibidas
- FROM identifica a tabela que contém essas colunas

1-3

Comando SELECT

Um comando SELECT deve conter no mínimo:

- Cláusula SELECT: especifica as colunas que devem ser exibidas
- Cláusula FROM: identifica a tabela que contém as colunas listadas na cláusula SELECT.

Na sintaxe:

SELECT	cláusula que indica início da lista de colunas retornadas
*	indica seleção de todas as colunas da tabela
DISTINCT	não exibe resultados duplicados
<i>coluna</i> <i>expressao</i>	seleciona a coluna especificada ou uma expressão
<i>apelido</i>	especifica um apelido para a coluna selecionada
FROM <i>tabela</i>	indica a tabela que deverá ser consultada

O comando SQL é finalizado no SQL*Plus com um sinal de ponto-e-vírgula (;)
Os comandos SELECT são comumente referidos como CONSULTA ou em inglês QUERY.

O usuário informa para o Oracle a maneira como ele deve processar o comando, apenas o que se deseja retornar, o caminho de acesso aos dados e a maneira como o Oracle processará o comando é definido automaticamente.

Comandos SELECT não fazem distinção entre maiúsculas e minúsculas e palavras-chave não podem ser abreviadas ou divididas em mais de uma linha

Selecionando Todos os Dados de uma Tabela

```
SQL> SELECT *  
2 FROM departamento;
```

COD_DEPARTAMENTO	NOME_DEPARTAMENTO	COD_GERENTE	COD_LOCALIDADE
10	Administração	200	1700
20	Marketing	201	1800
50	Logística	124	1500
60	Informática	103	1400
80	Vendas	149	2500
90	Executivo	100	1700
110	Contabilidade	205	1700
190	Contratos		1700

8 linhas selecionadas.

1-4

Selecionando Todos os Dados de uma Tabela

Para exibir todas as colunas e todas as linhas de uma tabela, você pode inserir um asterisco (*) após a palavra-chave `SELECT`. No exemplo do slide, a tabela `departamento` contém quatro colunas: `COD_DEPARTAMENTO`, `NOME_DEPARTAMENTO`, `COD_GERENTE` e `COD_LOCALIDADE`. A tabela contém oito linhas no total, uma para cada departamento.

Você também pode exibir todas as colunas da tabela listando-as após a palavra-chave `SELECT`. Por exemplo, o comando SQL a seguir (como o exemplo do slide) exibe todas as colunas e linhas da tabela

`DEPARTAMENTO`:

```
SELECT cod_departamento,  
       nome_departamento,  
       cod_gerente,  
       cod_localidade  
FROM   departamento;
```

Uma boa prática de programação de banco de dados é sempre explicitar todas as colunas que deseja-se retornar no comando `SELECT`, e só se deve retornar as colunas estritamente necessárias num determinado momento, pois isso reduz o tráfego de informações na rede e evita erros de aplicação caso a estrutura de uma tabela seja alterada no futuro. Por isso deve-se evitar a construção de comandos `SELECT` usando o asterisco (*) como lista de colunas.

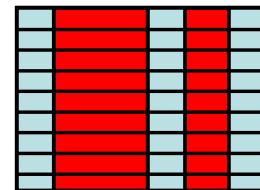
Selecionando Colunas Específicas

```
SQL> SELECT cod_departamento,  
2      nome_departamento  
3 FROM departamento;
```

```
COD_DEPARTAMENTO NOME_DEPARTAMENTO  
-----  
10 Administração  
20 Marketing  
50 Logística  
60 Informática  
80 Vendas  
90 Executivo  
110 Contabilidade  
190 Contratos
```

8 linhas selecionadas.

Projeção



1-5

Selecionando Colunas Específicas de uma Tabela

O comando SELECT pode ser utilizado para implementar a característica da álgebra relacional chamada de Projeção, onde apenas algumas colunas de uma tabela serão retornadas. Os nomes das colunas devem estar na cláusula SELECT e separadas por vírgulas.

O exemplo do slide exibe todos os códigos e nomes dos departamentos da tabela DEPARTAMENTO.

A ordem de listagem das colunas é a ordem que será exibida no resultado do comando.

Para exibir o nome do departamento antes do código, o exemplo do slide deve ser alterado da seguinte forma:

```
SELECT nome_departamento, cod_departamento  
FROM departamento;
```

NOME_DEPARTAMENTO	COD_DEPARTAMENTO
Administração	10
Marketing	20
Logística	50
Informática	60
Vendas	80
Executivo	90
Contabilidade	110
Contratos	190

Padrões de Cabeçalho e Alinhamento

- SQL*Plus:
 - Os cabeçalhos e o resultado das colunas tipo caractere ou data são alinhados à esquerda
 - Os cabeçalhos e o resultado das colunas tipo numérico são alinhados à direita
 - Se não for especificado um apelido entre aspas duplas (“) a exibição do nome da coluna no cabeçalho se dará com letras maiúsculas

1-6

Padrões de Cabeçalho e Alinhamento

No SQL*Plus, os cabeçalhos e o resultado das colunas tipo caractere ou data são alinhados à esquerda, as colunas tipo numérico são alinhadas à direita.

```
SQL> SELECT cod_funcionario, nome, data_admissao  
2 FROM funcionario;
```

COD_FUNCIONARIO	NOME	DATA_ADMISSAO
100	Roberto	17/06/1997
101	Nair	21/09/1999
102	Leonardo	13/01/2003
103	Alexandre	03/01/2000
104	Pedro	21/05/2001
...		
201	William	17/02/2006
202	Cosmo	17/08/2007
205	Sheila	07/06/2004
206	Roberto	07/06/2004

20 linhas selecionadas.

Por padrão, o nome das colunas que é exibido no cabeçalho do resultado aparece com letras maiúsculas.

Uso de Expressões Aritméticas

- Use expressões aritméticas nos comandos SQL.
- Uma expressão aritmética consiste de dois operandos (numérico ou data) e um operador aritmético.
- Os 4 operadores aritméticos são os seguintes:

Operador	Descrição
+	Somar
-	Subtrair
*	Multiplicar
/	Dividir

1-7

Uso de Expressões Aritméticas

Talvez você precise modificar a forma como os dados são exibidos, realizar cálculos ou examinar cenários hipotéticos. Todas essas ações são possíveis com o uso de expressões aritméticas. Uma expressão aritmética pode conter nomes de colunas, valores numéricos constantes e operadores aritméticos.

Operadores Aritméticos

O slide lista os 4 operadores aritméticos disponíveis na linguagem SQL. Você pode usar operadores aritméticos em qualquer cláusula de uma instrução SQL (exceto na cláusula `FROM`).

Nos tipos de dados `DATE` e `TIMESTAMP`, só é possível usar os operadores de adição e subtração.

Uso de Operadores Aritméticos

```
SQL> SELECT nome, salario, salario + 500  
2 FROM funcionario;
```

NOME	SALARIO	SALARIO+500
Roberto	24000	24500
Nair	17000	17500
Leonardo	17000	17500
Alexandre	9000	9500
...		
Elias	8600	9100
Sadako	7000	7500
Luciana	4400	4900
William	13000	13500
Cosmo	6000	6500
Sheila	12000	12500
Roberto	8300	8800

20 linhas selecionadas.

1-8

Uso de Operadores Aritméticos

No exemplo do slide o operador de adição é usado para calcular um aumento de salário de R\$ 500,00 para os funcionários, exibindo uma coluna `SALARIO+500` na saída.

A coluna `SALARIO+500` resultante da soma não é uma nova coluna da tabela `FUNCIONARIO`, ela existe apenas no resultado da consulta. Por padrão, o nome de uma nova coluna origina-se do cálculo que a gerou, neste caso, `salario+500`.

O Oracle ignora os espaços em branco antes e depois do operador aritmético.

Precedência de Operadores

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

Você pode usar parênteses para impor a avaliação da expressão entre parênteses primeiro e tornar a instrução mais clara.

Precedência de Operadores

```
SELECT nome,  
       salario,  
       13*salario+500  
FROM   funcionario;
```

NOME	SALARIO	13*SALARIO+500
Roberto	24000	312500
Nair	17000	221500
Leonardo	17000	221500
Alexandre	9000	117500
Pedro	6000	78500
Dilma	4200	55100
Manuela	5800	75900
Tomé	3500	46000
Cristóvão	3100	40800
Rafael	2600	34300

```
SELECT nome,  
       salario,  
       13*(salario+500)  
FROM   funcionario;
```

NOME	SALARIO	13*(SALARIO+500)
Roberto	24000	318500
Nair	17000	227500
Leonardo	17000	227500
Alexandre	9000	123500
Pedro	6000	84500
Dilma	4200	61100
Manuela	5800	81900
Tomé	3500	52000
Cristóvão	3100	46800
Rafael	2600	40300

1-9

Precedência de Operadores (continuação)

O primeiro exemplo do slide exibe o nome, o salário e a remuneração anual dos funcionários. Para calcular a remuneração anual, ele multiplica o salário mensal por 13 e acrescenta um bônus único de R\$ 500,00. Observe que a multiplicação é executada antes da adição.

O parênteses reforça a ordem padrão de precedência e torna a expressão mais clara. Por exemplo, a expressão do slide pode ter a forma $(13*\text{salario})+500$ sem alteração no resultado final.

Usando Parênteses

Você pode sobrepor as regras de precedência usando parênteses para especificar a ordem na qual os operadores devem ser executados.

O segundo exemplo do slide exibe o nome, o salário e a remuneração anual dos funcionários. Ele calcula a remuneração anual da seguinte forma: adiciona um bônus de R\$ 500,00 ao salário mensal e ao décimo terceiro salário e multiplica esse subtotal por 13. Por conta dos parênteses, a adição passou a ter prioridade sobre a multiplicação.

Valores Nulo

- Uma coluna pode conter o valor Nulo.
- O valor Nulo é caracterizado pela ausência de valor. A ausência de valor pode ocorrer porque o valor não é conhecido ou porque o valor não é aplicável
- Um valor nulo é diferente de zero ou de um espaço em branco.

```
SELECT nome, cod_cargo, comissao  
FROM funcionario;
```

NOME	COD_CARGO	COMISSAO
Roberto	AD_PRES	
Nair	AD_VP	
Leonardo	AD_VP	
Alexandre	IT_PROG	
Pedro	IT_PROG	
Dilma	IT_PROG	
Manuela	ES_GER	
Tomé	ES_AUX	
Cristóvão	ES_AUX	
Rafael	ES_AUX	
Pedro	ES_AUX	
Paula	VE_GER	.2
Pamela	VF_RFP	3

1-10

Valores Nulos

Quanto a coluna de uma linha não possui valor de dados específico, podemos dizer que o valor é *nulo*.

O valor nulo é caracterizado pela ausência de valor. A ausência de valor pode ocorrer porque o valor não é conhecido ou porque o valor não se aplica àquela coluna naquele registro.

No exemplo do slide, o valor da comissão só existe para os funcionários que trabalham com vendas. Para todos os outros funcionários não se aplica o valor da comissão.

Um valor nulo é diferente de zero ou de um espaço. Zero é um número e um espaço em branco é um caractere.

As colunas de qualquer tipo de dados podem conter valores nulos. No entanto, algumas constraints (`NOT NULL` e `PRIMARY KEY`) impedem o uso de valores nulos em colunas.

Valores Nulos em Expressões Aritméticas

- O resultado de expressões aritméticas que contenham valores nulo também é nulo.

```
SELECT nome,  
       cod_cargo,  
       13*(salario+comissao)  
FROM   funcionario
```

NOME	COD_CARGO	13*(SALARIO+COMISSAO)
Roberto	AD_PRES	
Nair	AD_VP	
Leonardo	AD_VP	
Alexandre	IT_PROG	
Pedro	IT_PROG	
Dilma	IT_PROG	
Manuela	ES_GER	
Tomé	ES_AUX	
Cristóvão	ES_AUX	
Rafael	ES_AUX	
Pedro	ES_AUX	
Paula	VE_GER	136502,6
Pamela	VE_REP	143003,9
Paula	VE_REP	111803,6

1-11

Valores Nulos em Expressões Aritméticas

Se o valor de uma coluna na expressão aritmética for nulo, o resultado será nulo. Por exemplo, se você tentar realizar uma divisão por zero, será gerado um erro. Entretanto, se você dividir um número por um valor nulo, o resultado será nulo.

No exemplo do slide, os funcionários que não trabalham com vendas não recebem comissão. Como a coluna COMISSAO na expressão aritmética é nula, o resultado da operação é nulo.

Tabela DUAL

- Tabela especial que contém apenas uma linha e uma coluna.

```
SQL> DESCRIBE dual
Nome      Nulo?     Tipo
-----
DUMMY          VARCHAR2(1)
```

- Usada quando se deseja retornar apenas um resultado de uma expressão.

```
SQL> SELECT 2400/4
2 FROM dual;
```

```
2400/4
-----
600
```

```
SQL> SELECT sysdate
2 FROM dual;
```

```
SYSDATE
-----
01/03/2009
```

```
SQL> SELECT user
2 FROM dual;
```

```
USER
-----
LOJA
```

1-12

Tabela Dual

A tabela `DUAL` é uma tabela que pertence ao esquema `SYS`, mas que todos os usuário do banco de dados possuem permissão de `SELECT`.

Ela é uma tabela que contém apenas uma coluna e uma linha e que pode ser utilizada quando se deseja retornar apenas um resultado de uma expressão aritmética ou o retorno de execução de uma função.

O slide mostra a estrutura do tabela `DUAL` e alguns exemplos de uso dessa tabela.

Apelido de Coluna

- Renomeia um cabeçalho de coluna
- Deve ser especificado imediatamente após o nome da coluna
- Também é possível incluir a palavra-chave opcional `AS` entre o nome e o apelido da coluna
- Requer aspas duplas quando contém espaços ou caracteres especiais, ou quando faz distinção entre maiúsculas e minúsculas

1-13

Apelido de Coluna

Quando exibe o resultado de uma consulta, o SQL*Plus usa o nome da coluna selecionada como seu cabeçalho. Como esse cabeçalho pode não ser descritivo, talvez seja difícil compreendê-lo. Para alterar um cabeçalho de coluna você pode utilizar o artifício de criar um apelido.

Especifique o apelido após a coluna na lista `SELECT` usando um espaço como separador ou inclua a palavra-chave opcional `AS` entre o nome e o apelido da coluna.

Os cabeçalhos de apelidos aparecem em maiúsculas. Se o apelido contiver espaços ou caracteres especiais (como `#` ou `$`), ou se fizer distinção entre maiúsculas e minúsculas, delimite-o por aspas duplas (`""`).

Apelido de Coluna

```
SELECT nome AS funcionario,  
       comissao valor_comissao  
FROM   funcionario;
```

FUNCIONARIO	VALOR_COMISSAO
Roberto	
Nair	
Leonardo	
Alexandre	
Pedro	
Dilma	
Manuela	
Tomé	
Cristóvão	
Rafael	
Pedro	
Paula	,2
Pamela	,3
Paula	,2

```
SELECT nome "Funcionario",  
       13*salario "Salário Anual"  
FROM   funcionario;
```

Funcionario	Salário Anual
Roberto	312000
Nair	221000
Leonardo	221000
Alexandre	117000
Pedro	78000
Dilma	54600
Manuela	75400
Tomé	45500
Cristóvão	40300
Rafael	33800
Pedro	32500
Paula	136500

1-14

Apelido de Coluna (continuação)

O primeiro exemplo exibe os nomes e a comissão dos funcionários. Observe que a palavra-chave opcional `AS` foi usada antes do apelido da coluna `nome`. O resultado da consulta será o mesmo com ou sem a inclusão da palavra-chave `AS`. Observe também que, na instrução SQL, os apelidos das colunas, `nome` e `comissao`, estão em minúsculas, enquanto o resultado da consulta exibe os cabeçalhos das colunas em maiúsculas. Como mencionado anteriormente, os cabeçalhos das colunas aparecem em maiúsculas por padrão.

O segundo exemplo exibe os nomes e os salários anuais dos funcionários. Como `Salário Anual` contém um espaço, ele foi delimitado por aspas duplas. Observe que o cabeçalho da coluna no resultado é exatamente igual ao apelido da coluna especificado no comando SQL.

A palavra-chave `AS` antes do apelido facilita a leitura da cláusula `SELECT`.

Operador de Concatenação

- Concatena duas colunas ou expressões do tipo caractere
- É representado por duas barras verticais (||)
- A coluna resultante da concatenação também é do tipo caractere

```
SELECT nome || sobrenome AS "Funcionario"  
FROM funcionario;
```

Funcionario
SheilaAlmeida
DilmaBarata
ManuelaBrunni
CristóvãoCabral
RobertoCarlos
PedroChaves
AlexandreHonorato
CosmoKramer
ToméLopes
NairMartins
RafaelMiranda
RobertoNascimento

1-15

Operador de Concatenação

Você pode concatenar duas colunas ou expressões caractere usando o *operador de concatenação* (||). As colunas nos dois lados do operador são combinadas para formar uma única coluna de saída com o conteúdo dos dois operadores concatenados.

No slide, NOME e SOBRENOME são concatenadas e recebem o apelido `Funcionario`. Observe que o nome e o sobrenome do funcionário são combinados para formar uma única coluna de saída.

Valores Nulos com o Operador de Concatenação

Se você concatenar um valor nulo com uma expressão caractere, o resultado será uma string de caracteres.

Por exemplo: `SOBRENOME || NULO` resulta em `SOBRENOME`.

Literais

- Um literal é um valor fixo do tipo caractere, data ou numérico que pode ser incluído na instrução `SELECT`
- Literais do tipo caractere e data precisam estar delimitados por aspas simples
- Para cada linha da tabela será retornado o valor repetido da literal

```
SQL> SELECT 'Isto é uma literal' AS texto
      2 FROM      departamento;

TEXTO
-----
Isto é uma literal
Isto é uma literal
Isto é uma literal
...
Isto é uma literal
Isto é uma literal

8 linhas selecionadas.
```

1-16

Literais

Um literal é um valor fixo do tipo caractere, data ou numérico incluído na lista `SELECT` que não constitui um nome ou um apelido de coluna. Ele é impresso para cada linha retornada.

É possível incluir vários valores literais separados por vírgula num comando `SELECT`. Cada literal será retornado como se fosse uma coluna.

Os literais do tipo caractere e data *devem* ser delimitados por aspas simples (' ').

Em literais do tipo número, as aspas não são necessárias.

Uso de Literais para Compor Resultados

```
SELECT nome || ' trabalha como ' || cod_cargo
       AS "Detalhe do Funcionário"
FROM   funcionario;
```

Detalhe do Funcionário
Roberto trabalha como AD_PRES
Nair trabalha como AD_VP
Leonardo trabalha como AD_VP
Alexandre trabalha como IT_PROG
Pedro trabalha como IT_PROG
Dilma trabalha como IT_PROG
Manuela trabalha como ES_GER
Tomé trabalha como ES_AUX
Cristóvão trabalha como ES_AUX
Rafael trabalha como ES_AUX

1-17

Uso de Literais para Compor Resultados

Você pode usar literais para compor resultados a partir dos registros de uma tabela.

O exemplo do slide exhibe os nomes e os cargos de todos os funcionários dispostos dentro de uma frase e o cabeçalho da coluna é “Detalhe do Funcionário”. Observe os espaços entre as aspas simples na instrução `SELECT`. Esses espaços melhoram a legibilidade da saída.

No exemplo a seguir, o sobrenome e o salário de cada funcionário são concatenados com um literal para que as linhas retornadas sejam mais significativas:

```
SELECT nome || ': Salário Anual = R$ ' || 13*salario
       AS "Salario Anual"
FROM   funcionario;
```

Salario Anual
Roberto: Salário Anual = R\$ 312000
Nair: Salário Anual = R\$ 221000
Leonardo: Salário Anual = R\$ 221000
Alexandre: Salário Anual = R\$ 117000
Pedro: Salário Anual = R\$ 78000
Dilma: Salário Anual = R\$ 54600
Manuela: Salário Anual = R\$ 75400

Uso de Aspas na Literal

- O que acontece quando uma literal possui uma aspas simples?

```
SQL> SELECT nome || ' quer beber um copo d'água.' AS texto  
2  
ERROR:  
ORA-01756: string entre aspas não finalizada adequadamente
```

- O banco de dados interpreta a aspas como o final da literal e o comando causa um erro

1-18

Uso de Aspas na Literal

Como o delimitador de literais caractere e data é a aspas simples, o que acontece quando a própria literal é composta um uma aspas simples?

O exemplo do slide demonstra o erro que é gerado pelo comando SQL. A aspas simples da literal é interpretada como o fechamento dela, causando um erro de sintaxe na execução do comando.

Uso de Aspas na Literal

- Especifique seu próprio delimitador de aspas com o operador (q) ou utilize duas aspas simples seguidas
- Usando o operador (q) escolha qualquer delimitador
- Melhore a legibilidade e a utilização

```
SELECT nome || q'[ quer beber um copo d'água.]' AS texto
FROM funcionario;
```

```
SELECT nome || ' quer beber um copo d'água.' AS texto
FROM funcionario;
```

TEXTO
Sheila quer beber um copo d'água.
Dilma quer beber um copo d'água.
Manuela quer beber um copo d'água.
Cristóvão quer beber um copo d'água.
Roberto quer beber um copo d'água.
Pedro quer beber um copo d'água.
Alexandre quer beber um copo d'água.

1-19

Uso de Aspas na Literal (continuação)

Quando o próprio literal contiver aspas simples, você poderá usar o operador de aspas (q) e escolher o delimitador de aspas que desejar, ou então poderá utilizar duas aspas simples seguidas.

Com o operador de aspas (q), você pode escolher qualquer caractere delimitador que seja conveniente, ou pode escolher um destes pares de caracteres: [], { }, () ou < >.

No exemplo do slide, com o uso do operador q ou a duplicação de aspas simples, o literal que contém aspas simples é exibido sem causar erros de sintaxe na execução do comando

Entretanto, o uso do operador q requer a utilização de colchetes [] como o delimitador de aspas. A string entre os delimitadores de colchetes é interpretada como uma string de caracteres literais.

Linhas Duplicadas

```
SQL> SELECT cod_cargo  
2 FROM funcionario;
```

```
COD_CARGO  
-----  
AD_ASST  
AD_PRES  
AD_VP  
AD_VP  
CTPUB_GER  
...  
MK_ANA  
MK_GER  
VE_GER  
VE_REP  
VE_REP  
VE_REP
```

20 linhas selecionadas.

```
SQL> SELECT DISTINCT cod_cargo  
2 FROM funcionario;
```

```
COD_CARGO  
-----  
AD_ASST  
AD_PRES  
AD_VP  
CTPUB_GER  
CT_GER  
ES_AUX  
ES_GER  
IT_PROG  
MK_ANA  
MK_GER  
VE_GER  
VE_REP
```

12 linhas selecionadas.

1-20

Linhas Duplicadas

O banco de dados irá retornar todos os valores de uma coluna mesmo se estiverem duplicados. O primeiro exemplo do slide exibe todos os códigos de cargos existentes na tabela `FUNCIONARIO` repetindo os valores iguais.

Para eliminar linhas duplicadas do resultado, inclua a palavra-chave `DISTINCT` na cláusula `SELECT` logo após a palavra-chave `SELECT`. No segundo exemplo do slide, a tabela `FUNCIONARIO` contém, na verdade, 20 linhas, mas existem apenas 12 cargos diferentes.

Você pode especificar várias colunas após o qualificador `DISTINCT`. Esse qualificador afeta todas as colunas selecionadas e o resultado são todas as combinações distintas das colunas.

```
SELECT DISTINCT cod_cargo,  
                cod_gerente  
FROM funcionario;
```

COD_CARGO	COD_GERENTE
ES_AUX	124
VE_GER	100
MK_ANA	201
AD_PRES	
VE_REP	149
AD_ASST	101
MK_GER	100
IT_PROG	102
IT_PROG	103
CTPUB_GER	205
AD_VP	100
ES_GER	100
CT_GER	101

Exercício 1

1-21

Exercício 1

1. Inicie uma sessão do SQL*Plus com o ID de usuário e a senha fornecidos pelo instrutor.

2. Os comandos SQL*Plus acessam o banco de dados.
Verdadeiro/Falso

3. Esta instrução SELECT é executada com êxito:

```
SELECT nome, sobrenome, salario AS Vencimentos
FROM funcionario;
```

Verdadeiro/Falso

4. Esta instrução SELECT é executada com êxito:

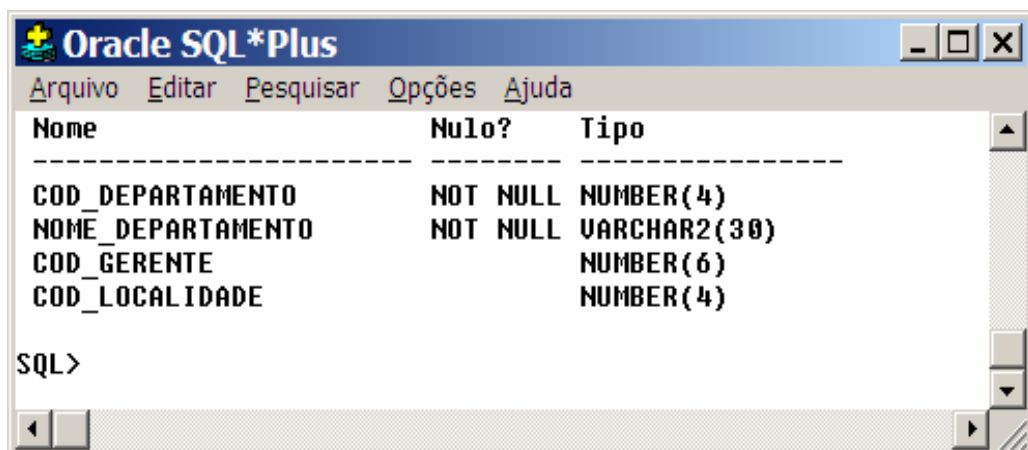
```
SELECT *
FROM faixa_salario;
```

Verdadeiro/Falso

5. Quais os erros do comando abaixo?

```
SELECT      cod_funcionario, nome
            salario x 13 SALARIO ANUAL
FROM        funcionario;
```

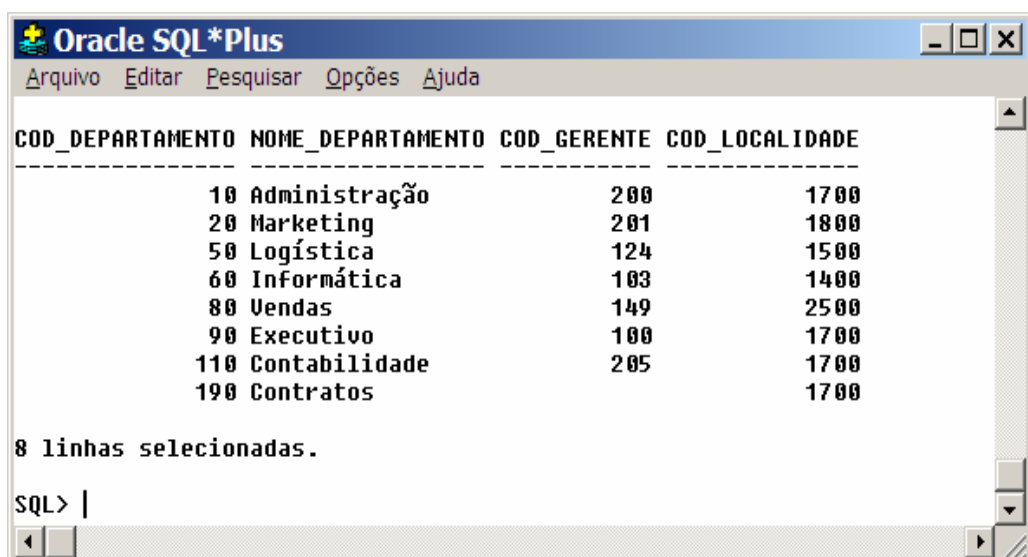
6. Determine a estrutura da tabela DEPARTAMENTO e seu conteúdo.



The screenshot shows the Oracle SQL*Plus interface with the command prompt at the bottom. The table structure is displayed as follows:

Nome	Nulo?	Tipo
COD_DEPARTAMENTO	NOT NULL	NUMBER(4)
NOME_DEPARTAMENTO	NOT NULL	VARCHAR2(30)
COD_GERENTE		NUMBER(6)
COD_LOCALIDADE		NUMBER(4)

SQL>



The screenshot shows the Oracle SQL*Plus interface with the command prompt at the bottom. The table content is displayed as follows:

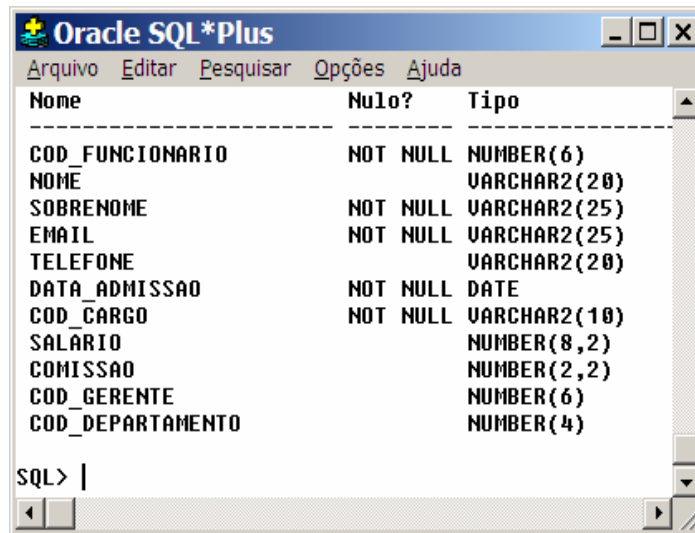
COD_DEPARTAMENTO	NOME_DEPARTAMENTO	COD_GERENTE	COD_LOCALIDADE
10	Administração	200	1700
20	Marketing	201	1800
50	Logística	124	1500
60	Informática	103	1400
80	Vendas	149	2500
90	Executivo	100	1700
110	Contabilidade	205	1700
190	Contratos		1700

8 linhas selecionadas.

SQL> |

Exercício 1 (continuação)

7. Determine a estrutura da tabela `FUNCIONARIO`.

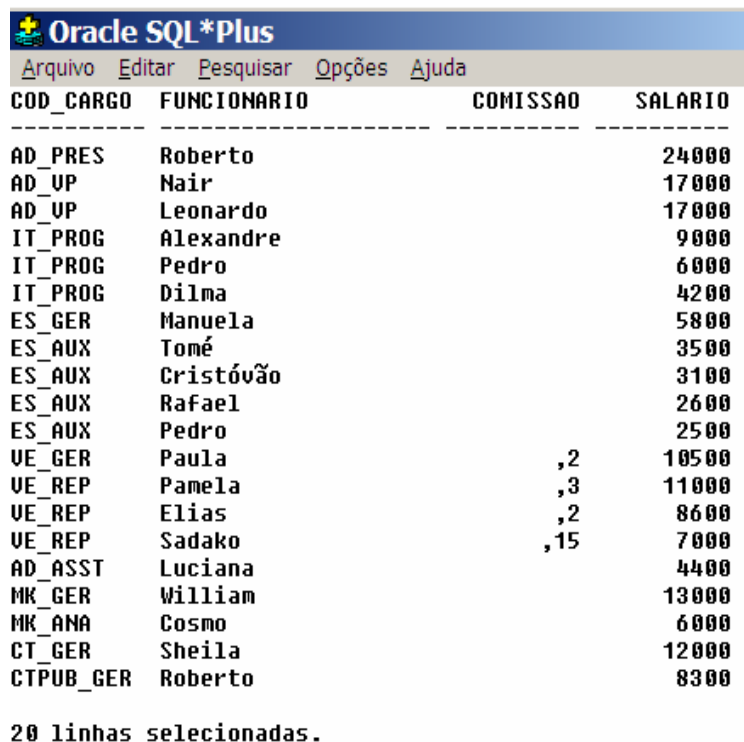


The screenshot shows the Oracle SQL*Plus interface with the table structure of `FUNCIONARIO` displayed. The table has the following columns:

Nome	Nulo?	Tipo
<code>COD_FUNCIONARIO</code>	NOT NULL	NUMBER(6)
<code>NOME</code>		VARCHAR2(20)
<code>SOBRENOME</code>	NOT NULL	VARCHAR2(25)
<code>EMAIL</code>	NOT NULL	VARCHAR2(25)
<code>TELEFONE</code>		VARCHAR2(20)
<code>DATA_ADMISSAO</code>	NOT NULL	DATE
<code>COD_CARGO</code>	NOT NULL	VARCHAR2(10)
<code>SALARIO</code>		NUMBER(8,2)
<code>COMISSAO</code>		NUMBER(2,2)
<code>COD_GERENTE</code>		NUMBER(6)
<code>COD_DEPARTAMENTO</code>		NUMBER(4)

Execute uma consulta para exibir o nome, o código do cargo, a comissão e o salário de cada funcionário, com o cargo do funcionário sendo exibido primeiro. Forneça o apelido `FUNCIONARIO` para o nome do funcionário. Salve o comando SQL no arquivo `cap_01_07.sql`.

8. Teste a consulta do arquivo `cap_01_07.sql` para verificar se é executada corretamente.



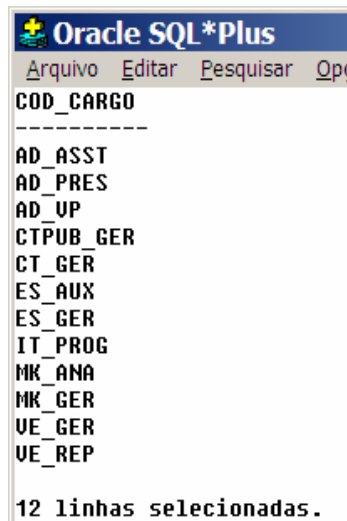
The screenshot shows the Oracle SQL*Plus interface displaying the results of a query. The query returns 20 rows of data. The columns are `COD_CARGO`, `FUNCIONARIO`, `COMISSAO`, and `SALARIO`.

<code>COD_CARGO</code>	<code>FUNCIONARIO</code>	<code>COMISSAO</code>	<code>SALARIO</code>
<code>AD_PRES</code>	Roberto		24000
<code>AD_UP</code>	Nair		17000
<code>AD_UP</code>	Leonardo		17000
<code>IT_PROG</code>	Alexandre		9000
<code>IT_PROG</code>	Pedro		6000
<code>IT_PROG</code>	Dilma		4200
<code>ES_GER</code>	Manuela		5800
<code>ES_AUX</code>	Tomé		3500
<code>ES_AUX</code>	Cristóvão		3100
<code>ES_AUX</code>	Rafael		2600
<code>ES_AUX</code>	Pedro		2500
<code>VE_GER</code>	Paula	,2	10500
<code>VE_REP</code>	Pamela	,3	11000
<code>VE_REP</code>	Elias	,2	8600
<code>VE_REP</code>	Sadako	,15	7000
<code>AD_ASST</code>	Luciana		4400
<code>MK_GER</code>	William		13000
<code>MK_ANA</code>	Cosmo		6000
<code>CT_GER</code>	Sheila		12000
<code>CTPUB_GER</code>	Roberto		8300

20 linhas selecionadas.

Exercício 1 (continuação)

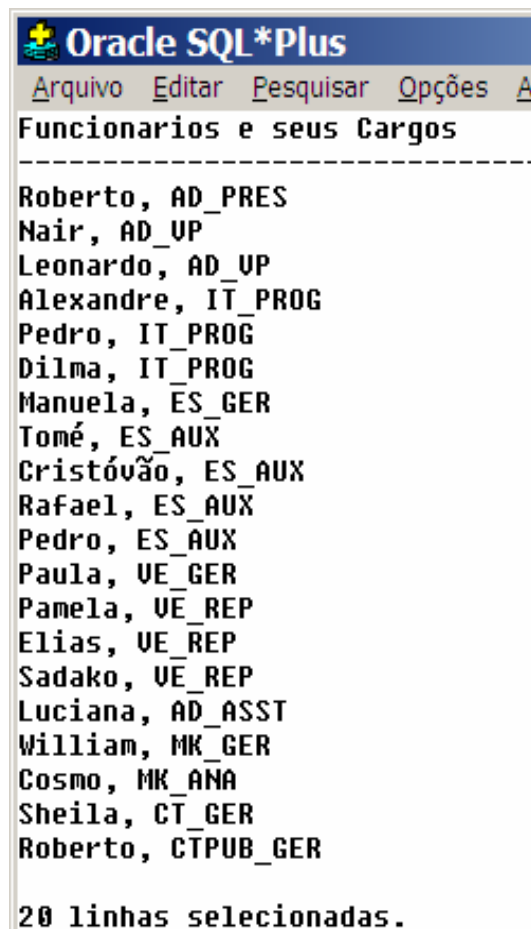
9. Crie uma consulta para exibir todos códigos dos cargo exclusivos da tabela `FUNCIONARIO`.



```
Oracle SQL*Plus
Arquivo Editar Pesquisar Opções
COD_CARGO
-----
AD_ASST
AD_PRES
AD_UP
CTPUB_GER
CT_GER
ES_AUX
ES_GER
IT_PROG
MK_ANA
MK_GER
UE_GER
UE_REP

12 linhas selecionadas.
```

10. Crie um relatório de todos os funcionários e os respectivos códigos de cargos. Exiba o nome concatenado com o código do cargo e separado-os por uma vírgula e um espaço. Nomeie a coluna como `Funcionarios e seus Cargos`.

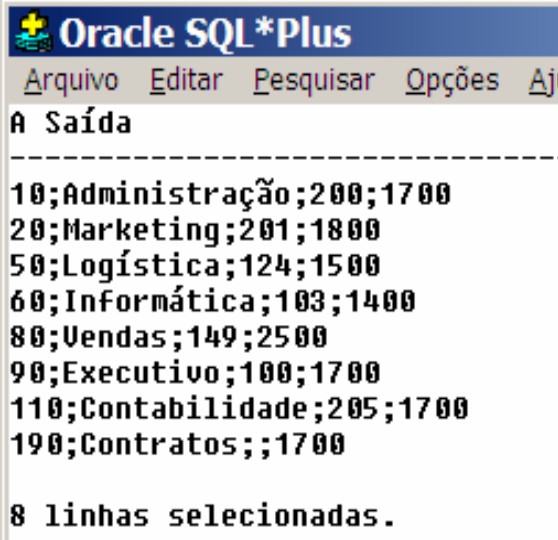


```
Oracle SQL*Plus
Arquivo Editar Pesquisar Opções A
Funcionarios e seus Cargos
-----
Roberto, AD_PRES
Nair, AD_UP
Leonardo, AD_UP
Alexandre, IT_PROG
Pedro, IT_PROG
Dilma, IT_PROG
Manuela, ES_GER
Tomé, ES_AUX
Cristóvão, ES_AUX
Rafael, ES_AUX
Pedro, ES_AUX
Paula, UE_GER
Pamela, UE_REP
Elias, UE_REP
Sadako, UE_REP
Luciana, AD_ASST
William, MK_GER
Cosmo, MK_ANA
Sheila, CT_GER
Roberto, CTPUB_GER

20 linhas selecionadas.
```


Exercício 1 (continuação)

11. Crie uma consulta para exibir todos os dados da tabela DEPARTAMENTO. Separe cada coluna com um ponto-e-vírgula (;). Nomeie o título da coluna como A Saída.



The screenshot shows the Oracle SQL*Plus interface. The title bar reads "Oracle SQL*Plus". The menu bar includes "Arquivo", "Editar", "Pesquisar", "Opções", and "Ajuda". The command prompt shows "A Saída" followed by a dashed line. The output displays eight rows of data, each separated by semicolons. The last line indicates "8 linhas selecionadas."

```
A Saída
-----
10;Administração;200;1700
20;Marketing;201;1800
50;Logística;124;1500
60;Informática;103;1400
80;Vendas;149;2500
90;Executivo;100;1700
110;Contabilidade;205;1700
190;Contratos;;1700

8 linhas selecionadas.
```