

Sistemas de Informação

Design e Desenvolvimento de Banco de Dados

PROF. MILTON

Restringindo e Classificando Dados

Objetivos

- Depois de completar esta lição, você poderá fazer o seguinte:
 - Limitar linhas recuperadas por uma consulta
 - Classificar linhas recuperadas por uma consulta

Objetivo da Lição

Ao recuperar dados do banco de dados, pode ser preciso restringir as linhas de dados exibidas ou especificar a ordem de exibição das mesmas. Essa lição explica as instruções SQL que você utiliza para executar essas ações.

Limitando Linhas Usando uma Seleção

EMP

| EMPNO | ENAME | JOB | ... | DEPTNO |
|-------|-------|-----------|-----|--------|
| 7839 | KING | PRESIDENT | | 10 |
| 7698 | BLAKE | MANAGER | | 30 |
| 7782 | CLARK | MANAGER | | 10 |
| 7566 | JONES | MANAGER | | 20 |
| ... | | | | |

"...recuperar todos os funcionários do departamento 10"



EMP

| EMPNO | ENAME | JOB | ... | DEPTNO |
|-------|--------|-----------|-----|--------|
| 7839 | KING | PRESIDENT | | 10 |
| 7782 | CLARK | MANAGER | | 10 |
| 7934 | MILLER | CLERK | | 10 |

Limitando Linhas Usando uma Seleção

No exemplo do slide, suponha que você deseje exibir todos os funcionários do departamento 10.

O conjunto de linhas realçadas com um valor 10 na coluna DEPTNO são as únicas retornadas. Esse método de restrição é a base da cláusula WHERE na linguagem SQL.

Limitando Linhas Selecionadas

- Restringe as linhas retornadas usando a cláusula WHERE.

```
SELECT      [DISTINCT] {*} coluna [apelido], ...
FROM        tabela
[WHERE      condição (ões)];
```

- A cláusula WHERE segue a cláusula FROM.

Limitando Linhas Selecionadas

É possível restringir as linhas retornadas da consulta utilizando a cláusula **WHERE**. Uma cláusula WHERE contém uma condição que deve coincidir e seguir diretamente a cláusula FROM.

Na sintaxe:

| | |
|-----------------|--|
| WHERE | restringe a consulta às linhas que atendem uma condição. |
| <i>condição</i> | é composta por nomes de colunas, expressões, constantes e um operador de comparação. |

A cláusula WHERE pode comparar valores em colunas, valores literais, expressões aritméticas ou funções. A cláusula WHERE é formada por três elementos:

- Nome de coluna
- Operadores de comparação
- Nome da coluna, constante ou lista de valores

Usando a Cláusula WHERE

```
SELECT ename, job, deptno
FROM emp
WHERE job='CLERK';
```

| ENAME | JOB | DEPTNO |
|--------|-------|--------|
| JAMES | CLERK | 30 |
| SMITH | CLERK | 20 |
| ADAMS | CLERK | 20 |
| MILLER | CLERK | 10 |

Usando a Cláusula WHERE

No exemplo, a instrução SELECT recupera o nome, o cargo e o número do departamento de todos os funcionários cujo cargo é CLERK.

Observe que o cargo CLERK foi especificado em letras maiúsculas para garantir que a correspondência seja feita com a coluna do cargo na tabela EMP. As *strings* de caractere não fazem distinção entre maiúsculas de minúsculas.

Strings de Caractere e Datas

- As *strings* de caractere e valores de data aparecem entre aspas simples.
- Os valores de caractere fazem distinção entre maiúsculas e minúsculas e os valores de data diferenciam formatos.
- O formato de data default é DD-MON-YY.

```
SELECT  ename, job, deptno
FROM    emp
WHERE   ename = 'JAMES';
```

Strings de Caractere e Datas

Strings de caractere e datas na cláusula WHERE devem estar entre aspas simples (' '). Constantes de número, no entanto, não precisam.

Todas as pesquisas de caractere fazem distinção entre maiúsculas e minúsculas. No exemplo a seguir, nenhuma linha retornou pois a tabela EMP armazena todas as datas em letras maiúsculas:

```
SELECT ename, empno, job, deptno
FROM emp
WHERE job='clerk';
```

Os produtos da Oracle armazenam datas em um formato numérico interno, representando o século, ano, mês, dia, horas, minutos e segundos. A exibição da data *default* é DD-MON-YY.

Observação: A alteração do formato da data default será abordada na lição subsequente.

Valores de número não aparecem entre aspas.

Operadores de Comparação

| Operador | Significado |
|----------|-------------------------|
| = | Igual a |
| > | Maior do que |
| >= | Maior do que ou igual a |
| < | Menor que |
| <= | Menor ou igual a |
| <> | Diferente de |

Operadores de Comparação

Usa-se os **operadores de comparação** em condições que comparam uma expressão a outra. Eles são usados na cláusula WHERE no seguinte formato:

Sintaxe

```
... WHERE expr valor operador
```

Exemplos

```
... WHERE hiredate='01-JAN-95'
```

```
... WHERE sal>=1500
```

```
... WHERE ename='SMITH'
```


Usando Operadores de Comparação

```
SELECT ename, sal, comm
FROM emp
WHERE sal <= comm;
```

| ENAME | SAL | COMM |
|--------|------|------|
| MARTIN | 1250 | 1400 |

Usando Operadores de Comparação

No exemplo, a instrução SELECT recupera o nome, salário e comissão da tabela EMP, em que o salário do funcionário é menor ou igual à quantia da comissão. Observe que não há valor explícito fornecido para a cláusula WHERE. Os dois valores que estão sendo comparados são retirados das colunas SAL e COMM na tabela EMP.

Outros Operadores de Comparação

| Operador | Significado |
|------------------------------|---|
| BETWEEN ...AND... | Entre dois valores (inclusive) |
| IN(list) | Vincula qualquer um de uma lista de valores |
| LIKE | Vincula um padrão de caractere |
| IS NULL | É um valor nulo |

Usando o Operador BETWEEN

- Use o operador BETWEEN para exibir linhas baseadas em uma faixa de valores.

```
SQL> SELECT  ename, sal
2  FROM      emp
3  WHERE     sal BETWEEN 1000 AND 1500;
```

| ENAME | SAL | | |
|--------|-------|-----------------|-----------------|
| ----- | ----- | | |
| MARTIN | 1250 | Limite inferior | Limite superior |
| TURNER | 1500 | | |
| WARD | 1250 | | |
| ADAMS | 1100 | | |
| MILLER | 1300 | | |

O Operador BETWEEN

Você pode exibir linhas baseadas em uma faixa de valores usando o operador **BETWEEN**. A faixa que você especificar possuirá uma faixa inferior e uma superior.

A instrução SELECT no slide retorna as linhas da tabela EMP para qualquer funcionário cujo salário esteja entre US\$1.000 e US\$1.500.

Valores especificados com o operador BETWEEN são inclusivos. Você deve especificar primeiro o limite inferior.

Usando o Operador IN

- Use o operador IN para testar os valores de uma lista.

```
SQL> SELECT empno, ename, sal, mgr
2 FROM emp
3 WHERE mgr IN (7902, 7566, 7788);
```

| EMPNO | ENAME | SAL | MGR |
|-------|-------|------|------|
| 7902 | FORD | 3000 | 7566 |
| 7369 | SMITH | 800 | 7902 |
| 7788 | SCOTT | 3000 | 7566 |
| 7876 | ADAMS | 1100 | 7788 |

O Operador IN

Para testar os valores em uma determinada lista, use o operador **IN**.

O exemplo do slide exibe o número do funcionário, o nome, o salário e o número de funcionário do gerente de todos os funcionários cujo número de funcionário do gerente for 7902, 7566 ou 7788.

O operador IN pode ser usado com qualquer tipo de dados. O exemplo seguinte retorna uma linha da tabela EMP para qualquer funcionário cujo nome estiver incluído na lista de nomes na cláusula WHERE:

```
SELECT empno, ename, mgr, deptno
FROM emp
WHERE ename IN ('FORD' , 'ALLEN');
```

Se forem utilizados caracteres ou datas na lista, eles devem estar entre aspas simples (' ').

Usando o Operador LIKE

- Use o operador LIKE para executar pesquisas curinga de valores de string de pesquisa válidos.
- As condições de pesquisa podem conter caracteres literais ou números.
 - % denota zero ou muitos caracteres.
 - _ denota um caractere.

```
SQL> SELECT  ename
      2 FROM    emp
      3 WHERE   ename LIKE 'S%';
```

O Operador LIKE

Talvez nem sempre você saiba o valor exato pelo qual procurar. É possível selecionar linhas que vinculem um padrão de caractere usando o operador **LIKE**. A operação de vinculação de um padrão de caractere refere-se a uma pesquisa de curinga. Dois símbolos podem ser utilizados para construir a string de pesquisa.

| Símbolo | Descrição |
|---------|---|
| % | Representa qualquer seqüência de zero ou mais caracteres. |
| _ | Representa qualquer caractere único. |

A instrução SELECT acima retorna o nome do funcionário da tabela EMP para qualquer funcionário cujo nome começa com "S". Note o "S". maiúsculo. Os nomes iniciados com "s" não retornarão.

O operador LIKE pode ser usado como um atalho para algumas comparações BETWEEN. O exemplo a seguir exibe os nomes as datas de admissão de todos os funcionários admitidos entre janeiro e dezembro de 1981:

```
SELECT ename, hiredate
FROM emp
WHERE hiredate LIKE '%1981';
```

Usando o Operador LIKE

- Você pode combinar caracteres de vinculação de padrão.

```
SQL> SELECT  ename
      2  FROM    emp
      3  WHERE   ename LIKE '_A%';
```

| ENAME |
|--------|
| MARTIN |
| JAMES |
| WARD |

Combinando Caracteres Curinga

Os símbolos **%** e **_** podem ser usados em qualquer combinação com caracteres literais. O exemplo do slide exibe os nomes de todos os funcionários cujos nomes possuem a letra "A" como segundo caractere.

A Opção ESCAPE

Quando for necessário ter uma correspondência exata para os caracteres '%' e '_' reais, use a opção **ESCAPE**. Essa opção especifica qual é o caractere **ESCAPE**. Caso possua HEAD_QUARTERS como um nome de departamento, deve procurar por ele utilizando a seguinte instrução SQL:

```
SELECT * FROM dept
WHERE dname LIKE '%\_%' ESCAPE '\\';
```

| DEPTNO | DNAME | LOC |
|--------|---------------|---------|
| 50 | HEAD_QUARTERS | ATLANTA |

A opção **ESCAPE** identifica a barra invertida (\) como o caractere de escape. No padrão, o caractere de escape vem antes do sublinhado (_). Isso faz com que o Oracle Server interprete o sublinhado literalmente.

Usando o Operador IS NULL

- Teste para valores nulos com o operador IS NULL.

```
SELECT  ename, mgr
FROM    emp
WHERE   mgr IS NULL;
```

| ENAME | MGR |
|-------|-------|
| ----- | ----- |
| KING | |

O Operador IS NULL

O operador **IS NULL** testa valores que são nulos. Um valor nulo significa que o valor não está disponível, não-atribuído, desconhecido ou não-aplicável. Assim, não é possível testar com (=) porque um valor nulo não pode ser igual ou desigual a qualquer valor. O exemplo do slide recupera o nome e o gerente de todos os funcionários que não possuem um gerente.

Por exemplo, para exibir o nome, o cargo e a comissão de todos os funcionários que não estão nomeados para obter uma comissão, use a seguinte instrução SQL:

```
SELECT  ename, job, comm
FROM    emp
WHERE   comm IS NULL;
```

| ENAME | JOB | COMM |
|-------|-----------|-------|
| ----- | ----- | ----- |
| KING | PRESIDENT | |
| BLAKE | MANAGER | |
| CLARK | MANAGER | |
| ... | | |

Operadores Lógicos

| Operador | Significado |
|----------|--|
| AND | Retorna TRUE se as condições de componentes forem TRUE |
| OR | Retorna TRUE se cada condição de componente for TRUE |
| NOT | Retorna TRUE se a condição seguinte for FALSE |

Operadores Lógicos

Um operador lógico combina o resultado de duas condições de componente para produzir um único resultado com base neles ou inverter o resultado para uma condição única. Três **operadores lógicos** estão disponíveis no SQL:

AND
OR
NOT

Todos os exemplos até aqui especificaram somente uma condição na cláusula WHERE. Você pode usar várias condições em uma cláusula WHERE usando operadores AND e OR.

Usando o Operador AND

AND exige que ambas as condições sejam TRUE

```
SELECT empno, ename, job, sal
FROM emp
WHERE sal >= 1100
AND job = 'CLERK';
```

| EMPNO | ENAME | JOB | SAL |
|-------|--------|-------|------|
| 7876 | ADAMS | CLERK | 1100 |
| 7934 | MILLER | CLERK | 1300 |

O Operador AND

No exemplo, as duas condições devem ser verdadeiras para cada registro a ser selecionado. Assim, um funcionário que possua o cargo CLERK e receba mais de US\$1.100 será selecionado.

Todas as pesquisas de caractere fazem distinção entre maiúsculas de minúsculas. Nenhuma linha retornará se CLERK não estiver em letra maiúscula. As *strings* de caractere devem estar entre aspas.

Tabela de Verdade AND

A tabela a seguir mostra os resultados da combinação de duas expressões com **AND**:

| AND | TRUE | FALSE | NULL |
|-------|-------|-------|-------|
| TRUE | TRUE | FALSE | NULL |
| FALSE | FALSE | FALSE | FALSE |
| NULL | NULL | FALSE | NULL |

Usando o Operador OR

OR exige que uma das condições seja TRUE

```
SELECT empno, ename, job, sal
FROM emp
WHERE sal >= 1100
OR job = 'CLERK';
```

| EMPNO | ENAME | JOB | SAL |
|-------|--------|-----------|------|
| 7839 | KING | PRESIDENT | 5000 |
| 7698 | BLAKE | MANAGER | 2850 |
| 7782 | CLARK | MANAGER | 2450 |
| 7566 | JONES | MANAGER | 2975 |
| 7654 | MARTIN | SALESMAN | 1250 |
| ... | | | |
| 7900 | JAMES | CLERK | 950 |
| ... | | | |

14 rows selected.

O Operador OR

No exemplo, cada condição pode ser verdadeira para qualquer registro a ser selecionado. Assim, um funcionário que possua o cargo CLERK *ou* que receba mais de US\$1.100 será selecionado.

A Tabela de Verdade OR

A tabela a seguir mostra os resultados da combinação de duas expressões com **OR**:

| OR | TRUE | FALSE | NULL |
|-------|------|-------|------|
| TRUE | TRUE | TRUE | TRUE |
| FALSE | TRUE | FALSE | NULL |
| NULL | TRUE | NULL | NULL |

Usando o Operador NOT

```
SELECT ename, job
FROM emp
WHERE job NOT IN ('CLERK', 'MANAGER', 'ANALYST');
```

| ENAME | JOB |
|--------|-----------|
| KING | PRESIDENT |
| MARTIN | SALESMAN |
| ALLEN | SALESMAN |
| TURNER | SALESMAN |
| WARD | SALESMAN |

O Operador NOT

O exemplo do slide exibe o nome e o cargo de todos os funcionários que *não* possuem os cargos CLERK, MANAGER ou ANALYST.

A Tabela de Verdade NOT

A tabela a seguir mostra o resultado da aplicação do operador **NOT** para uma condição:

| NOT | TRUE | FALSE | NULL |
|------|-------|-------|------|
| TRUE | FALSE | TRUE | NULL |

Observação: O operador NOT pode ser utilizado também com outros operadores SQL, como BETWEEN, LIKE e NULL.

```
... WHERE job NOT IN ('CLERK', 'ANALYST')
... WHERE sal NOT BETWEEN 1000 AND 1500
... WHERE ename NOT LIKE '%A%'
... WHERE comm IS NOT NULL
```


Regras de Precedência

| Ordem de Avaliação | Operador |
|--------------------|-----------------------------------|
| 1 | Todos os operadores de comparação |
| 2 | NOT |
| 3 | AND |
| 4 | OR |

- Sobreponha regras de precedência usando parênteses.

Regras de Precedência

```
SELECT      ename, job, sal
FROM        emp
WHERE       job='SALESMAN'
  OR        job='PRESIDENT'
  AND       sal>1500;
```



| ENAME | JOB | SAL |
|--------|-----------|------|
| KING | PRESIDENT | 5000 |
| MARTIN | SALESMAN | 1250 |
| ALLEN | SALESMAN | 1600 |
| TURNER | SALESMAN | 1500 |
| WARD | SALESMAN | 1250 |

Exemplo de Precedência do Operador AND

No exemplo do slide, há duas condições:

A primeira condição é que o cargo seja PRESIDENT e o salário maior que 1500.

A segunda condição é que o cargo seja SALESMAN.

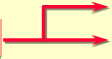
Portanto, a instrução SELECT faz a leitura do seguinte modo:

"Selecione a linha se o funcionário for um PRESIDENT e receber mais que US\$1.500 ou se o funcionário for um SALESMAN".

Regras de Precedência

Use parênteses para forçar a prioridade

```
SELECT      ename, job, sal
FROM        emp
WHERE       (job='SALESMAN'
OR          job='PRESIDENT')
AND        sal>1500;
```



| ENAME | JOB | SAL |
|-------|-----------|------|
| KING | PRESIDENT | 5000 |
| ALLEN | SALESMAN | 1600 |

Usando Parênteses

No exemplo, há duas condições:

A primeira condição é que o cargo seja PRESIDENT ou SALESMAN.

A segunda é que o salário sejam maior que 1500.

Portanto, a instrução SELECT faz a leitura do seguinte modo:

"Selecione a linha se um funcionário for um PRESIDENT ou um SALESMAN e se o funcionário receber mais de US\$1500".

Cláusula ORDER BY

- Classificar as linhas com a cláusula ORDER BY
 - ASC: ordem crescente, default
 - DESC: ordem decrescente
- A cláusula ORDER BY vem depois na instrução SELECT.

```
SELECT      ename, job, deptno, hiredate
FROM        emp
ORDER BY    hiredate;
```

| ENAME | JOB | DEPTNO | HIREDATE |
|-------|----------|--------|-----------|
| SMITH | CLERK | 20 | 17-DEC-80 |
| ALLEN | SALESMAN | 30 | 20-FEB-81 |
| ... | | | |

14 rows selected.

A Cláusula ORDER BY

A ordem das linhas retornadas em um resultado de consulta é indefinida. A cláusula **ORDER BY** pode ser utilizada para classificar as linhas. Se você usar a cláusula ORDER BY, deve colocá-la por último. É possível especificar uma expressão ou um apelido para classificação.

Sintaxe

```
SELECT      expr
FROM        tabela
[WHERE      condição(ões)]
[ORDER BY   {coluna, expr} [ASC|DESC]];
```

onde:

ORDER BY especifica a ordem em que as linhas recuperadas são exibidas
ASC ordena as linhas na ordem crescente (essa é a ordem default)
DESC ordena as linhas na ordem decrescente

Se a cláusula ORDER BY não for usada, a ordem de classificação será indefinida e o Oracle Server talvez não extraia as linhas na mesma ordem ao realizar a mesma consulta duas vezes. Use a cláusula ORDER BY para exibir as linhas em uma ordem específica.

Classificando em Ordem Decrescente

```
SELECT  ename, job, deptno, hiredate
FROM    emp
ORDER BY hiredate DESC;
```

| ENAME | JOB | DEPTNO | HIREDATE |
|--------|-----------|--------|-----------|
| ADAMS | CLERK | 20 | 12-JAN-83 |
| SCOTT | ANALYST | 20 | 09-DEC-82 |
| MILLER | CLERK | 10 | 23-JAN-82 |
| JAMES | CLERK | 30 | 03-DEC-81 |
| FORD | ANALYST | 20 | 03-DEC-81 |
| KING | PRESIDENT | 10 | 17-NOV-81 |
| MARTIN | SALESMAN | 30 | 28-SEP-81 |
| ... | | | |

14 rows selected.

Ordenação de Dados Default

A ordem de classificação default é crescente:

- Valores numéricos são exibidos primeiro com os valores mais baixos — por exemplo, 1–999.
- Valores de datas são exibidos primeiro com os valores mais recentes — por exemplo, 01-JAN-92 antes de 01-JAN-95.
- Valores de caracteres são exibidos em ordem alfabética — por exemplo, o A primeiro e o Z por último.
- Os valores nulos são exibidos por último em seqüências ascendentes e primeiro em seqüências descendentes.

Invertendo a Ordem Default

Para inverter a ordem de exibição das linhas, especifique a palavra-chave DESC após o nome da coluna na cláusula ORDER BY. O exemplo do slide classifica o resultado pelo funcionário contratado mais recentemente.

Classificando por Apelido de Coluna

```
SELECT empno, ename, sal*12 annsal
FROM emp
ORDER BY annsal;
```

| EMPNO | ENAME | ANNSAL |
|-------|--------|--------|
| 7369 | SMITH | 9600 |
| 7900 | JAMES | 11400 |
| 7876 | ADAMS | 13200 |
| 7654 | MARTIN | 15000 |
| 7521 | WARD | 15000 |
| 7934 | MILLER | 15600 |
| 7844 | TURNER | 18000 |

...

14 rows selected.

Classificando por Apelidos de Coluna

Você pode usar um apelido de coluna na cláusula ORDER BY. O exemplo do slide classifica os dados por salário anual.

Classificando por Várias Colunas

- A ordem da lista ORDER BY é a ordem de classificação.

```
SELECT      ename, deptno, sal
FROM        emp
ORDER BY    deptno, sal DESC;
```

| ENAME | DEPTNO | SAL |
|--------|--------|------|
| KING | 10 | 5000 |
| CLARK | 10 | 2450 |
| MILLER | 10 | 1300 |
| FORD | 20 | 3000 |
| ... | | |

14 rows selected.

- Você pode classificar por uma coluna que não esteja na lista SELECT.

Classificando por Várias Colunas

Você pode classificar os resultados da consulta por mais de uma coluna. O limite de classificação é o número de colunas de uma determinada tabela.

Na cláusula ORDER BY, especifique as colunas e separe seus nomes usando vírgulas. Se deseja inverter a ordem de uma coluna, especifique DESC após seu nome. É possível ordenar por colunas que não estão incluídas na cláusula SELECT.

Exemplo

Exiba o nome e o salário de todos os funcionários. Ordene o resultado por número de departamento e, em seguida, em ordem decrescente de salário.

```
SELECT      ename,  sal
FROM        emp
ORDER BY    deptno, sal DESC;
```

Sumário

```
SELECT      [DISTINCT] {*} | coluna [apelido], ...}
FROM        tabela
[WHERE      condição(ões)]
[ORDER BY   {coluna, expr, apelido} [ASC|DESC]];
```

Sumário

Nesta lição, você aprendeu sobre a restrição e classificação de colunas retornadas pela instrução SELECT. Também aprendeu como implementar vários operadores.

Visão Geral do Exercício

- Seleccionando dados e alterando a ordem das linhas exibidas
- Restringindo colunas utilizando a cláusula WHERE
- Usando as aspas nos apelidos de colunas



Visão Geral do Exercício

Esta seção oferece uma variedade de exercícios usando a cláusula WHERE e a cláusula ORDER BY.

Exercício 2

1. Crie uma consulta para exibir o nome e o salário dos funcionários que recebem mais de US\$2.850.

| ENAME | SAL |
|-------|-------|
| ----- | ----- |
| KING | 5000 |
| JONES | 2975 |
| FORD | 3000 |
| SCOTT | 3000 |

2. Crie uma consulta para exibir o nome do funcionário e o número do departamento para o número do funcionário 7566.

| ENAME | DEPTNO |
|-------|--------|
| ----- | ----- |
| JONES | 20 |

3. Exiba o nome e o salário de todos os funcionários cujos salários **não** estejam na faixa entre US\$1.500 e US\$2.850.

| ENAME | SAL |
|--------|-------|
| ----- | ----- |
| KING | 5000 |
| JONES | 2975 |
| MARTIN | 1250 |
| JAMES | 950 |
| WARD | 1250 |
| FORD | 3000 |
| SMITH | 800 |
| SCOTT | 3000 |
| ADAMS | 1100 |
| MILLER | 1300 |

10 rows selected.

Exercício 2 (continuação)

4. Exiba o nome do funcionário, o cargo e a data de admissão dos funcionários admitidos entre 20 de fevereiro de 1981 e 1 de maio de 1981. Ordene a consulta de modo crescente pela data inicial.

| ENAME | JOB | HIREDATE |
|-------|----------|-----------|
| ALLEN | SALESMAN | 20-FEB-81 |
| WARD | SALESMAN | 22-FEB-81 |
| JONES | MANAGER | 02-APR-81 |
| BLAKE | MANAGER | 01-MAY-81 |

5. Exiba o nome do funcionário e o número do departamento de todos os funcionários entre os departamentos 10 e 30 por ordem alfabética de nome.

| ENAME | DEPTNO |
|--------|--------|
| ALLEN | 30 |
| BLAKE | 30 |
| CLARK | 10 |
| JAMES | 30 |
| JONES | 10 |
| MARTIN | 30 |
| MILLER | 10 |
| TURNER | 30 |
| WARD | 30 |

9 rows selected.

6. Liste o nome e o salário dos funcionários que recebem mais de US\$1.500 e que estão nos departamentos 10 ou 30. Nomeie as colunas Employee e Monthly Salary, respectivamente.

| Employee | Monthly Salary |
|----------|----------------|
| KING | 5000 |
| BLAKE | 2850 |
| CLARK | 2450 |
| ALLEN | 1600 |

Exercício 2 (continuação)

7. Exiba o nome e a data de admissão de cada funcionário admitido em 1982.

| ENAME | HIREDATE |
|--------|-----------|
| ----- | ----- |
| SCOTT | 09-DEC-82 |
| MILLER | 23-JAN-82 |

8. Exiba o nome e o cargo de todos os funcionários que não possuem um gerente.

| ENAME | JOB |
|-------|-----------|
| ----- | ----- |
| KING | PRESIDENT |

9. Exiba o nome, o salário e a comissão de todos os funcionários que recebem comissão. Classifique os dados em ordem decrescente de salários e comissões.

| ENAME | SAL | COMM |
|--------|-------|-------|
| ----- | ----- | ----- |
| ALLEN | 1600 | 300 |
| TURNER | 1500 | 0 |
| MARTIN | 1250 | 1400 |
| WARD | 1250 | 500 |

10. Exiba os nomes de todos os funcionários que possuem um A na terceira letra de seus nomes.

| ENAME |
|-------|
| ----- |
| BLAKE |
| CLARK |
| ADAMS |

11. Exiba todos os funcionários que possuem duas letras L em seus nomes e estão no departamento 30 ou seu gerente seja o 7782.

| ENAME |
|--------|
| ----- |
| ALLEN |
| MILLER |

Exercício 2 (continuação)

12. Exiba o nome, o cargo e o salário de todos os funcionários cujos cargos seja CLERK ou ANALYST e que seus salários não sejam iguais a US\$1.000, US\$3.000 ou US\$5.000.

| ENAME | JOB | SAL |
|--------|-------|-------|
| ----- | ----- | ----- |
| JAMES | CLERK | 950 |
| SMITH | CLERK | 800 |
| ADAMS | CLERK | 1100 |
| MILLER | CLERK | 1300 |

13. Exiba o nome, o salário e a comissão de todos os funcionários cuja quantia de comissão seja maior que seus salários com 10% de aumento.

| Employee | Monthly Salary | COMM |
|----------|----------------|-------|
| ----- | ----- | ----- |
| MARTIN | 1250 | 1400 |

