

Bases de Dados

SQL- Restringir e Ordenar Dados



Limitar as Linhas Usando uma Selecção

EMP

NEMP	NOME	FUNCAO	...	NDEP
7839	KING	PRESIDENT		10
7698	BLAKE	MANAGER		30
7782	CLARK	MANAGER		10
7566	JONES	MANAGER		20
...				

"...obtem todos os empregados do departamento 10"

EMP

NEMP	NOME	FUNCAO	...	NDEP
7839	KING	PRESIDENT		10
7782	CLARK	MANAGER		10
7934	MILLER	CLERK		10



Limitar as Linhas Selecionadas

- ▶ Restringir as linhas devolvidas usando a clausula WHERE.

```
SELECT      [DISTINCT] { * | column [alias], ... }
FROM        table
[WHERE      condition(s)];
```

- ▶ A clausula WHERE segue a clausula FROM.



Usar a Cláusula WHERE

```
SQL> SELECT nome, funcao, ndep
2 FROM emp
3 WHERE funcao = 'CONTABILISTA';
```

NOME	FUNCAO	NDEP
JAMES	CONTABILISTA	30
SMITH	CONTABILISTA	20
ADAMS	CONTABILISTA	20
MILLER	CONTABILISTA	10



Strings de caracteres e Datas

- ▶ Valores de caracteres e datas vão para dentro de plicas.
- ▶ Valores caracter são case sensitive e valores de data são format sensitive.
- ▶ O formato de data default é DD-MON-YY .

```
SQL> SELECT nome, funcao, ndep
2 FROM emp
3 WHERE nome = 'JAMES';
```



Operadores de comparação

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



Usar os Operadores de comparação

```
SQL> SELECT nome, sal, premios
2 FROM emp
3 WHERE sal<=premios;
```

NOME	SAL	PREMIOS
MARTIN	1250	1400



Outros Operadores de Comparação

Operador	Significado
BETWEEN ...AND...	Entre dois valores (inclusivé)
IN(list)	Igual a qualquer dos valores da listas
LIKE	Igual ao padrão de caracteres
IS NULL	É um valor null



Usar o Operador BETWEEN

- ▶ Usar o operador BETWEEN para mostrar linhas baseadas numa gama de valores.

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

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



Usar o Operador IN

- ▶ Usar o operador IN para testar os valores numa lista.

```
SQL> SELECT  nemp, nome, sal, encar
2 FROM      emp
3 WHERE     encar IN (7902, 7566, 7788);
```

NEMP	NOME	SAL	ENCAR
7902	FORD	3000	7566
7369	SMITH	800	7902
7788	SCOTT	3000	7566
7876	ADAMS	1100	7788



Usar o Operador LIKE

- ▶ Usar o operador LIKE para fazer pesquisas com os wildcards de valores de string válidos.
- ▶ As condições de pesquisa podem conter caracteres literais ou números.
 - ▶ % designa zero ou mais caracteres.
 - ▶ _ designa um caracter.

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



Usar o Operador LIKE

- Pode combinar os dois caracteres.

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

NOME
MARTIN
JAMES
WARD



Usar o Operador IS NULL

- Testar se há valores nulos com o operador IS NULL.

```
SQL> SELECT  nome, encar
2  FROM      emp
3  WHERE     encar IS NULL;
```

NOME	ENCAR
KING	



Operadores Lógicos

Operador	Significado
AND	Devolve TRUE se ambas as condições forem TRUE
OR	Devolve TRUE se alguma das condições for TRUE
NOT	Devolve TRUE se a condição for FALSE

Usar o Operador AND

- o AND requer que ambas as condições sejam TRUE.

```
SQL> SELECT nemp, nome, funcao, sal
2 FROM emp
3 WHERE sal >= 1100
4 AND funcao = 'CONTABILISTA';
```

NEMP	NOME	FUNCAO	SAL
7876	ADAMS	CONTABILISTA	1100
7934	MILLER	CONTABILISTA	1300

Usar o Operador OR

- o OR requer que uma das condições seja TRUE.

```
SQL> SELECT nemp, nome, funcao, sal
2 FROM emp
3 WHERE sal >= 1100
4 OR funcao = 'CONTABILISTA';
```

NEMP	NOME	FUNCAO	SAL
7839	KING	PRESIDENTE	5000
7698	BLAKE	GESTOR	2850
7782	CLARK	GESTOR	2450
7566	JONES	GESTOR	2975
7654	MARTIN	VENDEDOR	1250
...			
7900	JAMES	CONTABILISTA	950
...			

14 rows selected.



Usar o Operador NOT

```
SQL> SELECT nome, funcao
2 FROM emp
3 WHERE funcao NOT IN ('CONTABILISTA', 'GESTOR', 'ANALISTA');
```

NOME	FUNCAO
KING	PRESIDENTE
MARTIN	VENDEDOR
ALLEN	VENDEDOR
TURNER	VENDEDOR
WARD	VENDEDOR



Regras de Precedência

Ordem de cálculo	Operador
1	Todos os operadores de comparação
2	NOT
3	AND
4	OR

- ▶ Alterar as regras de precedência usando parênteses.



Regras de Precedência

```
SQL> SELECT nome, funcao, sal
2 FROM emp
3 WHERE funcao= 'VENDEDOR'
4 OR funcao= 'PRESIDENTE'
5 AND sal>1500;
```

NOME	FUNCAO	SAL
-----	-----	-----
KING	PRESIDENTE	5000
MARTIN	VENDEDOR	1250
ALLEN	VENDEDOR	1600
TURNER	VENDEDOR	1500
WARD	VENDEDOR	1250



Regras de Precedência

- Uso de parênteses para forçar a prioridade.

```
SQL> SELECT  nome, funcao, sal
2 FROM      emp
3 WHERE      (funcao='VENDEDOR'
4 OR         funcao='PRESIDENTE')
5 AND        sal>1500;
```

NOME	FUNCAO	SAL
KING	PRESIDENTE	5000
ALLEN	VENDEDOR	1600



Clausula ORDER BY

- ▶ Ordena as linhas com a clausula ORDER BY
 - ▶ ASC: ordem ascendente, default
 - ▶ DESC: ordem descendente
- ▶ A clausula ORDER BY vem no fim da instrução SELECT.

```
SQL> SELECT  nome, funcao, ndep, data_entrada
2 FROM      emp
3 ORDER BY  data_entrada;
```

NOME	FUNCAO	NDEP	DATA_ENTRADA
SMITH	CONTABILISTA	20	17-DEC-80
ALLEN	VENDEDOR	30	20-FEB-81
...			
14 rows selected.			



Ordenar por Ordem Descendente

```
SQL> SELECT  nome, funcao, ndep, data_entrada
2  FROM      emp
3  ORDER BY  data_entrada DESC;
```

NOME	FUNCAO	NDEP	DATA_ENTRADA
ADAMS	CONTABILISTA	20	12-JAN-83
SCOTT	ANALISTA	20	09-DEC-82
MILLER	CONTABILISTA	10	23-JAN-82
JAMES	CONTABILISTA	30	03-DEC-81
FORD	ANALISTA	20	03-DEC-81
KING	PRESIDENTE	10	17-NOV-81
MARTIN	VENDEDOR	30	28-SEP-81
...			

14 rows selected.



Ordenar por Alias da Coluna

```
SQL> SELECT  nemp, nome, sal*12 salano
2  FROM      emp
3  ORDER BY  salano;
```

NEMP	NOME	SALANO
7369	SMITH	9600
7900	JAMES	11400
7876	ADAMS	13200
7654	MARTIN	15000
7521	WARD	15000
7934	MILLER	15600
7844	TURNER	18000
...		

14 rows selected.



Ordenar por Colunas Múltiplas

- ▶ A ordem da lista ORDER BY é a ordem da ordenação.

```
SQL> SELECT  nome, ndep, sal
2  FROM      emp
3  ORDER BY  ndep, sal DESC;
```

NOME	NDEP	SAL
-----	-----	-----
KING	10	5000
CLARK	10	2450
MILLER	10	1300
FORD	20	3000
...		

14 rows selected.

- Pode ordenar por uma coluna que não esteja na lista do SELECT.



Sumário

```
SELECT      [DISTINCT] {*| column [alias], ...}
FROM        table
[WHERE      condition(s)]
[ORDER BY  {column, expr, alias} [ASC|DESC]];
```

