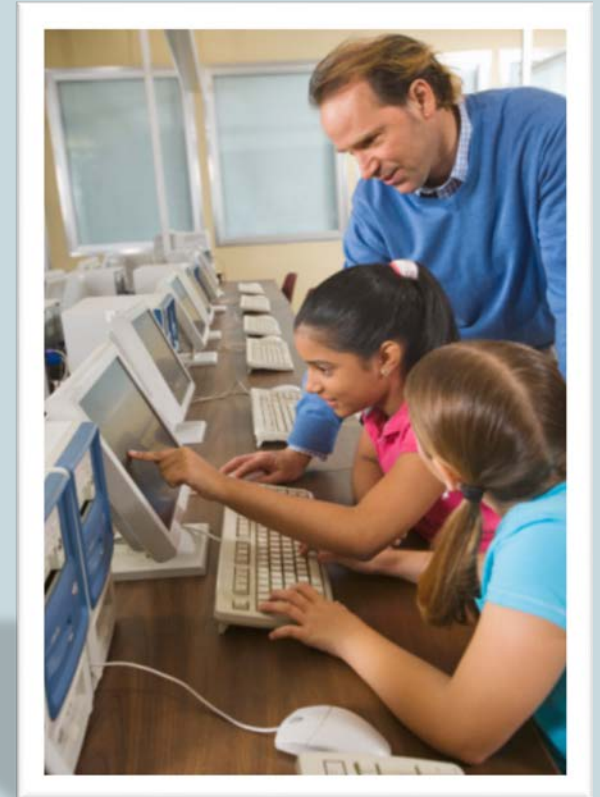




Programação de Banco de Dados com SQL

8-2

COUNT, DISTINCT, NVL



Objetivos

Esta lição abrange os seguintes objetivos:

- Construir e executar uma consulta SQL usando a função de grupo COUNT
- Usar DISTINCT e a função NVL com funções de grupo



Finalidade

- A capacidade de agregar (agrupar) dados usando funções SQL permite às empresas fazer cálculos que, do contrário, teriam de ser feitos manualmente.
- Lembra do exemplo no qual era preciso contar todos os alunos da sua escola? Uma tarefa difícil!
- Não existem mãos suficientes para fazer essa conta.
- Felizmente, as funções de grupo SQL conseguem processar facilmente esse tipo de solicitação.

COUNT

- COUNT(expressão) retorna o número de valores não nulos na coluna de expressões.

```
SELECT COUNT(job_id)
FROM employees;
```

COUNT(JOB_ID)
20

COUNT e Valores NULOS

- Vinte linhas de funcionários estão listadas na tabela de funcionários. Se você selecionar `commission_pct`, vinte linhas serão retornadas.
- O acréscimo de uma função `COUNT` à consulta retornou somente quatro linhas.
- `COUNT` conta especificamente a coluna `commission_pct`, mas ignora os valores nulos que ela contém.

```
SELECT commission_pct  
FROM employees;
```

20 linhas retornadas em 0,01 segundos

```
SELECT COUNT(commission_pct)  
FROM employees;
```

COUNT(COMMISSION_PCT)
4

Contar Todas as Linhas

- COUNT(*) retorna o número de linhas em uma tabela.
- Ele não especifica uma coluna (que pode ou não conter valores nulos), apenas conta o número de linhas retornadas no conjunto de resultados.
- Por exemplo, para descobrir quantos funcionários foram contratados antes de 01/Jan/1996, pode-se usar COUNT na instrução SELECT.

```
SELECT COUNT(*)  
FROM employees  
WHERE hire_date < '01-Jan-1996 ';
```

COUNT (*)
9

Contar Todas as Linhas

- Usamos COUNT(*) quando queremos garantir a contagem de todas as linhas (incluindo duplicatas), bem como daquelas que têm valores nulos em uma ou mais colunas.

```
SELECT COUNT(*)  
FROM employees  
WHERE hire_date < '01-Jan-1996';
```

COUNT (*)
9

DISTINCT

- A palavra-chave DISTINCT é usada para retornar somente valores não duplicados ou combinações de valores não duplicados em uma consulta.
- Examine a consulta abaixo.
- Sem o uso da palavra-chave DISTINCT, a consulta retornou todos os valores de id_cargo da tabela de funcionários, incluindo os valores duplicados.

```
SELECT job_id  
FROM employees;
```

JOB_ID
CO_CONTAD
CO_GER
AD_ASST
AD_PRES
AD_VP
AD_VP
TI_PROG
...

20 linhas retornadas em 0,01 segundos

Exemplo de DISTINCT

- Para eliminar linhas duplicadas, use a palavra-chave DISTINCT conforme mostrado aqui.
- O uso da palavra-chave DISTINCT retornou todos os IDs de cargo exatamente uma vez, sem valores duplicados.

```
SELECT DISTINCT job_id  
FROM employees;
```

JOB_ID
CO_CONTAD
CO_GER
AD_ASST
AD_PRES
AD_VP
TI_PROG
MK_GER
...

12 linhas retornadas em 0,01 segundos

DISTINCT: Valores Não Duplicados

- A palavra-chave DISTINCT, quando usada em uma consulta que seleciona mais de uma coluna, retornará as combinações não duplicadas das colunas selecionadas.
- Examine o conjunto de resultados mostrado aqui.
- Observe que não há duplicatas na combinação entre job_id e department_id, mesmo que elas existam nas colunas separadas.

```
SELECT DISTINCT job_id,  
department_id  
FROM employees;
```

JOB_ID	DEPARTMENT_ID
TI_PROG	60
VE_REP	80
AL_GER	50
AD_VP	90
AD_ASST	10
MK_GER	20
MK_REP	20
VE_GER	80
VE_REP	-
...	...

13 linhas retornadas em 0,01 segundos

Usando DISTINCT

- A palavra-chave DISTINCT pode ser usada com todas as funções de grupo.
- O uso de DISTINCT faz a função considerar somente valores não duplicados.
- As duas instruções à direita produzem resultados diferentes porque a segunda considera somente uma ocorrência de 17000

```
SELECT SUM(salary)
FROM employees
WHERE department_id = 90;
```

SALARY	SUM(SALARY)
24000	58000
17000	
17000	
....	

```
SELECT SUM(DISTINCT salary)
FROM employees
WHERE department_id = 90;
```

SALARY	SUM(DISTINCT SALARY)
24000	41000
17000	
17000	
....	

DISTINCT e COUNT

- Quando se usa DISTINCT com uma função de grupo como COUNT, o conjunto de resultados retorna o número de valores de coluna não duplicados.

```
SELECT COUNT (DISTINCT job_id)
FROM employees;
```

COUNT (DISTINCT job_id)
12

Quantos cargos diferentes estão atribuídos aos funcionários?

```
SELECT COUNT (DISTINCT salary)
FROM employees;
```

COUNT (DISTINCT salary)
18

Quantos salários diferentes são pagos aos funcionários?

NVL

- Às vezes, deseja-se incluir valores nulos em funções de grupo.
- Por exemplo, saber a média de pedidos de clientes servidos por dia ajudaria a julgar quanta comida é necessário encomendar por mês.
- Em certos dias, o restaurante está fechado e nenhum cliente é servido, mas o proprietário percebeu que calcular a média incluindo os dias que o restaurante está fechado é um indicador melhor do que apenas contar os dias em que há clientes.

NVL

- A instrução SELECT para incluir valores nulos poderia ser gravada começando com:

```
SELECT AVG(NVL(customer_orders, 0))
```

- Outro exemplo na tabela de funcionários:

```
SELECT AVG(commission_pct)  
FROM employees;
```

AVG(COMMISSION_PCT)
.2125

```
SELECT AVG(NVL(commission_pct, 0))  
FROM employees;
```

AVG(NVL(COMMISSION_PCT,0))
.0425

NVL

- Compare os resultados das duas consultas a seguir.

```
SELECT AVG(commission_pct)
FROM employees;
```

AVG(COMMISSION_PCT)
.2125

```
SELECT AVG(NVL(commission_pct, 0))
FROM employees
```

AVG(NVL(COMMISSION_PCT,0))
.0425

Terminologia

Estes são os principais termos usados nesta lição:

- Agregar
- COUNT (expressão)
- COUNT (DISTINCT expressão)
- DISTINCT

Resumo

Nesta lição, você deverá ter aprendido a:

- Construir e executar uma consulta SQL usando a função de grupo COUNT
- Usar DISTINCT e a função NVL com funções de grupo

