

Consultas simples para recuperação de informações em SQL

Jackson Antonio do Prado Lima (jacksonpradolima at gmail dot com)

1. O que foi visto na última aula?
2. Consultas simples em SQL
3. O que teremos na próxima aula?
4. Exercícios

O que foi visto na última aula?

Na última aula

1. O que é SQL (*Structured Query Language*)
 - ❖ Uma linguagem de BD
 - ❖ Instruções para definições de dados, consultas e atualizações
2. Comandos de DDL (*Data Definition Language*) da SQL para criação, modificação, remoção de tabelas e seus atributos, além dos tipos de dados básicos em SQL
3. Comandos de DML (*Data Manipulation Language*) da SQL para inserção, atualização e deleção de tuplas das tabelas

Consultas simples em SQL

Contexto

- ❖ Alguns autores costumam denominar como DQL (*Data Query Language*)
- ❖ Instrução básica de recuperação de informações do BD: **SELECT**

SELECT-FROM-WHERE

Forma básica

```
SELECT <lista atributos>  
FROM <lista tabelas>  
WHERE <condição>
```

onde:

- ❖ **lista atributos** é uma lista de nomes de atributo cujos valores devem ser recuperados pela consulta.
- ❖ **lista tabelas** é uma lista dos nomes de relação exigidos para processar a consulta.
- ❖ **condição** é uma expressão condicional (booleana) que identifica as tuplas a serem recuperadas pela consulta.

SELECT-FROM-WHERE

PRODUTO

| id | pnome | preco | categoria | quantidade |
|----|-----------|--------|-------------|------------|
| 1 | lapis | 19.99 | papelaria | 30 |
| 2 | lapiseira | 29.99 | papelaria | 55 |
| 3 | camera | 103.99 | fotografia | 10 |
| 4 | televisao | 203.99 | eletronicos | 15 |

SELECT-FROM-WHERE

Exemplo 1

```
SELECT *  
FROM Produto
```

| id | pnome | preco | categoria | quantidade |
|----|-----------|--------|-------------|------------|
| 1 | lapis | 19.99 | papelaria | 30 |
| 2 | lapiseira | 29.99 | papelaria | 55 |
| 3 | camera | 103.99 | fotografia | 10 |
| 4 | televisao | 203.99 | eletronicos | 15 |

SELECT-FROM-WHERE

Exemplo 2

```
SELECT *  
FROM Produto  
WHERE categoria = 'papeleria'
```

| id | pnome | preco | categoria | quantidade |
|----|-----------|-------|-----------|------------|
| 1 | lapis | 19.99 | papelaria | 30 |
| 2 | lapiseira | 29.99 | papelaria | 55 |

SELECT-FROM-WHERE

Exemplo 3

```
SELECT *  
FROM Produto  
WHERE preco > 100
```

| id | pnome | preco | categoria | quantidade |
|----|-----------|--------|-------------|------------|
| 3 | camera | 103.99 | fotografia | 10 |
| 4 | televisao | 203.99 | eletronicos | 15 |

Eliminação de Duplicações

Devido a não haver eliminação de duplicatas no **SELECT** utilizamos o **DISTINCT**.

```
SELECT DISTINCT categoria  
FROM Produto
```

| categoria |
|-------------|
| papelaria |
| fotografia |
| eletronicos |

```
SELECT categoria  
FROM Produto
```

| categoria |
|-------------|
| papelaria |
| papelaria |
| fotografia |
| eletronicos |

Ordenação do Resultado

- ❖ A ordenação é ascendente a não ser que a palavra **DESC** seja especificada.
- ❖ No caso de igualdade no primeiro atributo da cláusula **ORDER BY**, o segundo atributo é utilizado e assim por diante.

```
SELECT pnome, preco, quantidade  
FROM Produto  
WHERE categoria='papeleria' AND preco > 50  
ORDER BY preco DESC, pnome ASC
```

Exercício de fixação

1. O resultado ao aplicar a consulta: `SELECT categoria FROM Produto ORDER BY categoria`
2. O resultado ao aplicar a consulta: `SELECT DISTINCT categoria FROM Produto ORDER BY categoria DESC`

Facilidades na projeção dos resultados

- ❖ Uso de aritmética nas consultas (+, -, *, /)
- ❖ Renomeação

Exemplo

```
SELECT pnome AS Nome, (preco*quantidade) AS Total  
FROM Produto
```

| Nome | Total |
|-----------|---------|
| lapis | 599.7 |
| lapiseira | 1649.45 |
| camera | 1039.9 |
| televisao | 3059.85 |

Funções de agregação

SQL dá suporte a diversas operações de agregação:

- ❖ **SUM**: soma dos valores de um atributo
- ❖ **MIN**: valor mínimo de um atributo
- ❖ **MAX**: valor máximo de um atributo
- ❖ **AVG**: média dos valores de um atributo
- ❖ **COUNT**: contagem de ocorrências [de um atributo]

Observações:

- ❖ Com exceção de **COUNT**, todas as outras operações de agregação aplicam-se a um único atributo.
- ❖ **COUNT** não conta valores nulos.

Funções de agregação

Exemplo

```
SELECT COUNT(pnome) AS count_nome, MIN(preco) AS min_preco,  
MAX(preco) AS max_preco, AVG(quantidade) AS avg_qtd, SUM(quantidade),  
sum_qtd  
FROM Produto
```

| count_nome | min_preco | max_preco | avg_qtd | sum_qtd |
|------------|-----------|-----------|---------|---------|
| 4 | 19.99 | 203.99 | 27.5 | 110 |

Funções de agregação

Exemplo com **DISTINCT**

```
SELECT COUNT(DISTINCT pnome) AS count_nome  
FROM Produto
```

Incrementando a cláusula WHERE

- ❖ Busca por padrões: **Cláusula [NOT] LIKE**
- ❖ Teste de existência de valores nulos: **Cláusula IS [NOT] NULL**
- ❖ Busca por intervalos de valores: **Cláusula [NOT] BETWEEN valor1 AND valor2**
- ❖ Teste de pertinência elemento-conjunto: **Cláusula [NOT] IN**

Incrementando a cláusula WHERE

LIKE pode ser traduzido como uma busca em um atributo que “contém”, “começa” ou “termina” com um padrão, onde:

- ❏ % = qualquer sequência de caracteres
- ❏ _ = um único carácter

Entretanto, não é possível testar padrões em atributos DATETIME.

Exemplo

```
SELECT pnome
FROM Produto
WHERE pnome LIKE 'p%' OR pnome LIKE '_a%' OR pnome LIKE '%p%'
OR pnome LIKE '%a'
```

Incrementando a cláusula WHERE

Exemplo valores não nulos

```
SELECT pnome  
FROM Produto  
WHERE pnome IS NOT NULL
```

Exemplo intervalo de valores

```
SELECT pnome  
FROM Produto  
WHERE quantidade BETWEEN 10 and 100
```

Exemplo pertinência elemento-conjunto

```
SELECT pnome  
FROM Produto  
WHERE id IN (1,3)
```

O que teremos na próxima aula?

The complex queries are coming...

- ❖ Consultas com mais de uma tabela: Junções em SQL
 - ❖ União, Interseção, Diferença
 - ❖ Subconsultas
 - ❖ INTERSECT, EXCEPT e EXISTS
- ❖ Agrupamento
 - ❖ GROUP BY
 - ❖ HAVING
- ❖ Aula prática no laboratório
- ❖ Trabalho com entrega de relatório

Exercícios

Exercícios para entrega

Entregar até a próxima aula a Lista de exercícios 4.

Bibliografia

Bibliografia

Básica:

ELMASRI, Ramez, NAVATHE, Shamkant B. Sistemas de banco de dados. 6 ed., 2011.
Cap 4.

Total de exemplares disponíveis: 9 (CEPLAN), 10 (CCT), 2 (CEAVI).

Leitura:

Capítulo 3, “Simple Queries” do livro **SQL for Web Nerds**, de Philip Greenspun
<http://philip.greenspun.com/sql/>

Consultas simples para recuperação de informações em SQL

Obrigado!

Jackson Antonio do Prado Lima (jacksonpradolima at gmail dot com)