

Comando SELECT

Agradecimentos aos professores
Caetano Traina Jr, Enzo Seraphim,
Humberto Razente e Maria Camilla
N. Barioni Razente, que cederam
partes deste material.

- Permite a realização de consultas em um banco de dados
- Forma básica de um comando SELECT


```
SELECT <lista de atributos>
FROM <lista de tabelas>
WHERE <condições>
```
- Onde
 - <lista de atributos>: possui os nomes dos atributos cujos valores devem ser retornados pela consulta
 - <lista de tabelas>: possui os nomes das tabelas necessárias para executar a consulta
 - <condição>: possui uma expressão condicional (booleana) que identifica as tuplas que devem ser retornadas pela consulta

1

Relações usadas nos exemplos

```
AUTOR = { COD_AUTOR, NOME, NASCIMENTO }
      ↑
ESCREVE = { COD_AUTOR, COD_LIVRO }
      ↙
LIVRO = { COD_LIVRO, TITULO, VALOR, VOLUME,
          COD_EDITORA }
      ↓
EDITOR = { COD_EDITORA, RAZAO, ENDereco,
           CIDADE }
```

2

Comando SELECT

Exemplos

```
/* Selecionar todos os nomes e datas de nascimento da
tabela autor */
SELECT NOME, NASCIMENTO
FROM AUTOR;

/* Selecionar todos os campos da tabela autor */
SELECT * FROM AUTOR;

/* Selecionar todos os livros cujos valores sejam
maiores que R$24,00 */
SELECT TITULO, VALOR
FROM LIVRO
WHERE VALOR > 24;
```

3

Comando SELECT

Para tratar nomes de atributos ambíguos

```
/* Selecionar todos os livros da editora TABAJARA */
SELECT LI.TITULO
FROM LIVRO LI, EDITORA ES
WHERE ES.RAZAO = 'TABAJARA' AND
      LI.COD_EDITORA = ES.COD_EDITORA;
```

Renomeação de atributos

```
/* Selecionar todos os nomes e datas de nascimento da
tabela autor renomeando a coluna nome para autor*/
SELECT NOME AS AUTOR, NASCIMENTO
FROM AUTOR;
```

4

Comando SELECT

Para tratar tuplas duplicadas

```
/* Selecionar os valores de todos os livros */
SELECT ALL VALOR
FROM LIVRO;

/* Selecionar apenas os valores de livros
distintos */
SELECT DISTINCT VALOR
FROM LIVRO;
```

5

Cláusula WHERE

Especifica quais tuplas das tabelas listadas na cláusula FROM são afetados por uma instrução SELECT, UPDATE ou DELETE

Se uma cláusula WHERE não for especificada, a consulta retornará todas as linhas da tabela

Exemplo

```
/* Selecionar as editoras da cidade de São Paulo */
SELECT RAZAO
FROM EDITORA
WHERE CIDADE = 'SÃO PAULO';
```

6

Cláusula WHERE

Exemplos

```
/* Selecionar todos os títulos dos livros da editora
TABAJARA renomeando as tabelas */
SELECT LI.TITULO
FROM EDITORA ED, LIVRO LI
WHERE (ED.RAZAO = 'TABAJARA') AND
      (ED.COD_EDITORA = LI.COD_EDITORA);

/* Selecionar os títulos da tabela livro e os nomes
da tabela autor */
SELECT AU.NOME, LI.TITULO
FROM AUTOR AU, LIVRO LI, ESCRIVE ES
WHERE (AU.COD_AUTOR = ES.COD_AUTOR_ESC) AND
      (LI.COD_LIVRO = ES.COD_LIVRO_ESC);
```

7

Cláusula WHERE

Exercício:

- Especifique uma consulta que selecione todos os livros escritos por Fulano de Tal

```
SELECT AU.NOME, LI.TITULO
FROM AUTOR AU, ESCRIVE ES, LIVRO LI
WHERE (AU.NOME = 'Fulano de Tal') AND
      (AU.COD_AUTOR = ES.COD_AUTOR_ESC) AND
      (ES.COD_LIVRO_ESC = LI.COD_LIVRO);
```

8

Outras Cláusulas

```
SELECT <lista de atributos>
FROM <lista de tabelas>
WHERE <condições>
GROUP BY <atributos de agrupamento>
HAVING <condição de agrupamento>
ORDER BY <lista de atributos>
```

9

Cláusula ORDER BY

- Classifica as tuplas resultantes de uma consulta em ordem crescente ou decrescente de acordo com uma lista de atributos especificada
- As tuplas são classificadas pelo primeiro atributo listado após ORDER BY
- As tuplas que possuam valores iguais no primeiro atributo são então classificadas pelo valor do segundo atributo listado e assim por diante

10

Cláusula ORDER BY

Exemplos

```
/* Selecione nome e nascimento da tabela autor
em ordem dos autores mais novos */
SELECT NOME, NASCIMENTO
FROM AUTOR
ORDER BY NASCIMENTO DESC;

/* Selecione título e valor da tabela livro em
ordem dos valores mais baixos */
SELECT TITULO, VALOR
FROM LIVRO
ORDER BY VALOR ASC;
```

11

Características Adicionais

- Existem várias características adicionais que permitem a especificação de consultas mais elaboradas como
 - Operadores
 - =, !=, >, <, >=, <=
 - IN
 - BETWEEN...AND
 - LIKE
 - IS NULL ou IS NOT NULL
 - Funções
 - AVG, COUNT, MIN, MAX e SUM

12

Operadores =, !=, >, <, >=, <=

Exemplos:

```
/* Selecionar os livros que não sejam da editora com
COD_EDITORA igual a 1 */
SELECT *
FROM LIVRO
WHERE COD_EDITORA != 1;

/* Selecionar os livros cujo VALOR seja maior ou igual
a R$ 24,00 */
SELECT *
FROM LIVRO
WHERE VALOR >= 24;
```

13

Operador IN

- Determina se o valor de uma expressão é igual a algum dos valores em uma lista especificada
- Se o valor da expressão for encontrado na lista de valores, o operador IN retornará True; caso contrário, retornará False
- Exemplo:


```
/* Selecionar as editoras localizadas nas cidades de
São Paulo, São Carlos e Rio de Janeiro */
SELECT *
FROM EDITORA
WHERE CIDADE IN ('SÃO PAULO', 'SÃO CARLOS', 'RIO DE
JANEIRO');
```

14

Operador BETWEEN...AND

- Determina se o valor de uma expressão se encontra dentro de um intervalo especificado de valores
- Se o valor da expressão estiver entre valor1 e valor2 (inclusive), o operador BETWEEN...AND retornará True; caso contrário, retornará False
- Exemplo:


```
/* Selecionar livros com valores entre R$ 30,00 e
R$ 60,00 */
SELECT TITULO, VALOR
FROM LIVRO
WHERE VALOR BETWEEN 30 AND 60;
```

15

Operador LIKE

- Compara uma expressão formada por uma sequência de caracteres com um padrão, formado por partes de cadeias de caracteres, em um comando SQL
- Padrões podem ser especificados
 - '%' (substitui um número arbitrário de caracteres)
 - '_' (substitui um único caractere)

16

Operador LIKE

- Exemplo:


```
/* Selecionar nome de autor que inicia com
a letra 'J' */
SELECT NOME
FROM AUTOR
WHERE NOME LIKE 'J%';
```

17

Operador IS NULL

- Determina se o valor de uma expressão é nulo
- Exemplo:


```
/* Selecionar as editoras sem endereço*/
SELECT RAZAO
FROM EDITORA
WHERE ENDEREÇO IS NULL;
```

18

Função AVG

- Calcula a média aritmética de um conjunto de valores contido em um campo especificado de uma consulta
- Exemplo:

```
/* Média de valores dos livros */
SELECT AVG (VALOR) AS MEDIA
FROM LIVRO;
```

19

Cláusula GROUP BY

- Agrupar as tuplas que possuam valores idênticos para algum(s) atributo(s) especificado(s)
- Um valor de resumo é criado para cada tupla se for incluída uma função agregada SQL, como SUM ou COUNT, no comando SELECT
- Exemplo:

```
/* Média de valores dos livros de cada editora */
SELECT ED.RAZAO, AVG (LI.VALOR) AS MEDIA
FROM EDITORA ED, LIVRO LI
WHERE ED.COD_EDITORA = LI.COD_EDITORA
GROUP BY ED.RAZAO;
```

20

Função COUNT

- Calcula o número de tuplas retornado por uma consulta
- A função COUNT não conta tuplas que tenham campos Null, exceto quando a expressão for o caractere curinga asterisco (*)
- Se a expressão identificar vários campos, a função COUNT contará uma tupla somente se um dos campos não for Null

21

Função COUNT

- Exemplo:
- ```
/* Contar autores que nasceram antes de 01/01/1960 */
SELECT COUNT (*)
FROM AUTOR
WHERE NASCIMENTO < '1/1/1960';
```
- A função COUNT também pode ser usada para contar valores em um campo no resultado de uma consulta
  - Exemplo:

```
/* Contar os valores de livros distintos */
SELECT COUNT (DISTINCT VALOR)
FROM LIVRO;
```

22

## Função COUNT

- Exemplos:
- ```
/* Contar quantos livros não tem volume com null */
SELECT COUNT (VOLUME)
FROM LIVRO;
```
- ```
/* Contar total de editoras */
SELECT COUNT (*)
FROM EDITORA;
```
- ```
/* Contar os livros publicados por cada editora */
SELECT ED.RAZAO, COUNT (LI.COD_EDITORA)
FROM EDITORA ED, LIVRO LI
WHERE ED.COD_EDITORA = LI.COD_EDITORA
GROUP BY ED.RAZAO;
```

23

Cláusula HAVING

- Especifica quais tuplas agrupadas são exibidas no comando SELECT com uma cláusula GROUP BY
- Depois que GROUP BY combina as tuplas, apenas as tuplas que satisfazem às condições da cláusula HAVING são exibidas
- Exemplo:

```
/* Selecionar as editoras que possuam mais de 2 livros publicados */
SELECT ED.RAZAO, COUNT (LI.COD_EDITORA)
FROM EDITORA ED, LIVRO LI
WHERE ED.COD_EDITORA = LI.COD_EDITORA
GROUP BY ED.RAZAO;
HAVING COUNT (LI.COD_EDITORA) > 2;
```

24

Funções MIN, MAX

- Retornam o mínimo ou o máximo de um conjunto de valores contido em um campo especificado em uma consulta
- Exemplos:


```
/* Selecionar o valor do livro mais caro */
SELECT MAX (VALOR) AS VR_ALTO
FROM LIVRO;

/* Selecionar o valor do livro mais barato */
SELECT MIN (VALOR) AS VR_BARATO
FROM LIVRO;
```

25

Função SUM

- Retorna a soma de um conjunto de valores contido em um campo especificado em uma consulta
- A função SUM ignora os registros que contenham campos Null
- Exemplo:


```
/* Soma dos valores dos livros publicados pela
editora com COD_EDITORA = 1 */
SELECT SUM (VALOR) AS SOMA_EDITORA
FROM LIVRO
WHERE COD_EDITORA = 1;
```

26

Função SUM

- Exemplo:


```
/* Somar os valores dos livros por cada editora */
SELECT ED.RAZAO, SUM (LI.VALOR)
FROM EDITORA ED, LIVRO LI
WHERE ED.COD_EDITORA = LI.COD_EDITORA
GROUP BY ED.RAZAO;
```

27

Subconsultas - SELECTs aninhados

- São blocos SELECT...FROM...WHERE completos dentro da cláusula WHERE de outra consulta
- Exemplo:


```
/* Selecionar o título com valor mais alto */
SELECT TITULO, VALOR
FROM LIVRO
WHERE VALOR IN (
  SELECT MAX (VALOR)
  FROM LIVRO );
```

28

Subconsultas - SELECTs aninhados

- Exemplo:


```
/* Selecionar os títulos de livros da editora com
COD_EDITORA = 2 que possuam valor menor do que todos
os livros da editora com COD_EDITORA = 1 */
SELECT TITULO, VALOR
FROM LIVRO
WHERE COD_EDITORA = 2 AND
  VALOR < ALL (SELECT VALOR
               FROM LIVRO
               WHERE COD_EDITORA = 1);
```
- Operador ALL: Compara um valor a todos os valores retornados por uma consulta

29

Subconsultas - SELECTs aninhados

- Exercícios:
 - Especifique uma consulta que selecione o nome e a data de nascimento do autor mais velho
 - Modifique a consulta anterior para selecionar o nome e a data de nascimento do(s) autor(es) mais velho(s) do que o(s) autor(es) da editora com COD_EDITORA = 1

```
SELECT NOME, NASCIMENTO
FROM AUTOR
WHERE NASCIMENTO <= ALL (SELECT NASCIMENTO
                        FROM AUTOR);

SELECT NOME, NASCIMENTO
FROM AUTOR
WHERE NASCIMENTO <= ALL (SELECT AU.NASCIMENTO
                        FROM AUTOR AU, EDITORA ES, LIVRO LI
                        WHERE (AU.COD_AUTOR = ES.COD_AUTOR_ESC) AND
                              (ES.COD_LIVRO_ESC = LI.COD_LIVRO) AND
                              (LI.COD_EDITORA = 1));
```

30

Subconsultas - SELECTs aninhados

■ Exercícios:

- 3) Especifique uma consulta que retorne os nomes dos autores que possuam apenas um livro registrado no sistema

```
SELECT NOME AS ESCRITOR
FROM AUTOR
WHERE COD_AUTOR IN (SELECT COD_AUTOR_ESC
                     FROM ESCRIVE
                     GROUP BY COD_AUTOR_ESC
                     HAVING COUNT(COD_AUTOR_ESC) = 1);
```