INSTITUTO FEDERAL
ESPÍRITO SANTO

BANCOS DE DADOS

BD-I

Prof. Moisés Omena

INSTITUTO FEDERAL
ESPÍRITO SANTO

SELEÇÃO DE DADOS

DQL (Data Query Language)

- Select: **Seleção** dados (múltiplas linhas)

```
SELECT   *
FROM <table_name>
```

Execução de código:

```
select * from projeto;
```

Exercícios iniciais

Utilize a declaração SELECT para verificar se os dados inseridos anteriormente em cada uma das tabelas estão corretos

1) Selecione todos os campos e registros da tabela projeto
2) Selecione todos os campos e registros da tabela departamento_projeto
3) Selecione todos os campos e registros da tabela empregado
4) Selecione todos os campos e registros da tabela empregado_projeto
5) Selecione todos os campos e registros da tabela departamento
6) Selecione todos os campos e registros da tabela dependente
7) Selecione todos os campos e registros da tabela historico_salario

Filtrando dados com operadores de comparação

=	→	igualdade
<>	→	diferença
<=	→	menor ou igual
<	→	menor
>=	→	maior ou igual
>	→	maior
IS NULL	→	Valores nulos
IS NOT NULL	→	Valores não nulos

DQL (Data Query Language)

- Select: **Seleção** dados (múltiplas linhas)

```
SELECT  select_list
FROM <table_name>
[ WHERE search_conditions]
```

DQL (Data Query Language)

- Select: **Seleção** dados (múltiplas linhas)

SELECT [all | distinct] select_list
[**FROM** <table_name>
[**WHERE** search_conditions]

Execução de código:

```
select * from projeto where numero=5;
OU
select localizacao from projeto where numero>5;
OU
select * from empregado where RG_SUPERVISOR is not null;
```

Obs:

- Campos do tipo data e tipo money devem aparecer no where entre aspas simples

Exercícios

(obs: quando necessário, visualize todos os dados da tabela para conferir os resultados)

1) Selecione todos os registros dos campos (atributos) nome, localização e número da tabela projeto (mostre os atributos nessa ordem).
2) selecione todos os registros e campos (atributos) da tabela projeto onde a localização é igual a "Vitória"
3) Mostre nome e número, para os projetos onde a localização consta como "Vitória"
4) Mostre todos campos (atributos) e registros da tabela projeto com número maior que 10
5) Mostre todos os campos e registros da tabela empregado com salário acima de 3000.
6) Mostre todos os campos e registros da tabela empregado onde RG_SUPERVISOR é igual a 1010.
7) Mostre todos os campos e registros da tabela empregado onde o campo RG corresponde a 4040
8) Mostre todos os campos e registros da tabela historico_salario onde o campo RG corresponde a 4040
9) Mostre todos os campos e registros da tabela empregado onde RG_SUPERVISOR é menor que 2020.
10) selecione todos os campos e registros da tabela projeto com valores de número maior ou igual a 10

Exercícios

11) Mostre todos os registros da tabela empregado onde RG_SUPERVISOR é menor ou igual 2020.
12) Mostre todos os registros da tabela empregado onde RG_SUPERVISOR é nulo
13) Mostre todos os registros da tabela empregado onde RG_SUPERVISOR não é nulo
14) Mostre todos os registros da tabela empregado onde DEPTO é diferente de 2
15) Mostre o campos nome e cpf e depto da tabela empregado onde o campo DEPTO corresponde a 2
16) Mostre o campos rg e nome tabela empregado onde o campo DEPTO corresponde a 1
17) Mostre apenas os nomes dos empregados que possuem salario maior que 5500
18) Mostre apenas os nomes dos empregados que possuem salario maior ou igual 5500
19) Mostre todos os empregados que data de salario inicial maior que 01-02-2012
20) Mostre todos os empregados que data de salario inicial maior que 01-05-2012
21) Mostre apenas os nomes dos empregados que data de salario inicial maior ou igual a 01-05-2012
22) Mostre apenas os cpfs dos empregados que data de salario inicial maior ou igual a 01-05-2012

INSTITUTO FEDERAL
ESPÍRITO SANTO

ATUALIZAÇÃO, EXCLUSÃO E CÓPIA DE DADOS

DML (Data Manipulation Language)

- UPDATE: **Atualização** de dados (múltiplas linhas)

UPDATE <table_name>
SET column_name = {expression | null }
[, column_name = {expression | null}]
[**WHERE** search_conditions]

Definições:
table_name → nome da tabela que sofrerá a alteração
column_name→ nome da coluna que sofrerá alteração
search_condition → indica as restrições que deverão ser aplicadas ao conjunto de linhas envolvidas de forma a restringir o universo de ação do comando UPDATE.

DML (Data Manipulation Language)

- UPDATE: **Atualização** de dados (múltiplas linhas)

```
UPDATE <table_name>
SET column_name = {expression | null }
    [, column_name = {expression | null}]
[ WHERE search_conditions ]
```

Execução de código:

```
UPDATE projeto set nome = 'Aguas limpas' where numero=25;
OU
UPDATE dependente set sexo='M' where codigo=5;
```

DML (Data Manipulation Language)

- DELETE: **Exclusão** de dados (múltiplas linhas)

```
DELETE  FROM {table_name}
[ WHERE search_conditions ]
```

Definições:
table_name → nome da tabela que sofrerá a alteração/exclusão.
search_condition → indica as restrições que deverão ser aplicadas ao conjunto de linhas envolvidas de forma a restringir o universo de ação do comando DELETE.

DML (Data Manipulation Language)

- DELETE: **Exclusão** de dados (múltiplas linhas)

```
DELETE  FROM {table_name}
[ WHERE search_conditions ]
```

Execução de código:

```
DELETE FROM PROJETO WHERE NUMERO=50;
OU
DELETE FROM PROJETO WHERE NUMERO>=50;
```

Cópia de dados e estrutura de uma tabela

- Para garantir a proteção dos dados e possível restauração destes na manipulação por atualizações e deleções, vamos fazer uma cópia (backup) do dados das tabelas a serem manipuladas
- CREATE TABLE NovaTabela SELECT * FROM AntigaTabela;

DML (Data Manipulation Language)

- Copiando estrutura e dados de uma tabela

```
CREATE TABLE {new_table_name} AS SELECT * FROM
{old_table_name};
```

Execução de código:

```
CREATE TABLE nova_projeto  AS SELECT * FROM projeto;
OU
CREATE TABLE nova_projeto AS (SELECT * FROM projeto )
OU
CREATE TABLE nova_projeto AS (SELECT * FROM projeto
WHERE numero>10 )
```

DML (Data Manipulation Language)

- Inserindo dados selecionados a partir de outra tabela

```
INSERT INTO {target_table_name} (list fields target table,...)
SELECT (list fields source table,...)
FROM {source_table_name}
[WHERE search_conditions ]
```

Execução de código:

```
INSERT INTO DEPARTAMENTO_PROJETO (codigo, numero_depto,numero_projeto)
SELECT codigo, numero_depto,numero_projeto
FROM DEPARTAMENTO_PROJETO2
OU
INSERT INTO DEPARTAMENTO_PROJETO (codigo, numero_depto,numero_projeto)
SELECT codigo, numero_depto,numero_projeto
FROM DEPARTAMENTO_PROJETO2 WHERE codigo>10
```

Exercícios de inclusão, alteração e exclusão

1. Faça uma cópia das seguintes tabelas com todos os seus registros por meio da instrução create table:
    1. DEPARTAMENTO_PROJETO para DEPARTAMENTO_PROJETO2,
    2. PROJETO para PROJETO2,
    3. EMPREGADO para EMPREGADO2,
    4. DEPARTAMENTO para DEPARTAMENTO2
2. Insira na tabela correspondente os seguintes projetos
    1. Ginásio de Esportes, de código 50 e localização Serra
    2. Teatro de código 51 e localização Vitória
3. Atualize o nome do projeto Águas Limpas para “Aguas Claras”
4. Retorne com o nome do projeto Aguas Claras para “Águas Limpas”
5. Defina a localização do projeto motor igual a “Serra”
6. Apague todos os registros da tabela departamento_projeto
7. Apague todos os registros da tabela projeto (houve algum ploblema, qual? Qual a solução para a questão?)
8. Reinsira todos os registros da tabela departamento_projeto com um único comando insert
9. Elimine as tabelas de backup criadas anteriormente (tabelas com nome terminado em ‘2’)