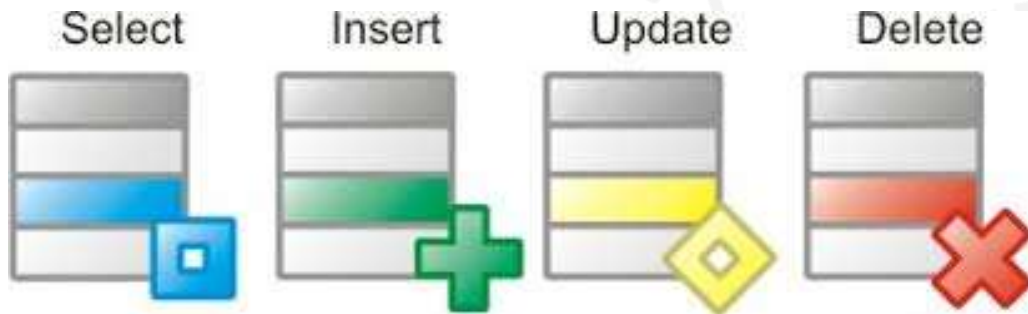


Unidade 5.2: DML (*Data Manipulation Language*): INSERT, SELECT, UPDATE e DELETE

Prof. Rodrigo Baroni

SQL-DML

- DML possui uso muito mais frequente do que DDL
- Uso de DML por SQL embutido em aplicações ou via janelas de acesso ao banco de dados
- 4 operações básicas:
 - INSERT
 - SELECT
 - UPDATE
 - DELETE



SQL-DML: INSERT

- **Sintaxe 1 - Inserção Unitária:**

INSERT INTO <tabela>

(<lista-de-colunas>)

VALUES (<lista-de-valores>)

- **Sintaxe 2 - Inserção em Massa:**

INSERT INTO <tabela1>

(<lista-de-colunas>)

SELECT ...

SQL-DML: INSERT

- **O valor dos atributos devem ser informados na mesma ordem da criação da tabelas** ou os nomes dos atributos devem ser informados.
- As regras de integridade são observadas no momento da inclusão da linha na tabela.
- São observados os campos que aceitam nulos ou que têm *default*.

```
create table Projeto
(numero number(5) constraint projeto_numdep_pk primary key,
Nome varchar2(40) constraint projeto_nome_uk unique,
Localizacao varchar2(50),
Dep_responsavel number(2) constraint projeto_dep_reponsavel_fk
references departamento(numero));
```

```
INSERT INTO projeto
VALUES (22, 'Rede Wi-Fi corporativa', 'Predio P1', 10);
```

SQL-DML: SELECT

- Característica não-procedural
- Filtros de colunas e/ou linhas
- **Sintaxe:**

SELECT <lista de atributos>

FROM <nome das tabelas>

[WHERE <condição de pesquisa / filtro>]



SQL-DML: SELECT

- **Montagem de Filtros:**

- Na expressão WHERE, especificam-se as condições para seleção das linhas da tabela. Qualquer expressão lógica envolvendo os campos das tabelas é válida. Os campos que fazem parte da expressão WHERE **não** precisam necessariamente ter sido selecionados na expressão SELECT.

TABELA Funcionario

matricula	nome	cargo	salario	cod-depto	comissao	estado	cidade
10	Ana	analista	2800	1	100	null	null
20	Jose	prog.	1500	1	200	MG	BH
30	Paulo	gerente	4000	1	100	RJ	Rio
40	Lucas	contador	1400	2	300	RJ	Rio
50	Luiz	gerente	3500	2	200	MG	Betim
60	Joel	analista	3200	1	100	ES	Vitoria

TABELA Depto

cod-depto	nome-depto
1	Informatica
2	Contabilidade

SQL-DML: SELECT

- Selecionar o matrícula e o nome dos funcionários com a comissão menor do que 150:

```
SELECT matricula,nome
```

```
FROM Funcionario
```

```
WHERE comissao < 150
```

- Selecionar todas as informações de todos os funcionários:

```
SELECT *
```

```
FROM Funcionario
```


SQL-DML: Operadores

- Comparação: =, <>, >, <, >=, <=
- Lógicos: AND, OR, NOT
- BETWEEN <expressão1> AND <expressão2>: testa intervalo
- IN (<lista de valores>): testa presença na lista
- IS NULL: testa nulo
- LIKE: testa conteúdo de *string* de caracteres

SQL-DML: Operadores

- Ex: Selecione os funcionários do estado de MG ou RJ que ganham uma comissão maior ou igual a R\$ 100,00.

Select nome

From funcionario

Where estado = 'MG' or 'RJ' and comissao >= 100

Funcionario

matricula	nome	cargo	salario	cod-depto	comissao	estado	cidade
20	Jose	prog.	1500	1	200	MG	BH
30	Paulo	gerente	4000	1	100	RJ	Rio
40	Lucas	contador	1400	2	300	RJ	Rio
50	Luiz	gerente	3500	2	200	MG	Betim

Operador BETWEEN

- Permite testar se um determinado campo assume o valor dentro de um intervalo especificado. É utilizado por ser mais prático do que o teste de \geq e \leq .

```
SELECT *  
FROM FUNCIONARIOS  
WHERE SALARIO BETWEEN 1800 AND 2000
```

(é bem mais legível do que)

```
SELECT *  
FROM FUNCIONARIOS  
WHERE SALARIO  $\geq$  1800 AND SALARIO  $\leq$  2000
```

TABELA Funcionario

matricula	nome	cargo	salario	cod-depto	comissao	estado	cidade
10	Ana	analista	2800	1	100	null	null
20	Jose	prog.	1500	1	200	MG	BH
30	Paulo	gerente	4000	1	100	RJ	Rio
40	Lucas	contador	1400	2	300	RJ	Rio
50	Luiz	gerente	3500	2	200	MG	Betim
60	Joel	analista	3200	1	100	ES	Vitoria

TABELA Depto

cod-depto	nome-depto
1	Informatica
2	Contabilidade

Operador BETWEEN

- Ex: Selecione os funcionários que ganham entre 2000 e 3500 no departamento de informática

Select nome,salario

From funcionario

Where salario between 2000 and 3500 and cod-depto = 1

Funcionario

matricula	nome	cargo	salario	cod-depto	comissao	estado	cidade
10	Ana	analista	2800	1	100	null	null
60	Joel	analista	3200	1	100	ES	Vitoria

Operador IN

- Verifica se um elemento pertence a um conjunto, isto é, se um campo assume o valor de algum membro de uma lista de valores. É utilizado para substituir grandes expressões de OR para o mesmo campo.

```
SELECT *
```

```
FROM Funcionario
```

```
WHERE cod_depto IN (1, 2)
```

(é bem mais legível do que)

```
SELECT *
```

```
FROM Funcionario
```

```
WHERE cod_depto =1 OR cod_depto =2
```

TABELA Funcionario

matricula	nome	cargo	salario	cod-depto	comissao	estado	cidade
10	Ana	analista	2800	1	100	null	null
20	Jose	prog.	1500	1	200	MG	BH
30	Paulo	gerente	4000	1	100	RJ	Rio
40	Lucas	contador	1400	2	300	RJ	Rio
50	Luiz	gerente	3500	2	200	MG	Betim
60	Joel	analista	3200	1	100	ES	Vitoria

TABELA Depto

cod-depto	nome-depto
1	Informatica
2	Contabilidade

Operador IN

- Ex: Selecione os funcionarios cujo cargo é contador ou gerente.

Select nome,cargo

From funcionario

Where cargo in ('gerente', 'contador')

Funcionario

matricula	nome	cargo	salario	cod-depto	comissao	estado	cidade
30	Paulo	gerente	4000	1	100	RJ	Rio
40	Lucas	contador	1400	2	300	RJ	Rio
50	Luiz	gerente	3500	2	200	MG	Betim

Operador IS NULL

- IS NULL / IS NOT NULL
- Ex: Selecione os funcionarios cujo estado ou cidade é nula.

Select nome

Where estado is null or cidade is null
Funcionario

matricula	nome	cargo	salario	cod-depto	comissao	estado	cidade
10	Ana	analista	2800	1	100	Null	null

Operador LIKE

- O operador LIKE é usado quando se deseja obter colunas de um registro que sigam um determinado padrão pré-especificado.
- Quando se quer saber os nomes de todos os funcionários cujo nome começa com JOAO ou termina com SILVA, usa-se o LIKE.
- O caractere % dentro da expressão LIKE tem a mesma função do caractere * no Windows, assim como o _ (underscore) tem a semelhança com o ? do Windows.

Operador LIKE

```
SELECT NOME  
FROM FUNCIONARIOS  
WHERE NOME LIKE 'JOAO%'
```

```
SELECT *  
FROM FUNCIONARIOS  
WHERE NOME LIKE '__ R%'
```

(Obs: a terceira letra do nome do funcionário deve ser R)



Operador LIKE

- Ex: Selecione os funcionários cujo o nome tem a letra u como segundo caractere.

Select nome

From Funcionario

Where nome Like '_u%

Funcionario

matricula	nome	cargo	salario	cod-depto	comissao	estado	cidade
40	Lucas	contador	1400	2	300	RJ	Rio
50	Luiz	gerente	3500	2	200	MG	Betim

SQL-DML: UPDATE

```
UPDATE <tabela>  
SET <coluna1> = <expressão1>,  
<coluna2> =<expressão2>, ...  
WHERE <condição-de-alteração>
```



- Dica 1: para maiores atualizações, faça o SELECT antes do UPDATE para visualizar os registros
- Dica 2: se quiser atualizar um único registro, informe a chave primária no filtro do WHERE

SQL-DML: DELETE

```
DELETE FROM <tabela>  
WHERE <condição-de-exclusão>
```



Exemplos de SQL-DML

Insert into Funcionario

(matricula,nome,cargo,salario,cod-depto)

Values (70,'Abel','analista',2000,1)

Insert into Funcionario

Values (80,'Dora','analista',2500,1,250,'RJ','Rio')

Update Funcionario

Set salario= salario * 1.10

Where cargo= 'contador'

Update Funcionario

Set cargo = 'desenvolvedor',salario=1800

Where cargo= 'prog'

Exemplos de SQL-DML

```
Insert into Funcionario_TI  
Select * from Funcionario  
Where cod_depto = 1
```

```
Delete from Funcionario  
Where cargo= 'contador'
```




PUC Minas
Virtual