

Atividade: Update

Atualizar as tabelas :

Atualizar a cidade de um dos alunos para Feira de Santana.

Atualizar da titulação de um dos professor para Mestre.

Atualizar a quantidade de alunos em duas disciplinas para 40 e 35.

Atualizar a menção de um aluno para B.

DML - Delete

- O comando DELETE é usado para remover linhas de uma tabela.
- Sua forma geral:

DELETE

FROM nome_da_tabela

WHERE condição;

Exemplo:

```
delete from aluno  
where id_aluno = 1
```

- Você **não** pode excluir parcialmente uma linha, portanto não precisa especificar os nomes das colunas no comando DELETE.
- A cláusula WHERE determina que linhas serão eliminadas.
- Ela pode ser complexa e incluir várias condições, conectores e/ou sub-consultas.

Atividade: Delete

Atualizar as tabelas :

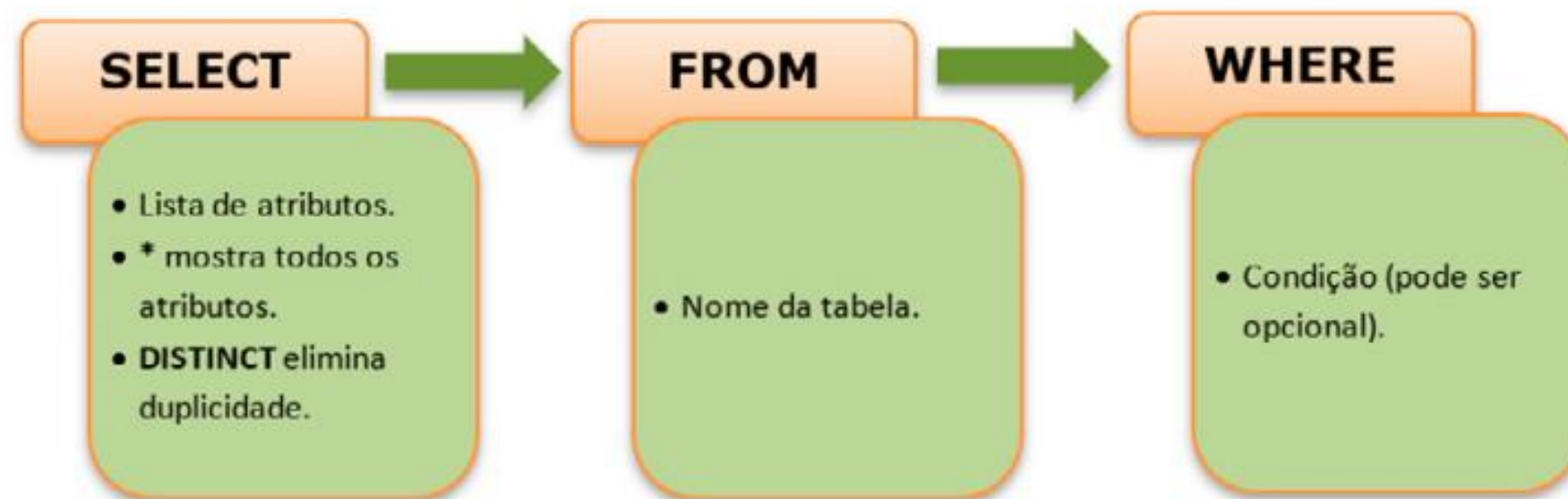
Delete um dos alunos.

Delete um dos professores.

Delete uma disciplina.

DML - Select

- É utilizado para resgatar valores salvos nas tabelas do banco de dados:



SELECT coluna1, coluna2,..., colunaN

FROM nome_da_tabela

WHERE condição;

Exemplos de utilização:

1-select * from aluno

2-select nome from aluno

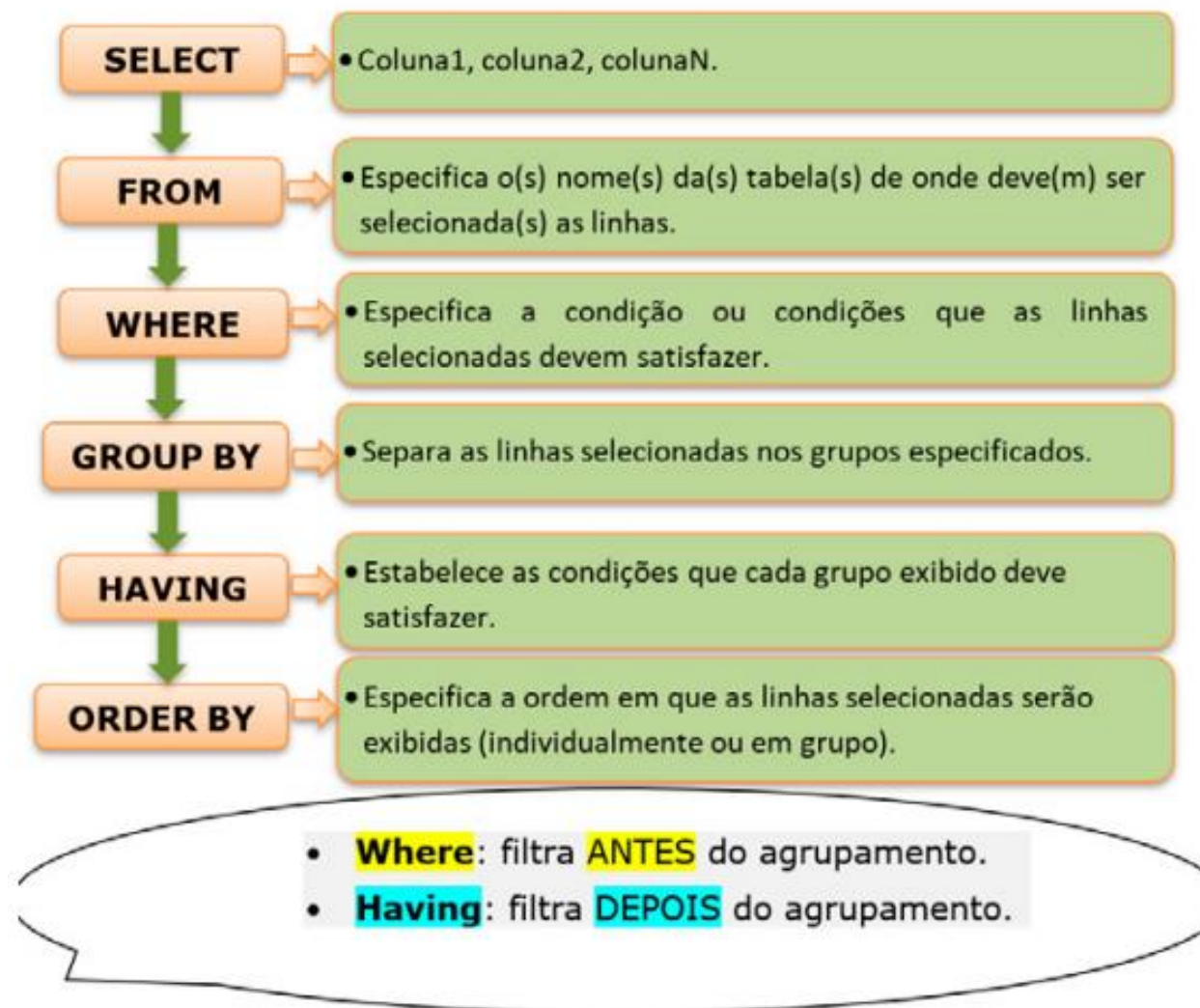
where id_aluno=1

3-select * from aluno

where id_aluno=1

DML - Select

- Uma consulta SQL pode conter outras cláusulas como esquematizadas a seguir:



A cláusula **FROM** é necessária para uma consulta SQL; as cláusulas **WHERE**, **GROUP BY**, **HAVING** e **ORDER BY** são opcionais.

DML - Select

- Exemplos:

```
SELECT * FROM PESSOA;
```

```
SELECT CPF, NOME, SEXO, IDADE  
FROM PESSOA
```

```
WHERE sexo = 'M' OR sexo = 'F';
```

```
SELECT CPF, NOME, SEXO, IDADE  
FROM PESSOA
```

```
WHERE IDADE > ANY (10,20,30);
```

OU

```
WHERE IDADE > ALL (10,20,30);
```

O **ALL** operador:

- retorna um valor booleano como resultado.
- retorna TRUE se TODOS os valores da sub-consulta atenderem à condição.
- é usado com SELECT, WHERE e HAVING declarações

ALL significa que a condição será verdadeira somente se a operação for verdadeira para todos os valores no intervalo.

O **ANY** operador:

- retorna um valor booleano como resultado.
- retorna TRUE se QUALQUER um dos valores da sub-consulta atender à condição.

ANY significa que a condição será verdadeira se a operação for verdadeira para qualquer um dos valores no intervalo.

DML - Select

- LIKE e NOT LIKE

- Exemplo:

```
SELECT CPF, NOME, SEXO
FROM PESSOA
WHERE nome LIKE 'P%';
```

LIKE determina se uma cadeia de caracteres específica corresponde a um padrão especificado. Um padrão pode incluir **caracteres normais** e **coringas**.

Os operadores **NOT LIKE** são usados de forma oposta ao operador **LIKE**.

DML - Select

- LIKE e NOT LIKE

Expressão	Explicação
LIKE 'A%'	Todas as palavras que iniciem com a letra A .
LIKE '%A'	Todas as palavras que terminem com a letra A .
LIKE '%A%'	Todas que tenham a letra A em qualquer posição .
LIKE 'A_'	<i>String</i> de dois caracteres que tenham a primeira letra A e o segundo caractere seja qualquer outro .

Expressão	Explicação
LIKE '_A'	<i>String</i> de dois caracteres cujo primeiro caractere seja qualquer um e a última letra seja A .
LIKE '_A_'	<i>String</i> de três caracteres cuja segunda letra seja A , independentemente do primeiro ou do último caractere .
LIKE '%A_'	Todos que tenham a letra A na penúltima posição e a última seja qualquer outro caractere .
LIKE '_A%'	Todos que tenham a letra A na segunda posição e o primeiro <i>caracter</i> seja qualquer um .

(%) indica zero ou vários caracteres.
(_) indica um caractere

DML – Select into

- Seleciona dados de uma ou mais tabelas e os insere em uma tabela diferente.
- Pode ser usado, para criar cópias de backup de tabelas, ou ainda tabelas para relatório, contendo campos selecionados especialmente provenientes de várias tabelas distintas..

Sintaxe

```
SELECT coluna(s)  
INTO nova_tabela  
FROM tabela_atual  
[WHERE campo = valor];
```

```
INSERT INTO table2 (column1, column2, column3, ...)  
SELECT column1, column2, column3, ...  
FROM table1  
WHERE condition;
```

DML – Condição BETWEEN

- Use a condição BETWEEN para exibir linhas com base em uma faixa de valores:

```
SELECT last_name, salary  
FROM employees  
WHERE salary BETWEEN 2500 AND 3500 ;
```

Limite inferior Limite superior

LAST_NAME	SALARY
Rajs	3500
Davies	3100
Matos	2600
Vargas	2500

*Também é possível usar a condição BETWEEN em valores de caractere:

DML – Condição IN

- Use a condição de associação IN para testar os valores de uma lista:

```
SELECT employee_id, last_name, salary, manager_id
FROM employees
WHERE manager_id IN (100, 101, 201) ;
```

EMPLOYEE_ID	LAST_NAME	SALARY	MANAGER_ID
202	Fay	6000	201
200	Whalen	4400	101
205	Higgins	12000	101
101	Kochhar	17000	100
102	De Haan	17000	100
124	Mourgos	5800	100
149	Zlotkey	10500	100
201	Hartstein	13000	100

8 rows selected.

Se forem usados caracteres ou datas na lista, eles deverão ser delimitados por aspas simples (")

DML – Condição Lógicas

Operador	Significado
AND	Retornará TRUE se as duas condições componentes forem verdadeiras
OR	Retornará TRUE se uma das condições componentes for verdadeira
NOT	Retornará TRUE se a condição seguinte for falsa

DML – Regras de Precedência

Operador	Significado
1	Operadores aritméticos
2	Operador de concatenação
3	Condições de comparação
4	IS [NOT] NULL, LIKE, [NOT] IN
5	[NOT] BETWEEN
6	Diferente de
7	Condição lógica NOT
8	Condição lógica AND
9	Condição lógica OR

DML – Regras de Precedência

```
SELECT last_name, job_id, salary
FROM   employees
WHERE  job_id = 'SA_REP'
OR     job_id = 'AD_PRES'
AND    salary > 15000;
```

1

LAST_NAME	JOB_ID	SALARY
King	AD_PRES	24000
Abel	SA_REP	11000
Taylor	SA_REP	8600
Grant	SA_REP	7000

```
SELECT last_name, job_id, salary
FROM   employees
WHERE  (job_id = 'SA_REP'
OR     job_id = 'AD_PRES')
AND    salary > 15000;
```

2

LAST_NAME	JOB_ID	SALARY
King	AD_PRES	24000

DML – Cláusula ORDER BY

- Classifique as linhas recuperadas com a cláusula ORDER BY:
 - – **ASC**: ordem crescente, default
 - – **DESC**: ordem decrescente
- A cláusula ORDER BY é inserida por último na instrução SELECT:

Sintaxe

```
SELECT          expr
FROM            table
[WHERE          condition(s)]
[ORDER BY      {column, expr, numeric_position} [ASC|DESC]] ;
```

Na sintaxe:

ORDER BY	especifica a ordem na qual as linhas recuperadas são exibidas
ASC	ordena as linhas em ordem crescente (essa é a ordem default)
DESC	ordena as linhas em ordem decrescente

Atividade: Select

Selecionar :

Selecionar todos os alunos.

Selecionar apenas dois alunos

Selecionar três professores.

Selecionar todas as disciplinas ordenadas pelo nome da disciplina.

*Enviar os **insert**, **update**, **delete** e **select** feitos na sala de aula com base nos registros criados para o seu banco de dados pessoal.*

murphyprof@gmail.com