

Cláusulas ORDER BY, SELECT DISTINCT, e LIMIT

Banco de Dados

Charles Tim Batista Garrocho

Instituto Federal de Minas Gerais – IFMG
Campus Ponte Nova

`garrocho.github.io/BDD`

`charles.garrocho@ifmg.edu.br`

Técnico em Informática



Claúsula ORDER BY

O armazenando de linhas em uma tabela não acontece em uma **ordem** especificada. Isso significa que as linhas na tabela podem estar ou não na ordem em que foram inseridas.

Se você usar a instrução SELECT para consultar dados de uma tabela, a ordem das linhas no conjunto de resultados **não será especificada**. Para classificar o conjunto de resultados, você adiciona a cláusula **ORDER BY** na instrução SELECT da seguinte maneira:

```
SELECT * FROM TABELA ORDER BY COLUNA ASC;
```

Ordenação crescente em relação aos dados da coluna especificada.

ou

```
SELECT * FROM TABELA ORDER BY COLUNA DESC;
```

Ordenação decrescente em relação aos dados da coluna especificada.



Claúsula ORDER BY

Considerando a tabela Notas abaixo, o comando SQL para selecionar todos alunos ordenados da menor para a maior nota em Geografia é:

```
SELECT ALUNO FROM NOTAS ORDER BY GEOGRAFIA ASC;
```

Aluno	Matemática	Português	Ciências	História	Geografia
Adriana	9	10	9	10	9
Alexandre	8	8	9	8	7
Carlos	10	8	8	7	8
Fernanda	8	8	8	9	10
Joãozinho	5	4	5	3	3

Retorno da execução do comando:

Joãozinho

Alexandre

Carlos

Adriana

Fernanda



Claúsula ORDER BY

Considerando a tabela Notas abaixo, o comando SQL para selecionar todos alunos ordenados da maior para a menor nota em Geografia é:

```
SELECT ALUNO FROM NOTAS ORDER BY GEOGRAFIA DESC;
```

Aluno	Matemática	Português	Ciências	História	Geografia
Adriana	9	10	9	10	9
Alexandre	8	8	9	8	7
Carlos	10	8	8	7	8
Fernanda	8	8	8	9	10
Joãozinho	5	4	5	3	3

Retorno da execução do comando:

Fernanda

Adriana

Carlos

Alexandre

Joãozinho



Claúsula SELECT DISTINCT

A cláusula **DISTINCT** é opcional na instrução SELECT. Ela permite **remover as linhas duplicadas** no conjunto de resultados. A declaração a seguir ilustra a sintaxe da cláusula DISTINCT:

SELECT DISTINCT COLUNA FROM TABELA;

Considerando a tabela Notas abaixo, o comando SQL para selecionar as notas de Português descartando as notas repetidas é:

SELECT DISTINCT PORTUGUES FROM NOTAS;

Aluno	Matemática	Português	Ciências	História	Geografia
Adriana	9	10	9	10	9
Alexandre	8	8	9	8	7
Carlos	10	8	8	7	8
Fernanda	8	8	8	9	10
Joãozinho	5	4	5	3	3

Retorno da execução do comando: 10, 8, 4



Claúsula LIMIT

A cláusula **LIMIT** é opcional na instrução SELECT. Você usa a cláusula LIMIT para **restringir o número de linhas** retornadas pela consulta. O seguinte ilustra a sintaxe da cláusula LIMIT.

```
SELECT * FROM TABELA LIMIT QUANTIDADE;
```

Considerando a tabela Notas abaixo, o comando SQL para selecionar os nomes de dois alunos com nota em Matemática é:

```
SELECT ALUNO FROM NOTAS LIMIT 2;
```

Aluno	Matemática	Português	Ciências	História	Geografia
Adriana	9	10	9	10	9
Alexandre	8	8	9	8	7
Carlos	10	8	8	7	8
Fernanda	8	8	8	9	10
Joãozinho	5	4	5	3	3

Retorno da execução do comando: Adriana, Alexandre



Claúsulas ORDER BY e LIMIT

As cláusulas **ORDER BY** e **LIMIT** podem ser utilizadas em conjunto para retornar um resultado específico.

Considerando a tabela Notas abaixo, o comando SQL para selecionar o nome do aluno com maior nota em História é:

```
SELECT ALUNO FROM NOTAS ORDER BY HISTORIA DESC  
LIMIT 1;
```

Aluno	Matemática	Português	Ciências	História	Geografia
Adriana	9	10	9	10	9
Alexandre	8	8	9	8	7
Carlos	10	8	8	7	8
Fernanda	8	8	8	9	10
Joãozinho	5	4	5	3	3

Retorno da execução do comando: Adriana



Claúsula OFFSET

A cláusula **OFFSET** permite retornar os resultados **a partir** de uma linha.

Considerando a tabela Notas abaixo, o comando SQL para selecionar o nome do aluno com a segunda maior nota em História é:

```
SELECT ALUNO FROM NOTAS ORDER BY HISTORIA DESC  
LIMIT 1 OFFSET 2;
```

Aluno	Matemática	Português	Ciências	História	Geografia
Adriana	9	10	9	10	9
Alexandre	8	8	9	8	7
Carlos	10	8	8	7	8
Fernanda	8	8	8	9	10
Joãozinho	5	4	5	3	3

Retorno da execução do comando: Fernanda



Atividades Práticas

Utilizando o Shell SQLite3, implemente uma base de dados para gerenciar as cidades do brasil. Esta base de dados permitirá armazenar o nome, estado e população de várias cidades do brasil. Após isso, mostre os comandos necessários para:

- Inserir 6 cidades (sendo duas de Minas Gerais, e duas de São Paulo);
- Imprimir apenas duas cidades;
- Imprimir as cidades em ordem crescente de acordo com o nome;
- Imprimir todas as cidades (sem populações iguais);
- Imprimir a cidade com maior população de São Paulo;
- Imprimir as duas cidades com menores populações;
- Imprimir a segunda cidade com menor população de Minas Gerais;

Obs: Armazene o **log** de todos os comandos para a análise do professor.

