

BANCO DE DADOS – NOVOS COMANDO

Vamos criar O banco e a tabela abaixo funcionar.

```
DROP DATABASE IF EXISTS escola_155;
CREATE DATABASE escola_155;
USE escola_modulo01;
CREATE table alunoX(
  id_aluno INT AUTO_INCREMENT PRIMARY KEY,
  nome_aluno Varchar(100),
  datanasc_aluno date,
  nota01_aluno decimal(4,2),
  nota02_aluno decimal(4,2),
  nota03_aluno decimal(4,2),
  sexo_aluno ENUM("M","F"),
  turma_aluno int
)
```

Mas antes de adicionarmos os alunos na tabela, vamos fazer apenas algumas correções.

1 – O nome da tabela deve der “aluno” e não “alunoX”.

2 – Vamos adicionar um campo chamado cidade_aluno do tipo varchar(100) entre os campos nome_aluno e datanasc_aluno.

Para estas mudanças vamos usar os comando da categoria DDL

DDL (Data Definition Language) -> Linguagem de Definição de Dados.
Comandos DDL são usados para definir, modificar ou excluir a estrutura do banco de dados, como tabelas, índices e esquemas (bancos).

Vamos alterar o nome da tabela . Vamos usar o comando RENAME

O comando RENAME TABLE no MySQL pertence à categoria de Comandos de Definição de Dados (DDL - Data Definition Language).

Por que ele é um comando DDL?

Os comandos DDL são usados para definir e modificar a estrutura dos objetos do banco de dados, como tabelas, índices e esquemas. O RENAME TABLE altera a estrutura do banco de dados ao modificar o nome de uma tabela, sem afetar os dados armazenados.

Sua sintaxe

```
RENAME TABLE nome_antigo TO nome_novo;
```

```
Servidor: 127.0.0.1 » Banco de dados: escola173 » Tabela: alunox  
Executar consulta(s) SQL na tabela escola173.alunox:  
1 RENAME TABLE alunox TO aluno;
```

Vamos adicionar o nosso campos cidade_aluno lembrando ele é um varchar(100) e vai ficar entre os campos nome_aluno e anonasc_aluno.

Bom aqui iremos fazer uma alteração na tabela e o comando e ALTER.

Para adicionar uma coluna em uma posição específica em uma tabela no MySQL, você deve usar o comando **ALTER TABLE** junto com **ADD COLUMN** e as palavras-chave **FIRST** ou **AFTER**.

Sintaxe

```
ALTER TABLE nome_da_tabela  
ADD COLUMN nome_da_coluna tipo_dado [opções]  
[FIRST | AFTER nome_coluna_existente];
```

```
Servidor: 127.0.0.1 » Banco de dados: escola173 » Tabela: aluno  
Executar consulta(s) SQL na tabela escola173.aluno:  
1 ALTER TABLE aluno  
2 ADD COLUMN cidade_aluno VARCHAR(100) AFTER nome_aluno;
```

Agora vamos adicionar os dados na tabela “aluno”

Nome	Cidade	Ano Nasc.	Nota1	Nota2	Nota3	Sexo	Turma
Joao da Silva	Criciúma	10/10/2008	5,60	6,50	6,00	M	101
Maria Tereza	Içara	01/11/2008	6,00	7,00	7,50	F	101
Carlos Alberto	Içara	05/08/2009	7,00	7,50	8,00	M	102
Cristina Furlan	Criciúma	05/06/2008	8,00	8,50	7,00	F	202
Ana Carolina	Criciúma	02/02/2008	7,50	6,50	7,50	F	202
Marcos Lapoli	Criciúma	01/09/2007	8,50	9,00	7,50	M	301

```
1 INSERT INTO aluno(nome_aluno,cidade_aluno,datanasc_aluno,nota01_aluno,nota02_aluno,nota03_aluno,sexo_aluno,turma_aluno)VALUES  
2 ("Joao da Silva","Criciúma","2008-10-10",5.60,6.00,6.00,"M",101),  
3 ("Maria Tereza","Içara","2008-11-01",6.00,7.00,7.50,"F",101),  
4 ("Carlos Alberto","Içara","2009-08-05",7.00,7.50,8.00,"M",102),  
5 ("Cristina Furlan","Criciúma","2008-06-05",8.00,8.50,7.00,"F",202),  
6 ("Ana Carolina","Criciúma","2008-02-02",7.50,6.50,7.50,"F",202),  
7 ("Marcos Lapoli","Criciúma","2007-09-01",8.50,9.00,7.50,"M",301)  
8
```

Agora vamos brincar um pouco vamos fazer várias pesquisas.

1 – Mostre todos os alunos por ordem alfabética normal A-Z

- 2 – Mostre todos os alunos por ordem alfabética. Decrescente Z-A
- 3 – Mostre o nome, data nascimento e turma dos alunos do sexo Masculino ordenado em ordem alfabética.
- 4 – Mostre o nome, cidade e sexo dos alunos que morem em “Criciúma” que sejam do sexo Masculino”
- 5 – Mostre nome, cidade, sexo e turma dos alunos que morem “Criciúma” , que sejam do sexo “F” . Ordenados em ordem alfabética

1)

```
1 SELECT * FROM aluno ORDER BY nome_aluno
```

2)

```
1 SELECT * FROM aluno ORDER BY nome_aluno DESC
```

3)

```
1 SELECT nome_aluno,anonasc_aluno,turma_aluno FROM ALUNO ORDER BY nome_aluno
```

A

4)

```
1 SELECT nome_aluno,cidade_aluno,sexo_aluno FROM aluno WHERE cidade_aluno="Criciúma" and sexo_aluno="M"
```

5)

```
SELECT nome_aluno,cidade_aluno,sexo_aluno,turma_aluno FROM aluno WHERE cidade_aluno="Criciúma" and sexo_aluno="F" ORDER BY nome_aluno
```

VAMOS APRIMORAR AS PESQUISA NO NOSSO BANCO!!!!!!

Curte o que vamos fazer vamos trazer o nome de cada aluno e sua média geral e já mostrar ordenado alfabeticamente.

```
1 SELECT nome_aluno, ROUND((nota01_aluno + nota02_aluno + nota03_aluno) / 3, 2) AS media_notas
2 FROM aluno ORDER by nome_aluno
```

Resultado

nome_aluno	media_notas
Ana Carolina	7.17
Carlos Alberto	7.50
Cristina Furlan	7.83
Joao da Silva	5.87
Marcos Lapoli	8.33
Maria Tereza	6.83

Mas afinal que loucura aconteceu ali em cima naquele comando SQL????

Vamos lá → Primeiro selecionamos o campo nome_aluno, depois passamos a cláusula ROUND (arredondar) ele vai somar as 3 notas dividir por 3 e colocar somente duas casas decimais depois da virgula. Agora vem a MAGIA o “AS” cria um campo temporário na nossa tabela chamado media_aluno , conde vai guardar o resultado dos cálculos , depois dizemos a tabela que queremos fazer todos os comando e por último ordenamos pelo nome.

Sabendo um pouco destes comandos vamos fazer o seguinte vamos mostrar o nome, sexo e turma dos alunos que tiveram media $\geq 7,00$ e ordenar por ordem alfabética decrescente.

Mas aqui vamos usar palavra nova HAVING

O que é HAVING no MySQL?

O HAVING é usado para filtrar resultados após a agregação de dados com funções como AVG(), SUM(), COUNT(), etc.

Diferença entre HAVING e WHERE

WHERE

Filtra antes da agregação

Funciona em colunas individuais

HAVING

Filtra depois da agregação

Funciona em colunas agregadas (SUM(), AVG(), etc.)

```
1 SELECT nome_aluno,sexo_aluno,turma_aluno, ROUND((nota01_aluno + nota02_aluno + nota03_aluno) / 3, 2) AS media_notas FROM aluno HAVING media_notas  $\geq$  7 ORDER BY nome_aluno DESC
```

Resultado

nome_aluno	1	sexo_aluno	turma_aluno	media_notas
Marcos Lapoli		M	301	8.33
Cristina Furlan		F	202	7.83
Carlos Alberto		M	102	7.50
Ana Carolina		F	202	7.17

Teste:

Apresente o nome, sexo, turma, município dos alunos que tiverem media ≥ 7 e somente do sexo masculino.

E vamos ver mais uma coisa legal

Se quisermos apresentar o nome, data de nascimento e a idade dos alunos

```
1 SELECT nome_aluno,datanasc_aluno,TIMESTAMPDIFF(YEAR, datanasc_aluno, CURDATE()) AS idade FROM aluno;
```

nome_aluno	datanasc_aluno	idade
Joao da Silva	2008-10-10	16
Maria Tereza	2008-11-01	16
Carlos Alberto	2009-08-05	15
Cristina Furlan	2008-06-05	16
Ana Carolina	2008-02-02	17
Marcos Lapoli	2007-09-01	17

Mais um teste agora use o comando acima e apresente somente os maiores de 16 (>16)