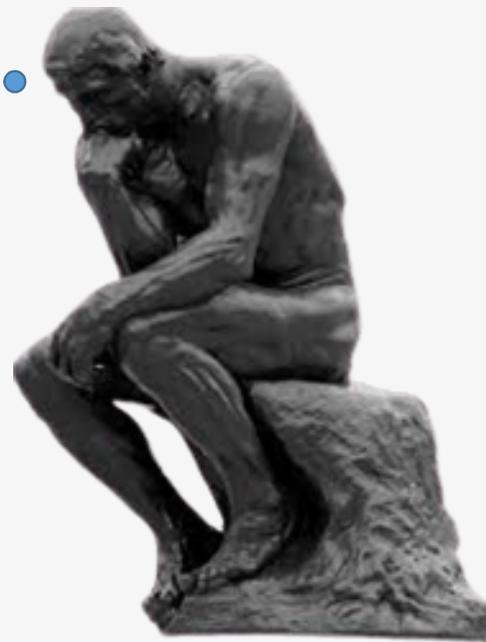


BANCO DE DADOS

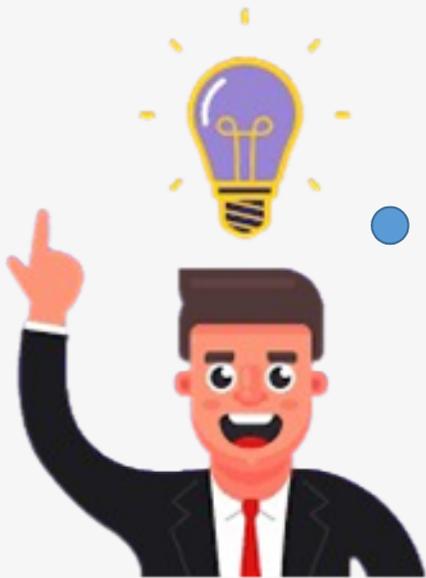
Vamos resolver uma solicitação de um cliente, precisamos fazer o cadastro de alunos de uma escola, então vamos pensar

Escola, Alunos,
Banco de Dados,
como vou
fazer????



BANCO DE DADOS

Vamos dar nome ao Banco, depois ver as Entidades e depois os Atributos !!!



BANCO DE DADOS

O nome do Banco até fica fácil pois vamos usar a estrutura principal ou seja podemos chama o banco de “escola” mesmo



BANCO DE DADOS

Vamos falar sobre “Entidades”



Não essa “Entidade” , Entidade de Banco de Dados

BANCO DE DADOS

Entidade → Em Banco de Dados, são objetos reais ou não que possuem características próprias

ALUNO 1



ALUNO 2



ALUNO 3



Como podemos notar temos um objeto que tem características próprias e por sinal é de extrema importância para o funcionamento da escola.

A ENTIDADE QUE VAMOS CHAMAR “ALUNO”

Na criação de nosso Banco de Dados as “Entidades” serão transformadas em “Tabelas”.

ENTÃO TEREMOS UMA TABELA QUE CHAMAREMOS DE “ALUNO”

BANCO DE DADOS

Atributo → São as características que definem uma entidade.



Nome: Maria Eduarda
Idade : 16
Altura: 1.60
Turma: 190

Nome: Carlos Alberto
Idade : 16
Altura: 1.78
Turma: 190

Nome: Silvana Torres
Idade : 17
Altura: 1.61
Turma: 280

Como podemos notar temos 3 alunos diferentes mas todos possuem algo em comum, eles possuem nome, idade, altura e uma turma que estudam.

Esses são os “ATRIBUTOS” que devemos ter em mente. Então a entidade “Aluno” tem estes atributos

Na criação de nosso Banco de Dados os “Atributos” serão conhecidos como “Campos”

ENTÃO TEREMOS UMA TABELA QUE CHAMAREMOS DE “ALUNO”

COM OS CAMPOS

NOME

IDADE

ALTURA

TURMA

BANCO DE DADOS

Registro → São os dados que vamos colocar dentro dos campos , e cada registro irá corresponder a um aluno individual.



Nome	Maria Eduarda]
Idade	16
Altura	1.60
Turma	190



Nome:	Carlos Alberto
Idade:	16
Altura:	1.78
Turma:	190



Nome:	Silvana Torres
Idade:	17
Altura:	1.61
Turma:	280

BANCO DE DADOS

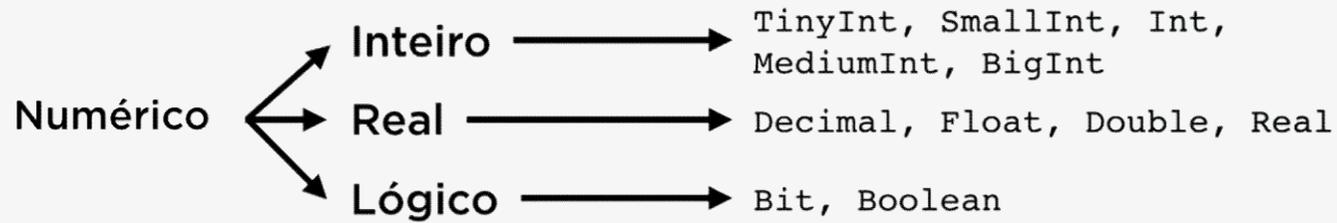
ATENÇÃO

Não podemos simplesmente criar “Campos” para guardar o registro de qualquer maneira temos algumas coisas antes fazer isso

Temos que saber que tipo de registro vamos guardar palavra, uma letra, um texto longo, um número pequeno, um peso, valor financeiro, uma data,etc.

VAMOS VER ALGUNS TIPOS , COM PASSAR DAS AULAS VAMOS USANDO OUTROS TIPOS

BANCO DE DADOS





Mãos
à Obra

Banco de Dados - SQL

SQL (Structured Query Language - Linguagem de Consulta Estruturada) é uma linguagem padrão para gerenciar e manipular bancos de dados relacionais. Ela permite que você interaja com o banco de dados para realizar operações como consultar, inserir, atualizar e excluir dados, além de criar e modificar a estrutura do banco de dados.

O SQL é dividido em várias categorias de comandos, cada uma com um propósito específico. Vamos explorar essas categorias:

1. 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.

CREATE: Cria novos objetos no banco de dados, como tabelas, índices ou visões.

```
CREATE TABLE usuarios (  
  id INT PRIMARY KEY,  
  nome VARCHAR(50),  
  idade INT  
);
```

ALTER: Modifica a estrutura de uma tabela existente, como adicionar, remover ou alterar colunas.

```
ALTER TABLE usuarios ADD COLUMN email VARCHAR(100);
```

DROP: Exclui objetos do banco de dados, como tabelas ou índices.

```
DROP TABLE usuarios;
```

TRUNCATE: Remove todos os dados de uma tabela, mas mantém a estrutura.

```
TRUNCATE TABLE usuarios;
```

2. DML (Data Manipulation Language) - Linguagem de Manipulação de Dados.

Comandos DML são usados para manipular os dados armazenados nas tabelas.

SELECT: Recupera dados de uma ou mais tabelas.

```
SELECT nome, idade FROM usuarios WHERE idade > 18;
```

INSERT: Adiciona novos registros a uma tabela.

```
INSERT INTO usuarios (id, nome, idade) VALUES (1, 'João', 25);
```

UPDATE: Modifica dados existentes em uma tabela.

```
UPDATE usuarios SET idade = 26 WHERE id = 1;
```

DELETE: Remove registros de uma tabela.

```
DELETE FROM usuarios WHERE id = 1;
```

3. DCL (Data Control Language) - Linguagem de Controle de Dados

Comandos DCL são usados para controlar o acesso aos dados e gerenciar permissões de usuários.

GRANT: Concede permissões a usuários ou roles.

```
GRANT SELECT, INSERT ON usuarios TO 'usuario1';
```

REVOKE: Revoga permissões concedidas anteriormente.

```
REVOKE INSERT ON usuarios FROM 'usuario1';
```

4. TCL (Transaction Control Language) - Linguagem de Controle de Transações

Comandos TCL são usados para gerenciar transações no banco de dados, garantindo a consistência dos dados.

BEGIN ou START TRANSACTION: Inicia uma nova transação.

```
BEGIN;
```

COMMIT: Confirma as alterações feitas durante uma transação.

```
COMMIT;
```

ROLLBACK: Desfaz as alterações feitas durante uma transação, revertendo ao estado anterior.

```
ROLLBACK;
```

SAVEPOINT: Define um ponto de salvamento dentro de uma transação, permitindo rollback parcial.

```
SAVEPOINT ponto1;
```

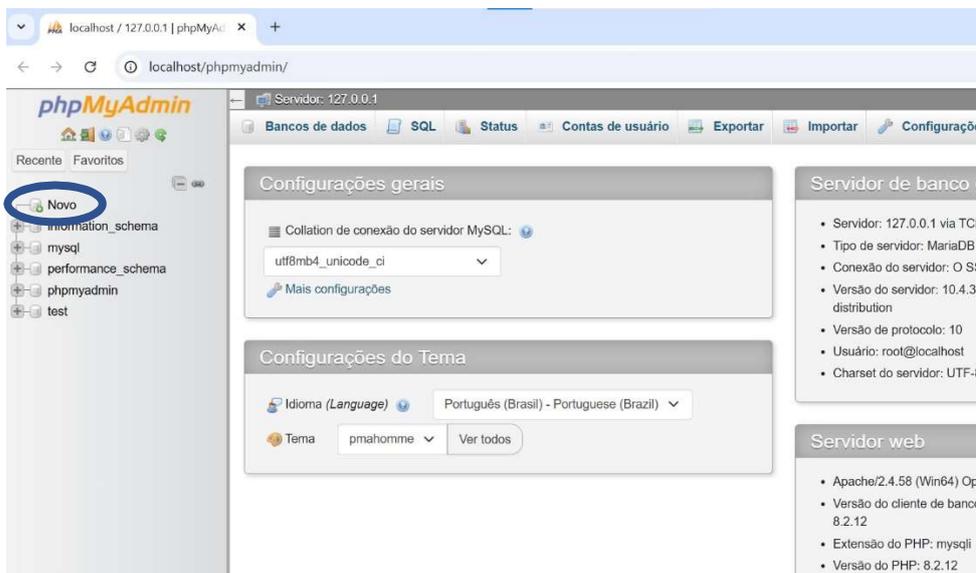
Resumo das Categorias:

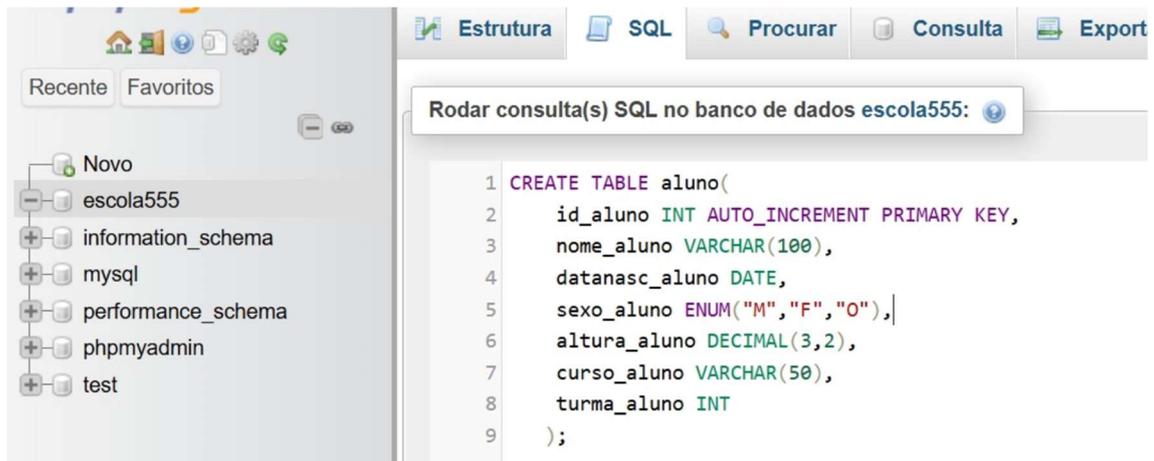
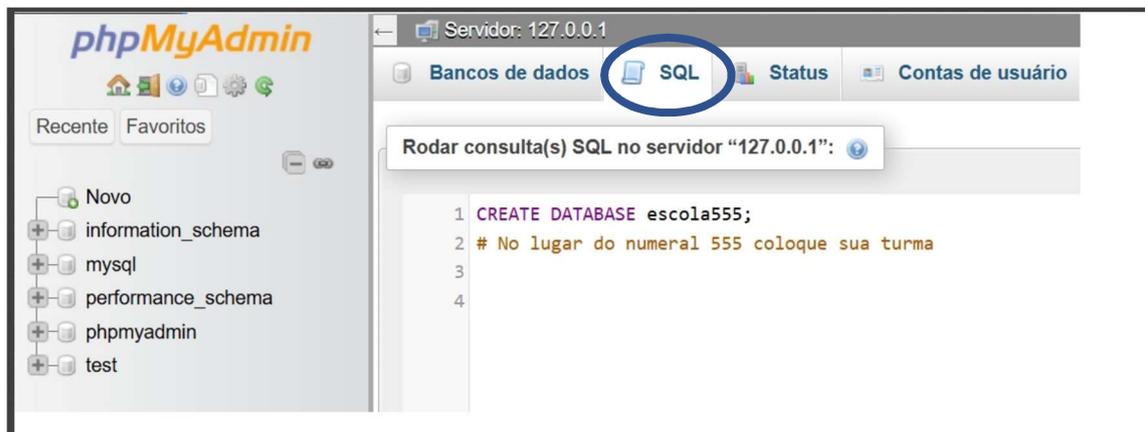
Categoria	Descrição	Exemplos de Comandos
DDL	Define a estrutura do banco de dados	CREATE, ALTER, DROP, TRUNCATE
DML	Manipula os dados nas tabelas	SELECT, INSERT, UPDATE, DELETE
DCL	Controla permissões de acesso	GRANT, REVOKE
TCL	Gerencia transações	BEGIN, COMMIT, ROLLBACK, SAVEPOINT

VAMOS METER A MÃO NA MASSA !!!!

Nos foi solicitado criar um banco de dados para uma escola, e supersimples vamos apenas guardar o nome de alunos, sua data de nascimento, sexo, altura, curso que está estudando e sua turma.

LEMBRE-SE ESTAMOS O USANDO O SOFTWARE XAMPP E PHPMYADMIN, ENTÃO SEMPRE QUE FORMOS TRABALHAR VEJA SE OS MESMOS ESTAO FUNCIONANDO.





Vamos entender o que está acontecendo com os comandos acima
`id_aluno` → Estamos criando um atributo identificador ou seja, seja único e não pode repetir. O do tipo numérico Inteiro, `AUTO_INCREMENT` (significa que se incrementará sozinho) e `PRIMARY KEY` (EXPLICAÇÃO ABAIXO)

Uma Primary Key (ou Chave Primária, em português) é um conceito fundamental em bancos de dados relacionais. Ela é um campo (ou conjunto de campos) em uma tabela que identifica de forma única cada registro (linha) naquela tabela. A Primary Key tem as seguintes características:

- **Unicidade:** Cada valor na Primary Key deve ser único. Não pode haver dois registros com o mesmo valor na chave primária.
- **Não nula (NOT NULL):** A Primary Key não pode conter valores nulos (NULL). Cada registro deve ter um valor definido para a chave primária.
- **Imutabilidade:** Idealmente, o valor da Primary Key não deve mudar ao longo do tempo. Isso garante a integridade dos dados e dos relacionamentos com outras tabelas

`nome_aluno Varchar(100) →`

VARCHAR é um tipo de dado usado em bancos de dados para armazenar textos (cadeias de caracteres) de comprimento variável. O nome "VARCHAR" vem de Variable Character, ou seja, "caracteres variáveis". Ele é amplamente utilizado em sistemas de gerenciamento de banco de dados (SGBDs), como MySQL, PostgreSQL, SQL Server, Oracle, entre outros.

Características do VARCHAR:

- Comprimento variável: Diferente de um tipo de dado como CHAR, que sempre ocupa um espaço fixo (mesmo que o texto seja menor), o VARCHAR armazena apenas o espaço necessário para o texto inserido, até um limite máximo definido.
- Definição de tamanho máximo: Ao criar uma coluna do tipo VARCHAR, você deve especificar o número máximo de caracteres que ela pode armazenar. Por exemplo, VARCHAR(50) permite armazenar até 50 caracteres.
- Eficiência de armazenamento: Como o VARCHAR só ocupa o espaço necessário para o texto armazenado, ele é mais eficiente em termos de armazenamento quando comparado ao CHAR, especialmente para textos que variam muito em tamanho.
- Uso comum: É frequentemente usado para armazenar dados como nomes, endereços, descrições, e-mails, etc., onde o comprimento do texto pode variar.

`datanasc_aluno → DATE`

O campo DATE em SQL é um tipo de dado usado para armazenar datas (ano, mês e dia) em bancos de dados. Ele é amplamente utilizado para registrar informações temporais, como datas de nascimento, datas de pedidos, prazos, entre outros.

Características do tipo DATE:

- Formato: O tipo DATE armazena datas no formato YYYY-MM-DD (Ano-Mês-Dia), que é o padrão ISO 8601.
- Exemplo: 2023-10-05 representa 5 de outubro de 2023.
- Sem horário: O tipo DATE armazena apenas a data (ano, mês e dia), sem informações de horário. Se você precisar armazenar horas, minutos e segundos, deve usar o tipo DATETIME ou TIMESTAMP.
- Operações com datas: O tipo DATE permite realizar operações como:
- Comparação de datas (maior, menor, igual).
- Cálculo de diferenças entre datas.
- Extração de partes da data (ano, mês, dia).

`sexo_aluno ENUM("M", "F", "O") →`

O ENUM em SQL é um tipo de dado que permite definir uma lista de valores pré-definidos para uma coluna em uma tabela. Ele é útil quando você sabe que os valores de uma coluna só podem ser um conjunto específico de opções. O nome "ENUM" vem de enumeração, pois ele enumera os valores permitidos.

Características do ENUM:

- **Valores pré-definidos:** Ao criar uma coluna do tipo ENUM, você define uma lista de valores possíveis que essa coluna pode armazenar.
- **Exemplo:** `ENUM('ativo', 'inativo', 'pendente')` significa que a coluna só pode ter os valores 'ativo', 'inativo' ou 'pendente'.
- **Valor padrão:** Se nenhum valor for especificado ao inserir um registro, o ENUM usará o primeiro valor da lista como padrão (a menos que você defina outro valor padrão explicitamente).
- **Restrição de valores:** Se você tentar inserir um valor que não está na lista de opções do ENUM, o banco de dados retornará um erro (ou um valor vazio, dependendo da configuração).

`altura_aluno Decimal(3,2)` →

O tipo DECIMAL em SQL é usado para armazenar números exatos com uma precisão definida, especialmente valores monetários ou outros dados que exigem cálculos precisos. Diferente de tipos como FLOAT ou DOUBLE, que são aproximados, o DECIMAL armazena números de forma exata, sem arredondamentos.

Características do DECIMAL:

Precisão e escala:

Precisão (p): Número total de dígitos que o DECIMAL pode armazenar (incluindo dígitos antes e depois da vírgula).

Escala (s): Número de dígitos após a vírgula.

Exemplo: DECIMAL (5,2) pode armazenar números com até 5 dígitos no total, sendo 2 deles após a vírgula (ex.: 123.45).

Armazenamento exato: O DECIMAL não faz arredondamentos, o que é ideal para cálculos financeiros ou onde a precisão é crítica.

Intervalo de valores: O intervalo de valores que um DECIMAL pode armazenar depende da precisão e escala definidas.

Uso comum: É frequentemente usado para armazenar valores monetários, quantidades, medidas e outros dados que exigem exatidão.

ATE AQUI USAMOS 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.

DDL Define a estrutura do banco de dados **CREATE, ALTER, DROP, TRUNCATE**

Ao clicarmos na tabela depois de executarmos os comandos teremos a imagem abaixo

The screenshot shows the phpMyAdmin interface for a table named 'aluno' in the 'escola555' database. The table structure is displayed in a grid format with the following columns:

#	Nome	Tipo	Colaço	Atributos	Nulo	Padrão	Comentários	Extra	Ação
1	id_aluno	int(11)			Não	Nenhum		AUTO_INCREMENT	Alterar Eliminar Mais
2	nome_aluno	varchar(100)	utf8mb4_general_ci		Sim	NULL			Alterar Eliminar Mais
3	datanasc_aluno	date			Sim	NULL			Alterar Eliminar Mais
4	sexo_aluno	enum('M', 'F', 'O')	utf8mb4_general_ci		Sim	NULL			Alterar Eliminar Mais
5	altura_aluno	decimal(3,2)			Sim	NULL			Alterar Eliminar Mais
6	curso_aluno	varchar(50)	utf8mb4_general_ci		Sim	NULL			Alterar Eliminar Mais
7	turma_aluno	int(11)			Sim	NULL			Alterar Eliminar Mais

AGORA VAMOS USAR COMANDOS DA CATEGORIA DML (Data Manipulation Language) - Linguagem de Manipulação de Dados.

Comandos DML são usados para manipular os dados armazenados nas tabelas.

DML Manipula os dados nas tabelas **SELECT, INSERT, UPDATE, DELETE**

Vamos agora incluir alguns alunos para podemos fazer teste.

Nome Aluno	Data Nasc	Sexo	Altura	Curso	Turma
João da Silva	10-10-2007	M	1.78	Informática	390
Maria Tereza	05-02-2009	F	1.62	Informática	380
Carlos Alberto	01-06-2006	M	1.70	Administração	242
Estela Souza	13-10-2005	F	1.58	Direito	185

Clique na tabela "aluno" e depois em SQL e vamos digitar os dados abaixo

The screenshot shows the phpMyAdmin interface with the SQL query editor open for the 'aluno' table. The following SQL query is entered:

```
1 INSERT INTO aluno(nome_aluno,datanasc_aluno,sexo_aluno,altura_aluno,curso_aluno,turma_aluno)VALUES
2 ("João da Silva","2007-10-10","M",1.78,"Informática",390),
3 ("Maria Tereza","2009-02-05","F",1.62,"Informática",380),
4 ("Carlos Alberto","2006-06-01","M",1.70,"Administração",242),
5 ("Estela Souza","2005-10-13","F",1.58,"Direito",185)
6
```

Vamos trabalhar com o comando "select" para ver o que podemos fazer.

Clique na tabela desejada e de novo no SQL e depois vamos executar os comandos abaixo e ver o que eles retornam.

1 - Vamos dar um SELECT normal para mostrar todos os itens da tabela

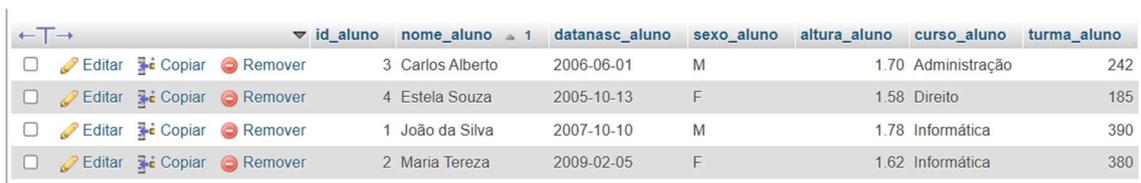
```
SELECT * FROM aluno
```



	id_aluno	nome_aluno	datanasc_aluno	sexo_aluno	altura_aluno	curso_aluno	turma_aluno
<input type="checkbox"/> Editar Copiar Remover	1	João da Silva	2007-10-10	M	1.78	Informática	390
<input type="checkbox"/> Editar Copiar Remover	2	Maria Tereza	2009-02-05	F	1.62	Informática	380
<input type="checkbox"/> Editar Copiar Remover	3	Carlos Alberto	2006-06-01	M	1.70	Administração	242
<input type="checkbox"/> Editar Copiar Remover	4	Estela Souza	2005-10-13	F	1.58	Direito	185

2 - Vamos dar outro SELECT agora vamos ordenar pelo nome em ordem alfabética. A-Z

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



	id_aluno	nome_aluno	datanasc_aluno	sexo_aluno	altura_aluno	curso_aluno	turma_aluno
<input type="checkbox"/> Editar Copiar Remover	3	Carlos Alberto	2006-06-01	M	1.70	Administração	242
<input type="checkbox"/> Editar Copiar Remover	4	Estela Souza	2005-10-13	F	1.58	Direito	185
<input type="checkbox"/> Editar Copiar Remover	1	João da Silva	2007-10-10	M	1.78	Informática	390
<input type="checkbox"/> Editar Copiar Remover	2	Maria Tereza	2009-02-05	F	1.62	Informática	380

3 - Vamos dar outro SELECT agora vamos ordenar pelo nome em ordem alfabética, mas em ordem decrescente de Z-A

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



	id_aluno	nome_aluno	datanasc_aluno	sexo_aluno	altura_aluno	curso_aluno	turma_aluno
<input type="checkbox"/> Editar Copiar Remover	2	Maria Tereza	2009-02-05	F	1.62	Informática	380
<input type="checkbox"/> Editar Copiar Remover	1	João da Silva	2007-10-10	M	1.78	Informática	390
<input type="checkbox"/> Editar Copiar Remover	4	Estela Souza	2005-10-13	F	1.58	Direito	185
<input type="checkbox"/> Editar Copiar Remover	3	Carlos Alberto	2006-06-01	M	1.70	Administração	242

4 - Mostrar os funcionários de um determinado sexo

```
SELECT * FROM aluno WHERE sexo_aluno="F"
```



	id_aluno	nome_aluno	datanasc_aluno	sexo_aluno	altura_aluno	curso_aluno	turma_aluno
<input type="checkbox"/> Editar Copiar Remover	2	Maria Tereza	2009-02-05	F	1.62	Informática	380
<input type="checkbox"/> Editar Copiar Remover	4	Estela Souza	2005-10-13	F	1.58	Direito	185

5 - Vamos selecionar um determinado sexo e altura, pessoas do sexo Masculino com mais de 1.70 de altura.

```
SELECT * FROM aluno WHERE sexo_aluno="M" AND altura_aluno > 1.70
```



	id_aluno	nome_aluno	datanasc_aluno	sexo_aluno	altura_aluno	curso_aluno	turma_aluno
<input type="checkbox"/> Editar Copiar Remover	1	João da Silva	2007-10-10	M	1.78	Informática	390

6 - Vamos selecionar nome que iniciem com um caracter especifico digamos a letra "e"

SELECT * FROM aluno WHERE nome_aluno LIKE "e%"

id_aluno	nome_aluno	datanasc_aluno	sexo_aluno	altura_aluno	curso_aluno	turma_aluno
4	Estela Souza	2005-10-13	F	1.58	Direito	185

CASO QUEIRA IMPRIMIR CAMPOS ESPECIFICOS E SO COLOCAR NO LUGAR * O NOME DOS CAMPOS QUE DESEJA IMPRIMIR.

EXERCÍCIO DE FIXAÇÃO

- 1 - Crie o banco de dados chamado "empresa_seunome"
- 2 - Crie a tabela funcionário com os seguintes campos.
nome_func
datanasc_func
sexo_func
setor_func
salario_func
Atenção não esquecer o atributo identificador.

- 3 - Cadastre os seguintes dados.

Nome	Data	Sexo	Setor	Salário
Pedro Felisbino	10-10-1999	M	Vendas	5622,30
João da Silva	20-05-2000	M	Vendas	4680,25
Maiara Silvestre	30-03-2001	F	RH	6320,58
Maria Eduarda	01-08-2002	F	Vendas	6240,65
Estela Campos	13-10-1999	F	Direção	9580,95

- 4 - Faça as seguintes pesquisas

- a) Mostre todos os funcionários de sexo Masculino em ordem alfabética.
- b) Mostre todos os funcionários que ganham acima de 6000,00
- c) Mostre todos os funcionários do sexo "F" e do departamento de "Vendas"
- d) Mostre todos os funcionários que contenham a letra "s" no nome