

## SQL - Standard Query Language

### MySQL

Aprender:

- Criar uma tabela
- Inserir dados
- Consultar dados
- Deletar dados

#### MODELO RELACIONAL

Surgiu década 60

- Começou implementação dos computadores nas empresas

As tabelas são compostas por: ENTIDADES - ATRIBUTOS - CHAVES

Entidades: cada entidade tem seus atributos (são únicos) - A entidade é o que é de fato a tabela

Pq usar tabelas?

- Organizar dados de forma estruturada
- Dados atômicos: dado é único e não existe em nenhum outro lugar - organizar dados nas colunas
- Manipular dados simplificados

#### FERRAMENTAS

MySQL - banco de dados gratuitos (principais utilizadas hoje em dia) - Base pra você começar a trabalhar com banco de dados

phpMyAdmin - sistema gerenciador de banco de dados - Ferramenta mais práticas e praticamente todo computador consegue rodar

#### **EXEMPLO DE TABELA - (converter situações do dia a dia em tabela)**

Lista de compras

Atributos -	ITEM (tomate, arroz)
	QTDADA (1, 2)
	PREÇO

XAMPP - pilha de aplicações que envolve apache, mySQL, phpMyAdmin...

**Comandos** no MySQL por convenção e boas práticas usa-se sempre em Caps Lock, mas funciona também em letras minúsculas

Tipos: INT, VARCHAR, DATETIME

Exemplo:

CREATE TABLE *pessoa* [que é uma tabela], com o [atributo] nome VARCHAR (30) [recebe texto], atributo nascimento DATE [recebe uma data]

### **Criar tabela**

```
CREATE TABLE nome tabela (  
    nome VARCHAR (30),  
    nascimento DATE,
```

### **Inserção de dados**

INSERT INTO *nome tabela* (informações que quer cadastrar, colunas, ex: nome e nascimento) VALUES ('nome', 'data nasc') data nasci sempre formato americano.  
exemplo: INSERT INTO pessoas

Primary key: valor único que nunca se repetirá. Exemplo: ID é um nome por padrão  
Toda tabela tem que ter um campo primário, uma primary key. Ela é fundamental.  
Mesmo que a informação tenha sido deletada, a id será sempre única, um novo cadastro virá com um novo primary key, ele pula a key que foi deletada.

### **Mais comandos QUERY**

**SELECT:** SELECT\*FROM nome tabela - com esse comando ele retorna todos os dados da tabela

SELECT nome FROM nome tabela

**UPDATE:** comando responsável pela atualização

UPDATE *nome tabela* SET *qual o campo quer atualizar* = " texto a ser adicionado " WHERE *id=número da primary key*

Cláusula WHERE: fazer a manipulação certa dos dados

Você pode criar um arquivo .sql no VS CODE, para salvar as informações e compartilhar no github

```
CREATE TABLE nome tabela  
    id INT NOT NULL PRIMARY KEY AUTOINCREMENT  
    nome VARCHAR (45) NOT NULL  
    nascimento DATE
```

### **Cláusula DELETE**

Deletou já era, não tem ctrl+z, não tem como recuperar o dado.

Boas práticas: Para evitar deletar um dado errado, escreve primeiro um SELECT \* FROM e confirma o dado, verifica o dado antes de deletar.

DELETE FROM nome tabela WHERE id=numero key

### **Ordenando os dados**

ORDER BY : ordena as informações de forma crescente ou decrescente. Por padrão ele ordena de forma crescente.

SELECT \* FROM *nome tabela* ORDER BY *atributo*

SELECT \* FROM *nome tabela* ORDER BY *atributo* DESC - para ordem decrescente

### **Agrupando dados**

GROUP BY = agrupar

COUNT = conta quantos elementos são inseridos dentro de determinado grupamento

SELECT COUNT (passar parâmetro ex: id), *nome coluna* FROM *nome tabela* GROUP BY *atributo que eu quero agrupar (ex: genero)*

ALTER TABLE - serve para qualquer alteração na tabela

Exemplo:

ALTER TABLE *nome tabela* ADD *nome nova coluna* VARCHAR(1) NOT NULL AFTER *nome coluna já existente*