



# \_Banco de Dados

LET'S CODE

| \_Dúvidas



## |\_Constraints – NOT NULL

```
CREATE TABLE cliente (  
    cod_cli INTEGER NOT NULL,  
    cpf CHAR(11) UNIQUE,  
    nome_cli VARCHAR(40) NOT NULL,  
    idade INTEGER CHECK (idade = 18)  
    endereco VARCHAR(40) null,  
    cidade VARCHAR(20) null,  
    cep CHAR(8) null,  
    uf CHAR(2) null,  
    ativo BIT null  
);
```

## |\_Constraints – UNIQUE

```
CREATE TABLE cliente (  
    cod_cli INTEGER NOT NULL,  
    cpf CHAR(11) UNIQUE,  
    nome_cli VARCHAR(40) NOT NULL,  
    idade INTEGER CHECK (idade = 18)  
    endereco VARCHAR(40) null,  
    cidade VARCHAR(20) null,  
    cep CHAR(8) null,  
    uf CHAR(2) null,  
    ativo BIT null  
);
```

## |\_Constraints – DEFAULT

```
CREATE TABLE conta (  
    cod_conta BIGINT NOT NULL,  
    saldo DOUBLE DEFAULT 0.0  
);
```

## |\_Constraints – CHECK

```
CREATE TABLE cliente (  
    cod_cli INTEGER NOT NULL,  
    cpf CHAR(11) UNIQUE,  
    nome_cli VARCHAR(40) NOT NULL,  
    idade INTEGER CHECK (idade = 18)  
    endereco VARCHAR(40) null,  
    cidade VARCHAR(20) null,  
    cep CHAR(8) null,  
    uf CHAR(2) null,  
    ativo BIT null  
);
```



## | \_Constraints – CHAVES PRIMARIAS

```
CREATE TABLE letscode_genero (  
    id_genero INTEGER NOT NULL,  
    nome VARCHAR(40) NOT NULL,  
    PRIMARY KEY (id_genero)  
);
```

## | \_Constraints – CHAVES ESTRANGEIRAS

```
CREATE TABLE letscore_participacao (  
    id_participacao INTEGER NOT NULL,  
    id_filme INTEGER NOT NULL,  
    id_ator INTEGER NOT NULL,  
    FOREIGN KEY (id_filme)  
        REFERENCES letscore_filme (id_filme),  
    FOREIGN KEY (id_ator)  
        REFERENCES letscore_ator (id_ator),  
    PRIMARY KEY (id_participacao)  
);
```



## | \_Funções Agregadas - MIN() e MAX()

```
SELECT MIN(valor_unitario) FROM tb_entrada_produto;
```

```
SELECT MAX(valor_unitario) FROM tb_entrada_produto;
```

## | \_Funções Agregadas - COUNT()

```
SELECT COUNT(*) FROM tb_saida_produto;
```

## | \_Funções Agregadas - SUM()

```
SELECT qtde, SUM(valor_unitario) FROM tb_saida_produto  
GROUP BY 1;
```

## | \_Funções Agregadas - AVG()

```
SELECT valor_unitario, AVG(qtde) FROM tb_entrada_produto  
GROUP BY 1;
```

## | \_Funções Agregadas - HAVING

```
SELECT valor_unitario, COUNT(*) FROM tb_estoque  
GROUP BY 1  
HAVING COUNT(*) > 2;
```

# |\_Consulta

Consultas

Na tabela Customers:

1) Gere uma relação com os nomes dos clientes, suas cidades e países, em ordem alfabética de nome.