



escola
britânica de
artes criativas
& tecnologia

SQL para análise de dados



SELEÇÃO E ORDENAÇÃO EM SQL



GUIA DA AULA 1



Restrinja os dados – Teoria



Acompanhe aqui os temas que serão tratados na videoaula

● **Introdução**

● **Restrição de colunas**

● **Tipos de restrições**

● **Checando restrições**



Introdução

Nessa aula, usaremos a seguinte tabela:

```
CREATE TABLE
  transacoes ( id_cliente
  INT, id_transacao INT,
  data_compra DATE,
  valor FLOAT,
  id_loja varchar(25)
  );
```



Também temos os seguintes valores inseridos na tabela:

```

INSERT INTO transacoes VALUES (1,768805383,2021-06-10,50.74,'magalu');
INSERT INTO transacoes VALUES (2,768805399,2021-06-13,30.90,'giraffas');
INSERT INTO transacoes VALUES (3,818770008,2021-06-05,110.00,'postoshell');
INSERT INTO transacoes VALUES (1,76856563,2021-07-10,2000.90,'magalu');
INSERT INTO transacoes VALUES (1,767573759,2021-06-20,15.70,'subway');
INSERT INTO transacoes VALUES (3,818575758,2021-06-25,2.99,'seveneleven');
INSERT INTO transacoes VALUES (4,764545534,2021-07-11,50.74,'extra'); INSERT
INTO transacoes VALUES (5,76766789,2021-08-02,10.00,'subway'); INSERT INTO
transacoes VALUES (3,8154567758,2021-08-15,1100.00,'shopee');
  
```



Como resultado da função SELECT, temos a seguinte tabela:

id_cliente	id_transacao	data_compra	valor	id_loja
1	768805383	2021-06-10	50.74	magalu
2	768805399	2021-06-13	30.90	giraffas
3	818770008	2021-06-05	110.00	postoshell
1	76856563	2021-07-10	2000.90	magalu
1	767573759	2021-06-20	15.70	subway
3	818575758	2021-06-25	2.99	seveneleven
4	764545534	2021-07-10	50.74	extra
5	76766789	2021-08-92	10.00	subway
3	8154567758	2021-08-15	1100.00	shopee

Para realizar os testes no SQL, acesse o *link* <https://sqliteonline.com/>.



Restrição de colunas

1. Tipos de chaves: Chave primária “primary key”

Para identificar uma tabela, podemos utilizar uma chave primária. Existem algumas regras que devem ser respeitadas na coluna designada a ser chave primária:

- O valor da coluna não pode se **repetir** na tabela;
- O valor da coluna não pode ser **nulo**.



Você pode utilizar a chave primária de uma tabela para identificar uma instância em outra tabela.

```

CREATE TABLE <nome_tabela> (
  <nome_da_coluna_primaria> <tipo_da_coluna_primaria> PRIMARY KEY,
  <nome_da_coluna_2> <tipo_da_coluna_2>,
  ...
);
  
```



No exemplo, podemos colocar a chave primeira como o **id_transacao**, já que toda transação deve ser única (índice não se repete) e não nula (não pode existir um valor de transação que não foi preenchido):

```
CREATE TABLE transacoes (
  id_cliente INT,
  id_transacao INT PRIMARY KEY,
  data_compra DATE,
  valor FLOAT,
  id_loja varchar(25)
);
```

Nota: existe a opção de utilizar o AUTO_INCREMENT para que o valor de id_transacao seja automaticamente adicionado de 1 a cada nova transação no banco de dados. Entenda mais acessando o link https://www.w3schools.com/sql/sql_autoincrement.asp.



2. **Tipos de chaves:** Chave estrangeira “foreign key”:

Quando queremos relacionar duas tabelas, utilizamos a chave estrangeira. Isso significa que uma coluna na tabela atual, está relacionada com alguma instância de outra tabela. No exemplo para o MySQL:

```

CREATE TABLE <nome_tabela> (
  <nome_da_coluna> <tipo_da_coluna>,
  <nome_da_coluna> <tipo_da_coluna>,
  ...,
  <nome_da_coluna> <tipo_da_coluna>,
  FOREIGN KEY <nome_da_coluna> REFERENCES <nome_tabela_da_chave_relacionada>(<nome_da_coluna_na_tabe
);
  
```



Exemplo: Na nossa tabela transações, faz sentido ter a chave primária como id_transacoes como fizemos anteriormente e deixar a chave estrangeira como o id_cliente. Isso porque, deve existir uma tabela clientes onde esse id_cliente é a chave primeira.

```

CREATE TABLE transacoes(
  id_cliente INT,
  id_transacao INT PRIMARY KEY,
  data_compra DATE,
  valor FLOAT,
  id_loja varchar(25),
  FOREIGN KEY (id_cliente) REFERENCES clientes(id_cliente)
);
  
```



Nota: Não há alteração visual quando adicionamos primary key ou foreign key na tabela. Por isso não mostraremos o resultado aqui.

Para mais informações e outros formatos, acesse o **link**

https://www.w3schools.com/sql/sql_foreignkey.asp



Tipos de restrições

Algumas regras podem ser estabelecidas nos dados da sua tabela. Definindo-as na hora da criação da tabela, garantirá segurança a inserção dos dados.

1. **Valores não nulos (not null):** Os valores da coluna não podem ser nulos.

```

CREATE TABLE <nome_tabela> (
  <nome_da_coluna> <tipo_da_coluna> NOT NULL ,
  <nome_da_coluna_2> <tipo_da_coluna_2> ,
  ...
);
  
```



No nosso exemplo (unindo tudo que já aprendemos):

```

CREATE TABLE
    transacoes ( id_cliente
    INT,
    id_transacao INT PRIMARY
    KEY, data_compra DATE,
    valor FLOAT NOT NULL,
    id_loja varchar(25),
    FOREIGN KEY (id_cliente) REFERENCES clientes(id_cliente)
    );
    
```



2. **Valores únicos (unique):** Define que os valores da coluna devem ser únicos.

```

CREATE TABLE <nome_tabela> (
  <nome_da_coluna> <tipo_da_coluna> UNIQUE,
  <nome_da_coluna_2> <tipo_da_coluna_2>,
  ...
);
  
```



No nosso exemplo (unindo tudo que já aprendemos):

```

CREATE TABLE transacoes (
  id_cliente INT,
  id_transacao INT PRIMARY KEY,
  data_compra DATE UNIQUE,
  valor FLOAT NOT NULL,
  id_loja varchar(25),
  FOREIGN KEY (id_cliente) REFERENCES clientes(id_cliente)
);
  
```



Checando restrições (check)

Esse comando certifica que algumas condições devem ser satisfeitas ao inserir um dados na tabela.

```

CREATE TABLE <nome_tabela> (
  <nome_da_coluna> <tipo_da_coluna>,
  <nome_da_coluna_2> <tipo_da_coluna_2>,
  CHECK (<condicao>)
  ...
);
  
```



No nosso caso, podemos avaliar se não há valor negativo nas transações:

```

CREATE TABLE transacoes (
  id_cliente INT,
  id_transacao INT PRIMARY KEY,
  data_compra DATE UNIQUE,
  valor FLOAT NOT NULL,
  id_loja varchar(25),
  FOREIGN KEY (id_cliente) REFERENCES clientes(id_cliente)
  CHECK (valor > 0)
);
  
```

Nota: Não há alteração visual quando adicionamos primary key ou foreign key na tabela. Por isso não mostraremos o resultado aqui.

