

Administrador de Banco de Dados (extra Abril) - Turma 2024A

3.7 Criando e populando as tabelas: Festival

Na próxima atividade, você irá praticar seus conhecimentos de SQL através do exemplo de tabelas relacionadas a um festival.

A tabela `artist` armazena informações sobre os artistas que se apresentam no festival, como o ID e o nome. A tabela `show` armazena informações sobre os shows, como o ID do artista que se apresenta, a data e hora do show, e o palco. A tabela `ticket` armazena informações sobre os ingressos vendidos, como o ID do show, o preço do ingresso e o tipo de ingresso. A tabela `customer` armazena informações sobre os clientes que compram os ingressos, como o ID, nome e e-mail. A tabela `purchase` armazena informações sobre as compras feitas pelos clientes, como o ID do cliente, o ID do ingresso comprado e a data da compra.

As relações são estabelecidas através das chaves estrangeiras `artist_id` na tabela `show`, `show_id` na tabela `ticket`, `customer_id` na tabela `purchase` e `ticket_id` na tabela `purchase`. Essas chaves estrangeiras são utilizadas para relacionar as tabelas entre si, permitindo que as informações de uma tabela sejam acessadas a partir de outra.

Veja as informações e importe em seu banco de dados:

```
CREATE TABLE artist (  
    id INTEGER PRIMARY KEY,  
    name TEXT NOT NULL  
);  
  
CREATE TABLE show (  
    id INTEGER PRIMARY KEY,  
    artist_id INTEGER NOT NULL,  
    date_time DATETIME NOT NULL,  
    stage TEXT NOT NULL,  
    FOREIGN KEY (artist_id) REFERENCES artist (id)  
);  
  
CREATE TABLE ticket (  
    id INTEGER PRIMARY KEY,  
    show_id INTEGER NOT NULL,  
    price REAL NOT NULL,  
    type TEXT NOT NULL,  
    FOREIGN KEY (show_id) REFERENCES show (id)  
);  
  
CREATE TABLE customer (  
    id INTEGER PRIMARY KEY,  
    name TEXT NOT NULL,  
    email TEXT NOT NULL  
);  
  
CREATE TABLE purchase (  
    id INTEGER PRIMARY KEY,  
    customer_id INTEGER NOT NULL,  
    ticket_id INTEGER NOT NULL,  
    purchase_date DATETIME NOT NULL,  
    FOREIGN KEY (customer_id) REFERENCES customer (id),  
    FOREIGN KEY (ticket_id) REFERENCES ticket (id)  
);
```

```
INSERT INTO artist (id, name) VALUES
```

```
(1, 'Coldplay'),  
(2, 'Foo Fighters'),  
(3, 'Radiohead');
```

```
INSERT INTO show (id, artist_id, date_time, stage) VALUES
```

```
(1, 1, '2023-07-15 20:00:00', 'Main Stage'),  
(2, 2, '2023-07-15 22:00:00', 'Main Stage'),  
(3, 3, '2023-07-16 20:30:00', 'Second Stage'),  
(4, 1, '2023-07-16 22:00:00', 'Main Stage'),  
(5, 2, '2023-07-17 20:30:00', 'Main Stage'),  
(6, 3, '2023-07-17 22:00:00', 'Main Stage');
```

```
INSERT INTO ticket (id, show_id, price, type) VALUES
```

```
(1, 1, 250.0, 'VIP'),  
(2, 1, 120.0, 'Regular'),  
(3, 2, 300.0, 'VIP'),  
(4, 2, 150.0, 'Regular'),  
(5, 3, 200.0, 'VIP'),  
(6, 3, 100.0, 'Regular'),  
(7, 4, 250.0, 'VIP'),  
(8, 4, 120.0, 'Regular'),  
(9, 5, 300.0, 'VIP'),  
(10, 5, 150.0, 'Regular'),  
(11, 6, 200.0, 'VIP'),  
(12, 6, 100.0, 'Regular');
```

```
INSERT INTO customer (id, name, email) VALUES
```

```
(1, 'Ana Silva', 'ana.silva@gmail.com'),  
(2, 'João Oliveira', 'joao.oliveira@hotmail.com'),  
(3, 'Maria Santos', 'maria.santos@yahoo.com');
```

```
INSERT INTO purchase (id, customer_id, ticket_id, purchase_date) VALUES
```

```
(1, 1, 1, '2023-06-15 15:00:00'),  
(2, 2, 3, '2023-06-15 16:00:00'),  
(3, 3, 5, '2023-06-16 10:30:00'),
```

```
(4, 1, 9, '2023-06-17 14:00:00'),  
(5, 2, 12, '2023-06-17 16:30:00');
```

Última atualização: quarta, 15 mar 2023, 15:02

◀ 3.6 Teste seus conhecimentos: Hotel

Seguir para...

3.8 Teste seus conhecimentos: Festival ▶