

## Online Bookstore Database Setup Script - MYSQL Challenge

Rishabh Vyas

**Data Analyst** 

#### INTRODUCTION

"Explore the 'Online Bookstore Database Setup Script' to efficiently create and populate a database for managing book titles, customer information, orders, and more. This script simplifies the process of setting up an online bookstore's

backend data structure."



## 1. Create Database

-- Create the Bookstore database CREATE DATABASE Bookstore; USE Bookstore;

## 2. Create Table

```
-- Create the 'books' table to store book information
• CREATE TABLE books (
     book id INT PRIMARY KEY AUTO INCREMENT,
     title VARCHAR(255) NOT NULL,
     author VARCHAR(255),
     genre VARCHAR(50),
     publication year INT,
     price DECIMAL(10, 2),
     copies sold INT
```

### 3. Insert Data into Books Table

```
-- Insert sample data into the 'books' table
INSERT INTO books (title, author, genre, publication year, price, copies sold)
VALUES
    ('The Great Gatsby', 'F. Scott Fitzgerald', 'Fiction', 1925, 9.99, 500),
    ('To Kill a Mockingbird', 'Harper Lee', 'Fiction', 1960, 10.99, 600),
    ('1984', 'George Orwell', 'Science Fiction', 1949, 8.99, 750),
    ('Pride and Prejudice', 'Jane Austen', 'Romance', 1813, 7.99, 900),
    ('The Hobbit', 'J.R.R. Tolkien', 'Fantasy', 1937, 12.99, 450),
    ('The Catcher in the Rye', 'J.D. Salinger', 'Fiction', 1951, 9.99, 550),
    ('The Hunger Games', 'Suzanne Collins', 'Science Fiction', 2008, 11.99, 700),
    ('Harry Potter and the Sorcerer''s Stone', 'J.K. Rowling', 'Fantasy', 1997, 14.99, 800),
    ('The Da Vinci Code', 'Dan Brown', 'Mystery', 2003, 10.99, 600),
    ('The Alchemist', 'Paulo Coelho', 'Fiction', 1988, 8.99, 950);
```

## 4. Create Customers Table

```
-- Create the 'customers' table to store customer information
• ○ CREATE TABLE customers (
      customer id INT PRIMARY KEY AUTO INCREMENT,
      first name VARCHAR(50) NOT NULL,
      last name VARCHAR(50) NOT NULL,
      email VARCHAR(100) UNIQUE NOT NULL,
      phone number VARCHAR(20),
      address VARCHAR(255)
```

### 5. INSERT Data into Customers Table

```
-- Insert sample data into the 'customers' table

INSERT INTO customers (first_name, last_name, email, phone_number, address)

VALUES

('John', 'Doe', 'johndoe@example.com', '+1234567890', '123 Main St, Anytown, USA'),

('Jane', 'Smith', 'janesmith@example.com', '+9876543210', '456 Elm St, Othercity, USA')

('Alice', 'Johnson', 'alice@example.com', NULL, '789 Oak St, Anothercity, USA');
```

### 6. Create Order Table

```
-- Create the 'orders' table to store order information
CREATE TABLE orders (
     order id INT PRIMARY KEY AUTO INCREMENT,
     customer id INT,
     order date DATE,
     total amount DECIMAL(10, 2),
     FOREIGN KEY (customer_id) REFERENCES customers(customer id)
```

#### 7. INSERT Data into Orders Table

```
-- Insert sample data into the 'orders' table

• INSERT INTO orders (customer_id, order_date, total_amount)

VALUES

(1, '2022-08-15', 45.97),
(2, '2022-08-16', 33.98),
(3, '2022-08-17', 28.99);
```

## 8. Create Order\_Items Table

```
-- Create the 'order items' table to store items within each order
order_item_id INT PRIMARY KEY AUTO_INCREMENT,
      order id INT,
     book_id INT,
      quantity INT,
      item_price DECIMAL(10, 2),
      FOREIGN KEY (order id) REFERENCES orders(order id),
      FOREIGN KEY (book id) REFERENCES books (book id)
```

## 9. INSERT Data into Order\_Items

```
-- Insert sample data into the 'order_items' table

• INSERT INTO order_items (order_id, book_id, quantity, item_price)

VALUES

(1, 1, 2, 19.98),
(1, 3, 1, 8.99),
(2, 2, 3, 32.97),
(3, 5, 2, 25.98);
```

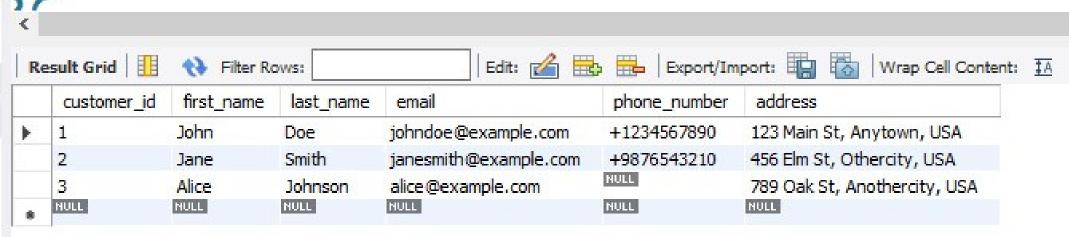
## 10. Books Table:

select \* from books;

Result Grid   🕕 \infty Filter Rows: Edit: 🚄 🖶 Export/Import: 🖫 🐻   Wrap Cell Content: 🖽							
	book_id	title	author	genre	publication_year	price	copies_sold
<b>&gt;</b>	1	The Great Gatsby	F. Scott Fitzgerald	Fiction	1925	9.99	500
	2	To Kill a Mockingbird	Harper Lee	Fiction	1960	10.99	600
	3	1984	George Orwell	Science Fiction	1949	8.99	750
	4	Pride and Prejudice	Jane Austen	Romance	1813	7.99	900
	5	The Hobbit	J.R.R. Tolkien	Fantasy	1937	12.99	450

### 11. Customers Table:

select \* from customers;

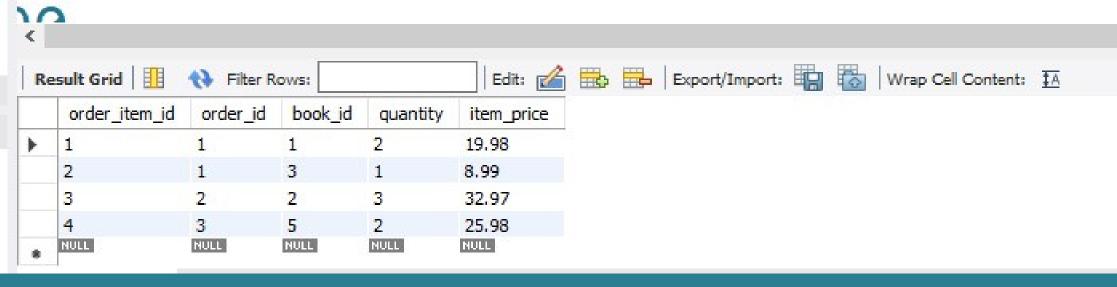


## 12. Orders Table:

select \* from orders;

## 13. Order\_Items Table:

# 39 \* select \* from order\_items;



## **Questions - Challenge**

```
1. What are the details of all books purchased in the year 2022?
--> SELECT o.order id, COUNT(oi.book id) AS total number of books,
       b.title, b.author, b.genre, b.price, SUM(b.price) AS total cost
FROM orders o
JOIN order_items oi ON o.order_id = oi.order_id
JOIN books b ON oi.book id = b.book id
WHERE YEAR(o.order date) = 2022
GROUP BY oi.book id;
```

```
2. What is the total number of books sold by each customer?
```

```
--> SELECT c.customer_id, CONCAT(c.first_name, ' ', c.last_name) AS customer_name, COUNT(oi.book_id) AS number_of_books_purchased
```

JOIN orders o ON c.customer\_id = o.customer\_id

FROM customers c

JOIN order\_items oi ON o.order\_id = oi.order\_id
GROUP BY c.customer\_id;

```
3. What is the total revenue generated by each customer?
```

```
--> SELECT c.customer_id, CONCAT(c.first_name, ' ', c.last_name) AS customer_name, SUM(b.price) AS total_revenue
```

JOIN orders o ON c.customer id = o.customer id

JOIN order\_items oi ON o.order\_id = oi.order\_id

JOIN books b ON oi.book\_id = b.book\_id

GROUP BY c.customer\_id;

FROM customers c

```
4. What are the details of the books purchased by each customer?
```

```
--> SELECT c.customer_id, CONCAT(c.first_name, ' ', c.last_name) AS customer_name, oi.book_id, b.title, b.author, b.genre, b.price
```

JOIN orders o ON c.customer\_id = o.customer\_id

FROM customers c

JOIN order\_items oi ON o.order\_id = oi.order\_id
JOIN books b ON oi.book id = b.book id;

```
5. What is the total revenue generated by each book genre?
--> SELECT b.genre, SUM(b.price) AS total revenue
FROM books b
JOIN order items oi ON b.book id = oi.book id
JOIN orders o ON oi.order id = o.order id
WHERE YEAR(o.order date) = 2022
GROUP BY b.genre;
```

```
6. What are the details of the books purchased in the year 2021 by customer 'Jane Smith'?
```

--> SELECT c.customer\_id, CONCAT(c.first\_name, ' ', c.last\_name) AS customer\_name,
oi.book\_id, b.title, b.author, b.genre, b.price

FROM customers c

JOIN orders o ON c.customer\_id = o.customer\_id

JOIN order\_items oi ON o.order\_id = oi.order\_id
JOIN books b ON oi.book id = b.book id

WHERE c.first\_name = 'Jane' AND c.last\_name = 'Smith' AND YEAR(o.order\_date) = 2021;

```
7. What is the total revenue generated by the sales of books in the 'Fiction' genre?
--> SELECT b.genre, SUM(b.price) AS total_revenue
FROM books b
JOIN order_items oi ON b.book_id = oi.book_id
```

JOIN orders o ON oi.order\_id = o.order\_id

WHERE b.genre = 'Fiction'

GROUP BY b.genre;

```
8. What is the total revenue generated by the sales of books in the 'Mystery' genre in the year 2022?
--> SELECT b.genre, SUM(b.price) AS total_revenue
```

JOIN order\_items oi ON b.book\_id = oi.book\_id
JOIN orders o ON oi.order\_id = o.order\_id

FROM books b

WHERE b.genre = 'Mystery' AND YEAR(o.order\_date) = 2022
GROUP BY b.genre;

```
9. Who is the customer who purchased the most number of books in the year 2023?
--> SELECT c.customer_id, CONCAT(c.first_name, ' ', c.last_name) AS customer_name,
       COUNT(oi.book id) AS number of books purchased
FROM customers c
JOIN orders o ON c.customer_id = o.customer_id
JOIN order_items oi ON o.order_id = oi.order_id
WHERE YEAR(o.order date) = 2023
GROUP BY c.customer id
ORDER BY COUNT(oi.book_id) DESC
LIMIT 1;
```

```
10. Who is the customer with the highest total spending in the year 2022?
--> SELECT c.customer_id, CONCAT(c.first_name, ' ', c.last_name) AS customer_name,
       SUM(b.price) AS total spending
FROM customers c
JOIN orders o ON c.customer id = o.customer id
JOIN order_items oi ON o.order_id = oi.order_id
JOIN books b ON oi.book id = b.book id
WHERE YEAR(o.order date) = 2022
GROUP BY c.customer id
ORDER BY SUM(b.price) DESC
LIMIT 1;
```

## Thank You!



Rishabh Vyas

**Data Scientisit** 



