SQL PROJECT

Source : www.youtube.com/@OfficialAccountAmitKumar

Online Book Store

BY AMIT KUMAR

Source : www.youtube.com/@OfficialAccountAmitKumar

SQL Project - Online Book Store

We have 3 CSV Files:

1. Books.CSV
2. Customers.CSV
3. Orders.CSV

Books.CSV

Book_ID	Title	Author	Genre	Published_Year	Price	Stock
1	Vision-oriented ze	David Hatfield	Science Fiction	1921	15.94	64
2	Compatible transi	Isaac Nelson	Biography	1905	6.94	64

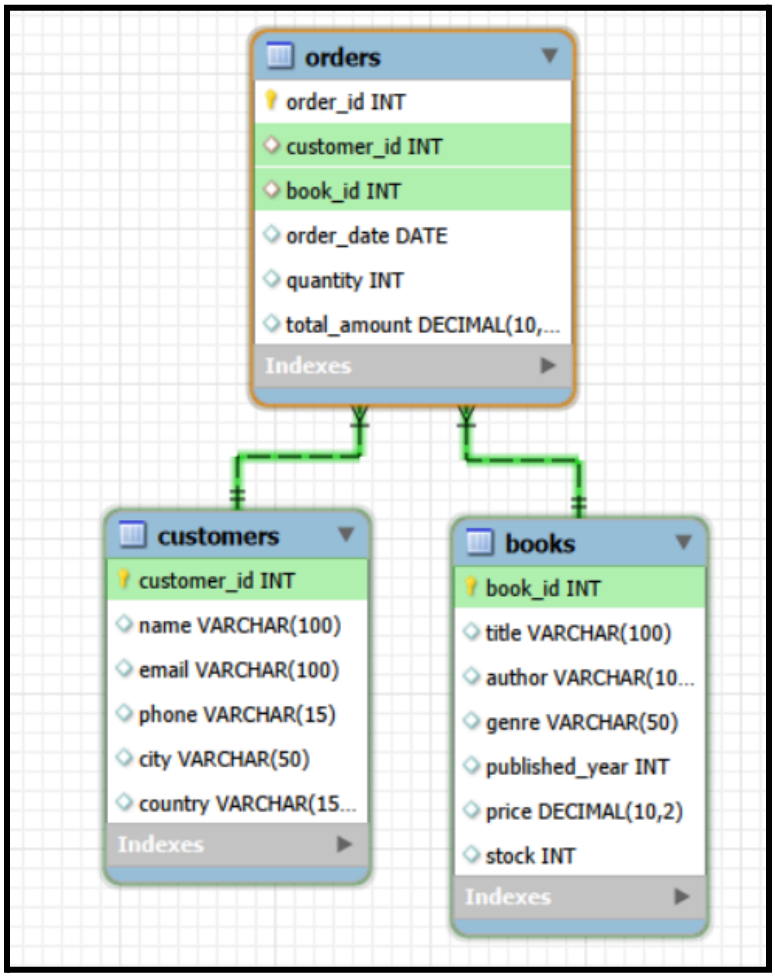
Customer.CSV

Customer_ID	Name	Email	Phone	City	Country
1	James York	James@gmail.com	1212121212	Port Angela	Moldova
2	Melissa Curtis	Melissa@gmail.com	1212121212	Ashleyport	Monaco

Orders.CSV

Order_ID	Customer_ID	Book_ID	Order_Date	Quantity	Total_Amount
1	297	137	2024-06-17	5	131.65
2	140	48	2023-08-26	1	24.7

Relationship Between these tables:



Questions:

Basic level:
1) Retrieve all books in the "Fiction" genre
2) Find books published after the year 1950
3) List all customers from the Canada
4) Show orders placed in November 2023
5) Retrieve the total stock of books available
6) Find the details of the most expensive book
7) Show all customers who ordered more than 1 quantity of a book
8) Retrieve all orders where the total amount exceeds $20
9) List all genres available in the Books table
10) Find the book with the lowest stock
11) Calculate the total revenue generated from all orders

Intermediate level:
1) Find the average price of books in the "Fantasy" genre
2) Show the top 3 most expensive books of 'Fantasy' Genre
3) Retrieve the total number of books sold for each genre
4) Retrieve the total quantity of books sold by each author

Advance level:
1) List customers who have placed at least 2 orders
2) Find the most frequently ordered book
3) List the cities where customers who spent over $30 are located
4) Find the customer who spent the most on orders
5) Calculate the stock remaining after fulfilling all orders

Solution:

```
-- Create Database
CREATE DATABASE OnlineBookstore;

-- Switch to the database
use OnlineBookstore;

-- Create Tables

CREATE TABLE Books (
    Book_ID SERIAL PRIMARY KEY,
    Title VARCHAR(100),
    Author VARCHAR(100),
    Genre VARCHAR(50),
    Published_Year INT,
    Price NUMERIC(10, 2),
    Stock INT
);

CREATE TABLE Customers (
    Customer_ID SERIAL PRIMARY KEY,
    Name VARCHAR(100),
    Email VARCHAR(100),
    Phone VARCHAR(15),
    City VARCHAR(50),
    Country VARCHAR(150)
);

CREATE TABLE Orders (
    Order_ID SERIAL PRIMARY KEY,
    Customer_ID INT REFERENCES Customers(Customer_ID),
    Book_ID INT REFERENCES Books(Book_ID),
    Order_Date DATE,
    Quantity INT,
    Total_Amount NUMERIC(10, 2)
);
```

-- View Tables

```
SELECT * FROM Books;
SELECT * FROM Customers;
SELECT * FROM Orders;
```

-- Import Data

```
-- Import Data into Books Table
-- Import Data into Customers Table
-- Import Data into Orders Table
```

 # Basic level:

-- 1) Retrieve all books in the "Fiction" genre:

```
SELECT * FROM books
WHERE Genre = 'Fiction';
```

-- 2) Find books published after the year 1950:

```
SELECT * FROM books
WHERE Published_year > 1950;
```

-- 3) List all customers from the Canada:

```
SELECT * FROM Customers
WHERE country='Canada';
```

-- 4) Show orders placed in November 2023:

```
SELECT * FROM Orders
WHERE order_date BETWEEN '2023-11-01' AND '2023-11-30';
```

-- 5) Retrieve the total stock of books available:

```
SELECT SUM(stock) AS Total_Stock
From Books;
```

-- 6) Find the details of the most expensive book:

```
SELECT * FROM Books
ORDER BY Price DESC
LIMIT 1;
```

-- 7) Show all customers who ordered more than 1 quantity of a book:

```
SELECT * FROM Orders
WHERE quantity > 1;
```

-- 8) Retrieve all orders where the total amount exceeds $20:

```
SELECT * FROM Orders
WHERE total_amount > 20;
```

-- 9) List all genres available in the Books table:

```
SELECT DISTINCT genre
FROM Books;
```

-- 10) Find the book with the lowest stock:

```
SELECT * FROM Books
ORDER BY stock
LIMIT 1;
```

-- 11) Calculate the total revenue generated from all orders:

```
SELECT SUM(total_amount) As Revenue
FROM Orders;
```

 # Intermediate level:

-- 1) Find the average price of books in the "Fantasy" genre:

```
SELECT AVG(price) AS Average_Price
FROM Books
WHERE Genre = 'Fantasy';
```

-- 2) Show the top 3 most expensive books of 'Fantasy' Genre :

```
SELECT * FROM books
WHERE genre ='Fantasy'
ORDER BY price DESC
LIMIT 3;
```

-- 3) Retrieve the total number of books sold for each genre:

```
SELECT b.Genre, SUM(o.Quantity) AS Total_Books_sold
FROM Orders o
JOIN Books b
ON o.book_id = b.book_id
GROUP BY b.Genre;
```

-- 4) Retrieve the total quantity of books sold by each author:

```
SELECT b.author, SUM(o.quantity) AS Total_Books_Sold
FROM orders o
JOIN books b
ON o.book_id=b.book_id
GROUP BY b.Author;
```

 # Advance level:

-- 1) List customers who have placed at least 2 orders:

```
SELECT o.customer_id, c.name, COUNT(o.Order_id) AS ORDER_COUNT
FROM orders o
JOIN customers c
ON o.customer_id=c.customer_id
GROUP BY o.customer_id, c.name
HAVING COUNT(Order_id) >= 2;
```

-- 2) Find the most frequently ordered book:

```
SELECT o.Book_id, b.title, COUNT(o.order_id) AS ORDER_COUNT
FROM orders o
JOIN books b
ON o.book_id=b.book_id
GROUP BY o.book_id, b.title
ORDER BY ORDER_COUNT DESC
LIMIT 1;
```

-- 3) List the cities where customers who spent over $30 are located:

```
SELECT DISTINCT c.city, total_amount
FROM orders o
JOIN customers c
ON o.customer_id=c.customer_id
WHERE o.total_amount > 30;
```

-- 4) Find the customer who spent the most on orders:

```
SELECT c.customer_id, c.name, SUM(o.total_amount) AS Total_Spent
FROM orders o
JOIN customers c
ON o.customer_id=c.customer_id
GROUP BY c.customer_id, c.name
ORDER BY Total_spent Desc
LIMIT 1;
```

-- 5) Calculate the stock remaining after fulfilling all orders:

```
SELECT b.book_id, b.title, b.stock, COALESCE(SUM(o.quantity),0) AS
Order_quantity,
        b.stock - COALESCE(SUM(o.quantity),0) AS Remaining_Quantity
FROM books b
LEFT JOIN orders o
ON b.book_id=o.book_id
GROUP BY b.book_id
ORDER BY b.book_id;
```

👋 **Congrats on completing the Complete SQL Course! You've leveled up your data skills—now go use them to build, analyze, and create amazing things. Keep learning, keep coding, and keep shining!**

Source : www.youtube.com/@OfficialAccountAmitKumar

Next Step:

1. **Subscribe My youtube channel: Click Here**
2. **Create More SQL Projects : Click Here**
3. **Practice SQL Interview Questions : Click Here**

जय बजरंग बली

Source : **www.youtube.com/@OfficialAccountAmitKumar**

📣Want to connect with me? Check out these links:📣

👉Youtube: https://www.youtube.com/@OfficialAccountAmitKumar

👉LinkedIn: https://www.linkedin.com/in/amit1990