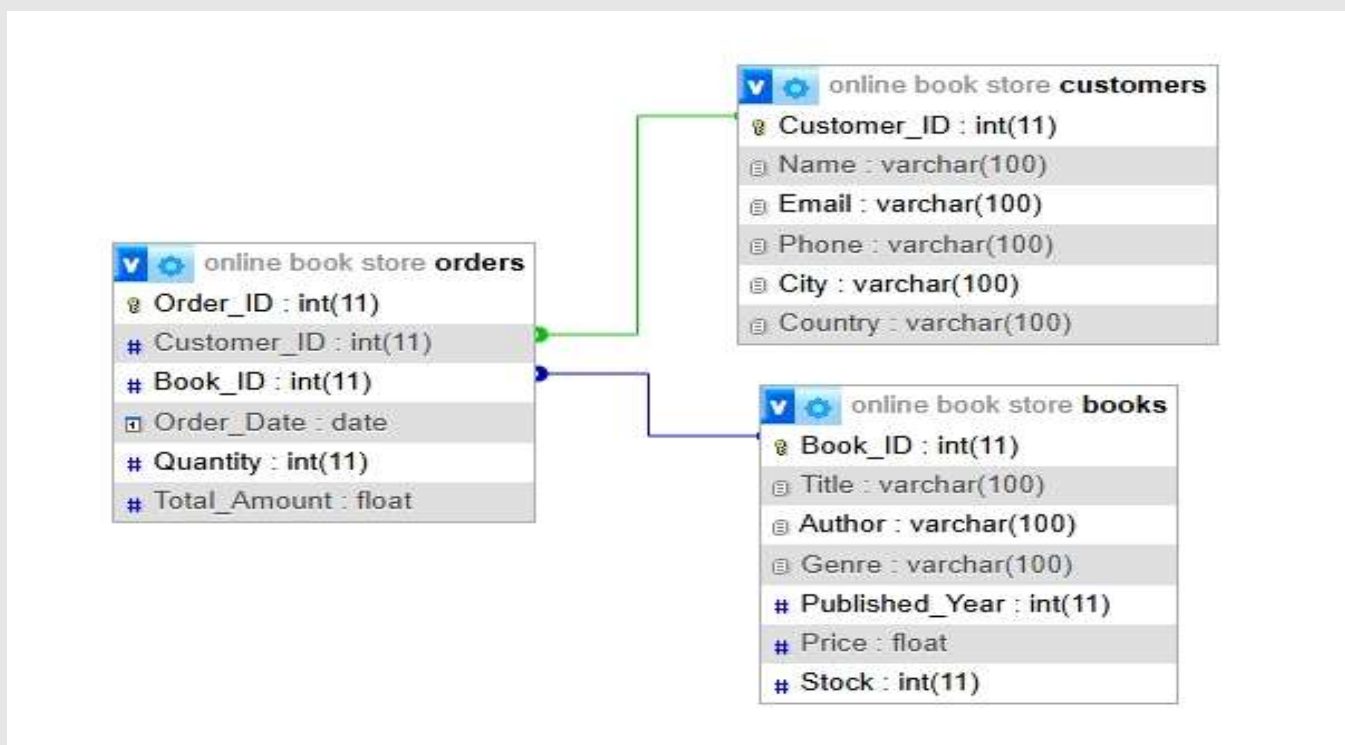


SQL Project On Online Book Store

Objective:- To derive valuable insights(using SQL query) so that organization will grow and revenue as well.

ER-Model:- one fact table(orders) and two dimensional tables(customers and books). ER-Model is given as below-



(1)- **Retrieve all books in the "Fiction" genre.**

Q= SELECT Title as "Books Fiction-Genre" FROM books WHERE Genre="Fiction";

(2)- **Books published after year 1950.**

Q= SELECT Title as "Books Published After 1950" FROM books WHERE Published_Year>1950;

(3)-**List all customers from Canada.**

Q= SELECT Name as "Customer From Canada" FROM customers WHERE Country="Canada";

(4)-Show orders placed in November 2023.

Q=SELECT SUM(Quantity) as "Order Placed in Nov 2023" FROM orders WHERE Order_Date BETWEEN "2023-11-01" AND "2023-11-30";

Order Placed in Nov 2023
118

(5)- Retrieve the total stock of books available.

Q=SELECT SUM(Stock) as "Total Stock Of Books Available" FROM books;

Total Stock Of Books Available
25056

(6)- The details of the most expensive book.

Q=SELECT * FROM books ORDER BY Price DESC LIMIT 1;

Book_ID	Title	Author	Genre	Published_Year	Price	Stock
340	Proactive system-worthy orchestration	Robert Scott	Mystery	1907	49.98	88

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

Q=SELECT customers.Name as "Customer Name",orders.Quantity as "Ordered MoreThan 1 Quantity" FROM customers LEFT JOIN orders ON customers.Customer_ID=orders.Customer_ID WHERE orders.Quantity>1;

(8)- Retrieve all orders where the total amount exceeds \$20.

Q=SELECT * FROM orders WHERE Total_Amount>20;

(9)- List all genres available in the Books table.

Q= SELECT DISTINCT(Genre) AS "All Genre in Book Table" FROM books;

All Genre in Book Table
Biography
Fantasy
Non-Fiction
Fiction
Romance
Science Fiction
Mystery

(10)- **The book with the lowest stock.**

Q= SELECT * FROM books ORDER BY Stock ASC LIMIT 1;

Book_ID	Title	Author	Genre	Published_Year	Price	Stock
44	Networked systemic implementation	Ryan Frank	Science Fiction	1965	13.55	0

(11)- **The total revenue generated from all orders.**

Q= SELECT SUM(Total_Amount) as "Total Revenue From All Orders" FROM orders;

Total Revenue From All Orders
75628.66016149521

(12)- **Retrieve the total number of books sold for each genre.**

Q= SELECT books.Genre AS "Genre",SUM(orders.Quantity) AS "Total Quantity" FROM orders JOIN books ON orders.Book_ID=books.Book_ID GROUP BY books.Genre;

Genre	Total Quantity
Biography	285
Fantasy	446
Fiction	225
Mystery	504
Non-Fiction	351
Romance	439
Science Fiction	447

(13)- **The average price of books in the "Fantasy" genre.**

Q= SELECT AVG(Price) AS "Average Price(Fantasy)" FROM books WHERE Genre="Fantasy";

Average Price(Fantasy)
25.981690111294597

(14)- **List customers who have placed at least 2 orders.**

```
Q=SELECT customers.Customer_ID,customers.Name,COUNT(orders.Order_ID)
AS "2 or More Orders Placed" FROM orders JOIN customers ON
orders.Customer_ID=customers.Customer_ID GROUP BY
customers.Customer_ID,customers.Name HAVING
COUNT(orders.Order_ID)>=2;
```

(15)- **The most frequently ordered book.**

```
Q= SELECT books.Book_ID,books.Title AS "Book(Most Frequently
Ordered)",COUNT(orders.Order_ID) AS "Total Order" FROM orders JOIN books
ON orders.Book_ID=books.Book_ID GROUP BY books.Book_ID,books.Title
ORDER BY COUNT(orders.Order_ID) DESC LIMIT 1;
```

Book_ID	Book(Most Frequently Ordered)	Total Order
31	Implemented encompassing conglomeration	4

(16)- **The top 3 most expensive books of 'Fantasy' Genre.**

```
Q= SELECT Book_ID,Title,Genre,Price AS "Total Price" FROM books WHERE
Genre="Fantasy" ORDER BY Price DESC LIMIT 3;
```

Book_ID	Title	Genre	Total Price
240	Stand-alone content-based hub	Fantasy	49.9
462	Innovative 3rdgeneration database	Fantasy	49.23
238	Optimized even-keeled analyzer	Fantasy	48.97

(17)- **Retrieve the total quantity of books sold by each author.**

```
Q= SELECT books.Author AS "Author",SUM(orders.Quantity) AS "Total
Quantity" FROM orders JOIN books ON orders.Book_ID=books.Book_ID GROUP
BY books.Author;
```

(18)- **List the cities where customers who spent over \$30 are located.**

```
Q=SELECT customers.City AS "City",SUM(orders.Total_Amount) AS "Total
Amount" FROM orders JOIN customers ON
```

orders.Customer_ID=customers.Customer_ID GROUP BY customers.City
HAVING SUM(orders.Total_Amount)>30;

(19)- Customer who spent the most on orders.

Q=SELECT customers.Name AS "Customer Name",SUM(orders.Total_Amount)
AS "Total Amount" FROM orders JOIN customers ON
orders.Customer_ID=customers.Customer_ID GROUP BY customers.Name
ORDER BY SUM(orders.Total_Amount) DESC LIMIT 1;

Customer Name	Total Amount
Kim Turner	1398.9000091552734

(20)- Calculate the stock remaining after fulfilling all orders.

Q=SELECT books.Book_ID,books.Title AS
"Book",books.Stock,COALESCE(SUM(orders.Quantity),0) AS
"Ordered(QTY)",books.Stock-COALESCE(SUM(orders.Quantity),0) AS
"Remaining(QTY)" FROM books LEFT JOIN orders ON
books.Book_ID=orders.Book_ID GROUP BY books.Book_ID LIMIT 5;

Book_ID	Book	Stock	Ordered(QTY)	Remaining(QTY)
1	Configurable modular throughput	100	3	97
2	Persevering reciprocal knowledge user	19	0	19
3	Streamlined coherent initiative	27	5	22
4	Customizable 24hour product	8	0	8
5	Adaptive 5thgeneration encoding	16	8	8
