SQL Practice – Online Bookstore Dataset

Basic SQL Queries

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 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

```
SELECT customers.Name, orders.Quantity
FROM customers
JOIN orders ON customers.Customer_id = orders.Customer_id
WHERE orders.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 ASC
LIMIT 1;
```

---

11) Calculate the total revenue generated from all orders

```
SELECT SUM(Total_Amount) AS Total_Revenue
FROM orders;
```

Advanced SQL Queries

---

1) Retrieve the total number of books sold for each genre

```
SELECT books.Genre, SUM(orders.Quantity) AS Total_Books_Sold
FROM books
JOIN orders ON books.Book_ID = orders.Book_ID
GROUP BY books.Genre;
```

---

2) Find the average price of books in the “Fantasy” genre

```
SELECT AVG(Price) AS Average_Price
FROM books
WHERE Genre = 'Fantasy';
```

---

3) List customers who have placed at least 2 orders

```
SELECT Customer_ID, COUNT(Order_ID) AS order_count
FROM orders
GROUP BY Customer_ID
HAVING COUNT(Order_ID) >= 2;
```

---

4) Find the most frequently ordered book

```
SELECT o.Book_ID, b.Title, COUNT(o.Order_ID) AS Total_Orders
FROM orders o
JOIN books b ON b.Book_ID = o.Book_ID
GROUP BY o.Book_ID, b.Title
ORDER BY Total_Orders DESC
LIMIT 1;
```

---

5) Show the top 3 most expensive books of the “Fantasy” genre

```
SELECT *
FROM books
WHERE Genre = 'Fantasy'
ORDER BY Price DESC
LIMIT 3;
```

---

6) Retrieve the total quantity of books sold by each author

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

---

7) List the cities where customers who spent over $30 are located

```
SELECT DISTINCT c.city, o.total_amount
FROM customers c
JOIN orders o ON o.customer_id = c.customer_id
WHERE o.total_amount > 300;
```

---

8) Find the customer who spent the most on orders

```
SELECT c.Customer_ID, c.Name, SUM(o.Total_Amount) AS total_spent
FROM customers c
JOIN orders o ON o.Customer_ID = c.Customer_ID
GROUP BY c.Customer_ID, c.Name
ORDER BY total_spent DESC
LIMIT 1;
```

---

9) Calculate the remaining stock 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 remain_stock
FROM books b
LEFT JOIN orders o ON o.Book_ID = b.Book_ID
GROUP BY b.Book_ID, b.Title, b.Stock
ORDER BY b.Book_ID;
```