# SQL PROJECT

1. Retrieve all books in the "Fiction" genre.

**Answer:-**

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

---

2. Find books published after the year 1950.

**Answer:-**

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

---

3. List all customers from the Canada.

**Answer:-**

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

---

4. Show orders placed in November 2023.

**Answer:-**

```
SELECT * FROM ordersg
WHERE order_date >='2023-11-01' AND
    order_date <= '2023-11-30';
```

---

5. Retrieve the total stock of books available.

**Answer:-**

```
SELECT SUM(stock)
FROM books;
```

---

6. Find the details of the most expensive book.

**Answer:-**

```
SELECT * FROM books
ORDER BY price DESC;
```

---

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

**Answer:-**

```
SELECT o.customer_id , o.quantity, c.name
FROM ordersg o
LEFT JOIN customers c
ON o.customer_id = c.customer_id
WHERE o.quantity > 1 ;
```

---

8. Retrieve all orders where the total amount exceeds $20.

**Answer:-**

```
SELECT * FROM ordersG
WHERE total_amount > 20
ORDER BY total_amount ASC;
```

---

9. List all genres available in the Books table.

**Answer:-**

```
SELECT DISTINCT genre FROM books;
```

# SQL PROJECT

10. Find the book with the lowest stock.

**Answer:-**

```
SELECT * FROM books
ORDER BY stock ASC;
```

---

11. Calculate the total revenue generated from all orders.

**Answer:-**

```
SELECT SUM (total_amount) AS total_revenue
FROM ordersg;
```

---

12. Retrieve the total number of books sold for each genre.

**Answer:-**

```
SELECT b.genre, COUNT(o.quantity)
FROM books b
JOIN ordersg o
ON b.book_id = o.book_id
GROUP BY (b.genre);
```

---

13. Find the average price of books in the "Fantasy" genre.

**Answer:-**

```
SELECT AVG(price)
FROM books
WHERE genre ='Fantasy';
```

---

14. List customers who have placed at least 2 orders.

**Answer:-**

```
SELECT customer_id,quantity
FROM ordersg
WHERE quantity >= 2
ORDER BY quantity;
```

---

15. Find the most frequently ordered book.

**Answer:-**

```
SELECT book_id,COUNT(book_id) AS order_frequency
FROM ordersg
GROUP BY book_id
ORDER BY order_frequency DESC;
```

---

16. Show the top 3 most expensive books of 'Fantasy' Genre.

**Answer:-**

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

---

# SQL PROJECT

17. Retrieve the total quantity of books sold by each author.

**Answer:-**

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

---

18. List the cities where customers who spent over $30 are located.

**Answer:-**

```
SELECT o.total_amount,c.city
FROM ordersg o
JOIN customers c
ON o.customer_id = c.customer_id
WHERE o.total_amount > 30
ORDER BY o.total_amount ;
```

---

19. Find the customer who spent the most on orders.

**Answer:-**

```
SELECT o.total_amount,c.name
FROM ordersg o
JOIN customers c
ON o.customer_id = c.customer_id
ORDER BY o.total_amount DESC;
```

---

20. Calculate the stock remaining after fulfilling all orders.

**Answer:-**

```
SELECT b.book_id, b.title ,b.stock , COALESCE (SUM(o.quantity),0) AS ordered_quantity,
b.stock - COALESCE (SUM(o.quantity),0) AS remaining_stock
FROM books b
LEFT JOIN ordersg o
ON b.book_id = o.book_id
GROUP BY b.book_id,b.title,b.stock
ORDER BY remaining_stock DESC;
```