SQL Queries

```sql
select * from books;
select * from customers;
select * from orders;

-- 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 * FROM books
WHERE "Stock" > 0;

-- 6) Find the details of the most expensive book:
SELECT * FROM books
WHERE "Price" = (
    SELECT MAX("Price") FROM books
);



-- 7) Show all customers who ordered more than 1 quantity of a book:
SELECT
  customers."Name" AS Cust_Name,
  orders."Quantity" AS Quantity
FROM
```

```
  customers
JOIN
  orders ON customers."Customer_ID" = orders."Customer_ID"
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
WHERE "Stock" <= (SELECT MIN("Stock") FROM books)+1;

-- Second
SELECT * FROM books
WHERE "Stock" = (SELECT MIN("Stock") FROM books);

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

select sum("Total_Amount") AS Revenue from orders;


-- Advance Questions :

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

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

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

```
SELECT "Genre", ROUND(AVG("Price")::numeric,2) AS AVG_Price FROM books
WHERE "Genre" = 'Fantasy'
GROUP BY
 "Genre";




-- 3) List customers who have placed at least 2 orders:
SELECT customers."Name", No_Order FROM
(select "Customer_ID" AS Customer_ID, COUNT("Order_ID") AS No_Order from orders
GROUP BY
 "Customer_ID") AS sub
JOIN
 customers ON customers."Customer_ID" = sub.Customer_ID
WHERE No_Order >= 2




-- 4) Find the most frequently ordered book:
SELECT
    books."Title",
    sub.No_Order
FROM (
    SELECT
        "Book_ID" AS book_id,
        COUNT("Order_ID") AS No_Order
    FROM
        orders
    GROUP BY
        "Book_ID"
    ORDER BY
        No_Order DESC
) AS sub
JOIN
    books ON books."Book_ID" = sub.book_id
ORDER BY
    sub.No_Order DESC
LIMIT 1;



-- 5) Show the top 3 most expensive books of '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" AS Author,
    SUM(orders."Quantity") AS Total_Quantity_Sold
FROM
    books
JOIN
    orders ON orders."Book_ID" = books."Book_ID"
GROUP BY
    books."Author"
ORDER BY
 Total_Quantity_Sold DESC;



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

SELECT
 DISTINCT customers."City" AS city,
 orders."Total_Amount" AS Total_Amt
FROM orders
JOIN
 customers ON customers."Customer_ID" = orders."Customer_ID"
WHERE
 "Total_Amount" > 30;



--8. Find the customer who spent the most on orders
SELECT
    customers."Name",
    SUM(orders."Total_Amount") AS spent
FROM
    orders
JOIN
    customers ON orders."Customer_ID" = customers."Customer_ID"
GROUP BY
    customers."Name"
```

```
ORDER BY
    spent DESC
LIMIT 1;



--9) Calculate the stock remaining after fulfilling all orders:
SELECT
    books."Title",
    books."Stock",
    COALESCE(SUM(orders."Quantity"), 0) AS Total_Ordered,
    books."Stock" - COALESCE(SUM(orders."Quantity"), 0) AS Remaining_Stock
FROM
    books
LEFT JOIN
    orders ON books."Book_ID" = orders."Book_ID"
GROUP BY
    books."Book_ID", books."Title", books."Stock";
```

SQL Queries

SQL Queries

SQL Queries