SQL Project on Online Book Store

```
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
        );


drop table if exists customers


create table Customers (
        Customer_ID serial primary key,
        Name varchar(100),
        Email varchar (100),
        Phone varchar (100),
        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 ,
        Quntity int ,
        Total_amount numeric (10,2));
```

```
select *from Books;
select *from Customers;
select *from Orders;
```

Question Basic Queries

1. Retrives all books in the table " Fiction " genere :

Answer :-

```
select *from  Books
where genre = 'Fiction';
```

2. Find book 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 order placed in November 2023:

Answer :-

```
select * from Orders
where Order_date  between '2023-11-01' and '2023-11-30';
```

5. Retrieve the total stock of books available:

Answer :-

```
select sum (Stock) as  total_stock
```

```
from books;
```

6. Find the details of the most expensive book:

Answer :-

```
select * from Books order by price desc
    limit 1;
```

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

Answer :-

```
select * from Orders
where  Quntity >1 ;
```

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

Answer :-

```
select * from Orders
where  total_amount > 20;
```

9. List all genres available in the books table:

Answer :-

```
select distinct genre from books;
```

10 Find the book with lower stock

Answer :-

```
select *from books order by stock limit 1 ;
```

11. Calculate the total revenue generated from all orders .

Answer :-

```
select  sum (Total_amount) as Total_revenue  from Orders;
```

-- advance Queries

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

Answer :-

```
select b.genre, sum(o.Quntity)
from  books b
join orders o on o.book_id = b.book_id
group by b.genre ;
```

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

Answer :-

```
select   avg(price) as  avg_price
from books
where  genre = 'Fantasy';
```

3. List customer who have placed at least 2 orders.

Answer :-

```
select  customer_id , count  (order_id) as  order_count
from orders
group by customer_id
having count  (order_id)>=2;
```

-- with customer name and id .

Answer :-

```
select o.Customer_ID, c.Name , count(o.order_id) as count_order
from orders o
join Customers c on o.Customer_ID = c.Customer_ID
group by o.Customer_ID , c.Name
having count (o.order_id) >=2;
```

4.find most Frequently ordered book.

Answer :-

```
select  book_id ,  count (order_id) as count_order
from orders
group by book_id
order by count_order desc limit 1;
```

-- with book name

Answer :-

```
select o.book_id , b.Title, count(o.order_id) as  cont_order_id
from orders o
join books b on o.book_id = b.book_id
group by o.book_id , b.Title
order by cont_order_id desc limit 1  ;
```

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

Answer :-

```
select  * from books
where genre = 'Fantasy'
order by price desc limit 3;
```

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

Answer :-

```
select  b.Author, sum(o.Quntity) as total_Quntity
from books b
```

```
join orders o on o.Book_ID = b.Book_ID
 group by b.Author;
```

7. List the cities where customer who spend the over $30 are located .

Answer :-

```
select  distinct c.city , o.Total_amount
from orders o
join customers c on c.Customer_ID = o.Customer_ID
where o.Total_amount  >30;
```

8. Find the customer who spend the most on order .

Answer :-

```
select  distinct c.name, o.customer_id , sum(o.Total_amount) as most_amout_used
 from orders o
 join customers c on c.customer_id =o.customer_id
 group by c.name , o.customer_id
 order by most_amout_used  desc ;
```

9. Calculate the stock remaining after fulfilling all orders

Answer :-

```
select b.book_id , b.title, b.stock, coalesce(sum(o.Quntity),0) ,
 b.stock - coalesce(sum(o.Quntity),0)  as remaning_book
 from books b
 left join orders o on  b.book_id = o.book_id
 group  by b.book_id
 order by b.book_id asc;
```