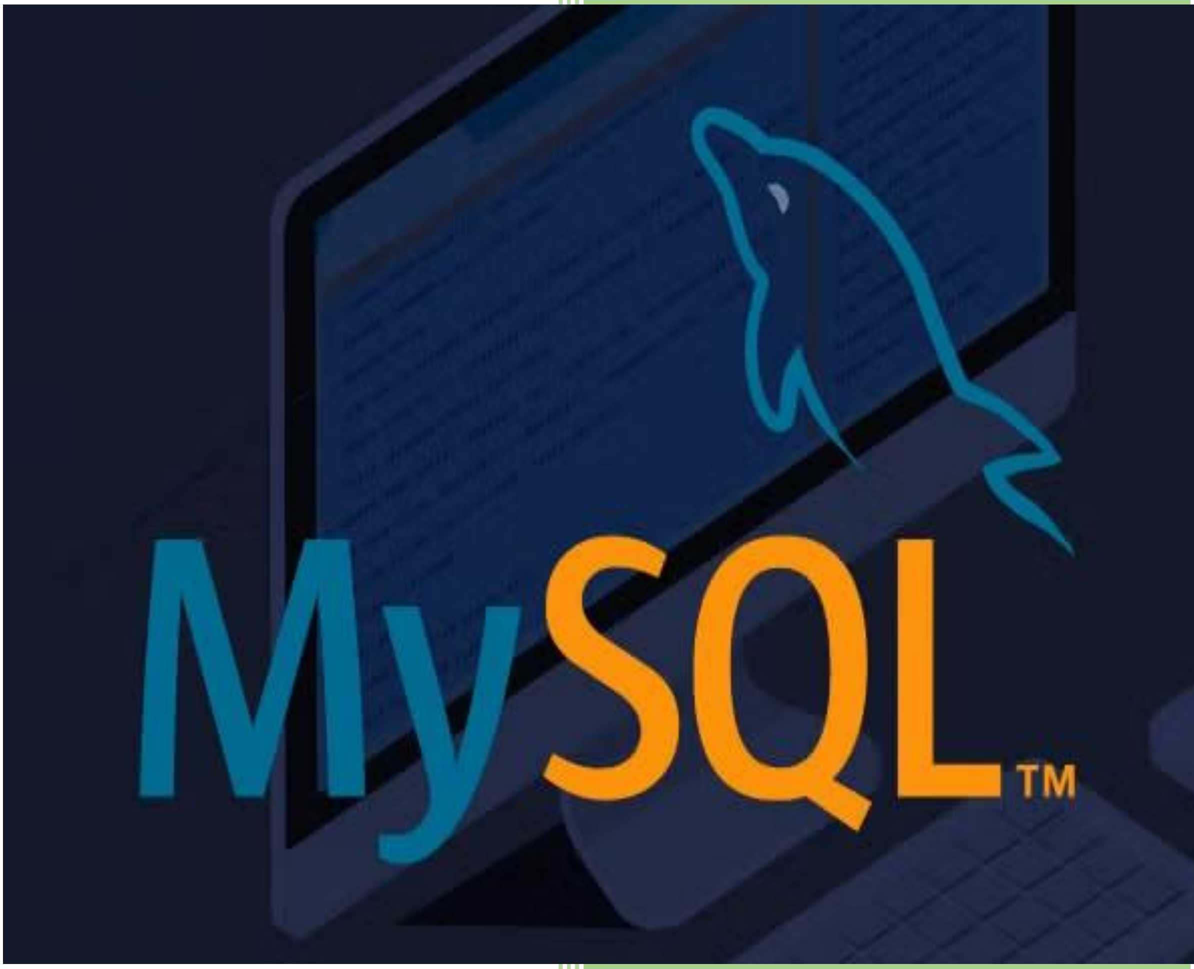
2025

Online_Book_Store_Analysis_using_SQL



VINAY KUMAR PANIKA

2/1/2025

Online_Book_Store_Analysis_using_SQL

```
create database online_book_store;

use online_book_store;

select * from books;

select * from customers;

select * from orders;
```

Basic Questions

Que 1) Retrieve all books in the "Fiction" genre:

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

Que 2) Find books published after the year 1950:

```
select * from books where Published_Year > 1950;
```

Que 3) List all customers from the Canada:

```
select * from customers where country = 'Canada';
```

Que 4) Show orders placed in November 2023:

```
select * from orders where year(order_date)=2023 and month(order_date)= 11;
```

Que 5) Retrieve the total stock of books available:

```
select sum(stock) as total_stock from books;
```

Que 6) Find the details of the most expensive book:

```
select * from books where price = (select max(price) from books);
```

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

```
select c.*,o.Quantity from customers as c
inner join orders as o on c.customer_id = o.customer_id
where o.Quantity >1 ;
```

Que 8) Retrieve all orders where the total amount exceeds $20:

```
select * from orders
where Total_Amount > 20;
```

Que 9) List all genres available in the Books table:

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

Que 10) Find the book with the lowest stock:

```
select * from books
where stock = (select min(stock) from books);
```

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

```
select round(sum(Total_Amount),2) as total_revenue from orders;
```

Advance Questions

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

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

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

```
select round(avg(price),2) as Average_price from books
where genre = "Fantasy";
```

Que 3) List customers who have placed at least 2 orders:

```
select c.Customer_id,c.name,c.city,c.country,count(o.customer_id) as no_of_orders from customers as c
inner join orders as o
on c.customer_id = o.Customer_id
group by customer_id
having count(o.customer_id)>=2;
```

Que 4) Find the most frequently ordered book:

```
select b.title,b.book_id,count(O.order_id) as no_of_orders from books as b
inner join orders as o
on b.book_id = o.book_id
group by b.title
order by count(O.order_id) desc;
```

Que 5) Show the top 3 most expensive books of 'Fantasy' Genre :

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

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

```
select b.author,sum(o.quantity) as Quantity_sold from books as b
inner join orders as o on
b.book_id = o.book_id
group by b.author
order by sum(o.quantity) desc ;
```

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

```
select distinct city,o.total_amount from customers as c
inner join orders as o on
c.customer_id = o.customer_id
where o.total_amount>30;
```

Que 8) Find the customer who spent the most on orders:

```
select c.customer_id,c.name,
round(sum(o.total_amount),2) as Total_spend from customers as c
inner join orders as o on c.customer_id= o.customer_id
group by c.customer_id
order by sum(o.total_amount) desc
limit 1;
```

Que 9) Calculate the stock remaining after fulfilling all orders:

```
SELECT  b.Book_ID, b.Title,b.Stock - COALESCE(SUM(o.Quantity), 0) AS Stock_Remaining
FROM Books as  b
LEFT JOIN Orders as  o ON b.Book_ID = o.Book_ID
GROUP BY b.Book_ID, b.Title, b.Stock
ORDER BY Stock_Remaining DESC;
```