SQL Learning



Nithish Gowda

# DAY - 30 - **End to End SQL Portfolio Project for Data analyst | SQL Project for Resume**

We have useed 3 tables books.csv, customers.csv, orders.csv

-- Create Database

CREATE DATABASE OnlineBookstore;

-- Switch to the database

\c OnlineBookstore;

-- Create Tables

DROP TABLE IF EXISTS Books;

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(15),

City VARCHAR(50),

Country VARCHAR(150)

);

DROP TABLE IF EXISTS orders;

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,

Quantity INT,

Total\_Amount NUMERIC(10, 2)

);

SELECT \* FROM Books;

SELECT \* FROM Customers;

SELECT \* FROM Orders;

-- Import Data into Books Table

COPY Books(Book\_ID, Title, Author, Genre, Published\_Year, Price, Stock)

FROM 'C:\Users\User\Desktop\SQL\Project\Books.csv'

CSV HEADER;

-- Import Data into Customers Table

COPY Customers(Customer\_ID, Name, Email, Phone, City, Country)

FROM 'D:\Course Updates\30 Day Series\SQL\CSV\Customers.csv'

CSV HEADER;

-- Import Data into Orders Table

COPY Orders(Order\_ID, Customer\_ID, Book\_ID, Order\_Date, Quantity, Total\_Amount)

FROM 'D:\Course Updates\30 Day Series\SQL\CSV\Orders.csv'

CSV HEADER;

-- 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

order by published\_year asc;

-- 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 sum(stock) as total\_stock

from books;

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

select book\_id,title,price

from books

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

select \* from books

order by price desc limit 1 ;

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

select \* from orders

where quantity >2;

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

select \* from orders

where total\_amount >20

order by total\_amount desc;

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

select distinct genre as genre\_list

from books;

-- 10) Find the book with the lowest stock:

select \* from books

order by stock asc

limit 10;

select \* from books

where stock<10

order by stock asc;

-- 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 b.genre,sum(o.quantity) as total\_books

from orders o

join books b

on o.book\_id=b.book\_id

group by b.genre;

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

select AVG(price) as AVG\_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;

select o.customer\_id, c.name,count(o.order\_id) as order\_count

from orders o

join customers c

on o.customer\_id=c.customer\_id

group by o.customer\_id, c.name

having count(order\_id)>=2;

-- 4) Find the most frequently ordered book:

select \* from orders;

select book\_id, count(order\_id) as order\_count

from orders

group by book\_id

order by order\_count desc

limit 3;

select o.book\_id,b.title, count(o.order\_id) as order\_count

from orders o

join books b

on o.book\_id=b.book\_id

group by o.book\_id,b.title

order by order\_count desc

limit 3;

-- 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 b.book\_id,b.author,Sum(o.quantity) as total\_quan

from orders o

join books b

on o.book\_id=b.book\_id

group by b.book\_id,b.author,o.quantity

order by total\_quan desc;

select b.author,Sum(o.quantity) as total\_quan

from orders o

join books b

on o.book\_id=b.book\_id

group by b.author ;

order by total\_quan desc;

select distinct author

from books;

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

select \* from customers;

select \* from orders;

select \* from books;

select distinct c.city , total\_amount

from orders o

join customers c

on o.customer\_id=c.customer\_id

where o.total\_amount>30;

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

select c.name,c.customer\_id ,sum(o.total\_amount) as spend\_amt

from orders o

join customers c

on o.customer\_id=c.customer\_id

group by c.customer\_id,c.name

order by spend\_amt desc

limit 3;

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

select b.book\_id,b.title,b.stock, coalesce(sum(o.quantity),0) as order\_quan,

b.stock- coalesce(sum(o.quantity),0) as reaining\_quan

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 ;