**SQL PROJECT (CUSTOMERS\_ORDERS\_PRODUCTS)**

Create Table Customers( CustomerID int Primary Key,Name Varchar(50),Email Varchar(100))

Insert into Customers(CustomerID,Name,Email) Values (1,'John Doe','johndoe@example.com'),

(2,'Jane Smith','janesmith@example.com'),(3,'Robert Johnson','robertjohnson@example.com'),

(4,'Emily Brown','emilybrown@example.com'),(5,'Michael Davis','michaeldavis@example.com'),

(6,'Sarah Wilson','sarahwilson@example.com'),(7,'David Thompson','davidthompson@example.com'),

(8,'Jessica Lee','jessicalee@example.com'),(9,'William Turner','williamturner@example.com'),

(10,'Olivia Martinez','oliviamartinez@example.com')

Create Table Orders( OrderID int Primary Key,CustomerID int, ProductName Varchar(50),

OrderDate Date, Quantity int)

Insert into Orders ( OrderID,CustomerID,ProductName,OrderDate,Quantity) Values

(1, 1 , 'Product A','2023-07-01',5),(2, 2 , 'Product B','2023-07-02',3),(3, 3 , 'Product C','2023-07-03',2)

, (4, 4 , 'Product A','2023-07-04',1),(5, 5, 'Product B','2023-07-05',4),(6, 6 , 'Product C','2023-07-06',2),

(7, 7 , 'Product A','2023-07-07',3),(8, 8, 'Product B','2023-07-08',2),(9, 9 , 'Product C','2023-07-09',5),

(10, 10 , 'Product A','2023-07-10',1)

Create Table Products( ProductID int Primary Key, ProductName varchar(50), Price DECIMAL(10,2))

Insert into Products(ProductID,ProductName,Price) Values (1,'Product A',10.99),(2,'Product B',8.99),

(3,'Product C',5.99),(4,'Product D',12.99),(5,'Product E',7.99),(6,'Product F',6.99),

(7,'Product G',9.99),(8,'Product H',11.99),(9,'Product I',14.99),(10,'Product J',4.99)

-- Query to retrieve all records from customer table.

select\* from Customers

--Query to retrieve Name and Email of customers whose name start with 'J'.

select Name, Email from Customers where Name like 'J%'

-- Query to retrieve order details (OrderID, ProductName,Quantity) from orders.

select OrderID,ProductName,Quantity from Orders

--Query to calculate the total quantity of products ordered.

select SUM(Quantity) 'Total\_Quantity' from Orders

-- Query to retrieve names of customers who have placed an order.

select Name from Customers

-- Query to retrieve products with price greater than $10.

select\* from Products where Price > $10

--Query to retrieve customer Name and order date for all orders placed on or after '2023-07-05'.

select Customers.Name,Orders.OrderDate from Customers inner join Orders on

Customers.CustomerID=Orders.CustomerID where OrderDate> '2023-07-05'

-- Query to calculate Average Price of all Products.

select AVG(Price) 'Average\_ Price' from Products

--Query to retrieve customer name along with total quantity of products they have ordered.

Select Customers.Name,SUM(Orders.Quantity) As 'Total Quantity Ordered' from Customers

join Orders on Customers.CustomerID= Orders.CustomerID Group by Customers.Name

--Write a query to retrieve the products that have not been ordered.

select P. ProductName from Products P left join Orders O on O.ProductName=P.ProductName where O.ProductName is null

--Query to retrieve top 5 customers who have placed highest total quantity of orders.

select Customers.Name,SUM(Orders.Quantity) As 'TotalQuantityOrdered' from Customers join

Orders on Customers.CustomerID=Orders.OrderID group by Customers.Name order by TotalQuantityOrdered DESC

--Query to calculate average price of products for each product category.

select Products.ProductName, AVG(Products.Price) AS AveragePrice from Products join Orders

on Products.ProductName=Orders.ProductName group by Products.ProductName

--query to retrieve customers who have not placed any orders

select Customers.Name from Customers left join Orders on Customers.CustomerID= Orders.CustomerID

where Customers.CustomerID is null

--Write a query to retrieve the order details (OrderID, ProductName, Quantity) for orders placed by customers whose names start with 'M'.

select Orders.OrderID,Orders.ProductName,Orders.Quantity from Orders join Customers

On Customers.CustomerID=Orders.CustomerID where Customers.Name like 'M%'

--Write a query to calculate the total revenue generated from all orders.

Select SUM(Orders.Quantity\*Products.Price) AS Total\_Revenue FROM Orders join Products on

Orders.ProductName=Products.ProductName

--Write a query to retrieve the top 3 products with the highest average quantity ordered.

select Products.ProductName,AVG(Orders.Quantity) AS AverageQuantityOrdered from Products join

Orders on Orders.ProductName= Products.ProductName

group by Products.ProductName order by AverageQuantityOrdered DESC

--Write a query to retrieve the customer names along with the total revenue generated from their orders.

Select Customers.Name, SUM(Orders.Quantity\*Products.Price) AS TotalRevenue

FROM customers

JOIN orders ON Customers.customerID = Orders.customerID

JOIN products ON Orders.ProductName = Products.ProductName

GROUP BY Customers.Name

--Write a query to retrieve the customers who have placed at least one order for each product category.

select customerID,Quantity from orders where quantity >=1

--Write a query to retrieve the top 3 products with the highest average quantity ordered.

select TOP 3 ProductName ,COUNT(\*) As Number\_of\_products,AVG (Quantity) As AverageQuantity

from Orders GROUP BY ProductName ORDER BY AVG (Quantity) DESC

--Write a query to retrieve the products that have been ordered by all customers.

Select Distinct Products.ProductName from Products left join Orders

On Products.ProductName=Orders.ProductName where Orders.ProductName IS NOT NULL

--Write a query to retrieve the customers who have placed orders for all products in a specific category

select DISTINCT Name, ProductName From Orders left join Customers on

Customers.CustomerID=Orders.CustomerID

--Write a query to retrieve the top 5 customers who have spent the highest amount of money on orders

SELECT c.CustomerID, c.Name, SUM(p.Price \* o.Quantity) AS TotalSpent

FROM Customers c

JOIN Orders o ON c.CustomerID = o.CustomerID

JOIN Products p ON o.ProductName = p.ProductName

GROUP BY c.CustomerID, c.Name

ORDER BY TotalSpent DESC

--Write a query to calculate the running total of order quantities for each customer.

SELECT

o.CustomerID,

c.Name AS CustomerName,

o.OrderID,

o.ProductName,

o.OrderDate,

o.Quantity,

SUM(o.Quantity) OVER (PARTITION BY o.CustomerID ORDER BY o.OrderDate) AS RunningTotal

FROM

Orders o

JOIN

Customers c ON o.CustomerID = c.CustomerID

ORDER BY

o.CustomerID, o.OrderDate;

--Write a query to retrieve the top 3 most recent orders for each customer

WITH RankedOrders AS (

SELECT

o.CustomerID,

o.OrderID,

o.ProductName,

o.OrderDate,

o.Quantity,

ROW\_NUMBER() OVER (PARTITION BY o.CustomerID ORDER BY o.OrderDate DESC) AS RowNum

FROM

Orders o

)

SELECT

r.CustomerID,

c.Name AS CustomerName,

r.OrderID,

r.ProductName,

r.OrderDate,

r.Quantity

FROM

RankedOrders r

JOIN

Customers c ON r.CustomerID = c.CustomerID

WHERE

r.RowNum <= 3

ORDER BY

r.CustomerID, r.OrderDate DESC;

--Write a query to calculate the average revenue per order for each customer.

SELECT

c.CustomerID,

c.Name AS CustomerName,

AVG(p.Price \* o.Quantity) AS AvgRevenuePerOrder

FROM

Customers c

JOIN

Orders o ON c.CustomerID = o.CustomerID

JOIN

Products p ON o.ProductName = p.ProductName

GROUP BY

c.CustomerID, c.Name

ORDER BY

AvgRevenuePerOrder DESC;