Online Retail Project by SQL

Summary:
Analyzed Online Retail data (Customers, Products, Categories, Orders, OrderItems) in SQL to uncover sales trends, top products, and customer behavior.

Problem Statement:
Retailers struggle to identify top-selling products, valuable customers, and seasonal trends for better decisions.

Solution:
Used SQL joins, aggregations, and KPIs to find best-sellers, high-value customers, and sales patterns.

Key Highlights:

- Normalized relational database design.
- KPIs: Revenue, Avg Order Value, Repeat Buyer Rate.
- Seasonal and customer segmentation analysis.

Impact:

- Better marketing targeting.
- Inventory optimization.
- Improved sales strategy with data-driven insights.

create customer table

```
create Table Customers (
    CustomerID  INT PRIMARY KEY IDENTITY(1,1),
       FirstName NVARCHAR(50),
       LASTNAME NVARCHAR(50),
    Email NVARCHAR(100),
       Phone VARCHAR(50),
       Address NVARCHAR(225),
       City NVARCHAR(50),
       State NVARCHAR(50),
       Zipcode NVARCHAR(50),
       Country NVARCHAR(50),
       CreatedAt DATETIME DEFAULT Getdate()
)
```

create Products table

```
create table Products(
    ProductID INT PRIMARY KEY IDENTITY (1,1),
       ProductName NVARCHAR(100),
       CategoryID INT,
       Price DECIMAL(10,2),
```

```
       Stock INT,
       CreatedAt DATETIME DEFAULT getdate()
)
```

create Categories table

```
create table Categories(
    CategoryID INT PRIMARY KEY IDENTITY (1,1),
       CategoryName NVARCHAR(100),
       Description NVARCHAR(225),
       )
```

create orders table

```
create table Orders(
    OrderID INT PRIMARY KEY IDENTITY (1,1),
    CustomerID INT,
       TotalAmout DECIMAL(10,2),
       OrderDate DATETIME DEFAULT getdate(),
       FOREIGN KEY (CustomerID) REFERENCES Customers (customerID)
)
```

create the orderItems Table

```
create table Order_Items(
    OrderItemID INT PRIMARY KEY IDENTITY (1,1),
    OrderID INT,
       ProductID INT,
       Quantity INT,
       Price Decimal(10,2)
       FOREIGN KEY (ProductID) REFERENCES Products (ProductID),
    FOREIGN KEY (OrderID) REFERENCES Orders (OrderID))
```

Insert data into Categories table

```
INSERT INTO Categories  (CategoryName, Description) values ('Electronics', 'Devices and
Gadgets')
INSERT INTO Categories  (CategoryName, Description) values ('Clothing', 'Apparel and
Accessories')
INSERT INTO Categories  (CategoryName, Description) values ('Books', 'printed and
electronic books')
```

Insert data into Products table

```
INSERT INTO Products(ProductName, CategoryID, Price, Stock)
Values('Smartphone',1,699.99,50),
('Laptop', 1 , 999.99, 30),
('Tshirt', 2 , 19.99 , 100),
('Jeans', 2, 49.99, 60),
('Fiction Nove', 3, 14.99, 200),
('Science Journal', 3, 29.99, 150)
```

Insert data into customers table

```
INSERT INTO Customers(FirstName, LastName, Email, Phone, Address, City, State, Zipcode,
Country)
Values
('Sameer', 'Khanna', 'Sameer.khanna@exa.com', '123-456-7890', '123 EML ST.',
'Springfeild', 'IL', '62701', 'USA'),
('Jane', 'SMITH', 'Jane.smith@exa.com', '234-567-8945', '1456 Eoal RT.', 'Madison', 'WI',
'84566', 'USA'),
('Alice', 'Johnson', 'Alice.Johnson@exa.com', '741-852-8632', '852THB YU.', 'Mumbai',
'Maharastra', '41530', 'INDIA')
```

Insert data into orders table

```
INSERT INTO Orders(CustomerId, OrderDate, TotalAmount)
Values
(1, Getdate(), 719.98),
(2, getdate(), 49.99),
(3, getdate(), 44.98)
```

Insert data into orderItems table

```
INSERT INTO OrderItems(OrderID, ProductID, Quantity, Price)
Values
(1, 1, 1, 699.99),
(1, 3, 2, 19.99),
(2, 4, 1, 49.99),
(3, 5, 1, 14.99),
(3, 6, 1, 29.99)
```

Que1 Retrive all orders for specific customer

```
select o.OrderID, o.OrderDate, o.TotalAmount, oi.ProductID, p.ProductName, oi.Quantity,
oi.Price
from orders o
JOIN OrderItems oi ON o.OrderID = oi.OrderID
JOIN Products p on oi.ProductID = p.ProductID
where o.CustomerID = 1
```

	OrderID	OrderDate	TotalAmount	ProductID	ProductName	Quantity	Price
1	1	2025-07-19 23:07:52.020	719.98	1	Smartphone	1	699.99
2	1	2025-07-19 23:07:52.020	719.98	3	Tshirt	2	19.99

Que 2 Find total sales for each product

```
select p.ProductID, p.ProductName, sum(oi.Quantity * oi.Price) AS 'TotalSales'
from OrderItems oi
Join Products P on p.ProductID = oi.ProductID
group by p.productID, p.ProductName
```

	ProductID	ProductName	TotalSales
1	1	Smartphone	699.99
2	3	Tshirt	39.98
3	4	Jeans	49.99
4	5	Fiction Nove	14.99
5	6	Science Journal	29.99

Que 3. find average order value

```
select avg(totalAmount) as averageordervalue from orders
```

	averageordervalue
1	1562.970000

Que 4 Lit top 3 customers by total spending

```
select CustomerID, FirstName, LastName, Total_spending
from
(select c.CustomerID, c.FirstName, c.LastName, sum(TotalAmount) AS Total_spending,
Row_Number() over (order by sum(TotalAmount)  DESC) AS rn
from Customers c
join orders o on
o.CustomerID = c.CustomerID
group by c.CustomerID, c. FirstName, c.LastName)
sub where rn <= 3
```

	CustomerID	FirstName	LastName	Total_spending
1	4	Sam	Khan	6999.90
2	1	Sameer	Khanna	719.98
3	2	Jane	SMITH	49.99

Que 5 retrive most popular product category

```
select categoryID, categoryName, TotalquantitySold
from
(select c.categoryID, c.CategoryName, sum(oi.Quantity) as TotalQuantitySold,
row_Number() over(order by sum(oi.Quantity) DESC) AS rn
from OrderItems oi
join products p on
oi.ProductID = p.ProductID
Join Categories c on
p.CategoryID = c.CategoryID
Group by c.categoryID, c.CategoryName) sub
where rn = 1
```

	categoryID	categoryName	TotalquantitySold
1	2	Clothing	3

Que 6 list all products that are out of stock .e sstock = 0

added 1 product with zero stock

```
INSERT INTO Products(ProductName, CategoryID, Price, Stock)
Values('keyboard',1,39.99,0)
```

list all products that are out of stock .e sstock = 0

```
select ProductID, ProductName, stock from Products
where stock = 0
```

	ProductID	ProductName	stock
1	7	keyboard	0
2	1002	keyboard	0

Que 7 Find customers who placed orders in the last 30 days

```
select C.CustomerID, C.FirstName, C.Lastname
from Customers C
Join orders o on
o.CustomerID = C.CustomerID
where o.OrderDate >= DATEADD(Day, -30, getdate())
```

	CustomerID	FirstName	Lastname
1	1	Sameer	Khanna
2	2	Jane	SMITH
3	3	Alice	Johnson
4	4	Sam	Khan
5	4	Sam	Khan

Que 8 calculate total number of orders placed each month

```
select  year(orderDate) AS OrderYear, Month(orderDate) AS OrderMonth, count(orderID) as
TotalOrders
from orders
group by year(orderDate), Month(orderDate)
```

	OrderYear	OrderMonth	TotalOrders
1	2025	7	5

Que 9 retrive the details of the recent order

```
select Top 1 o.OrderID, o.OrderDate, o.TotalAmount, c.FirstName,c.LastName
from orders o
Join customers c on
o.customerID = c.CustomerID
order by orderDate desc
```

	OrderID	OrderDate	TotalAmount	FirstName	LastName
1	1003	2025-07-22 20:01:59.087	3499.95	Sam	Khan

10 find product price of each category

```
select c.categoryID, c.CategoryName, AVG(p.Price)  as AveragePrice
from categories c
Join Products p on
c.CategoryID = p.ProductID
group by c.categoryID, c.CategoryName
```

	categoryID	CategoryName	AveragePrice
1	1	Electronics	6999.990000
2	2	Clothing	999.990000
3	3	Books	19.990000

Que11 List customers who have never placed an order

add 1 customer data in customers who has not records in orders table

```
INSERT INTO Customers(FirstName, LastName, Email, Phone, Address, City, State, Zipcode,
Country)
Values
('Sam', 'Khan', 'Sam.khan@exa.com', '457-712-6542', '854 WDS RS.', 'Springfeild', 'IL',
'62701', 'USA')
select * from customers

select c.CustomerID, c.FirstName, c.lastName, o.TotalAmount, o.orderID
from Customers c
left JOIN orders o on
c.CustomerID = o.CustomerID
where o.orderID is null
```

	CustomerID	FirstName	lastName	TotalAmount	orderID
1	1002	Sam	Khan	NULL	NULL

Que 12 retrive the total quantity sold for each product

```
select p.ProductID, p.ProductName, sum(oi.Quantity) AS TotalQuantitySold
from orderItems as oi
Join Products p on
oi.productID = p.ProductID
group by p.ProductID, p.ProductName
```

	ProductID	ProductName	TotalQuantitySold
1	1	Smartphone	1
2	3	Tshirt	2
3	4	Jeans	1
4	5	Fiction Nove	1
5	6	Science Journal	1

Que 13 Calculate the total revenue generated from each category

```
select c.CategoryID, c.CategoryName, sum(oi.Quantity * oi.Price) AS TotalRevenue
from OrderItems oi
Join Products p on
oi.ProductID = p.ProductID
JOIN Categories c on
c.categoryID = p.CategoryID
group by c.CategoryID, c.CategoryName
order by TotalRevenue desc
```

	CategoryID	CategoryName	TotalRevenue
1	1	Electronics	699.99
2	2	Clothing	89.97
3	3	Books	44.98

Que 14 find highest priced product in category

```
select c.CategoryID, c.CategoryName, p1.ProductID, p1.ProductName, p1.Price
from categories c
Join Products p1 on
c.categoryID = p1.categoryID
where p1.price = (select max(price) from products p2 where p2.categoryID = p1.CategoryID)
order by p1.price desc
```

	CategoryID	CategoryName	ProductID	ProductName	Price
1	1	Electronics	1	Smartphone	6999.99
2	2	Clothing	4	Jeans	49.99
3	3	Books	6	Science Journal	29.99

Que 15 Retrive orders with a total amount greater than specific value (e.g - $500)

```
select o.OrderID, c.CustomerID, c.FirstName, c.LastName, o.TotalAmount
from Orders o JOIN Customers c ON
o.customerID = c.CustomerID
where o.TotalAmount >= 49.99
order by o.TotalAmount DESC
```

	OrderID	CustomerID	FirstName	LastName	TotalAmount
1	1002	4	Sam	Khan	3499.95
2	1003	4	Sam	Khan	3499.95
3	1	1	Sameer	Khanna	719.98
4	2	2	Jane	SMITH	49.99

Que 16 List Products along with the no of orders they appear in

```
select p.ProductID, p.ProductName, count(oi.OrderID) as OrderCount
from Products p
Join OrderItems oi
on p.ProductID = oi.ProductID
Group by p.ProductID, p.ProductName
order by orderCount DESC
```

	ProductID	ProductName	OrderCount
1	1	Smartphone	1
2	3	Tshirt	1
3	4	Jeans	1
4	5	Fiction Nove	1
5	6	Science Journal	1

Que 17 Find top 3 most frequently ordered products

```
select Top 3 p.ProductID, p.ProductName, count(oi.orderID) AS OrderCount
from OrderItems oi
join Products p on
oi.ProductID = p.ProductID
Group by p.ProductID, p.ProductName
order by OrderCount desc
```

	ProductID	ProductName	OrderCount
1	5	Fiction Nove	1
2	4	Jeans	1
3	3	Tshirt	1

Que 18 Calculate the total no of customers from each country

```
select Country, Count(CustomerID) AS TotalCustomers
from Customers
Group by Country
```

	Country	TotalCustomers
1	INDIA	1
2	USA	4

	Country	TotalCustomers
1	INDIA	1
2	USA	4

Que 19 retrive the list of customers along with their total spending

```
select c.CustomerID, c.FirstName, c.LastName, Sum(o.TotalAmount) AS TotalSpending
from Customers c
Join Orders o on
o.CustomerID = c.CustomerID
Group by  c.CustomerID, c.FirstName, c.LastName
```

	CustomerID	FirstName	LastName	TotalSpending
1	1	Sameer	Khanna	719.98
2	2	Jane	SMITH	49.99
3	3	Alice	Johnson	44.98
4	4	Sam	Khan	6999.90

Que 20 List Orders with more than a specified no of items(e.g 2 items)

```
select o.orderId, c.CustomerID, c.FirstName,c.LastName, Count(oi.OrderItemID) AS
NoOfItems
from Orders o
Join OrderItems oi
ON oi.OrderID = o.OrderID
Join Customers c ON
o.CustomerID = c.CustomerID
Group by o.orderId, c.CustomerID, c.FirstName,c.LastName
Having Count(oi.OrderItemID) >=1
Order by NoOfItems
```

	orderId	CustomerID	FirstName	LastName	NoOfItems
1	2	2	Jane	SMITH	1
2	3	3	Alice	Johnson	2
3	1	1	Sameer	Khanna	2