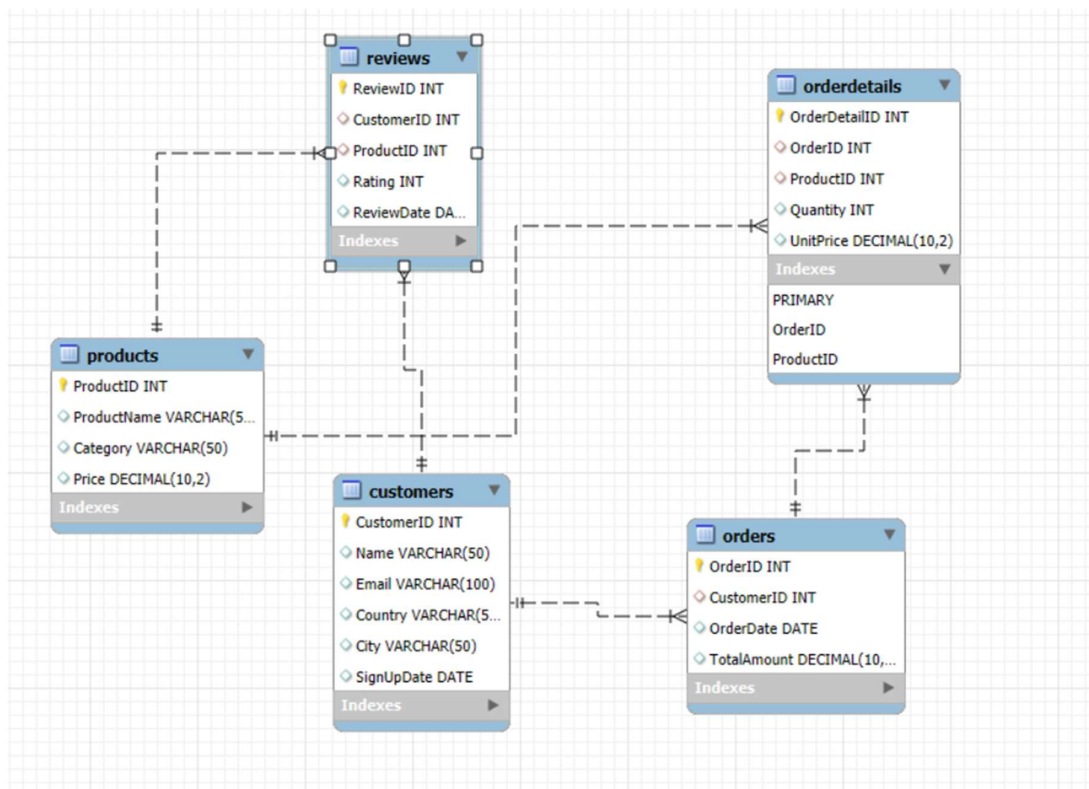


Project Title

SALES ANALYSIS AND INSIGHTS FOR AN E-COMMERCE COMPANY

Project Highlights:

- Designed and implemented a comprehensive relational database for an e-commerce platform, including tables for customers, products, orders, reviews, and order details.
- Developed and executed SQL queries for:
 - Top-Selling Products:** Identified products generating the highest revenue.
 - Customer Analysis:** Determined the most frequent and valuable customers.
 - Sales Trends:** Analyzed monthly and yearly sales performance.
 - Category Performance:** Identified the best-performing product categories.
 - Geographical Insights:** Analyzed regions/cities with the highest sales.
 - Product Rating Analysis:** Correlated product sales with average ratings.
- Applied EDA techniques to handle missing values and update inaccurate records.
- Used advanced SQL techniques like Common Table Expressions (CTEs), aggregations, and joins to derive actionable insights.
- Focused on customer retention by analyzing repeat purchase trends and categorizing buyers.



```

/* Creating Database */
CREATE DATABASE Ecommerce;
USE Ecommerce;

/* Creating tables */
CREATE TABLE Customers (
    CustomerID INT PRIMARY KEY,
    Name VARCHAR(50),
    Email VARCHAR(100),
    Country VARCHAR(50),
    City VARCHAR(50),
    SignUpDate DATE
);

CREATE TABLE Products (
    ProductID INT PRIMARY KEY,
    ProductName VARCHAR(50),
    Category VARCHAR(50),
    Price DECIMAL(10, 2)
);

CREATE TABLE Orders (
    OrderID INT PRIMARY KEY,
    CustomerID INT,
    OrderDate DATE,
    TotalAmount DECIMAL(10, 2),
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
);

CREATE TABLE OrderDetails (
    OrderDetailID INT PRIMARY KEY,
    OrderID INT,
    ProductID INT,
    Quantity INT,
    UnitPrice DECIMAL(10, 2),
    FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
);

CREATE TABLE Reviews (
    ReviewID INT PRIMARY KEY,
    CustomerID INT,
    ProductID INT,
    Rating INT,
    ReviewDate DATE,
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID),
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
);

/* Inserting values into tables */
INSERT INTO Customers VALUES
(1, 'Alice Green', 'alice@example.com', 'USA', 'New York', '2023-01-10'),
(2, 'Bob Brown', 'bob@example.com', 'Canada', 'Toronto', '2023-02-15'),
(3, 'Charlie Blue', 'charlie@example.com', 'UK', 'London', '2023-03-20'),
(4, 'Daisy White', 'daisy@example.com', 'India', 'Bangalore', '2023-04-10'),
(5, 'Eve Black', 'eve@example.com', 'Australia', 'Sydney', '2023-05-05');

```

```
INSERT INTO Products VALUES
(101, 'Smartphone', 'Electronics', 699.99),
(102, 'Laptop', 'Electronics', 1199.99),
(103, 'T-shirt', 'Apparel', 19.99),
(104, 'Sneakers', 'Footwear', 89.99),
(105, 'Blender', 'Home Appliances', 49.99);
```

```
INSERT INTO Orders VALUES
(201, 1, '2023-01-15', 719.98),
(202, 2, '2023-02-20', 1299.98),
(203, 3, '2023-03-25', 109.98),
(204, 4, '2023-04-15', 69.98),
(205, 5, '2023-05-10', 179.97);
```

```
INSERT INTO OrderDetails VALUES
(301, 201, 101, 1, 699.99),
(302, 201, 103, 1, 19.99),
(303, 202, 102, 1, 1199.99),
(304, 202, 104, 1, 89.99),
(305, 203, 103, 2, 19.99),
(306, 204, 104, 1, 89.99),
(307, 205, 103, 3, 19.99);
```

```
INSERT INTO Reviews VALUES
(401, 1, 101, 5, '2023-01-20'),
(402, 2, 102, 4, '2023-02-25'),
(403, 3, 103, 3, '2023-03-30'),
(404, 4, 104, 4, '2023-04-20'),
(405, 5, 103, 5, '2023-05-15');
```

/* Inserting more NULL data */

```
INSERT INTO Customers VALUES
(6, 'Ahmed Ali', NULL, 'UAE', 'Dubai', '2023-05-15'),
(7, 'Mei Ling', 'mei.ling@example.com', 'China', 'Beijing', NULL),
(8, 'Michael Johnson', 'michael.j@example.com', NULL, 'New York', '2023-06-10'),
(9, NULL, NULL, 'Canada', 'Toronto', '2023-07-01'),
(10, 'Rajesh Kumar', 'INVALID_EMAIL', 'India', 'Delhi', NULL);
```

```
SELECT * FROM Customers;
```

	CustomerID	Name	Email	Country	City	SignUpDate
▶	1	Alice Green	alice@example.com	USA	New York	2023-01-10
	2	Bob Brown	bob@example.com	Canada	Toronto	2023-02-15
	3	Charlie Blue	charlie@example.com	UK	London	2023-03-20
	4	Daisy White	daisy@example.com	India	Bangalore	2023-04-10
	5	Eve Black	eve@example.com	Australia	Sydney	2023-05-05
	6	Ahmed Ali	NULL	UAE	Dubai	2023-05-15
	7	Mei Ling	mei.ling@example.com	China	Beijing	NULL
	8	Michael Johnson	michael.j@example.com	NULL	New York	2023-06-10
	9	NULL	NULL	Canada	Toronto	2023-07-01
	10	Rajesh Kumar	INVALID_EMAIL	India	Delhi	NULL

```
SELECT * FROM Products;
```

	ProductID	ProductName	Category	Price
▶	101	Smartphone	Electronics	699.99
	102	Laptop	Electronics	1199.99
	103	T-shirt	Apparel	19.99
	104	Sneakers	Footwear	89.99
	105	Blender	Home Appliances	49.99
*	NULL	NULL	NULL	NULL

SELECT * FROM Orders;

	OrderID	CustomerID	OrderDate	TotalAmount
▶	201	1	2023-01-15	719.98
	202	2	2023-02-20	1299.98
	203	3	2023-03-25	109.98
	204	4	2023-04-15	69.98
	205	5	2023-05-10	179.97
*	NULL	NULL	NULL	NULL

SELECT * FROM OrderDetails;

	OrderDetailID	OrderID	ProductID	Quantity	UnitPrice
▶	301	201	101	1	699.99
	302	201	103	1	19.99
	303	202	102	1	1199.99
	304	202	104	1	89.99
	305	203	103	2	19.99
	306	204	104	1	89.99
	307	205	103	3	19.99
*	NULL	NULL	NULL	NULL	NULL

SELECT * FROM Reviews;

	ReviewID	CustomerID	ProductID	Rating	ReviewDate
▶	401	1	101	5	2023-01-20
	402	2	102	4	2023-02-25
	403	3	103	3	2023-03-30
	404	4	104	4	2023-04-20
	405	5	103	5	2023-05-15
*	NULL	NULL	NULL	NULL	NULL

/* EDA */

/* Handling Missing values in Names */

SELECT * FROM Customers;

	CustomerID	Name	Email	Country	City	SignUpDate
▶	1	Alice Green	alice@example.com	USA	New York	2023-01-10
	2	Bob Brown	bob@example.com	Canada	Toronto	2023-02-15
	3	Charlie Blue	charlie@example.com	UK	London	2023-03-20
	4	Daisy White	daisy@example.com	India	Bangalore	2023-04-10
	5	Eve Black	eve@example.com	Australia	Sydney	2023-05-05
	6	Ahmed Ali	NULL	UAE	Dubai	2023-05-15
	7	Mei Ling	mei.ling@example.com	China	Beijing	NULL
	8	Michael Johnson	michael.j@example.com	NULL	New York	2023-06-10
	9	NULL	NULL	Canada	Toronto	2023-07-01
	10	Rajesh Kumar	INVALID_EMAIL	India	Delhi	NULL

/* Filling Missing values in Name column */

UPDATE Customers

SET Name='Unknown'

WHERE Name IS NULL;

```
SELECT * FROM Customers;
```

	CustomerID	Name	Email	Country	City	SignUpDate
	2	Bob Brown	bob@example.com	Canada	Toronto	2023-02-15
	3	Charlie Blue	charlie@example.com	UK	London	2023-03-20
	4	Daisy White	daisy@example.com	India	Bangalore	2023-04-10
	5	Eve Black	eve@example.com	Australia	Sydney	2023-05-05
	6	Ahmed Ali	NULL	UAE	Dubai	2023-05-15
	7	Mei Ling	mei.ling@example.com	China	Beijing	NULL
	8	Michael Johnson	michael.j@example.com	NULL	New York	2023-06-10
	9	Unknown	NULL	Canada	Toronto	2023-07-01
	10	Rajesh Kumar	INVALID_EMAIL	India	Delhi	NULL

```
/* Filling Missing values in Email column */
```

```
UPDATE Customers
SET Email = 'unknown@example.com'
WHERE Email IS NULL;
SELECT * FROM Customers;
```

	CustomerID	Name	Email	Country	City	SignUpDate
▶	1	Alice Green	alice@example.com	USA	New York	2023-01-10
	2	Bob Brown	bob@example.com	Canada	Toronto	2023-02-15
	3	Charlie Blue	charlie@example.com	UK	London	2023-03-20
	4	Daisy White	daisy@example.com	India	Bangalore	2023-04-10
	5	Eve Black	eve@example.com	Australia	Sydney	2023-05-05
	6	Ahmed Ali	unknown@example.com	UAE	Dubai	2023-05-15
	7	Mei Ling	mei.ling@example.com	China	Beijing	NULL
	8	Michael Johnson	michael.j@example.com	NULL	New York	2023-06-10
	9	Unknown	unknown@example.com	Canada	Toronto	2023-07-01
	10	Rajesh Kumar	INVALID_EMAIL	India	Delhi	NULL

```
/* Filling Missing values in country column */
```

```
UPDATE Customers
SET Country='NoCountry'
WHERE Country is NULL;
SELECT * FROM Customers;
```

	CustomerID	Name	Email	Country	City	SignUpDate
▶	1	Alice Green	alice@example.com	USA	New York	2023-01-10
	2	Bob Brown	bob@example.com	Canada	Toronto	2023-02-15
	3	Charlie Blue	charlie@example.com	UK	London	2023-03-20
	4	Daisy White	daisy@example.com	India	Bangalore	2023-04-10
	5	Eve Black	eve@example.com	Australia	Sydney	2023-05-05
	6	Ahmed Ali	unknown@example.com	UAE	Dubai	2023-05-15
	7	Mei Ling	mei.ling@example.com	China	Beijing	NULL
	8	Michael Johnson	michael.j@example.com	NoCountry	New York	2023-06-10
	9	Unknown	unknown@example.com	Canada	Toronto	2023-07-01
	10	Rajesh Kumar	INVALID_EMAIL	India	Delhi	NULL

```
/* Filling Missing values in SignUpDate column */
```

```
UPDATE Customers
SET SignUpDate = (
SELECT ROUND(AVG(SignUpDate)) FROM
(SELECT SignUpDate FROM Customers WHERE SignUpDate IS NOT NULL) AS temp
)
WHERE SignUpDate IS NULL;
SELECT * FROM Customers;
```

	CustomerID	Name	Email	Country	City	SignUpDate
▶	1	Alice Green	alice@example.com	USA	New York	2023-01-10
	2	Bob Brown	bob@example.com	Canada	Toronto	2023-02-15
	3	Charlie Blue	charlie@example.com	UK	London	2023-03-20
	4	Daisy White	daisy@example.com	India	Bangalore	2023-04-10
	5	Eve Black	eve@example.com	Australia	Sydney	2023-05-05
	6	Ahmed Ali	unknown@example.com	UAE	Dubai	2023-05-15
	7	Mei Ling	mei.ling@example.com	China	Beijing	2023-04-23
	8	Michael Johnson	michael.j@example.com	NoCountry	New York	2023-06-10
	9	Unknown	unknown@example.com	Canada	Toronto	2023-07-01
	10	Rajesh Kumar	INVALID_EMAIL	India	Delhi	2023-04-23

/* Top-Selling products :- Which Product generate the highest revenue */

```
WITH CTE1 AS (
SELECT ProductID,
Quantity*UnitPrice AS Total
FROM OrderDetails
),
CTE2 AS (
SELECT ProductID, SUM(Total) AS Total
FROM CTE1 GROUP BY ProductID
)
SELECT p.ProductID,p.ProductName,
c.Total AS HightestRevenue
FROM Products p LEFT JOIN CTE2 c ON p.ProductID=c.ProductID
ORDER BY c.Total DESC LIMIT 1;
```

	ProductID	ProductName	HightestRevenue
▶	102	Laptop	1199.99

/* Customer Analysis:- Who are the most frequent or valuable customers? */

```
WITH CTE1 AS (
SELECT OrderID,
COUNT(OrderID) AS CountOfOrder,
SUM(Quantity*UnitPrice) as Spend
FROM OrderDetails
GROUP BY OrderID
)
SELECT
o.CustomerID,
cu.Name,
SUM(c.CountOfOrder) OVER( PARTITION BY o.CustomerID) AS Frequency,
c.Spend
FROM Orders o LEFT JOIN CTE1 c
ON o.OrderID=c.OrderID LEFT JOIN Customers cu
ON cu.CustomerID=o.CustomerID
ORDER BY c.Spend DESC
LIMIT 1;
```

	CustomerID	Name	Frequency	Spend
▶	2	Bob Brown	2	1289.98

/* Sales trends:- Analyze Monthly and Yearly sales performance */

```
WITH CTE1 AS (
SELECT
od.ProductID,
od.Quantity,
MONTHNAME(o.OrderDate) AS MonthName,
```



```

o.OrderDate
FROM OrderDetails od LEFT JOIN Orders o
ON od.OrderID=o.OrderID
)
SELECT ProductID,
SUM(CASE WHEN MonthName='January' THEN Quantity ELSE 0 END) AS January,
SUM(CASE WHEN MonthName='February' THEN Quantity ELSE 0 END) AS February,
SUM(CASE WHEN MonthName='March' THEN Quantity ELSE 0 END) AS March,
SUM(CASE WHEN MonthName='April' THEN Quantity ELSE 0 END) AS April,
SUM(CASE WHEN MonthName='May' THEN Quantity ELSE 0 END) AS May,
SUM(CASE WHEN MonthName='June' THEN Quantity ELSE 0 END) AS June,
SUM(CASE WHEN MonthName='July' THEN Quantity ELSE 0 END) AS July,
SUM(CASE WHEN MonthName='August' THEN Quantity ELSE 0 END) AS August,
SUM(CASE WHEN MonthName='September' THEN Quantity ELSE 0 END) AS September,
SUM(CASE WHEN MonthName='October' THEN Quantity ELSE 0 END) AS October,
SUM(CASE WHEN MonthName='November' THEN Quantity ELSE 0 END) AS November,
SUM(CASE WHEN MonthName='December' THEN Quantity ELSE 0 END) AS December,
SUM(Quantity) AS Yearly
FROM CTE1
GROUP BY ProductID
ORDER BY SUM(Quantity) DESC;

```

	ProductID	January	February	March	April	May	June	July	August	September	October	November	December	Yearly
▶	103	1	0	2	0	3	0	0	0	0	0	0	0	6
	104	0	1	0	1	0	0	0	0	0	0	0	0	2
	101	1	0	0	0	0	0	0	0	0	0	0	0	1
	102	0	1	0	0	0	0	0	0	0	0	0	0	1

```

/* Category Performance :- Which product categories perform the best ? */
WITH CTE1 AS(
SELECT
od.ProductID,
od.Quantity,
MONTHNAME(o.OrderDate) AS MonthName,
o.OrderDate
FROM OrderDetails od LEFT JOIN Orders o
ON od.OrderID=o.OrderID
)
SELECT c.ProductID, p.ProductName, p.Category,
SUM(c.Quantity) AS YearlySale
FROM CTE1 c LEFT JOIN Products p
ON c.ProductID=p.ProductID
GROUP BY c.ProductID
ORDER BY SUM(c.Quantity) DESC
LIMIT 1;

```

	ProductID	ProductName	Category	YearlySale
▶	103	T-shirt	Apparel	6

```

/* Geographical Insights :- Which regions/cities have the highest sales */
SELECT o.CustomerID,
COUNT(od.Quantity) AS ProductSale,
c.Country,
c.City
FROM OrderDetails od LEFT JOIN Orders o
ON od.OrderID=o.OrderID
INNER JOIN Customers c
ON o.CustomerID=c.CustomerID

```

```
GROUP BY o.CustomerID;
```

	CustomerID	ProductSale	Country	City
▶	1	2	USA	New York
	2	2	Canada	Toronto
	3	1	UK	London
	4	1	India	Bangalore
	5	1	Australia	Sydney

```
/* Revenue by Month */
```

```
SELECT  
MONTH(OrderDate) AS Month,  
SUM(TotalAmount) AS Revenue  
FROM Orders  
GROUP BY MONTH(OrderDate)  
ORDER BY Month;
```

	Month	Revenue
▶	1	719.98
	2	1299.98
	3	109.98
	4	69.98
	5	179.97

```
/* Top 5 Products by Revenue */
```

```
SELECT  
p.ProductName,  
SUM(od.Quantity*od.UnitPrice) AS TotalRevenue  
FROM OrderDetails od LEFT JOIN Products p  
ON od.ProductID=p.ProductID  
GROUP BY p.ProductName  
ORDER BY TotalRevenue DESC  
LIMIT 5;
```

	ProductName	TotalRevenue
▶	Laptop	1199.99
	Smartphone	699.99
	Sneakers	179.98
	T-shirt	119.94

```
/* Average Customer Spend */
```

```
SELECT  
c.Name,  
AVG(o.TotalAmount) as AVGSpend  
FROM Customers c JOIN Orders o  
ON c.CustomerID=o.CustomerID  
GROUP BY c.Name  
ORDER BY AVGSpend DESC;
```

	Name	AVGSpend
▶	Bob Brown	1299.980000
	Alice Green	719.980000
	Eve Black	179.970000
	Charlie Blue	109.980000
	Daisy White	69.980000

/* Product Rating Analysis:- Correlate product sales with average ratings */

$$\rho_{XY} = \frac{\text{Covariance}(X,Y)}{\sigma_X \sigma_Y}$$

```
SELECT * FROM Reviews;
```

```
WITH CTE1 AS (
SELECT
o.ProductID,
SUM(o.Quantity) as TotalSale,
ROUND(AVG(r.Rating)) as AVGRate
FROM OrderDetails o JOIN Reviews r
ON o.ProductID=r.ProductID
GROUP BY o.ProductID
),
CTE2 AS (
SELECT
ProductID,
TotalSale - (SELECT ROUND(AVG(TotalSale)) FROM CTE1) AS X,
AVGRate - (SELECT ROUND(AVG(AVGRate)) FROM CTE1) AS Y
FROM
CTE1
),
CTE3 AS (
SELECT
SUM(X*Y)/COUNT(*) AS Covariance,
POWER(SUM(X*X)/COUNT(*),0.5) AS SDX,
POWER(SUM(Y*Y)/COUNT(*),0.5) AS SDY
FROM CTE2
)
SELECT
ROUND(Covariance/(SDX*SDY),2) AS CorRelation
FROM CTE3;
```

	CorRelation
▶	-0.32

/* Customer Retention:- Analyze repeat customer trends DATADIFF or similar date features */

```
SELECT
o.CustomerID,
COUNT(od.OrderID) AS TotalOrders,
MIN(o.OrderDate) AS FirstPurchaseDate,
MAX(o.OrderDate) AS LastPurchaseDate,
DATEDIFF(MAX(o.OrderDate), MIN(o.OrderDate)) AS RetentionPeriod,
CASE WHEN COUNT(od.OrderID) = 1 THEN 'One-Time Buyer'
ELSE 'Long-Term Retention'
END AS CustomerCategory
FROM Orders o JOIN OrderDetails od
ON o.OrderID=od.OrderID
GROUP BY o.CustomerID;
```

	CustomerID	TotalOrders	FirstPurchaseDate	LastPurchaseDate	RetentionPeriod	CustomerCategory
▶	1	2	2023-01-15	2023-01-15	0	Long-Term Retention
	2	2	2023-02-20	2023-02-20	0	Long-Term Retention
	3	1	2023-03-25	2023-03-25	0	One-Time Buyer
	4	1	2023-04-15	2023-04-15	0	One-Time Buyer
	5	1	2023-05-10	2023-05-10	0	One-Time Buyer