Name – Gopi Patel

https://www.linkedin.com/in/gopi-patel-359229307

https://github.com/Gopi2515/sol-cafe-case-study-

Index

Table of Contents

https://github.com/Gopi2515/sol-cafe-case-study-

Introduction

In today's competitive marketplace, data plays a critical role in helping businesses make smart, strategic decisions. ***Sol Café****, a boutique coffee company known for its ethically sourced beans and handcrafted beverages, has experienced steady growth in recent years. With that growth comes the need for improved systems to manage information more effectively. This case study explores how Sol Café embarked on a* ***database design project*** *to enhance its business operations and support future expansion.*

Sol Café is more than just a coffee shop—it's a brand that values ***quality, community, and sustainability****. The company works closely with local and global farmers to source the best beans, while also aiming to provide customers with a warm and inviting café experience. However, as customer demand increased and operations became more complex, the management realized that relying on manual processes and fragmented records was no longer sustainable.*

To tackle these challenges, Sol Café launched a project to ***design and implement a comprehensive database system****. The purpose of this system is to centralize and organize key business data such as:*

- *Customer purchase behavior*
- *Employee performance*
- *Sales transactions*
- *Supplier inventory*
- *Product details*
- *Shipment records*
- *Store location data*

With this structured system in place, Sol Café can better ***analyze trends****,* ***evaluate performance****, and make* ***data-driven decisions****. For instance, by tracking which products sell the most and at which locations, the company can adjust its inventory and marketing strategies accordingly. Similarly, insights into employee sales performance can help improve training and recognition programs.*

This case study will walk through each phase of the database design process—from identifying business objectives to mapping out relationships between data tables—highlighting how Sol Café is transforming its operations using ***technology and smart data management****.*

Mission and Objectives

As Sol Café continues to grow, the leadership team identified the need for better information systems to guide their decisions and support long-term success. While the café had been managing operations using spreadsheets and manual tracking methods, this approach became inefficient and error-prone. The solution: a well-structured, relational database system.

Mission

*The **mission** behind the database project is to:*

"Develop a scalable and reliable data management system that supports operational efficiency and enables strategic, data-driven decisions at Sol Café."

*This mission reflects the company's desire to not only improve internal processes but also to become more responsive to customer needs, market changes, and performance insights. The database will act as the **backbone of information**, ensuring that the right data is available to the right people at the right time.*

Detailed Objectives

To fulfill this mission, the project was guided by four key objectives:

---

1. Understand Customer Buying Behavior

Knowing what your customers want is essential for any business. The database helps Sol Café track and analyze:

- *What products customers are buying*
- *How often they return*
- *Preferred store locations*
- *Seasonal buying trends*

*This information allows the marketing team to create targeted promotions, the inventory team to plan better stock levels, and the operations team to improve the overall customer experience. Over time, this leads to **higher customer retention and better satisfaction**.*

---

2. Evaluate Employee Performance

Employees are a vital part of the customer experience. Sol Café uses the database to:

- *Monitor sales generated by each employee*
- *Identify top performers*
- *Recognize training needs*
- *Ensure fair and consistent performance evaluations*

This helps foster a performance-based culture where effort and success are clearly measured and rewarded. It also ensures that staffing decisions are based on reliable data, not just assumptions.

3. Analyze Sales Trends

Sales are a key indicator of business health. The database allows Sol Café to:

- *Track sales by product, location, and date*
- *Compare current performance with past trends*
- *Forecast future demand*

With this knowledge, the company can plan product launches, adjust store offerings, and prepare for busy seasons. This kind of insight ensures ***more efficient resource use and better business decisions****.*

4. Improve Product Quality

Product performance is tracked through:

- *Sales velocity*
- *Return rates*
- *Customer complaints*
- *Supplier history*

By tying this information together, Sol Café can work with its suppliers to address quality issues, remove underperforming products, and improve the overall product mix. The goal is to ensure that every item on the menu meets the company's high standards.

Entity Identification and Data Dictionary

Key Entities Identified

1. ***Customers***
   Stores information about individuals who purchase from Sol Café.
   - *Example fields: Customer ID, Name, Email, Contact Number, Location*
2. ***Employees***
   Captures data about staff members responsible for handling sales and operations.
   - *Example fields: Employee ID, Name, Position, Store Location*
3. ***Products***
   Refers to the various coffee items, baked goods, and merchandise sold.
   - *Example fields: Product ID, Name, Category, Price, Supplier ID*
4. ***Sales***
   Records every transaction, linking products to customers and employees.
   - *Example fields: Sale ID, Date, Product ID, Customer ID, Employee ID, Quantity, Total Price*
5. ***Suppliers***
   Includes vendors who provide raw materials or finished products to Sol Café.
   - *Example fields: Supplier ID, Name, Contact Info, Product Supplied*
6. ***Shipments***
   Tracks incoming stock deliveries and inventory movement.
   - *Example fields: Shipment ID, Date, Supplier ID, Product ID, Quantity*
7. ***Store Locations***
   Represents the physical branches of Sol Café where sales and operations occur.
   - *Example fields: Store ID, Address, City, Manager*

Data Dictionary

Field Name	***Data Type***	***Description***	***Key Type***
CustomerID	*Integer*	*Unique ID assigned to each customer*	*PK*
FirstName	*Varchar (50)*	*Customer's first name*	
LastName	*Varchar (50)*	*Customer's last name*	
Email	*Varchar (100)*	*Customer's email*	
Phone	*Varchar (20)*	*Contact number*	

Customers Table

Field Name	***Data Type***	***Description***	***Key Type***
SuppliersID	*Integer*	*Unique ID for each supplier*	*PK*
Name	*Varchar (50)*	*Supplier's company or contact name*	
Phone	*Varchar (20)*	*Contact number*	
Country	*Varcha (20)*	*Country where the supplier is located*	

Suppliers Table

Field Name	***Data Type***	***Description***	***Key Type***
ProductID	*Integer*	*Unique ID for each product*	*PK*
Name	*Varchar (20)*	*Product name (e.g. Colombian Roast)*	
Type	*Varchar (50)*	*Describes the type of bean or drink*	
SupplierID	*Integer*	*References de supplier providing the product*	*FK*
Price	*Decimal (10, 2)*	*Price per unit*	

Products Table

Field Name	***Data Type***	***Description***	***Key Type***
StoreLocationID	*Integer*	*Unique ID for each store*	*PK*
City	*Varchar (25)*	*City where is located*	

Country	Varchar (25)	Country of the store	
Address	Varcha (100)	Physical address	
Phone	Varchar (20)	Store contact number	

Stores Location Table

Field Name	**Data Type**	**Description**	**Key Type**
EmployeeID	Integer	Unique ID for each employee	PK
FirstName	Varchar (50)	Employee’s first name	
LastName	Varchar (50)	Employee’s last name	
Role	Varchar (20)	Job title (e.g. barista, manager)	
Phone	Varchar (20)	Contact number	
Salary	Decimal (10, 2)	Base Salary per month	

Employees Table

Field Name	**Data Type**	**Description**	**Key Type**
ShipmentID	Integer	Unique ID for each shipment. Also tracking ID	PK
SaleID	Integer	References the sale this shipment processing	FK

Date	*Date*	*Date of dispatched*	
Status	*Varchar (20)*	*Shipped / delivered*	

Shipments Table

SalesTable

Field Name	***Data Type***	***Description***	***Key Type***
SaleID	*Integer*	*Unique ID for each sale transaction*	*PK*
CustomerID	*Integer*	*References the customer who made the purchase*	*FK*
ProductID	*Integer*	*References the producto being sold*	*FK*
EmployeeID	*Integer*	*References the employee who precess the sale*	*FK*
LocationID	*Integer*	*References the store where the sale was made*	*FK*
SaleDate	*Date*	*Date the sale occurred*	

Price	Decimal (10, 2)	Sale amount	

Final Table List

1. **Customers** – to store customer details
2. **Employees** – to track staff and performance
3. **Products** – to list all items sold
4. **Sales** – to record every transaction
5. **Suppliers** – to manage product sources
6. **Shipments** – to monitor incoming deliveries
7. **Store Locations** – to distinguish between different café branches

Preliminary Table List

1. Customers
2. Suppliers
3. Products
4. Store Locations
5. Shipments
6. Employees
7. Sales

1. Products → Suppliers
Relationship: Each product comes from one supplier, but a supplier can supply many products.

Type: One-to-Many

How: Products.SupplierID → Suppliers.SupplierID

Meaning:
If "Colombian Coffee" is supplied by "ABC Beans Inc.", that supplier may also supply other products like "Brazilian Coffee" or "Organic Espresso".

2. Sales → Products
Relationship: Each sale involves one product, but a product can be sold many times.

Type: One-to-Many

How: Sales.ProductID → Products.ProductID

Meaning:
If Product ID 101 is "Espresso Beans", this product can appear in many rows in the Sales table, each representing a new sale.

3. Sales → Customers
Relationship: Each sale is made to one customer, but a customer can make many purchases.

Type: One-to-Many

How: Sales.CustomerID → Customer.CustomerID

Meaning:
Customer "John Smith" can have multiple sales entries. Each sale entry connects to the same CustomerID.

4. Sales → Employees
Relationship: Each sale is handled by one employee, but an employee can handle many sales.

Type: One-to-Many

How: Sales.EmployeeID → Employees.EmployeeID

Meaning:
Employee "Sarah Jones" may have assisted in 50 sales. Each of those will be linked to her EmployeeID.

5. Sales → Store_Location
Relationship: Each sale happens at one store, but a store can host many sales.

Type: One-to-Many

How: Sales.LocationID → Store_Location.LocationID

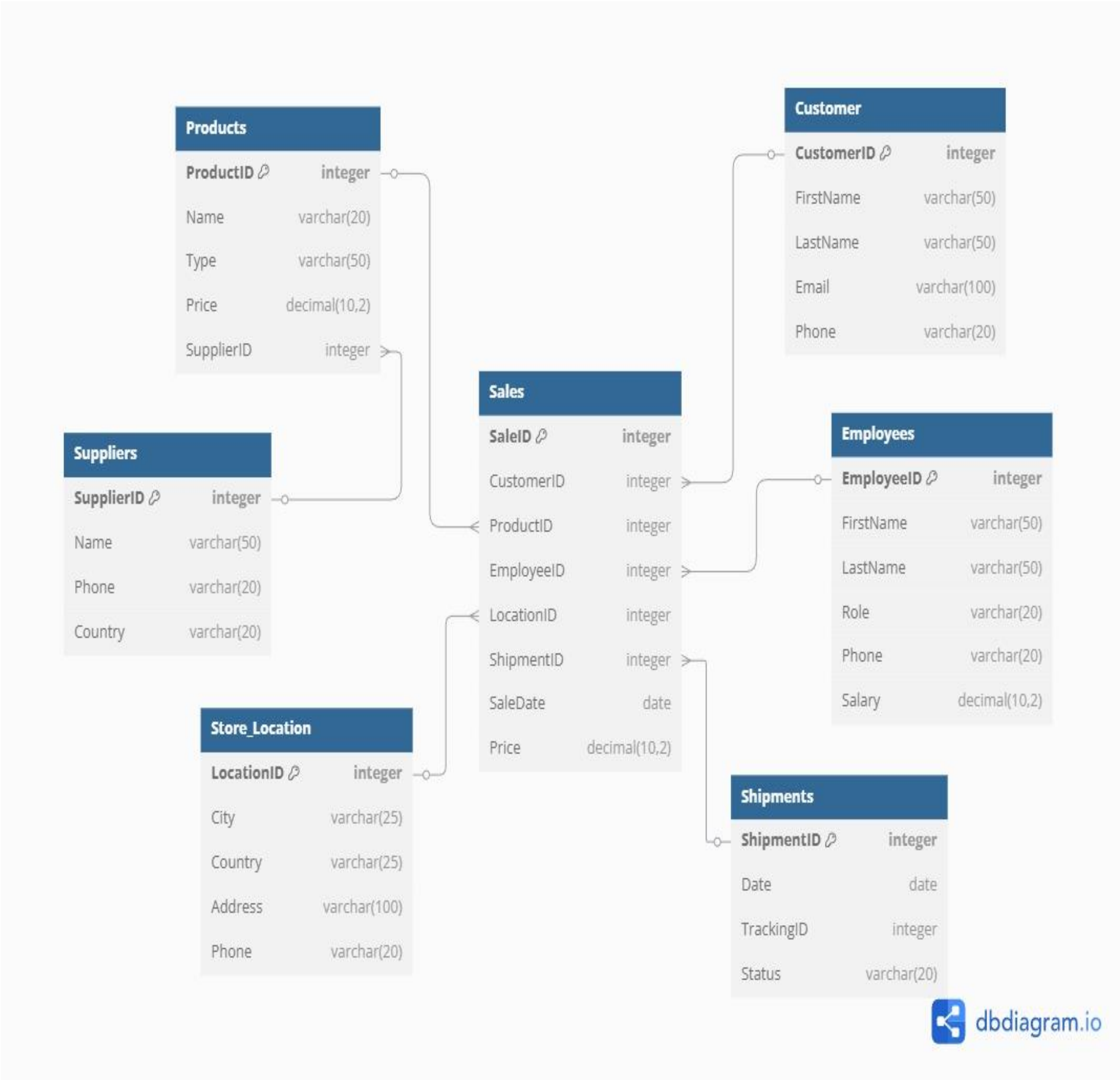
Meaning:
The Downtown Calgary branch (Location ID 3) may have handled hundreds of sales.

6. Sales → Shipments
Relationship: Each sale may be linked to one shipment, but a shipment may handle multiple sales (depending on implementation).

Type: Possibly One-to-Many or One-to-One depending on business rules.

How: Sales.ShipmentID → Shipments.ShipmentID

Meaning:
If you order 2 items in 1 sale, they may be shipped together under 1 tracking ID, or separately—depending on how this business handles shipments.



Products	
ProductID	integer
Name	varchar(20)
Type	varchar(50)
Price	decimal(10,2)
SupplierID	integer

Customer	
CustomerID	integer
FirstName	varchar(50)
LastName	varchar(50)
Email	varchar(100)
Phone	varchar(20)

Sales	
SaleID	integer
CustomerID	integer
ProductID	integer
EmployeeID	integer
LocationID	integer
ShipmentID	integer
SaleDate	date
Price	decimal(10,2)

Suppliers	
SupplierID	integer
Name	varchar(50)
Phone	varchar(20)
Country	varchar(20)

Employees	
EmployeeID	integer
FirstName	varchar(50)
LastName	varchar(50)
Role	varchar(20)
Phone	varchar(20)
Salary	decimal(10,2)

Store_Location	
LocationID	integer
City	varchar(25)
Country	varchar(25)
Address	varchar(100)
Phone	varchar(20)

Shipments	
ShipmentID	integer
Date	date
TrackingID	integer
Status	varchar(20)

Entity Relationship Diagram

Key Relationships in Sol Café Database

1. ***Customer → Sales***
   - ***One customer*** *can make* ***many purchases****.*
   - *Linked by* `Customer_ID`*.*
2. ***Employee → Sales***
   - ***One employee*** *can handle* ***many sales****.*
   - *Linked by* `Employee_ID`*.*
3. ***Product → Sales***
   - ***One product*** *can appear in* ***many sales transactions****.*
   - *Linked by* `Product_ID`*.*
4. ***Store Location → Sales***
   - ***One store*** *can process* ***many sales****.*
   - *Linked by* `Store_ID`*.*
5. ***Supplier → Products***
   - ***One supplier*** *can supply* ***many products****.*
   - *Linked by* `Supplier_ID`*.*
6. ***Supplier → Shipments***
   - ***One supplier*** *can send* ***multiple shipments****.*
   - *Linked by* `Supplier_ID`*.*
7. ***Product → Shipments***
   - ***One product*** *can be part of* ***many shipments****.*
   - *Linked by* `Product_ID`*.*
8. ***Store Location → Employees***
   - ***One store*** *employs* ***multiple staff members****.*
   - *Linked by* `Store_ID`*.*

Example Scenario

Imagine a customer walks into the downtown Sol Café and buys a cappuccino. That single transaction will:

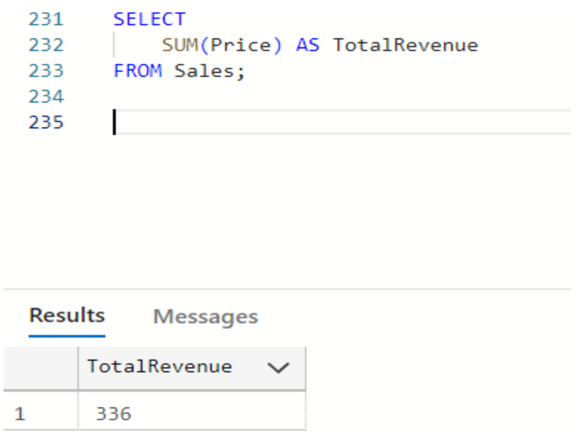
- *Link the* ***customer*** *to the* ***sale***
- *Record which* ***employee*** *made the sale*
- *Note which* ***product*** *(cappuccino) was sold*
- *Register the* ***store location***
- *Pull the price from the* ***product*** *table*
- *Reflect stock movement in the* ***shipment*** *record*

Database Creation

Customers, Employees, Products, Sales, Suppliers, Shipments, *and* ***Store Locations****.*

Each table was created with:

- *A* ***primary key*** *to uniquely identify each record*
- ***Foreign keys*** *to establish connections between related tables*
- *Proper* ***data types*** *to ensure accuracy and consistency*



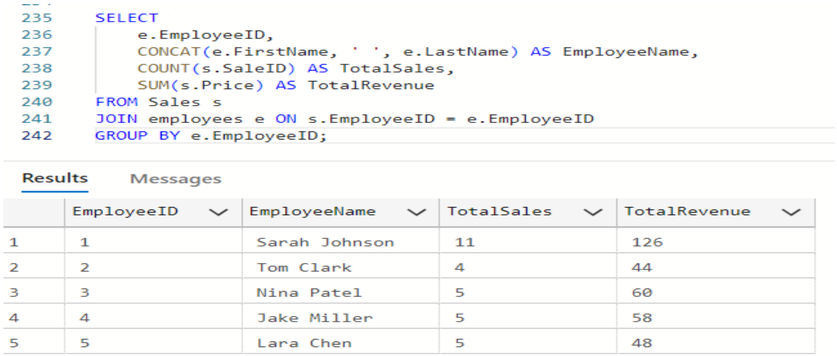
```
SELECT
    SUM(Price) AS TotalRevenue
FROM Sales;

```

	TotalRevenue
1	336

Views and Reports

Here are the key views developed:

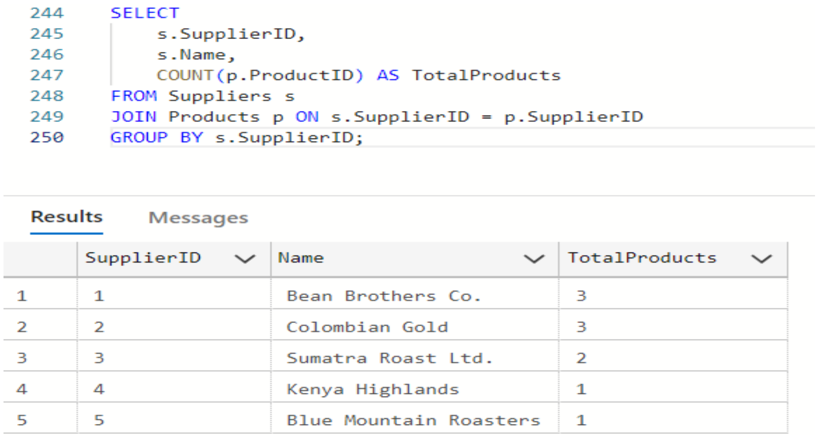
1. ***View #1: Total Revenue***
   - *Shows total income generated over a selected period.*
2. ***View #2: Sales by Each Employee***
   - *Helps track employee performance and reward high achievers.*
3. ***View #3: Total Products Supplied by Each Supplier***
   - *Allows the company to monitor supplier reliability and contribution.*
4. ***View #4: Store-wise Sales Analysis***
   - *Compares performance between different store locations*



```sql
SELECT
    e.EmployeeID,
    CONCAT(e.FirstName, ' ', e.LastName) AS EmployeeName,
    COUNT(s.SaleID) AS TotalSales,
    SUM(s.Price) AS TotalRevenue
FROM Sales s
JOIN employees e ON s.EmployeeID = e.EmployeeID
GROUP BY e.EmployeeID;
```

	EmployeeID	EmployeeName	TotalSales	TotalRevenue
1	1	Sarah Johnson	11	126
2	2	Tom Clark	4	44
3	3	Nina Patel	5	60
4	4	Jake Miller	5	58
5	5	Lara Chen	5	48

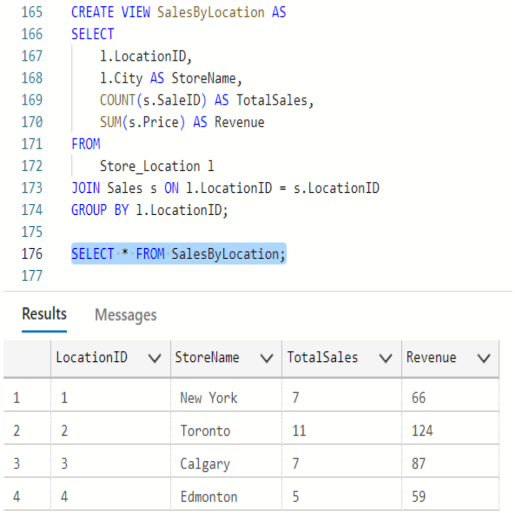
Sale By Each Employees



```sql
SELECT
    s.SupplierID,
    s.Name,
    COUNT(p.ProductID) AS TotalProducts
FROM Suppliers s
JOIN Products p ON s.SupplierID = p.SupplierID
GROUP BY s.SupplierID;
```

	SupplierID	Name	TotalProducts
1	1	Bean Brothers Co.	3
2	2	Colombian Gold	3
3	3	Sumatra Roast Ltd.	2
4	4	Kenya Highlands	1
5	5	Blue Mountain Roasters	1

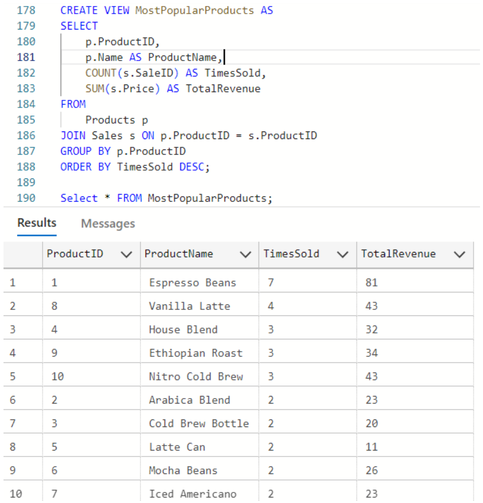
Total Products Supplied by Each Suppliers



```sql
CREATE VIEW SalesByLocation AS
SELECT
    l.LocationID,
    l.City AS StoreName,
    COUNT(s.SaleID) AS TotalSales,
    SUM(s.Price) AS Revenue
FROM
    Store_Location l
JOIN Sales s ON l.LocationID = s.LocationID
GROUP BY l.LocationID;

SELECT * FROM SalesByLocation;
```

	LocationID	StoreName	TotalSales	Revenue
1	1	New York	7	66
2	2	Toronto	11	124
3	3	Calgary	7	87
4	4	Edmonton	5	59

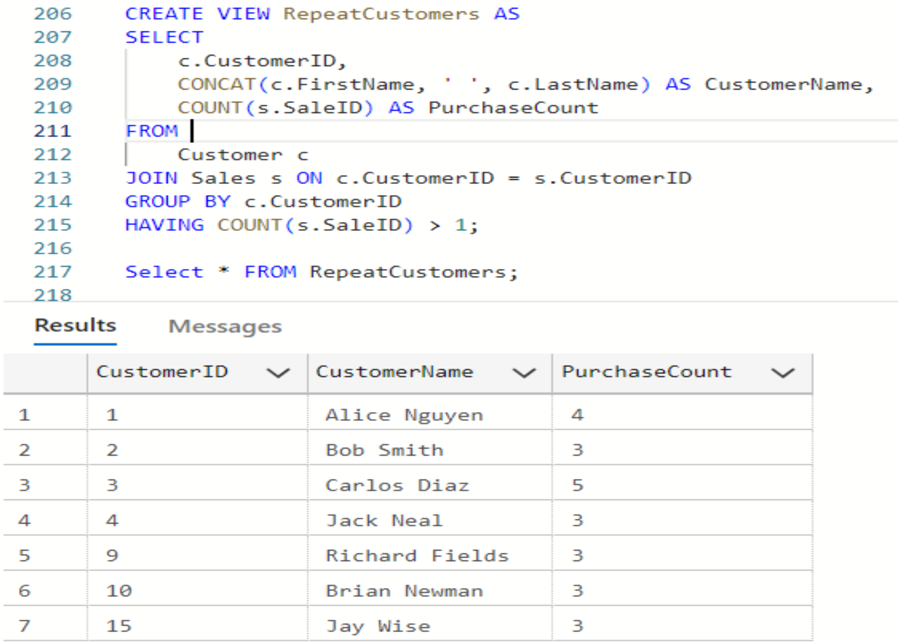
View 1



```sql
CREATE VIEW MostPopularProducts AS
SELECT
    p.ProductID,
    p.Name AS ProductName,
    COUNT(s.SaleID) AS TimesSold,
    SUM(s.Price) AS TotalRevenue
FROM
    Products p
JOIN Sales s ON p.ProductID = s.ProductID
GROUP BY p.ProductID
ORDER BY TimesSold DESC;

Select * FROM MostPopularProducts;
```

	ProductID	ProductName	TimesSold	TotalRevenue
1	1	Espresso Beans	7	81
2	8	Vanilla Latte	4	43
3	4	House Blend	3	32
4	9	Ethiopian Roast	3	34
5	10	Nitro Cold Brew	3	43
6	2	Arabica Blend	2	23
7	3	Cold Brew Bottle	2	20
8	5	Latte Can	2	11
9	6	Mocha Beans	2	26
10	7	Iced Americano	2	23

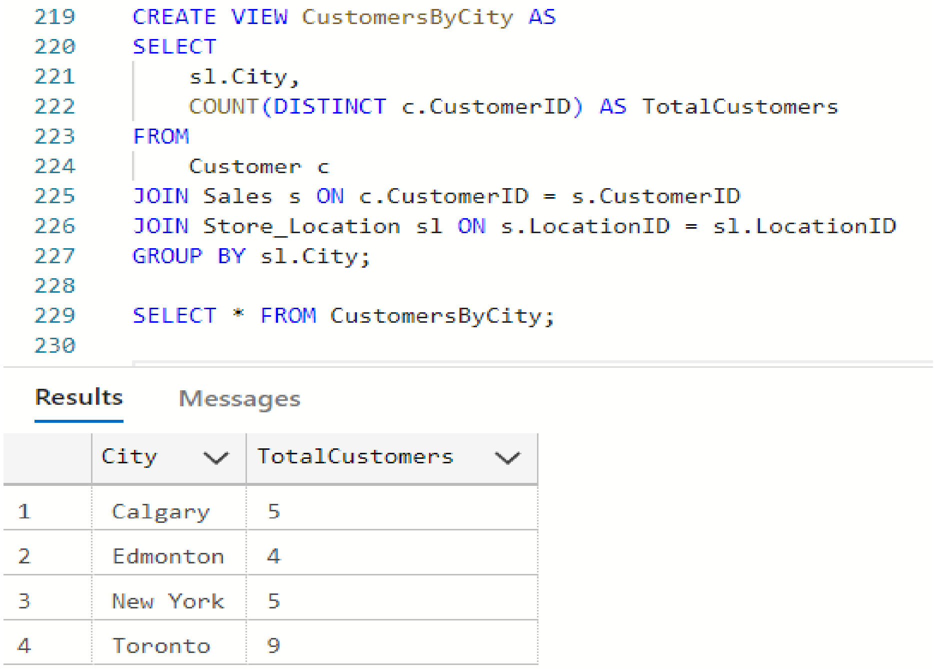
view 2



```sql
CREATE VIEW RepeatCustomers AS
SELECT
    c.CustomerID,
    CONCAT(c.FirstName, ' ', c.LastName) AS CustomerName,
    COUNT(s.SaleID) AS PurchaseCount
FROM
    Customer c
JOIN Sales s ON c.CustomerID = s.CustomerID
GROUP BY c.CustomerID
HAVING COUNT(s.SaleID) > 1;

Select * FROM RepeatCustomers;
```

	CustomerID	CustomerName	PurchaseCount
1	1	Alice Nguyen	4
2	2	Bob Smith	3
3	3	Carlos Diaz	5
4	4	Jack Neal	3
5	9	Richard Fields	3
6	10	Brian Newman	3
7	15	Jay Wise	3

View 3



```sql
CREATE VIEW CustomersByCity AS
SELECT
    sl.City,
    COUNT(DISTINCT c.CustomerID) AS TotalCustomers
FROM
    Customer c
JOIN Sales s ON c.CustomerID = s.CustomerID
JOIN Store_Location sl ON s.LocationID = sl.LocationID
GROUP BY sl.City;

SELECT * FROM CustomersByCity;
```

	City	TotalCustomers
1	Calgary	5
2	Edmonton	4
3	New York	5
4	Toronto	9

View 4

Conclusion

*The **Sol Café database project** was a critical step in transforming how the business operates. By building a well-structured relational database, the company can now manage, track, and analyze its operations with greater accuracy and efficiency.*

The database:

- *Supports better **sales tracking***
- *Enables deeper **customer insights***
- *Helps **evaluate employee performance***
- *Ensures **supplier coordination***
- *Simplifies **report generation** for management*

https://www.linkedin.com/in/gopi-patel-359229307

https://github.com/Gopi2515/sol-cafe-case-study-

Thank You