DATABASE DESIGN FOR COSTCO CORPORATION: A CASE STUDY

Course Project

OCTOBER 4, 2024

AKUTA TOCHUKWU MAURISON
mtakuta@yahoo.com | linkedin.com/in/akuta1

Table of Contents

Introduction

In today's competitive retail landscape, efficient data management is crucial to maintaining operational excellence and customer satisfaction. This case study explores the design of a robust database system for Costco Corporation, one of the world's leading wholesale retailers. With a focus on scalability, efficiency, and supporting Costco's core mission, this article delves into the structure, relationships, and key considerations in building a database that meets both current and future demands.

Mission

Costco is committed to continuously providing members with high-quality goods and services at the lowest possible prices. The organization strives to enhance customer satisfaction through efficient operations, underpinned by a membership-driven model that ensures long-term customer retention. To support this mission, Costco requires a reliable, scalable, and efficient database system to manage various aspects of its business, from inventory and supply chain management to customer transactions and member services.

Objectives

The primary objectives of designing a new database system for Costco are as follows:

- **Efficient Supply Chain Management and Procurement Processes:**
  The database should streamline inventory control and procurement, ensuring products are available at the right time and location to meet customer demand.

- **Seamless Customer Experience:**
  Ensuring accurate product availability, pricing, and inventory tracking across multiple locations is essential. A well-designed database helps maintain smooth customer interactions, both in-store and online.

- **Cost-Effective Operations:**
  The system should support Costco's low-pricing model by maintaining efficient operations, minimizing overhead costs, and improving data accuracy.

- Membership Satisfaction and Retention:
  The database must be capable of providing personalized services to Costco members, tracking preferences, purchase history, and ensuring a seamless shopping experience.

- **Scalable and Robust Architecture:**
  With Costco's rapid growth, the database must be capable of scaling to accommodate increased data loads, new warehouses, and expanded product offerings.

Database Design

Database design is critical for optimizing Costco's operational efficiency by organizing data related to members, products, orders, and warehouses in a scalable, structured way. It enables seamless inventory management, enhances customer satisfaction through accurate transaction tracking, and supports low-cost operations by improving data accuracy and accessibility. A well-designed database facilitates future growth, ensuring the system can scale to meet increased demand and complexity.

A. Database Structure

To meet Costco's operational needs, the database is designed with several key entities and tables, each fulfilling specific functions within the broader system. These include:

- **Members**: Stores personal and contact information for Costco members.
- **Products:** Captures details about items available for sale, such as product names, prices, and stock levels.
- **Suppliers:** Tracks companies that supply products to Costco.
- **Orders:** Logs individual customer purchases, including transaction dates and product details.
- **Inventory:** Monitors stock levels across Costco's warehouses and tracks restocking schedules.
- **Employees:** Contains records of Costco staff and their roles.
- **Transactions:** Logs payment details for each order, including payment method and total amount paid.
- **Warehouses:** Manages inventory distribution and stock levels across multiple locations.
- **Categories:** Groups products into categories such as groceries, electronics, and apparel for easier tracking.

B. Tables and Fields

1. Members			
Field Name	**Data Type**	**Size**	**Constraint**
MemberID	Integer	10	Primary Key
FirstName	Varchar	20	Not Null
LastName	Varchar	20	Not Null
Email	Varchar	30	Not Null
PhoneNumber	Number(10)	10	Not Null
Address	Varchar	50	Not Null
MembershipType (Executive, Gold Star & Business)	Varchar	20	Not Null
2. Products			
ProductID	Integer	10	Primary Key

ProductName	Varchar	20	Not Null
Description	Varchar	100	Not Null
Price	Float	10	Not Null
QuantityInStock	Number(10)	10	Not Null
CategoryID	Integer	10	Foreign Key
SupplierID	Integer	10	Foreign Key
3. Suppliers			
SupplierID	Integer	10	Primary Key
SupplierName	Varchar	20	Not Null
ContactPerson	Varchar	20	Not Null
Email	Varchar	30	Not Null
PhoneNumber	Number(10)	10	Not Null
Address	Varchar	50	Not Null
4. Orders			
OrderID	Integer	10	Primary
MemberID	Integer	10	Foreign Key
OrderDate	Date	10	Not Null
TotalAmount	Float	10	Not Null
Status	Varchar	100	Not Null
5. Inventory			
InventoryID	Integer	10	Primary Key
ProductID	Integer	10	Foreign Key
WarehouseID	Integer	10	Foreign Key
QuantityAvailable	Float	10	Not Null
LastRestockedDate	Date	10	Not Null
6. Employees			
EmployeesID	Integer	10	Primary Key
FirstName	Varchar	20	Not Null
LastName	Varchar	20	Not Null
Email	Varchar	30	Not Null
PhoneNumber	Number(10)	10	Not Null
Department	Varchar	20	Not Null
Position	Varchar	20	Not Null
HireDate	Date	10	Not Null
MangerID	Integer	10	Null
7. Transactions			
TransactionsID	Integer	10	Primary Key
OrderID	Integer	20	Foreign
TransactionDate	Date	10	Not Null
PaymentMethod (Cash, Master Card, Debit Card, shop card & CIBC Master card)	Varchar	30	Not Null
Amount	Float	10	Not Null
8. Warehouses			
WarehouseID	Varchar	10	Primary Key

ManagerID	Integer	20	Foreign
Location	Varchar	30	Not Null
Capacity	Varchar	15	Not Null
PhoneNumber	Number(10)	15	Not Null
9. Categories			
CategoryID	Integer	10	Primary Key
CategoryName	Varchar	20	Not Null
Description	Varchar	20	Not Null
10. Order Details			
OrderID	Integer	10	Primary Key
ProductID	Integer	20	Primary Key
Quantity	Integer	30	Not Null
PriceAtOrder	Decimal	(10,2)	Not Null

Table Relationships and Entity

A critical part of the database design is the relationships between different entities, which ensure that data is efficiently organized and retrieved. The relationships for Costco's database design is outlined below;

1. **Members Table**
    - **Primary Key**: MemberID
    - **Attributes**: FirstName, LastName, Email, PhoneNumber, Address, MembershipType
    - **Relationships**:
        - **One-to-Many**: A Member can place multiple Orders (linked via MemberID in Orders table).
2. **Products Table**
    - **Primary Key**: ProductID
    - **Attributes**: ProductName, Description, Price, QuantityInStock, CategoryID, SupplierID
    - **Relationships**:
        - **One-to-Many**: A Supplier can supply multiple Products (linked via SupplierID in Products table).
        - **Many-to-One**: Products belong to a Category (linked via CategoryID in Products table).
        - **Many-to-Many**: Products are associated with Orders (through the OrderDetails junction table).

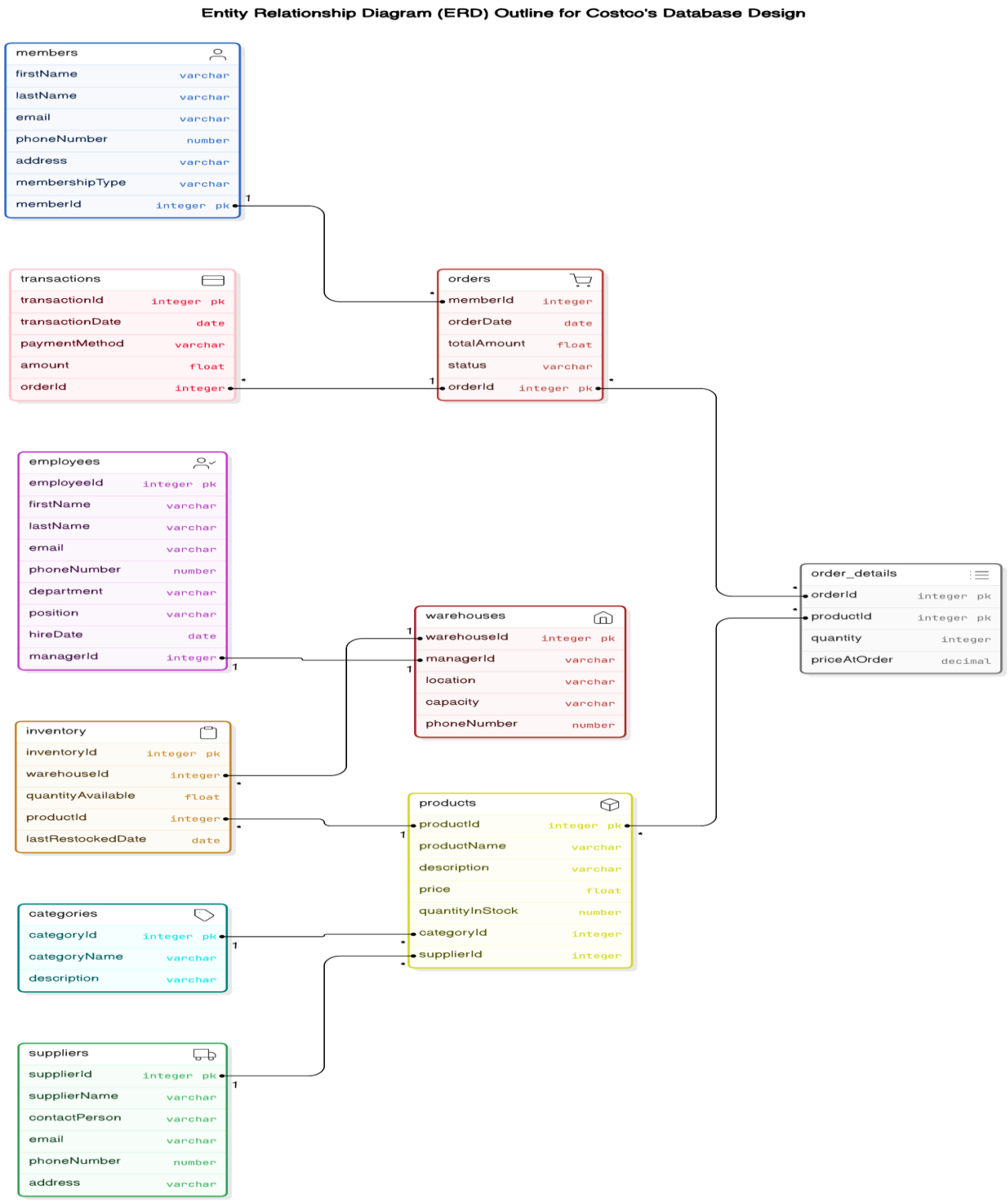
3. **Suppliers Table**
    - **Primary Key**: SupplierID
    - **Attributes**: SupplierName, ContactPerson, Email, PhoneNumber, Address
    - **Relationships**:
        - **One-to-Many**: One Supplier provides many Products (linked via SupplierID in Products table).
4. **Orders Table**
    - **Primary Key**: OrderID
    - **Attributes**: MemberID, OrderDate, TotalAmount, Status
    - **Relationships**:
        - **One-to-Many**: A Member can place multiple Orders (linked via MemberID).
        - **Many-to-Many**: Orders include multiple Products (linked through the OrderDetails table).
        - **One-to-Many**: An Order can have multiple Transactions (linked via OrderID in Transactions table).
5. **Inventory Table**
    - **Primary Key**: InventoryID
    - **Attributes**: ProductID, WarehouseID, QuantityAvailable, LastRestockedDate
    - **Relationships**:
        - **One-to-Many**: A Warehouse can hold multiple products (linked via WarehouseID in Inventory table).
        - **One-to-Many**: A Product can exist in multiple Warehouse locations (linked via ProductID).
6. **Employees Table**
    - **Primary Key**: EmployeesID
    - **Attributes**: FirstName, LastName, Email, PhoneNumber, Department, Position, HireDate, ManagerID
    - **Relationships**:
        - **One-to-One**: An Employee can manage one Warehouse (linked via ManagerID in Warehouses table).
7. **Transactions Table**

- **Primary Key**: TransactionsID
- **Attributes**: OrderID, TransactionDate, PaymentMethod, Amount
- **Relationships**:
  - **One-to-Many**: An Order can have multiple Transactions (linked via OrderID).

8. **Warehouses Table**
   - **Primary Key**: WarehouseID
   - **Attributes**: ManagerID, Location, Capacity, PhoneNumber
   - **Relationships**:
     - **One-to-Many**: A Warehouse holds multiple products in Inventory (linked via WarehouseID).
     - **One-to-One**: Each Warehouse is managed by a single Employee (linked via ManagerID).
9. **Categories Table**
   - **Primary Key**: CategoryID
   - **Attributes**: CategoryName, Description
   - **Relationships**:
     - **One-to-Many**: A Category can group multiple Products (linked via CategoryID).
10. **Order Details Table**
    - **Primary Keys**: OrderID, ProductID
    - **Attributes**: Quantity, PriceAtOrder
    - **Relationships**:
      - **Many-to-Many**: Links Orders and Products (via OrderID and ProductID).

Entity Relationship Diagram (ERD)

The Entity Relationship Diagram (ERD) illustrates how these tables interact with one another, providing a visual representation of the data flow within the system.



Conclusion

This case study presents a scalable, efficient database solution tailored to the specific needs of Costco Corporation. By focusing on key aspects such as supply chain management, customer experience, and cost-effectiveness, the proposed database design supports Costco's mission of delivering quality products at the lowest possible prices. Additionally, the system is designed to scale with Costco's growth, ensuring continued efficiency and customer satisfaction.

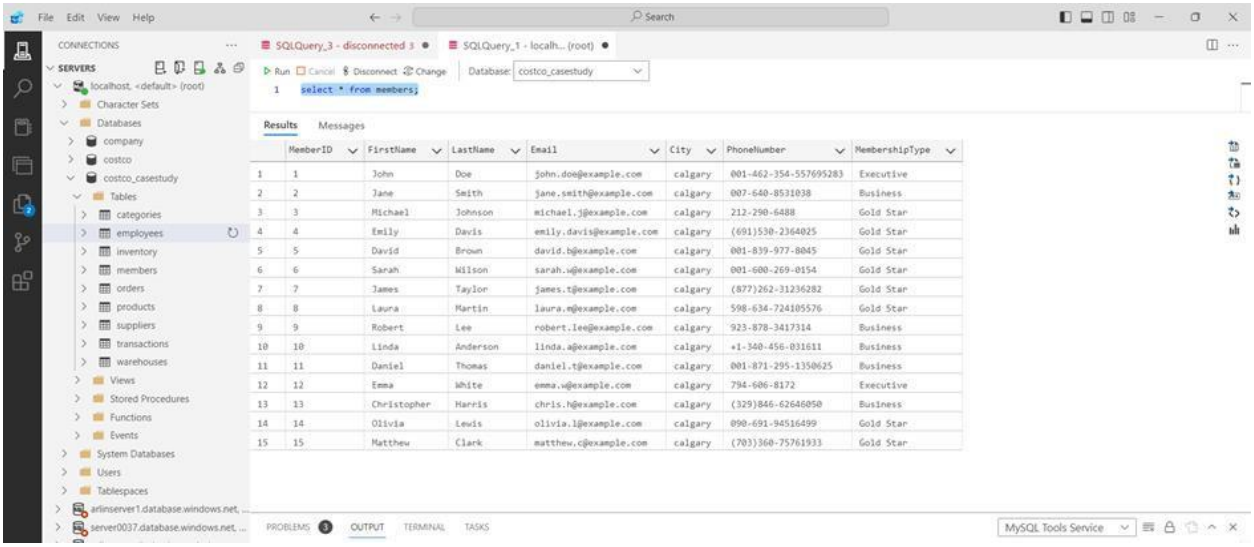
Appendix

SQL Queries and Database Management

To implement this database, SQL queries are used to create and manage tables, relationships, and data entries. For example:

- **Table Creation:** SQL scripts are used to define each table and its fields, including constraints like primary keys and foreign keys.

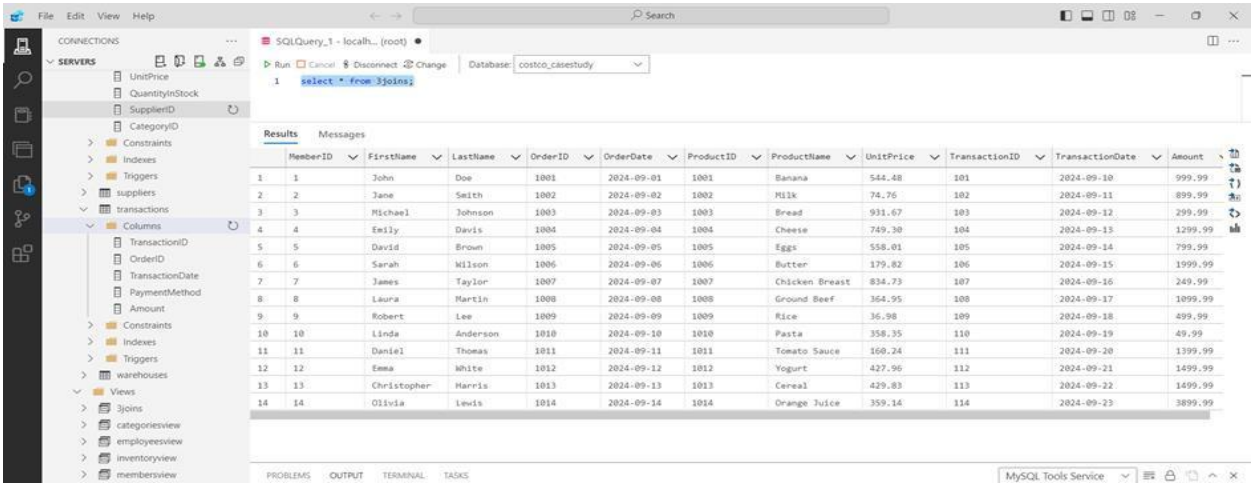
Queries – Creation of Database and Tables



	MemberID	FirstName	LastName	Email	City	PhoneNumber	MembershipType
1	1	John	Doe	john.doe@example.com	calgary	001-462-354-557695283	Executive
2	2	Jane	Smith	jane.smith@example.com	calgary	007-640-8531038	Business
3	3	Michael	Johnson	michael.j@example.com	calgary	212-290-6488	Gold Star
4	4	Emily	Davis	emily.davis@example.com	calgary	(691)530-2364025	Gold Star
5	5	David	Brown	david.b@example.com	calgary	001-839-977-8045	Gold Star
6	6	Sarah	Wilson	sarah.w@example.com	calgary	001-600-269-0154	Gold Star
7	7	James	Taylor	james.t@example.com	calgary	(877)262-31236282	Gold Star
8	8	Laura	Martin	laura.m@example.com	calgary	598-634-724105576	Gold Star
9	9	Robert	Lee	robert.lee@example.com	calgary	923-878-3417314	Business
10	10	Linda	Anderson	linda.a@example.com	calgary	+1-340-456-031611	Business
11	11	Daniel	Thomas	daniel.t@example.com	calgary	001-871-295-1350625	Business
12	12	Emma	White	emma.w@example.com	calgary	794-606-8172	Executive
13	13	Christopher	Harris	chris.h@example.com	calgary	(329)846-62646050	Business
14	14	Olivia	Lewis	olivia.l@example.com	calgary	090-691-94516499	Gold Star
15	15	Matthew	Clark	matthew.c@example.com	calgary	(703)360-75761933	Gold Star

- **Joins and Views:** Complex queries are employed to join tables such as Members, Orders, Products, and Transactions, allowing for detailed insights into customer purchasing behavior.

4 Joins & View Query: joins the Members, Orders, Products, and Transactions table



	MemberID	FirstName	LastName	OrderID	OrderDate	ProductID	ProductName	UnitPrice	TransactionID	TransactionDate	Amount
1	1	John	Doe	1001	2024-09-01	1001	Banana	544.48	101	2024-09-10	999.99
2	2	Jane	Smith	1002	2024-09-02	1002	Milk	74.76	102	2024-09-11	899.99
3	3	Michael	Johnson	1003	2024-09-03	1003	Bread	931.67	103	2024-09-12	299.99
4	4	Emily	Davis	1004	2024-09-04	1004	Cheese	749.30	104	2024-09-13	1299.99
5	5	David	Brown	1005	2024-09-05	1005	Eggs	558.01	105	2024-09-14	799.99
6	6	Sarah	Wilson	1006	2024-09-06	1006	Butter	179.82	106	2024-09-15	1999.99
7	7	James	Taylor	1007	2024-09-07	1007	Chicken Breast	834.73	107	2024-09-16	249.99
8	8	Laura	Martin	1008	2024-09-08	1008	Ground Beef	364.95	108	2024-09-17	1099.99
9	9	Robert	Lee	1009	2024-09-09	1009	Rice	36.98	109	2024-09-18	499.99
10	10	Linda	Anderson	1010	2024-09-10	1010	Pasta	358.35	110	2024-09-19	49.99
11	11	Daniel	Thomas	1011	2024-09-11	1011	Tomato Sauce	160.24	111	2024-09-20	1399.99
12	12	Emma	White	1012	2024-09-12	1012	Yogurt	427.96	112	2024-09-21	1499.99
13	13	Christopher	Harris	1013	2024-09-13	1013	Cereal	429.83	113	2024-09-22	1499.99
14	14	Olivia	Lewis	1014	2024-09-14	1014	Orange Juice	359.14	114	2024-09-23	3899.99

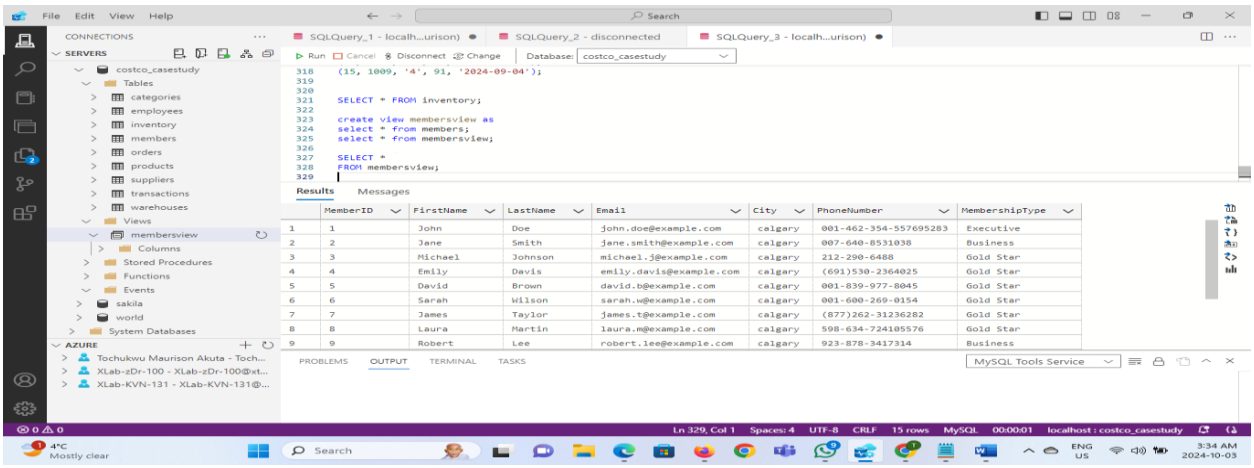
These queries enable Costco's management team to derive actionable insights from the data, such as identifying popular products, managing supplier relationships, and optimizing warehouse operations.

Views

create view membersview as

select * from members;

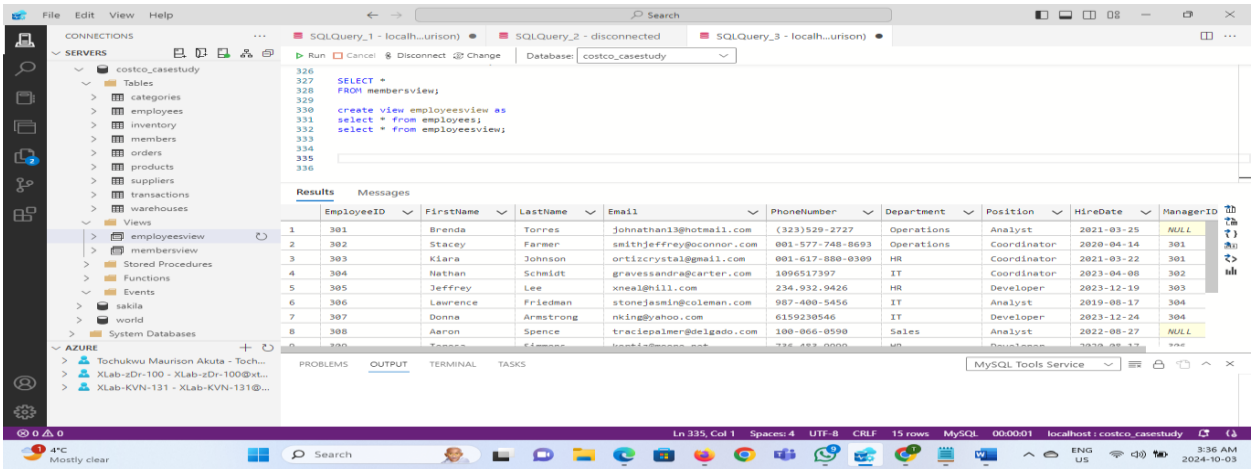
select * from membersview;



create view employeesview as

select * from employees;

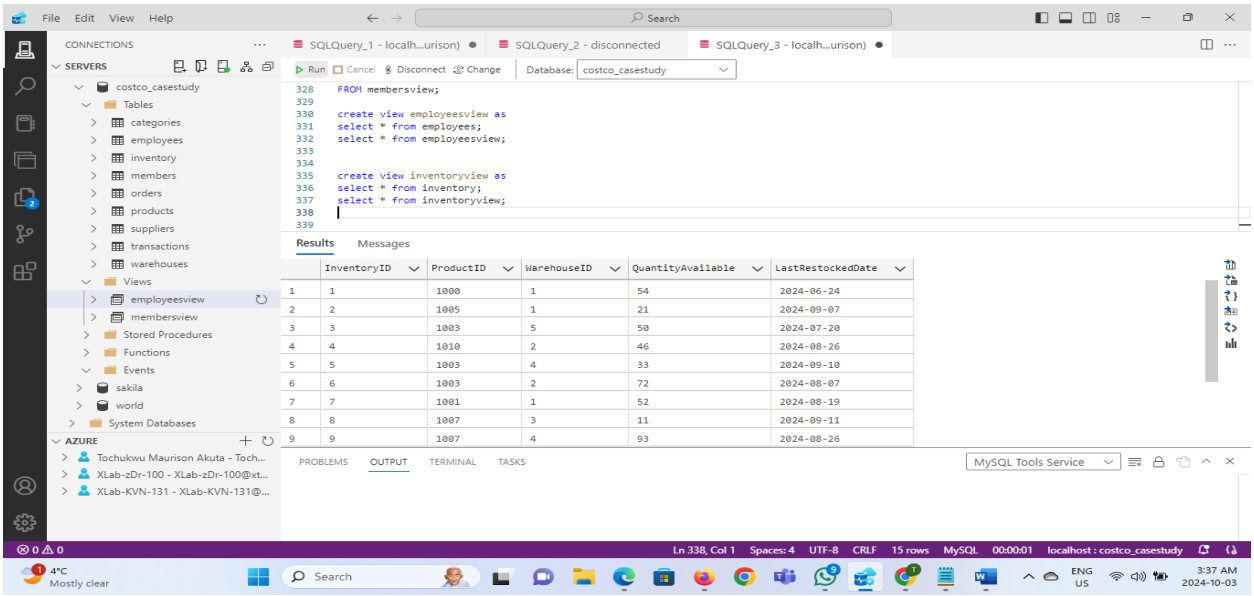
select * from employeesview;



create view inventoryview as

select * from inventory;

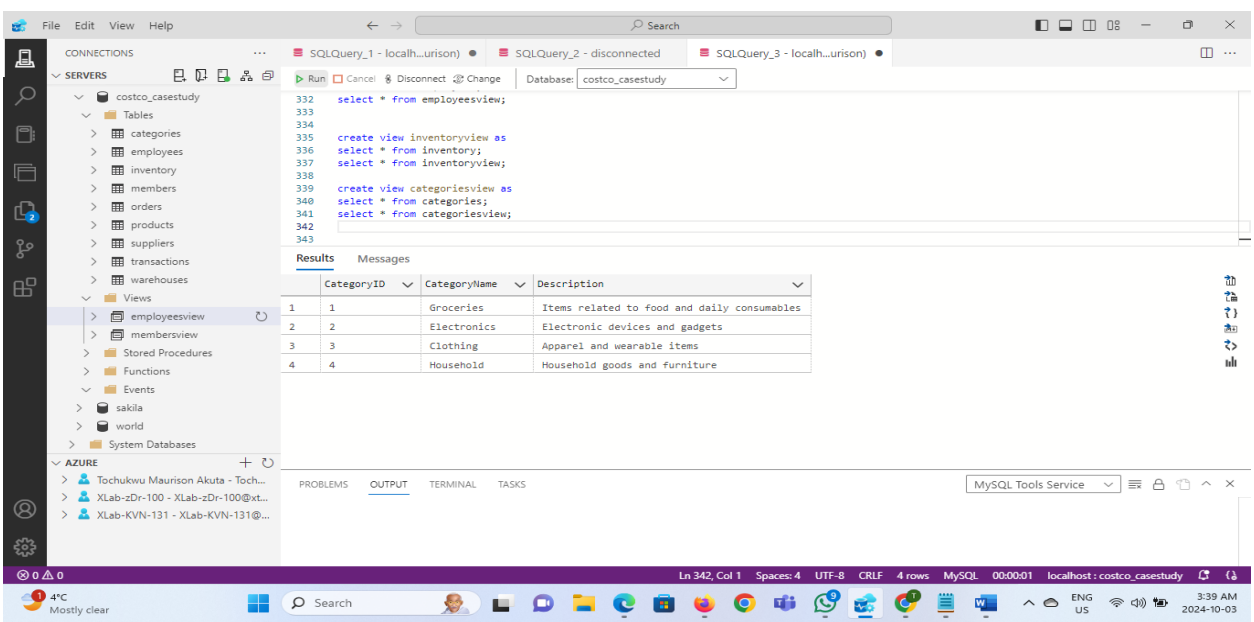
select * from inventoryview;



create view categoriesview as

select * from categories;

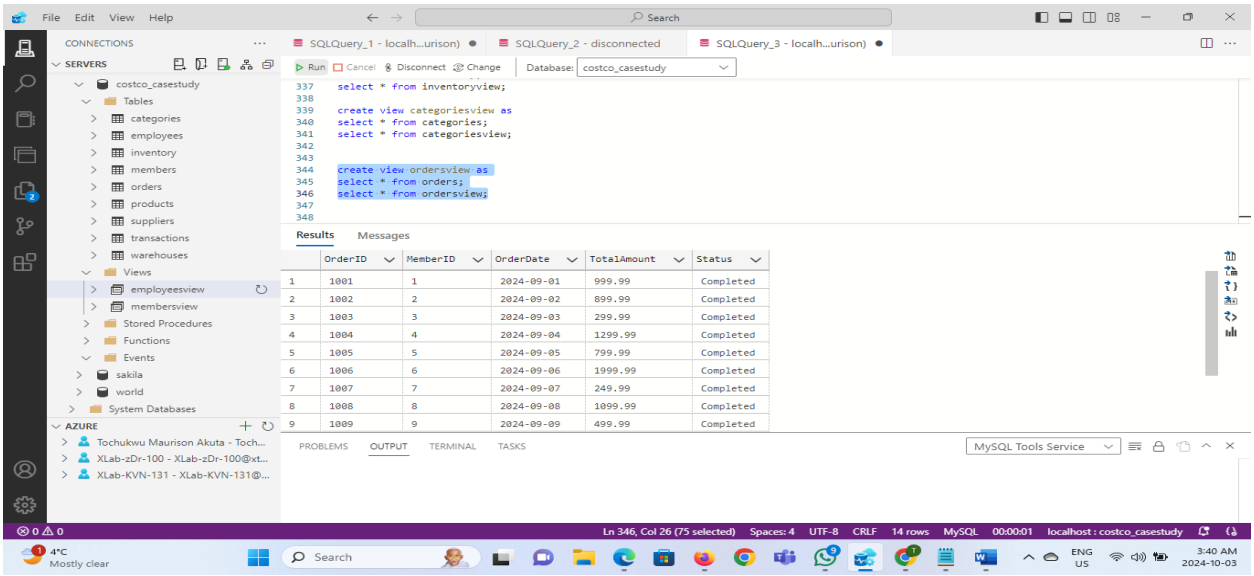
select * from categoriesview;



create view ordersview as

select * from orders;

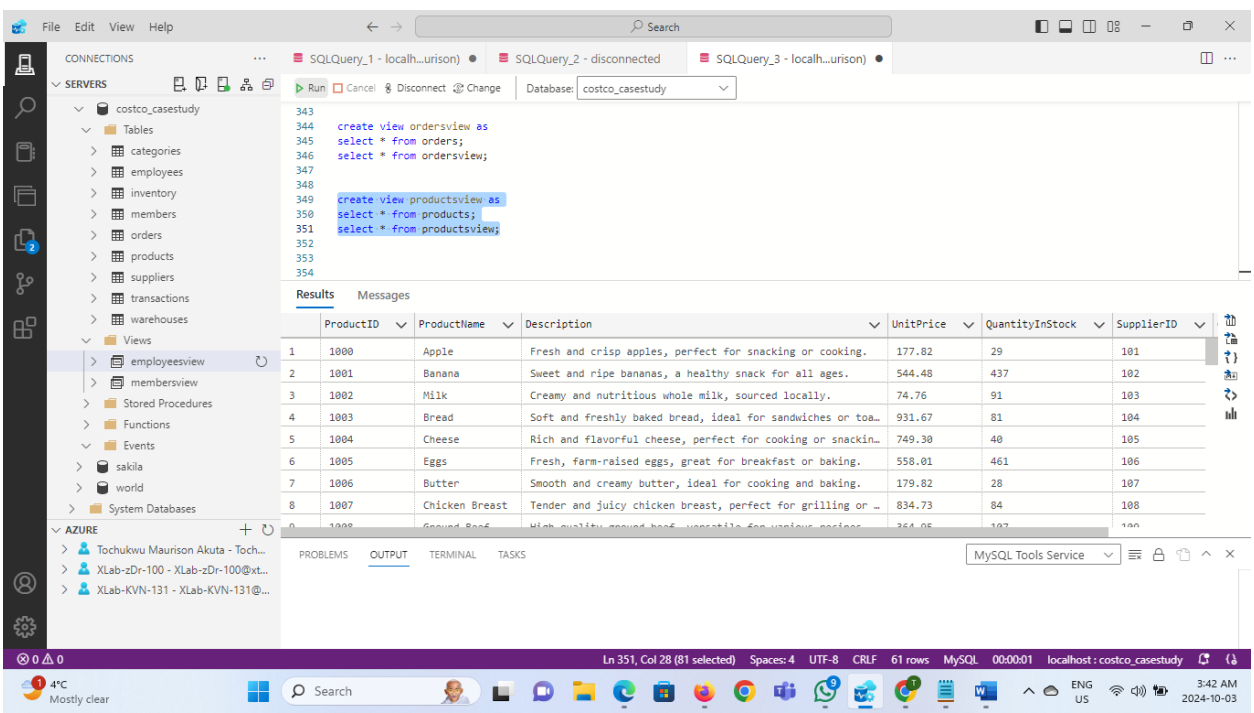
select * from ordersview;



create view productsview as

select * from products;

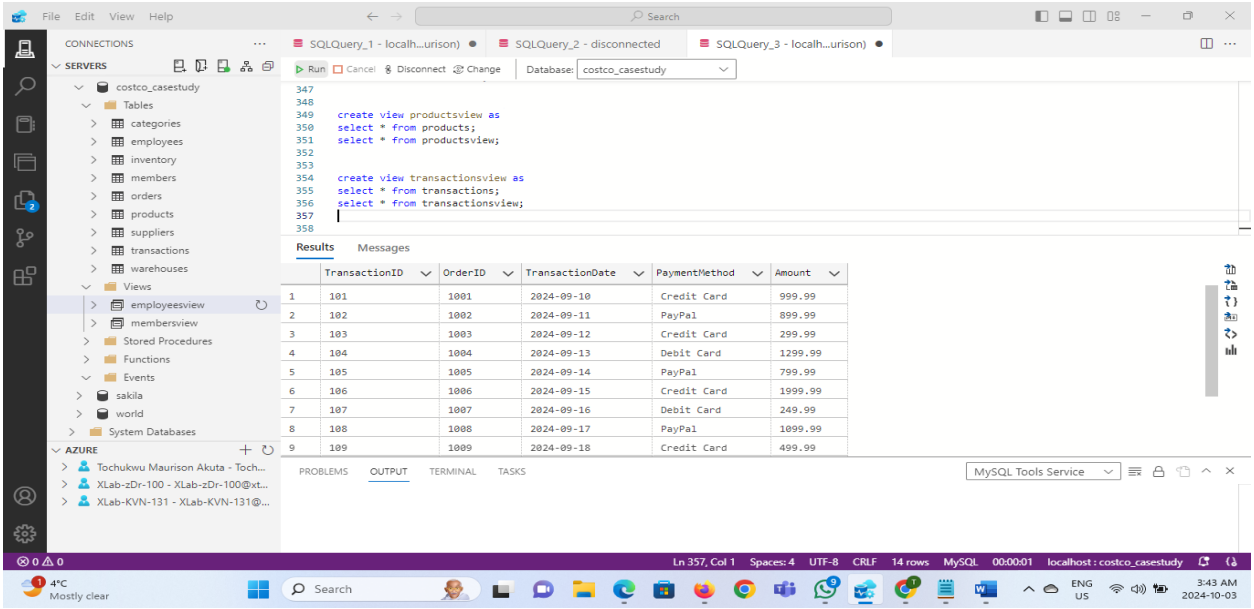
select * from productsview;



create view transactionsview as

select * from transactions;

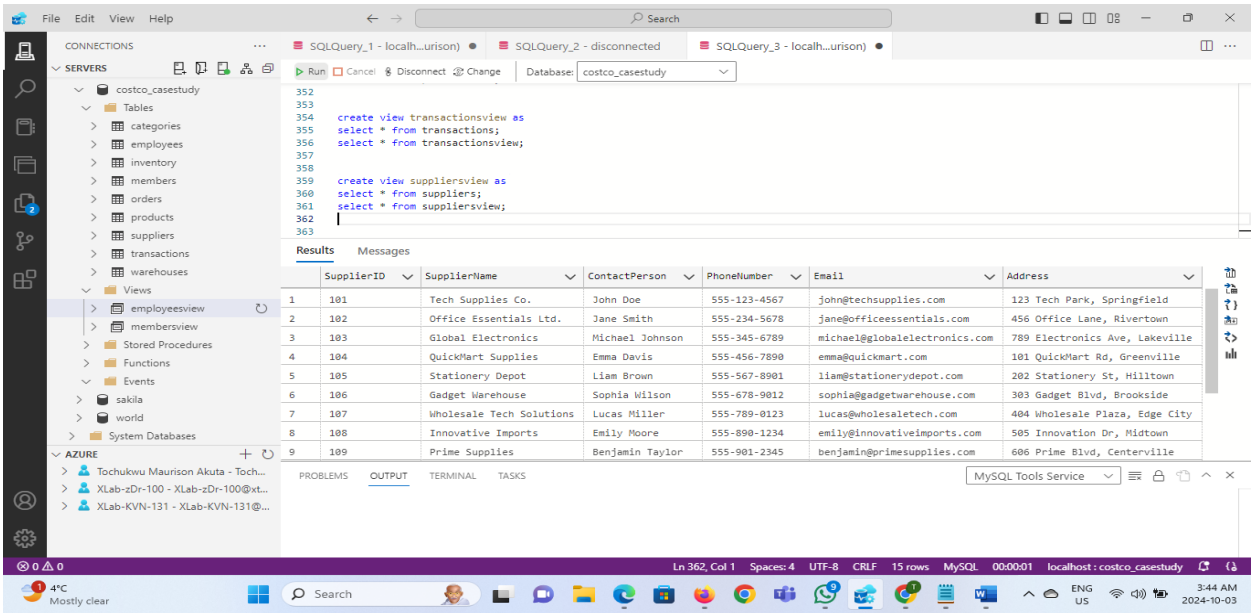
select * from transactionsview;



create view suppliersview as

select * from suppliers;

select * from suppliersview;



create view warehousesview as

select * from warehouses;

select * from warehousesview;