

Database Design for Costco Corporation



Submitted By: Naeem Patel

Case Study: Database Design for Costco Calgary

Table of Contents

1. Introduction
2. Mission and Objectives
3. Database & Design
 - 3.1 Key Tables and Data Entities
 - Members Table
 - Products Table
 - Suppliers Table
 - Orders Table
 - Inventory Table
 - Employees Table
 - Transactions Table
 - Warehouses Table
 - Categories Table
 - Order_Details Table
 - 3.2 Possible Relationships
 - One-to-Many Relationships
 - Many-to-Many Relationships
 - One-to-One Relationships
 - 3.3 Entity Relationship Diagram (ERD)
4. Testing Of Queries in SQL
5. Conclusion

1. Introduction

Costco Wholesale Corporation, founded in 1976, a leading warehouse retailer, operates on a membership-driven model to offer customers quality products at competitive prices. As the business expands, so do its operational demands, requiring a well-structured and scalable database to ensure seamless operations. This case study outlines a comprehensive database design tailored for Costco Calgary, emphasizing the importance of key business functions like inventory management, membership tracking, and transaction processing.

2. Mission and Objectives

Mission:

Costco is dedicated to continuously providing members with quality goods and services at the lowest possible prices, ensuring customer satisfaction through efficient operations and a membership-driven model.

Objectives:

- Ensure efficient supply chain management and procurement processes.
- Provide a seamless customer experience through accurate product availability, pricing, and inventory control.
- Maintain operational cost-effectiveness to support low-pricing models.
- Enhance membership satisfaction and retention through personalized services.
- Enable scalable and robust database architecture to support future growth and operational demands.

3. Database & Design:

3.1 Key Tables and Data Entities

- Members Table

The Members table stores all membership-related information, essential for managing customer interactions.

Field Name	Datatype	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

- Products Table

The Products table contains details of all items available for sale.

Field Name	Datatype	Size	Constraint
ProductID	Integer	10	Primary key
ProductName	Varchar	20	Not Null
Description	Varchar	100	Not Null
Price	Float	10	Not Null
QuantityInStock	Number	10	Not Null
CategoryID	Integer	10	Foreign Key
SupplierID	Integer	10	Foreign Key

- Suppliers Table

The Suppliers table tracks supplier information for effective procurement management.

Field Name	Datatype	Size	Constraint
SupplierID	Integer	10	Primary key
SupplierName	Varchar	20	Not Null
ContactPerson	Varchar	20	Not Null
Email	Varchar	30	Not Null
PhoneNumber	Number	10	Not Null
Address	Varchar	50	Not Null

- Orders Table

The Orders table tracks customer orders, both online and in-store.

Field Name	Datatype	Size	Constraint
OrderID	Integer	10	Primary key
MemberID	Integer	10	Foreign Key
OrderDate	Date	10	Not Null
TotalAmount	Float	10	Not Null
Status	Varchar	100	Not Null

- Inventory Table

The Inventory table manages product availability across various warehouse locations.

Field Name	Datatype	Size	Constraint
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

- Employees Table

The Employees table stores data of employees across different roles and departments.

Field Name	Datatype	Size	Constraint
EmployeesID	Integer	10	Primary key
FirstName	Varchar	20	Not Null
LastName	Varchar	20	Not Null
Email	Varchar	30	Not Null
PhoneNumber	Number	10	Not Null
Department	Varchar	20	Not Null
Position	Varchar	20	Not Null
HireDate	Date	10	Not Null
Mangerid	Integer	10	Null

- Transactions Table

The Transactions table records financial transactions made by customers.

Field Name	Datatype	Size	Constraint
TransactionsID	Integer	10	Primary key
OrderID	Integer	20	Foreign Key
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

- Warehouses Table

The Warehouses table stores information about Costco's warehouse locations.

Field Name	Datatype	Size	Constraint
WarehouseID	Varchar	10	Primary key
ManagerID	Integer	20	Foreign Key
Location	Varchar	30	Not Null
Capacity	Varchar	15	Not Null
PhoneNumber	Number(10)	15	Not Null

- Categories Table

The Categories table organizes products into categories for better management.

Field Name	Datatype	Size	Constraint
CategoryID	Integer	10	Primary key
CategoryName	Varchar	20	Not Null
Description	Varchar	20	Not Null

- Categories Table

Contains information about individual items within an order, including product ID, quantity, and price for each product in a specific order.

Field Name	Datatype	Size	Constraint
OrderID	Integer	10	Primary key
ProductID	Integer	20	Primary key
Quantity	Integer	30	Not Null
PriceAtOrder	Decimal	(10,2)	Not Null

3.2 Possible Relationships

In the proposed database design, several relationships exist between different entities, ensuring data integrity and smooth operations.

- One-to-Many Relationships

One-to-many relationships occur when a single record in one table is associated with multiple records in another table.

- **One Member can place many Orders:**
Explanation: A member (e.g., a Costco customer) can place several orders over time, but each order belongs to only one member. The **MemberID** is stored in the **Orders** table to track which member placed that order.
Example: Member *M123* places 3 separate orders (0001, 0002, and 0003), so these orders will have *M123* in the **MemberID** field.
- **One Supplier can supply many Products**
Explanation: A supplier provides many different products to Costco. The **SupplierID** is stored in the **Products** table to indicate which supplier provides each product.
Example: Supplier *S101* supplies Product A, Product B, and Product C.
- **One Category can have many Products**
Explanation: Each product belongs to a category (e.g., electronics, furniture). The **CategoryID** in the **Products** table identifies which category each product belongs to.
Example: The *Electronics* category has products like *TVs*, *Laptops*, and *Phones*.
- **One Product can appear in many Orders**

Explanation: A product can be ordered by many members. The **OrderID** in a junction table links a product to multiple orders.

Example: *Product A* appears in *Order 001*, *Order 002*, and *Order 005*.

➤ **One Warehouse can store many Products**

Explanation: Each warehouse can hold multiple products. The **WarehouseID** in the **Inventory** table shows which warehouse stores which products.

Example: *Warehouse 001* stores *TVs*, *Furniture*, and *Laptops*.

- Many-to-Many Relationships

Many-to-many relationships occur when records in one table are associated with multiple records in another table, and vice versa. This is usually managed through a junction table.

➤ **Orders and Products have a many-to-many relationship (via a junction table like OrderDetails)**

Explanation: A single order can contain multiple products, and each product can appear in multiple orders. This relationship is handled using a junction table (e.g., **OrderDetails**), which includes both the **OrderID** and **ProductID**.

Example: *Order 001* contains *Product A* and *Product B*. *Product A* also appears in *Order 003*.

- One-to-One Relationships

A one-to-one relationship occurs when a record in one table corresponds to exactly one record in another table.

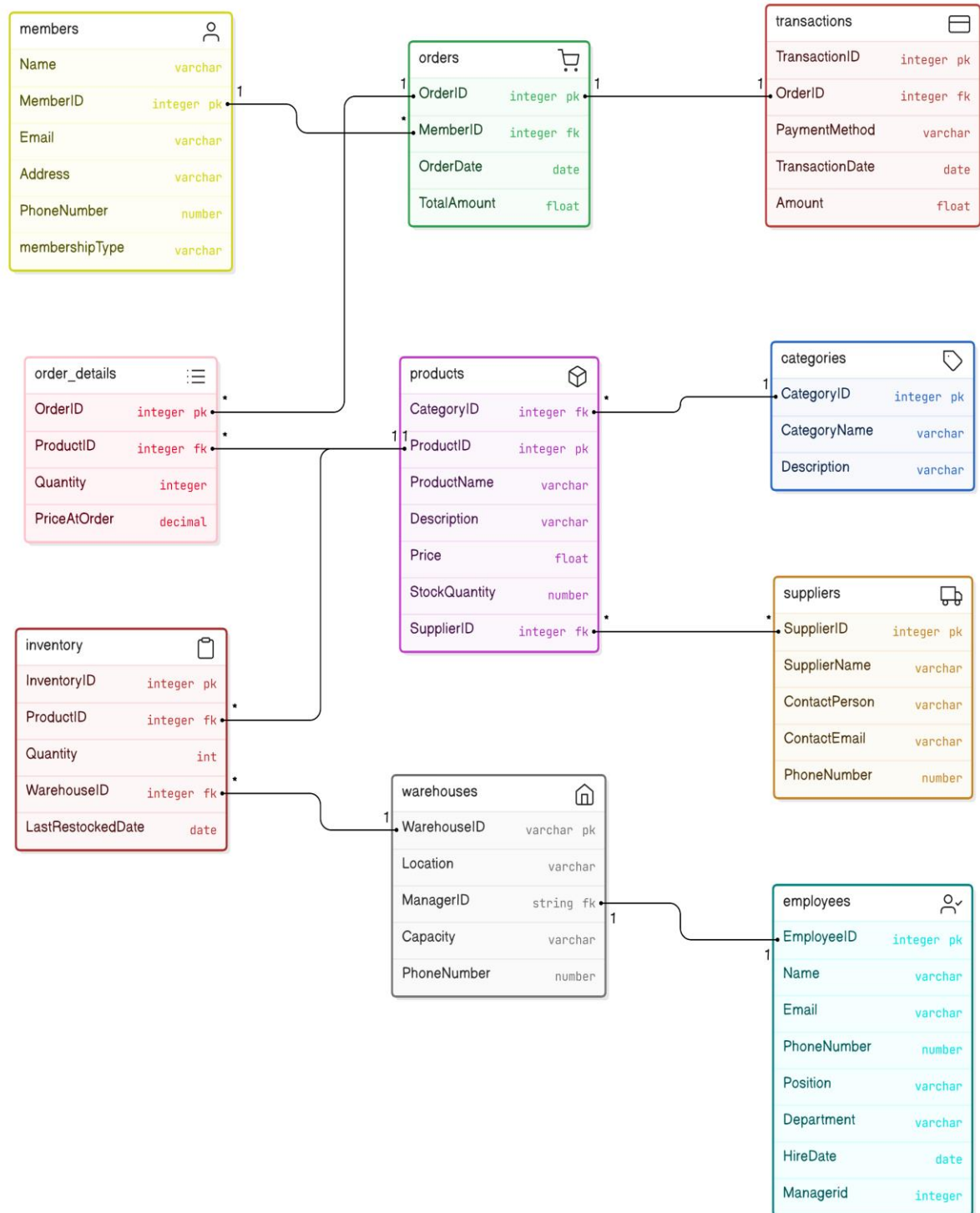
➤ **One Employee (Warehouse Manager) can manage one Warehouse**

Explanation: Each warehouse can be managed by only one manager. The **ManagerID** field in the **Warehouses** table links the warehouse to its manager. In turn, each manager (from the **Employees** table) is assigned to only one warehouse.

Example: *Employee E101* manages *Warehouse 001*.

3.3 Entity Relationship Diagram (ERD)

COSTCO ER DIAGRAM



In the ERD:

- Members are related to Orders, where one member can place multiple orders.
- Orders are linked to both Transactions and Products (via a junction table like OrderDetails).
- Products are related to Suppliers, Inventory, and Categories for better tracking and procurement.
- Inventory is connected to Warehouses, each managed by an employee (warehouse manager).

4. Testing Of Queries in SQL:

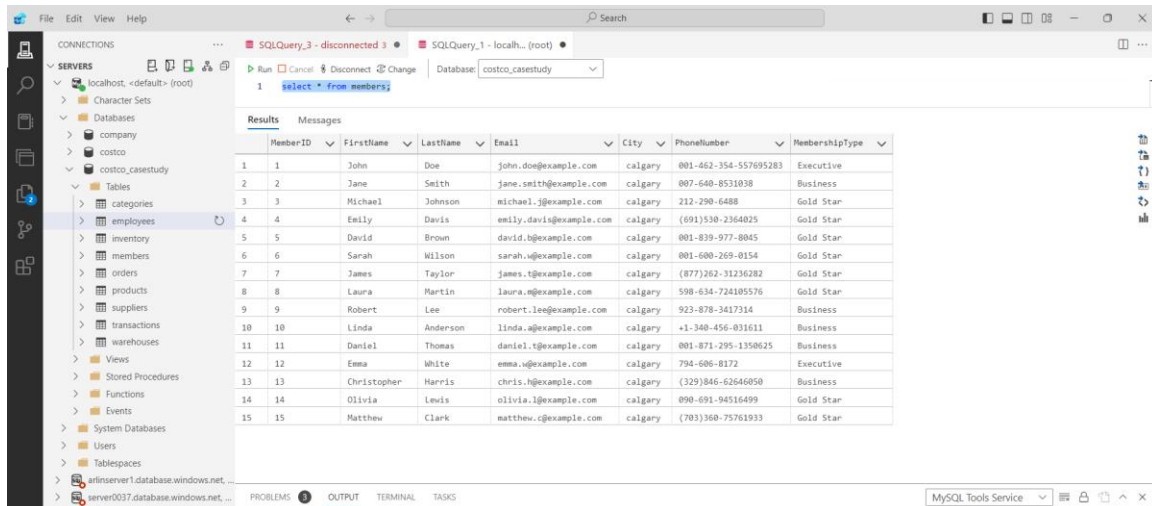
Creation of Database & Tables

-- Create Database

```
CREATE DATABASE Costco_CaseStudy;
```

-- Create Members Table

```
CREATE TABLE Members (
  MemberID INT PRIMARY KEY,
  FirstName VARCHAR(50),
  LastName VARCHAR(50),
  Email VARCHAR(100),
  City VARCHAR(50),
  PhoneNumber VARCHAR(20),
  MembershipType VARCHAR(50)
);
```



The screenshot shows the MySQL Workbench interface. On the left, the 'SERVERS' pane shows a connection to 'localhost' with the 'costco_casestudy' database selected. The 'TABLES' pane shows the 'members' table. The main window displays the 'Results' of a query: 'select * from members;'. The results are shown in a table with 15 rows and 7 columns: MemberID, Firstname, Lastname, Email, City, PhoneNumber, and MembershipType.

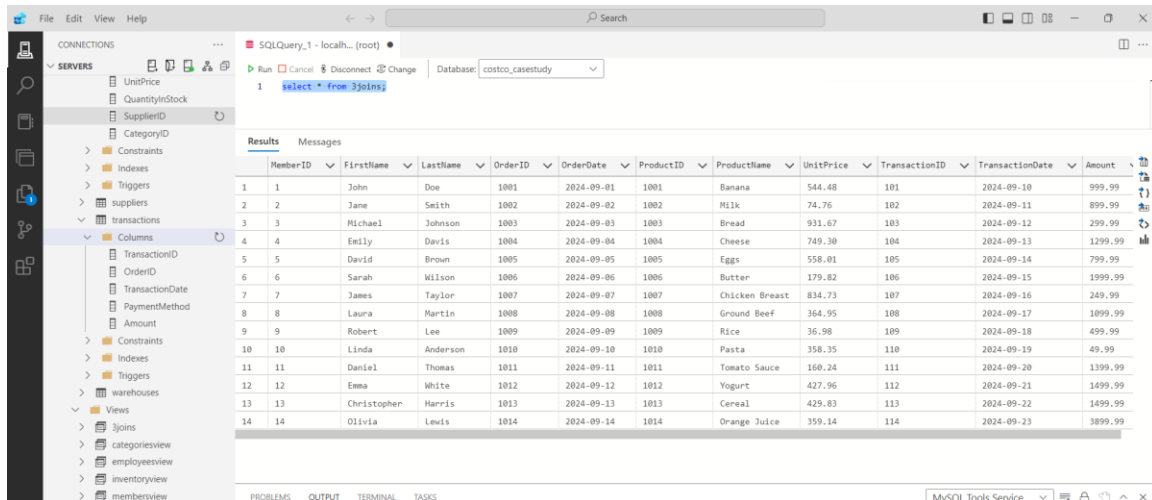
MemberID	Firstname	Lastname	Email	City	PhoneNumber	MembershipType
1	John	Doe	john.doe@example.com	calgary	001-462-354-55709283	Executive
2	Jane	Smith	jane.smith@example.com	calgary	007-640-8531038	Business
3	Michael	Johnson	michael.j@example.com	calgary	212-290-6488	Gold Star
4	Emily	Davis	emily.davis@example.com	calgary	(691)530-2364025	Gold Star
5	David	Brown	david.b@example.com	calgary	801-839-977-8045	Gold Star
6	Sarah	Wilson	sarah.w@example.com	calgary	801-800-269-0154	Gold Star
7	James	Taylor	james.t@example.com	calgary	(877)282-31236282	Gold Star
8	Laura	Martin	laura.m@example.com	calgary	508-634-724105576	Gold Star
9	Robert	Lee	robert.lee@example.com	calgary	923-878-3417314	Business
10	Linda	Anderson	linda.a@example.com	calgary	+1-340-456-031611	Business
11	Daniel	Thomas	daniel.t@example.com	calgary	801-871-295-1350625	Business
12	Emma	White	emma.w@example.com	calgary	794-606-8172	Executive
13	Christopher	Harris	chris.h@example.com	calgary	(329)846-62646050	Business
14	Olivia	Lewis	olivia.l@example.com	calgary	898-691-94516499	Gold Star
15	Matthew	Clark	matthew.c@example.com	calgary	(703)368-75761933	Gold Star

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

```

SELECT
    m.MemberID,
    m.FirstName,
    m.LastName,
    o.OrderID,
    o.OrderDate,
    p.ProductID,
    p.ProductName,
    p.Price AS UnitPrice,
    t.TransactionID,
    t.TransactionDate,
    t.Amount
FROM
    Members m
JOIN
    Orders o ON m.MemberID = o.MemberID      -- Join Members with Orders
JOIN
    Transactionst ON o.OrderID = t.OrderID      -- Join Orders with Transactions
JOIN
    OrderDetails od ON o.OrderID = od.OrderID  -- Join Orders with OrderDetails
(junction table)
JOIN
    Products p ON od.ProductID = p.ProductID  -- Join OrderDetails with Products
WHERE
    o.Status = 'Completed';                    -- Optional: Filter for completed orders

```



SQLQuery_1 - localh... (root)

Database: costco_casesstudy

1 select * from 3joins;

Results Messages

	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	168.24	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

MySQL Tools Service

6. Conclusion

The database design for Costco Calgary's warehouse operations is critical for maintaining efficient operations, from tracking inventory levels to processing transactions and ensuring seamless supply chain management. By establishing clear relationships between key entities, the proposed database structure aligns with Costco's operational goals, supports its commitment to member satisfaction, and allows for future growth.

This database model offers a robust foundation for Costco's operational and transactional needs, ensuring the organization can scale while maintaining cost-effectiveness, accurate data management, and customer satisfaction.