4/10/2024

Costco Database Design

Arlin George
SOUTHERN ALBERTA INSTITUTE OF TECHNOLOGY

https://github.com/arlingeo99/DATA_SCIENCE_PROJECTS

Contents

Costco Database Design

Abstract

This article depicts a detailed analysis of a database design for Costco Corporation, which supports core business operations such as member management, product inventory, supplier relations, and warehouse operations. The design contains essential tables such as Members, Products, Suppliers, Orders, OrderDetails, Transactions, Inventory, Employees, and Warehouses, with clear relationships and constraints. Proposed optimizations include the addition of another table named Department to better manage employee. These enhancements will strengthen Costco's data management, ensuring future scalability and operational efficiency.

Keywords

Costco, Database Design, ER Diagram, Supply Chain, Inventory Management, Employee Roles, Database Optimization, Data Scalability

Introduction

Costco's operational success depends heavily on its ability to manage its members, inventory, and warehouse operations flawless. Designing an efficient and scalable database structure is essential to support their business functions, maintain member satisfaction, and streamline supply chain management. The database design proposed here assembles the core elements of Costco's business into interconnected tables, ensuring robust data flow and consistency across operations.

Mission

The goal of Costco is to consistently offer its members premium products and services at the most competitive pricing. Through a simplified supply chain and efficient operations, the firm aims to retain client satisfaction. By focusing on a membership-driven business model, Costco ensures that its members receive special value.

Objectives

This database design of Costco plays a vital role in achieving its objectives through efficient data management, which justifies its importance in the following ways:

Objective 1: Ensure efficient supply chain management and procurement processes

- The **Suppliers Table**, allows Costco to efficiently manage its relationships with vendors, monitor supplier performance, and confirm that products are obtained at competitive prices. The ability to track supplier contracts, pricing agreements, and restocking details makes it easier to well run the supply chain operations and reduce procurement costs. **Inventory Table** enables the business to timely refill stock and

also to foresee or avoid stockouts and overstocking which ensures coming on schedule with goods for customers, and the overall effective management of all processes within the supply chain.

Objective 2: Provide a seamless customer experience through accurate product availability, pricing, and inventory control

- The Products Table and Inventory Table are essential for guaranteeing that Costco maintains up-to-date data on product price and availability. This is vital for maintaining Costco's commitment to deliver a seamless shopping experience Through the Inventory Table, Costco can monitor stock levels in each warehouse across different regions, to make sure that members can buy the products they want, both in the warehouses and online, ensuring customer satisfaction and brand loyalty.

Objective 3: Maintain operational cost-effectiveness to support low-pricing models

- By using the **Orders Table**, **OrderDetails Table**, and **Transactions Table**, Costco can monitor sales trends, evaluate the success of promotions, and adjust pricing strategies in real time. This cost-effective operational management supports Costco's mission of maintaining low prices for customers. The efficient flow of data between these tables enables Costco to reduce operational costs, particularly by optimizing inventory levels and supplier agreements.

Objective 4: Enhance membership satisfaction and retention through personalized services

- The **Members Table** plays a crucial role in tracking member purchases, preferences, and types of memberships (Executive, Gold Star, Business). This data allows Costco to offer personalized promotions, rewards, and other marketing strategies, all aimed at improving customer experience and retaining members. The ability to track individual member interactions helps Costco increase membership satisfaction, which is a key driver in its membership-based business model.

Objective 5: Enable scalable and robust database architecture to support future growth and operational demands

- The proposed design is built to be scalable, allowing for the addition of more members, products, warehouses, and employees as Costco grows. The relationships between the **Orders Table**, **OrderDetails Table**, **Transactions Table**, and **Inventory Table** are designed in a way that allows for smooth scaling of operations. Introducing the **Department Table** adds more flexibility in employee role management. This scalable architecture ensures that Costco can expand its business without being obstructed by database constraints.

DATABASE DESIGN

Core Table Structure and Relationships

- Justification for Each Table

1. Members Table:

The Members table holds critical information about Costco's customers, their contact details, and membership type (Executive, Gold Star, Business). This is essential for tracking purchases, issuing rewards, and providing personalized services.

2. Products Table:

This table contains product information like name, description, price, and stock quantity. It's linked to suppliers and categories, ensuring that each product has a supplier and fits into a specific category.

3. Suppliers Table:

Holding supplier details, this table ensures Costco can track where products come from, who to contact for procurement, and supplier performance.

4. Orders Table:

Orders capture customer transactions, storing details about the date, member ID, and status (pending, shipped, delivered). This table allows Costco to monitor order trends and fulfillment efficiency.

5. Order Details Table:

This table act as a bridge between Orders and Products, the Order Details table tracks the quantity and price of each product within an order.

6. Inventory Table:

The Inventory keep track of the product availability at different warehouse locations, including when they were last restocked.

7. Employees Table:

This table stores employee details, including their roles and department. It's linked to the Warehouses table through the ManagerID, identifying which employee manages which warehouse.

8. Transactions Table:

Every order generates a financial transaction. The Transactions table captures payment method (cash, card), transaction date, and amount.

9. Warehouses Table:

This table stores each location's capacity, manager, and contact details, ensuring Costco can manage inventory distribution effectively.

10. Categories Table:

Categories organize products into logical groups, making it easier for Costco to manage inventory and provide users with accurate, categorized listings.

Key Relationships

1. **Members and Orders**:
    - **Relationship**: One-to-many
    - **Explanation**: A member can place multiple orders, but each order is associated with only one member.
2. **Orders and OrderDetails**:
    - **Relationship**: One-to-many
    - **Explanation**: Each order can contain multiple products (order details), but each order detail belongs to one order.
3. **OrderDetails and Products**:
    - **Relationship**: Many-to-one
    - **Explanation**: An order detail refers to a single product, but each product can be part of multiple order details.
4. **Products and Categories**:
    - **Relationship**: Many-to-one
    - **Explanation**: Each product belongs to one category, but a category can contain multiple products.
5. **Products and Suppliers**:
    - **Relationship**: Many-to-one
    - **Explanation**: Each product is supplied by one supplier, but a supplier can supply multiple products.
6. **Orders and Transactions**:
    - **Relationship**: One-to-one
    - **Explanation**: Each order is associated with one transaction, and each transaction is linked to one order.
7. **Warehouses and Inventory**:
    - **Relationship**: One-to-many
    - **Explanation**: Multiple goods (inventory records) can be kept in a warehouse, but each inventory record only pertains to one warehouse.

8. **Products and Inventory**:
    - **Relationship**: One-to-many
    - **Explanation**: Although a product may be kept in more than one warehouse, only one product is mentioned in each inventory record.
9. **Warehouses and Employees**:
    - **Relationship**: One-to-many
    - **Explanation**: Although a warehouse may employ several people, each person is only connected to a single warehouse.
10. **Employees and Warehouses (via ManagerID)**:
    - **Relationship**: One-to-one (1:1)
    - **Explanation**: Every warehouse is assigned to a manager or employee, and each manager or employee is only able to manage one warehouse.

Proposed Optimization of the current database design: Department Table

Currently, the relationship between employees and warehouses assigns each employee as a manager. However, not all employees should be managers. To resolve this, I suggest adding a Department Table to define different roles and ensure that only some employees manage warehouses.

Department Table Structure:

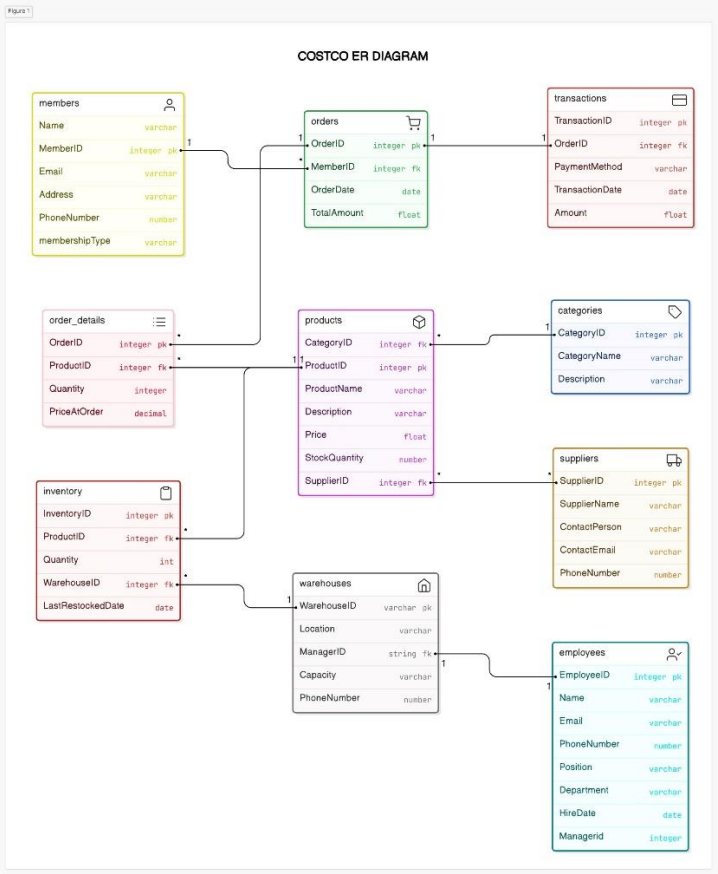
1. DepartmentID (Primary Key)
2. DepartmentName (e.g., Sales, Logistics, Management, etc.)
3. ManagerID (Foreign Key from Employees table)

By linking the Employees table to this Department table, only designated employees (with the ManagerID) would be linked to warehouse management, while others could be associated with various roles and departments. This avoids the assumption that all employees are managers.

ER Diagram

This ER diagram represents the database structure for Costco, showcasing relationships between various entities:

- **Members**: Contains details like name, email, address, phone number, and membership type, linked to the orders table through MemberID.
- **Orders**: Tracks each order's ID, date, member ID, and total amount, connected to order_details, transactions, and members.
- **Order Details**: Holds specific order information, such as the product ID, quantity, and price at the time of order, linked to products.
- **Products**: Stores product details, including price, stock quantity, category, and supplier, connected to both order_details and inventory.
- **Inventory**: Manages product stock levels at various warehouses, linking WarehouseID and ProductID.
- **Warehouses**: Consist of warehouse locations and information about their capacity and management, connected to inventory and employees.
- **Employees**: Lists employee information, including their name, position, and department, linked to warehouses through ManagerID.
- **Transactions**: Tracks payment and transaction details related to each order.
- **Suppliers**: Manages supplier details, linked to products.
- **Categories**: Stores product categories, linked to products.



Conclusion

The database structure designed for Costco captures the core elements of its business, ensuring efficient member management, inventory control, and warehouse operations. Adding the Department table will better represent employee roles, preventing the over-designation of employees as managers. This design not only supports current operations but also lays the foundation for future scalability as Costco continues to grow and evolve its membership-driven model.

APPENDIX

A. TABLE DETAILS

1. Members

- **MemberID**: INT, Primary Key
- **FirstName**: VARCHAR(50)
- **LastName**: VARCHAR(50)
- **Email**: VARCHAR(100)
- **City**: VARCHAR(50)
- **PhoneNumber**: NUMBER
- **MembershipType**: VARCHAR(50)

2. Orders

- **OrderID**: INT, Primary Key
- **MemberID**: INT, Foreign Key referencing Members(MemberID)
- **OrderDate**: DATE
- **TotalAmount**: DECIMAL(10, 2)
- **Status**: VARCHAR(50)

3. Suppliers

- **SupplierID**: INT, Primary Key
- **SupplierName**: VARCHAR(100)
- **ContactPerson**: VARCHAR(50)
- **PhoneNumber**: NUMBER
- **Email**: VARCHAR(100)
- **Address**: VARCHAR(255)

4. Categories

- **CategoryID**: INT, Primary Key
- **CategoryName**: VARCHAR(50)
- **Description**: TEXT

5. Products

- **ProductID**: INT, Primary Key
- **ProductName**: VARCHAR(50)
- **Description**: TEXT
- **UnitPrice**: DECIMAL(10, 2)

- **QuantityInStock**: INT
- **SupplierID**: INT, Foreign Key referencing Suppliers(SupplierID)
- **CategoryID**: INT, Foreign Key referencing Categories(CategoryID)

6. Transactions

- **TransactionID**: INT, Primary Key
- **OrderID**: INT, Foreign Key referencing Orders(OrderID)
- **TransactionDate**: DATE
- **PaymentMethod**: VARCHAR(50)
- **Amount**: DECIMAL(10, 2)

7. Inventory

- **InventoryID**: INT, Primary Key
- **ProductID**: INT, Foreign Key referencing Products(ProductID)
- **WarehouseID**: INT, Foreign Key referencing Warehouses(WarehouseID)
- **QuantityAvailable**: INT
- **LastRestockedDate**: DATE

8. Employees

- **EmployeeID**: INT, Primary Key
- **FirstName**: VARCHAR(50)
- **LastName**: VARCHAR(50)
- **Email**: VARCHAR(100)
- **PhoneNumber**: NUMBER
- **Department**: VARCHAR(50)
- **Position**: VARCHAR(50)
- **HireDate**: DATE

9. OrderDetails

- **OrderID**: INT, Foreign Key referencing Orders(OrderID)
- **ProductID**: INT, Foreign Key referencing Products(ProductID)
- **Quantity**: INT
- **PriceAtOrder**: DECIMAL(10, 2)
- **Primary Key**: (OrderID, ProductID)

10. Warehouses

- **WarehouseID**: VARCHAR(20), Primary Key

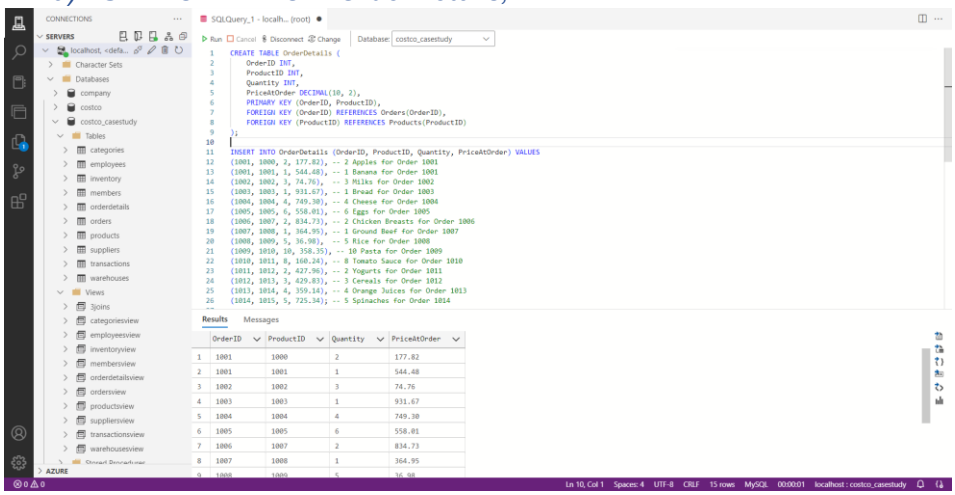
- **Location**: VARCHAR(100)
- **ManagerID**: INT, Foreign Key referencing Employees(EmployeeID)
- **Capacity**: VARCHAR(50)
- **PhoneNumber**: NUMBER

B. CREATING AND TESTING DATABASE WITH QUERIES

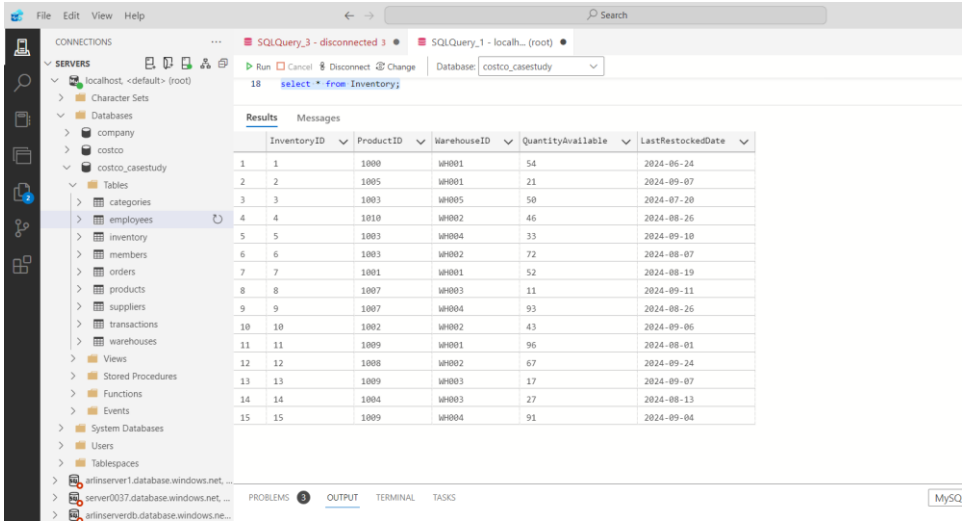
A database named Costco_CaseStudy has been created using all the mentioned tables, attributes and constraints, and has been populated with some sample data.

The following snapshot is the result of the SQL query

*a) SELECT * FROM OrderDetails;*



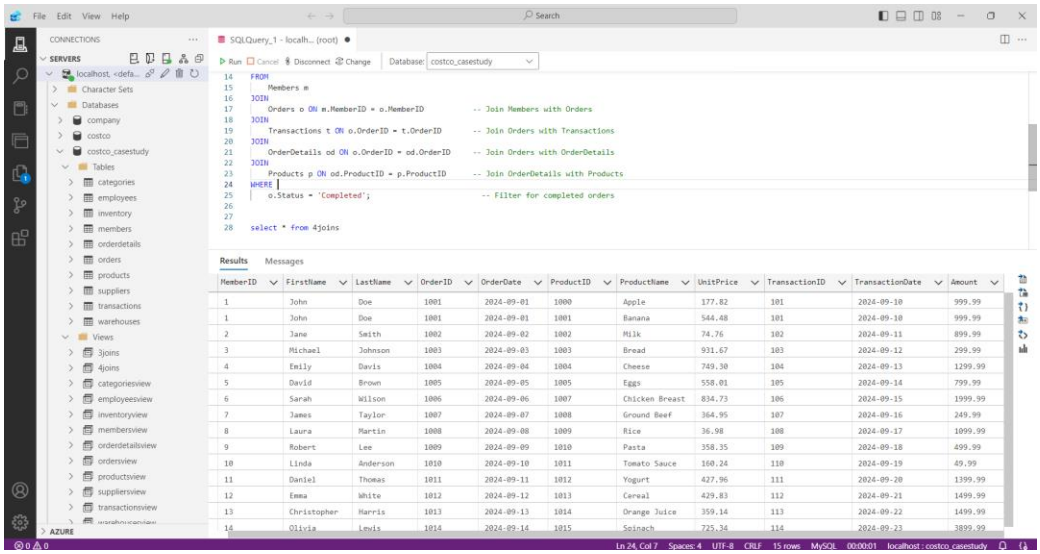
*b) SELECT * FROM INVENTORY;*



c) JOIN and VIEW QUERY

The following SQL join query creates a view that combines member, order, transaction, and product details, specifically for completed orders. It accurately links members to their respective orders and products through the order details, providing a comprehensive overview of completed transactions.

```
CREATE VIEW 4joins AS
SELECT
    m.MemberID,
    m.FirstName,
    m.LastName,
    o.OrderID,
    o.OrderDate,
    od.ProductID,
    p.ProductName,
    p.UnitPrice,
    t.TransactionID,
    t.TransactionDate,
    t.Amount
FROM
    Members m
JOIN
    Orders o ON m.MemberID = o.MemberID          -- Join Members with Orders
JOIN
    Transactions t ON o.OrderID = t.OrderID      -- Join Orders with Transactions
JOIN
    OrderDetails od ON o.OrderID = od.OrderID    -- Join Orders with OrderDetails
JOIN
    Products p ON od.ProductID = p.ProductID     -- Join OrderDetails with Products
WHERE
    o.Status = 'Completed';                      -- Filter for completed orders
```



d) GROUP BY QUERY

This SQL query calculates the total amount spent by each member on their completed orders. This query joins the Members table with Orders table to filter completed orders. This query helps to analyse the member spending behaviour within the Costco database.