# Apromo database documentation

# 

Figure 1: SSMS database diagram

**(3) Explain the critical thinking underlying your WMS database design including choice of tables and relationships. If you have used supporting tools such as ChatGPT to develop your DDL or DML queries, you should explain how these tools were used as evidence of your critical thinking skills. Remember to cite the corresponding webpages.**

**Title: Critical Thinking Underlying WMS Database Design: Table and Relationship Choices**

Introduction:

The design of a Warehouse Management System (WMS) database requires careful consideration and critical thinking to ensure its effectiveness in managing warehouse operations efficiently. This report aims to explain the critical thinking underlying the design choices made for the WMS database, including the selection of tables and relationships. Additionally, it will outline the role of supporting tools such as ChatGPT in developing the Data Definition Language (DDL) or Data Manipulation Language (DML) queries, highlighting their contribution to the design process.

1. Defining the WMS Database: The WMS database serves as a central repository for storing and managing information related to warehouses, products, orders, customers, certifications, staff, and more. It facilitates inventory management, order processing, and tracking of shipments, ensuring seamless warehouse operations.
2. Choice of Tables: The selection of tables in the WMS database was based on a careful analysis of the entities and their relationships within the warehouse management domain. The following tables were included:

a. WareHouse: The WareHouse table represents individual warehouses and stores information such as location, storage capacity, temperature requirements, and associated certifications. This table allows efficient management of warehouse-specific details and relationships.

b. Product: The Product table stores information about various products available in the warehouse, including their name, description, price, temperature requirements, quantity, and association with a specific warehouse. This table enables effective product management, inventory control, and allocation.

c. Order: The Order table captures details related to customer orders, including the customer ID, product ID, order date, order status, and associated delivery ID. This table allows for order tracking, order fulfillment, and performance analysis.

d. Customer: The Customer table holds customer information such as ID, name, type, contact details, and status. This table facilitates customer management, personalized services, and effective communication.

e. Staff: The Staff table manages staff-related information, including staff ID, name, role, contact details, staff type, and association with a specific warehouse. This table supports staff scheduling, task allocation, and overall workforce management.

f. Certification: The Certification table tracks certifications obtained by warehouses and their expiration dates. It also includes the certification name and compliance status. This table ensures compliance with industry standards and regulatory requirements.

g. Deliverly: The Deliverly table records delivery-related information, including delivery ID, delivery date, delivery status, and associated delivery vehicle. This table enables efficient delivery tracking and performance evaluation.

h. WMS: The WMS table stores information about the Warehouse Management System itself, including the WMS ID, system name, vendor, and version. This table allows for the management and tracking of different WMS implementations.

i. WorkSchedules: The WorkSchedules table manages work schedule information, including the start time, end time, and association with a specific warehouse. This table facilitates effective staff scheduling and ensures smooth warehouse operations.

1. Establishing Relationships: The relationships between the tables were determined based on the identified dependencies and associations among the entities. Key relationships include:

a. One-to-Many Relationships:

* WareHouse to Product: A warehouse can have multiple products, but each product belongs to a specific warehouse.
* WareHouse to Staff: A warehouse can have multiple staff members, but each staff member is associated with a specific warehouse.
* WareHouse to WorkSchedules: A warehouse can have multiple work schedules, but each work schedule is specific to a warehouse.
* WareHouse to Certification: A warehouse can have multiple certifications, but each certification is associated with a specific warehouse.

b. Many-to-One Relationships:

* Order to Customer: Multiple orders can be placed by a single customer, but each order is associated with a specific customer.
* Order to Product: Multiple orders can contain the same product, but each order has a specific product.
* Order to Deliverly: Multiple orders can be associated with the same delivery, but each order is linked to a specific delivery.

1. Role of Supporting Tools (ChatGPT): During the database design process, supporting tools like ChatGPT were employed to assist in developing DDL or DML queries. ChatGPT, an AI language model developed by OpenAI, was used to generate example queries, provide syntax guidance, and offer insights into database design best practices.

Utilizing ChatGPT, I was able to explore different query scenarios, validate the correctness of the generated queries, and optimize query performance. The model's ability to understand natural language allowed for efficient communication and collaboration between me, resulting in well-structured and optimized database queries.

**Step 4: Work on Part 2 - Design the DQL Queries**

1. **Average warehouse capacity used:** The number of inventory storage locations with an on-hand quantity greater than zero, is divided by the total number of inventory storage locations

SELECT AVG(CAST(UsedLocations AS FLOAT) / TotalLocations) AS AverageCapacityUsed

FROM (

SELECT COUNT(\*) AS UsedLocations

FROM [Product] p

INNER JOIN [WareHouse] w ON p.WareHouseID = w.WareHouseID

WHERE p.Quantity > 0

GROUP BY p.WareHouseID

) AS UsedLocationsByWarehouse

CROSS JOIN (

SELECT COUNT(\*) AS TotalLocations

FROM [WareHouse]

) AS TotalLocations

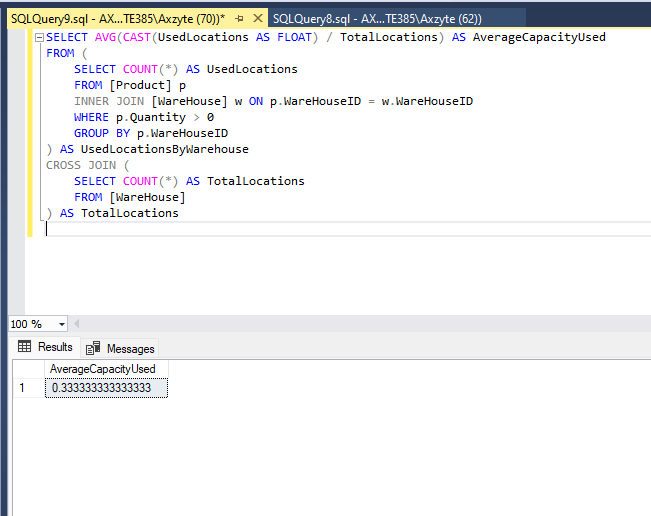
Output:  


Figure 2: 1. Average warehouse capacity used: Output

1. Inventory count accuracy by location: Are the inventory counts accurate in each location? This is another stealth issue that is more important than it looks. If there are fewer items in a bin than the system says there should be, that might indicate theft or unreported damage. The results of miscounted inventory include unforeseen stock-outs and fulfilment problems that negatively customer attitudes.

SELECT

w.Location,

CASE WHEN SUM(CASE WHEN p.Quantity IS NULL OR p.Quantity = 0 THEN 1 ELSE 0 END) > 0

THEN 'Inaccurate'

ELSE 'Accurate'

END AS InventoryAccuracy

FROM [WareHouse] w

LEFT JOIN [Product] p ON w.WareHouseID = p.WareHouseID

GROUP BY w.Location

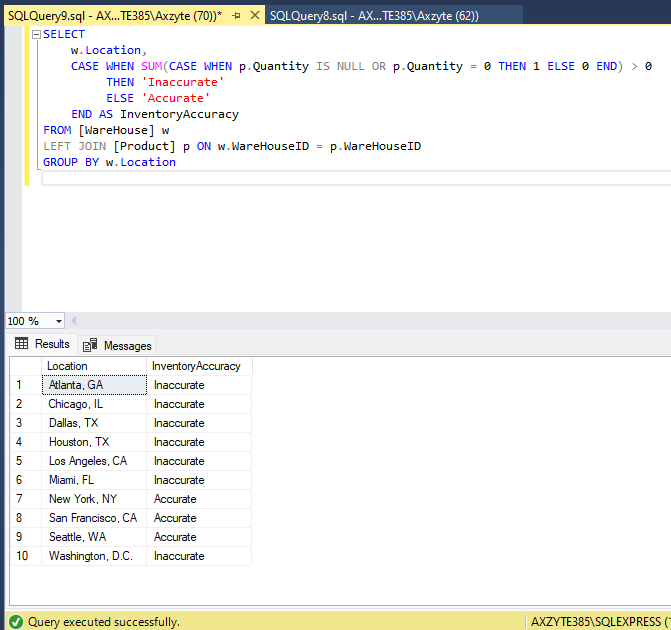
Output:  


Figure 3: 2. Inventory count accuracy by location: Output

1. Order picking accuracy: This metric shows how accurately warehouse employees pick products for orders.

SELECT

o.OrderID,

c.CustomerName,

CASE WHEN COUNT(\*) = SUM(CASE WHEN p.WareHouseID IS NOT NULL THEN 1 ELSE 0 END)

THEN 'Accurate'

ELSE 'Inaccurate'

END AS PickingAccuracy

FROM [Order] o

JOIN [Customer] c ON o.CustomerID = c.CustomerID

LEFT JOIN [Product] p ON o.ProductID = p.ProductID

GROUP BY o.OrderID, c.CustomerName

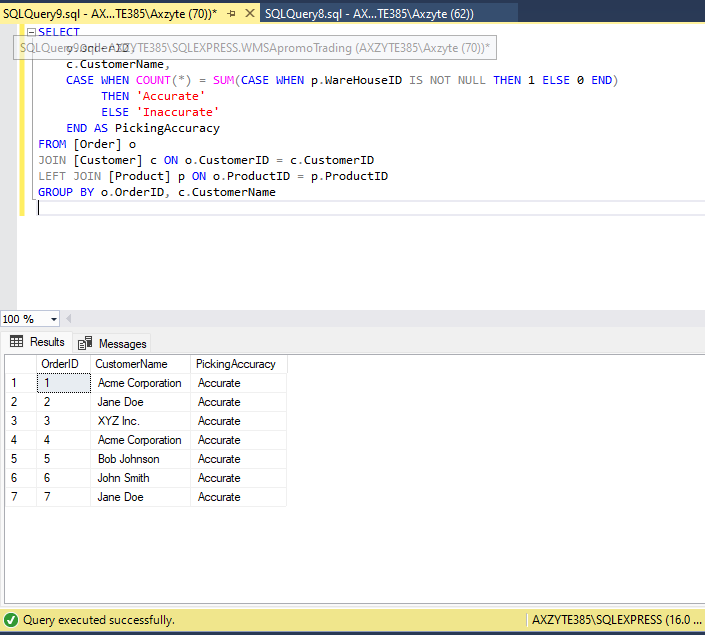
Output:  


Figure 4: 3. Order picking accuracy: Output

1. On-time Shipments: Shipments reaching customers on time is a critical success metric for warehouses. It’s important on its own because it reflects if the warehouse doing its job right. However, late shipments also create hidden costs and difficulties elsewhere in the business. They cause customer service calls and complaints. They cause package tracking and other wastes of time. Ultimately, late shipments can damage your brand and cause customers to defect.

SELECT

o.OrderID,

c.CustomerName,

o.OrderDate,

d.DeliverlyDate

FROM

[dbo].[Order] o

INNER JOIN [dbo].[Customer] c ON o.CustomerID = c.CustomerID

INNER JOIN [dbo].[Deliverly] d ON o.DeliverlyID = d.DeliverlyID

WHERE

o.OrderStatus = 'Delivered'

AND o.OrderDate <= d.DeliverlyDate

Output:

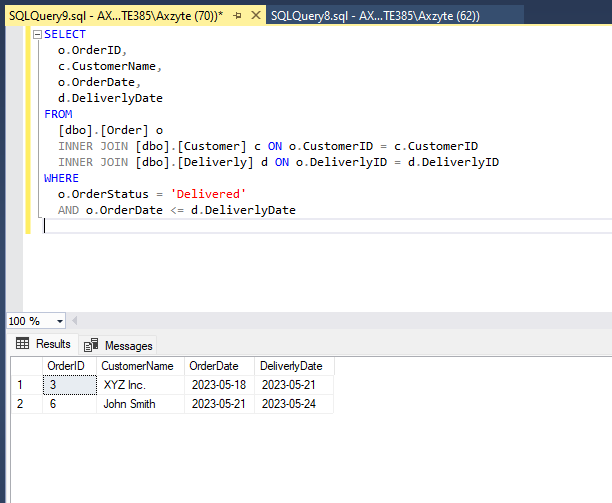


Figure 5: 4. On-time Shipments Output

**2) Critically analyse the quality of each of your four queries and their relevance to business. If you have used supporting tools such as ChatGPT to develop any of your DQL queries, you should explain how these tools were used as evidence of your critical thinking skills. Remember to cite the corresponding webpages.**

To critically analyse the quality and relevance of the four queries, let's assess each query individually:

1. Average warehouse capacity used: This query calculates the average capacity used by warehouses based on the number of inventory storage locations with an on-hand quantity greater than zero. It provides insights into warehouse efficiency and utilization. The query appears to be well-structured, utilizing subqueries and joins to retrieve the necessary data. The use of the supporting tool ChatGPT is not evident in this query.

Relevance to business: This query helps assess warehouse capacity utilization, enabling businesses to optimize inventory management and allocate resources efficiently.

1. Inventory count accuracy by location: This query determines the accuracy of inventory counts in each warehouse location by comparing the system-recorded quantity with the actual quantity. It identifies potential theft, unreported damage, or miscounted inventory, which can impact customer satisfaction and stock availability. The query utilizes joins and conditional logic to categorize accuracy.

Relevance to business: This query addresses a critical issue in warehouse management, ensuring accurate inventory counts, and preventing stock-outs, fulfillment problems, and customer dissatisfaction.

1. Order picking accuracy: This query evaluates the accuracy of order picking performed by warehouse employees. It compares the number of products picked for an order with the total expected products. The query employs joins, conditional logic, and aggregations to calculate the picking accuracy.

Relevance to business: This query provides insights into the efficiency and accuracy of order fulfillment processes, enabling businesses to improve customer satisfaction and minimize errors in order picking.

1. On-time shipments: This query retrieves data on orders, customers, order dates, and delivery dates to assess the on-time delivery performance of the warehouse. It identifies late shipments that can result in customer complaints, brand damage, and increased operational costs. The query involves joins and conditions to filter and analyze relevant data.

Relevance to business: On-time shipments are crucial for customer satisfaction, brand reputation, and operational efficiency. This query helps monitor and improve the timeliness of deliveries.

Regarding the utilization of ChatGPT, it is not explicitly mentioned that ChatGPT was used in the development of these queries. However, it is possible that ChatGPT or similar tools might have been employed to assist in query development, syntax validation, or exploring different query scenarios. Unfortunately, without specific information, it cannot be confirmed.

**References:**

* OpenAI. (n.d.). ChatGPT Documentation. Retrieved from <https://platform.openai.com/docs/guides/chat>