



**Course Title : Database Management System Lab**

**Project Name : Warehouse Management System**

**Submitted By**

Md: Sifat Hossen  
ID: **41230100957**

Anamul Islam  
ID: **41230100841**

**Submitted to**

Rumman Ahmed Prodhon  
Lecturer, Department of CSE  
Northern University Bangladesh

# Index

No.	Topic Name	Page No.
1	Information of the Database	3-4
2	Schema Diagram of the warehouse Management System	5
3	ER Diagram of the warehouse Management System	6
4	Insert data into tables	7-9
5	Running Query	10-16

# **WAREHOUSE MANAGEMENT SYSTEM**

**The information which are going to be available in the database are given below:**

A Warehouse Management System (WMS) is a software application designed to support and optimize warehouse or distribution center management. It facilitates various functions, including inventory management, order fulfillment, shipping and receiving and many more feature within the warehouse.

**Table:** Warehouses

**Attributes:** warehouse\_id, name, location

**Relations:**

1. Warehouses – warehouse\_id – Inventory
2. Warehouses – warehouse\_id – Transfers (as source\_warehouse\_id and destination\_warehouse\_id)
3. Warehouses – warehouse\_id – Orders
4. Warehouses – warehouse\_id – Supply\_Orders

**Table:** Products

**Attributes:** product\_id, name, category, price

**Relations:**

1. Products – product\_id – Inventory
2. Products – product\_id – Transfers
3. Products – product\_id – Orders
4. Products – product\_id – Supply\_Orders

**Table:** Inventory

**Attributes:** inventory\_id, warehouse\_id, product\_id, quantity, last\_updated, flight\_name, flight\_date, plane\_name

**Relations:**

1. Inventory – warehouse\_id – Warehouses
2. Inventory – product\_id – Products

**Table:** Transfers

**Attributes:** transfer\_id, source\_warehouse\_id, destination\_warehouse\_id, product\_id, transfer\_quantity, transfer\_date,

**Relations:**

1. Transfers – source\_warehouse\_id – Warehouses

2. Transfers – destination\_warehouse\_id – Warehouses
3. Transfers – product\_id – Products

**Table:** Suppliers

**Attributes:** supplier\_id, supplier\_name, contact\_info,

**Relations:**

1. Suppliers – supplier\_id – Supply\_Orders

**Table:** Orders

**Attributes:** order\_id, product\_id, warehouse\_id, order\_quantity, order\_date, delivery\_date, customer\_name, customer\_contact,

**Relations:**

1. Orders – product\_id – Products
2. Orders – warehouse\_id – Warehouses

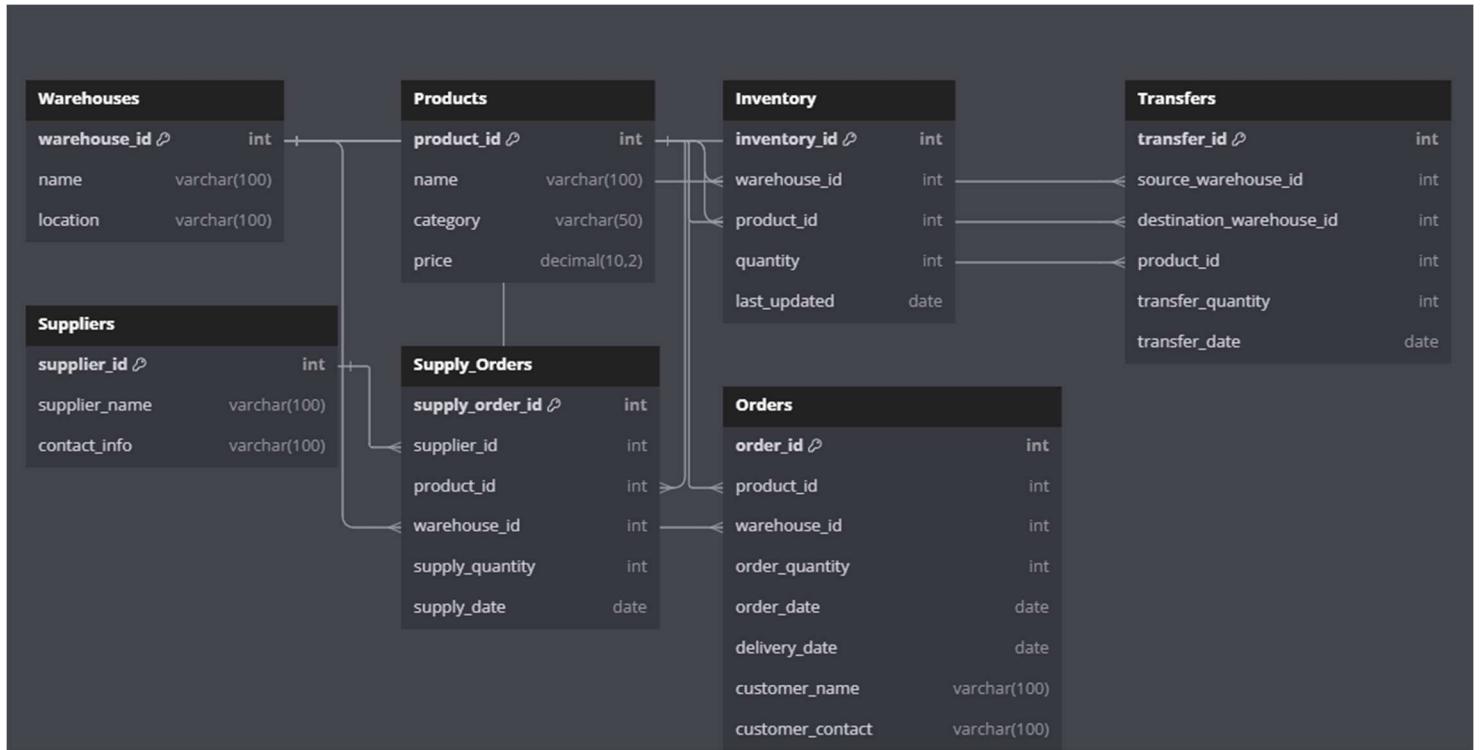
**Table:** Supply\_Orders

**Attributes:** order\_id, product\_id, warehouse\_id, order\_quantity, order\_date, delivery\_date, customer\_name, customer\_contact,

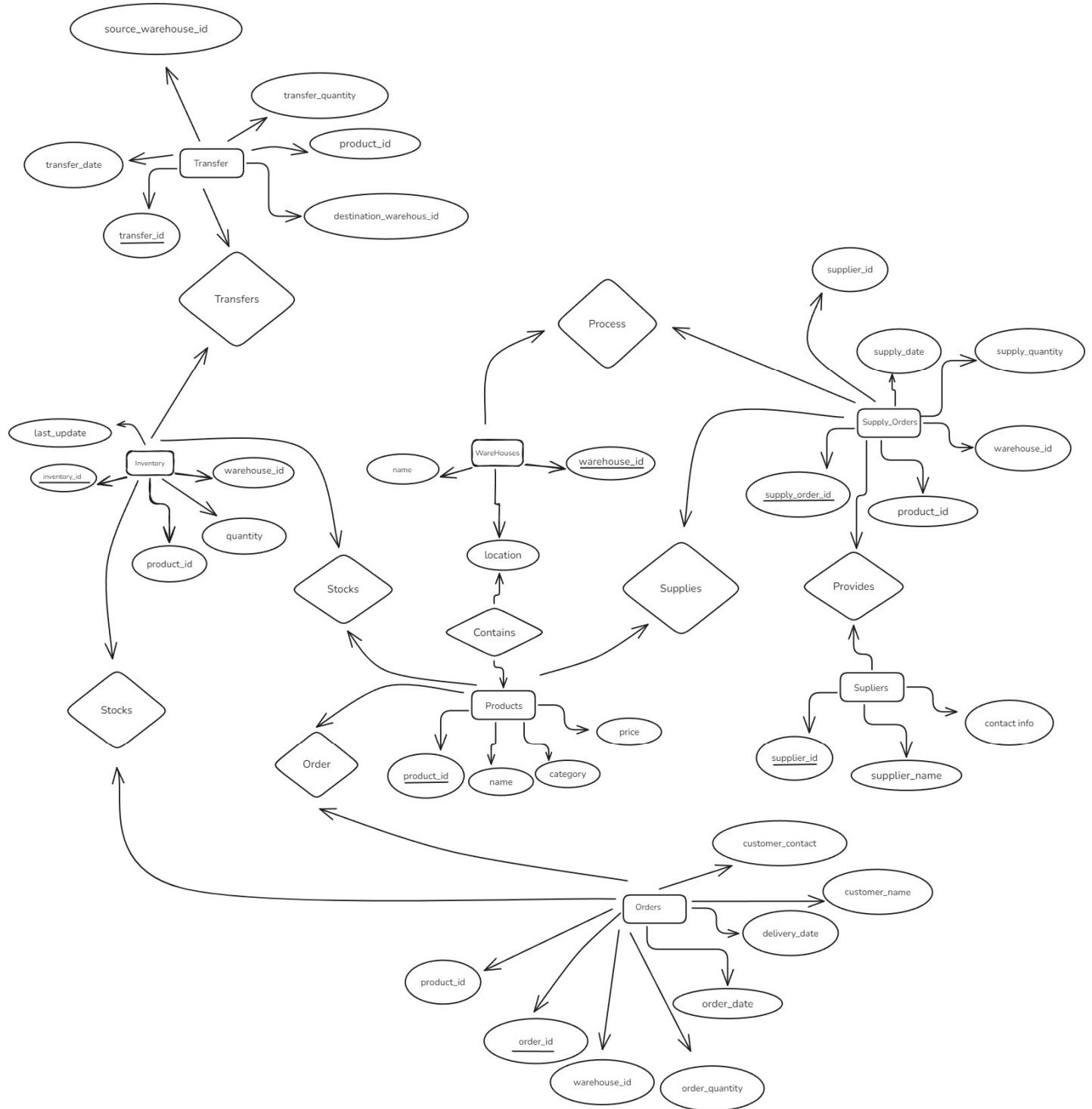
**Relations:**

1. Supply\_Orders – supplier\_id – Suppliers
2. Supply\_Orders – product\_id – Products
3. Supply\_Orders – warehouse\_id – Warehouses

# Schema Diagram For WareHouse Management System



# ER Diagram For WareHouse Management System



# Insert data into tables:

## Table: Warehouses

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "WareHouse\_Values\_Insertion.sql - SIFAT\SQLEXPRESS.WAREHOUSE\_MANAGEMENT (SIFAT\sifat (71))\* - Microsoft SQL Server Management Studio". The Object Explorer on the left shows the database structure under "SIFAT\SQLEXPRESS (SQL Server)". The main pane displays a query window with the following SQL code:

```
USE WAREHOUSE_MANAGEMENT;
INSERT INTO Warehouses (warehouse_id, name, location) VALUES
(1, 'Warehouse Kawla', 'Kawla Airport'),
(2, 'Warehouse Airport', 'Dhaka Airport'),
(3, 'Warehouse Narayanganj', 'Narayanganj'),
(4, 'Warehouse Fatulla', 'Dhaka Fatulla'),
(5, 'Warehouse Uttara', 'Uttara');
select * from Warehouses
```

The results pane shows a table with the inserted data:

	warehouse_id	name	location
1	1	Warehouse Kawla	Kawla Airport
2	2	Warehouse Airport	Dhaka Airport
3	3	Warehouse Narayanganj	Narayanganj
4	4	Warehouse Fatulla	Dhaka Fatulla
5	5	Warehouse Uttara	Uttara

At the bottom of the results pane, a message says "Query executed successfully." and the status bar indicates "SIFAT\SQLEXPRESS (16.0 RTM) | SIFAT\sifat (71) | WAREHOUSE\_MANAGEMENT | 00:00:00 | 5 rows".

## Table: Products

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the connection to SIFAT\SQLEXPRESS.WAREHOUSE\_MANAGEMENT. The main window has three tabs: WareHouse\_Tables.sq... (SIFAT\sifat (59)), WareHouse\_Queries.s... (SIFAT\sifat (73)), and WareHouse\_Values\_In... (SIFAT\sifat (71)\*). The query results tab displays the inserted data:

```
INSERT INTO Products (product_id, name, category, price) VALUES
(1, 'Samsung Galaxy S23', 'Electronics', 999.99),
(2, 'Apple iPhone 14', 'Electronics', 1099.99),
(3, 'Sony WH-1000XM4 Headphones', 'Electronics', 348.00),
(4, 'Instant Pot Duo 7-in-1', 'Home Appliances', 89.99),
(5, 'Dyson V11 Torque Drive', 'Home Appliances', 599.99),
(6, 'IKEA Malm Bed Frame', 'Furniture', 399.00),
(7, 'West Elm Sofa', 'Furniture', 1299.99),
(8, 'Nike Air Max 270', 'Footwear', 149.99),
(9, 'Adidas Ultraboost 21', 'Footwear', 179.99),
(10, 'The North Face Backpack', 'Sports', 89.99),
(11, 'Wilson Tennis Racket', 'Sports', 129.99),
(12, 'Levi's 501 Jeans', 'Clothing', 59.99),
(13, 'H&M Basic T-Shirt', 'Clothing', 14.99),
(14, 'Lego Star Wars Millennium Falcon', 'Toys', 169.99),
(15, 'Barbie Dreamhouse', 'Toys', 199.99),
(16, 'Nivea Moisturizing Cream', 'Beauty', 12.99),
(17, 'Maybelline Mascara', 'Beauty', 9.99),
(18, 'Becoming by Michelle Obama', 'Books', 24.99),
(19, 'The Silent Patient by Alex Michaelides', 'Books', 16.99),
```

The results grid shows 20 rows of product data. The status bar at the bottom indicates "Query executed successfully." and "20 rows".

## Table: Inventory

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the connection to SIFAT\SQLEXPRESS.WAREHOUSE\_MANAGEMENT. The main window has three tabs: WareHouse\_Tables.sq... (SIFAT\sifat (59)), WareHouse\_Queries.s... (SIFAT\sifat (73)), and WareHouse\_Values\_In... (SIFAT\sifat (71)\*). The query results tab displays the inserted data:

```
select * from Products
```

```
INSERT INTO Inventory (inventory_id, warehouse_id, product_id, quantity, last_updated) VALUES
(1, 1, 1, 500, '2024-09-01'),
(2, 1, 2, 300, '2024-09-02'),
(3, 1, 3, 250, '2024-09-03'),
(4, 1, 4, 600, '2024-09-04'),
(5, 1, 5, 700, '2024-09-05'),
(6, 1, 6, 150, '2024-09-06'),
(7, 1, 7, 400, '2024-09-07'),
(8, 2, 1, 800, '2024-09-08'),
(9, 2, 2, 900, '2024-09-09'),
(10, 2, 3, 300, '2024-09-10'),
(11, 2, 4, 350, '2024-09-11'),
```

The results grid shows 15 rows of inventory data. The status bar at the bottom indicates "Query executed successfully." and "34 rows".

## Table: Transfers

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "WareHouse\_Values\_Insertion.sql - SIFAT\SQLEXPRESS.WAREHOUSE\_MANAGEMENT (SIFAT\sifat (71)) - Microsoft SQL Server Management Studio". The Object Explorer on the left shows the connection to "SIFAT\SQLEXPRESS (SQL Server)". The central pane displays a script named "WareHouse\_Tables.sq... (SIFAT\sifat (59))" which contains SQL code for inserting data into the "Transfers" table. The results pane shows the output of the "select \* from Transfers" query, displaying 5 rows of data:

transfer_id	source_warehouse_id	destination_warehouse_id	product_id	transfer_quantity	transfer_date
1	1	2	1	100	2024-09-15
2	2	3	2	150	2024-09-16
3	3	4	3	200	2024-09-17
4	4	5	4	250	2024-09-18
5	5	1	5	300	2024-09-19

The status bar at the bottom indicates "Query executed successfully.", "SIFAT\SQLEXPRESS (16.0 RTM) | SIFAT\sifat (71) | WAREHOUSE\_MANAGEMENT | 00:00:00 | 5 rows".

## Table: Suppliers

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "WareHouse\_Values\_Insertion.sql - SIFAT\SQLEXPRESS.WAREHOUSE\_MANAGEMENT (SIFAT\sifat (71)) - Microsoft SQL Server Management Studio". The Object Explorer on the left shows the connection to "SIFAT\SQLEXPRESS (SQL Server)". The central pane displays a script named "WareHouse\_Tables.sq... (SIFAT\sifat (59))" which contains SQL code for inserting data into the "Suppliers" and "Supply\_Orders" tables. The results pane shows the output of the "select \* from Suppliers" query, displaying 9 rows of data:

supplier_id	supplier_name	contact_info
1	Samsung Electronics	samsung@example.com
2	Apple Inc.	apple@example.com
3	Sony Corp	sony@example.com
4	Dyson Ltd	dyson@example.com
5	Nike Inc.	nike@example.com
6	Adidas AG	adidas@example.com
7	H&M Group	hm@example.com
8	Lego Group	lego@example.com
9	Maybelline	maybelline@example.com

The status bar at the bottom indicates "Query executed successfully.", "SIFAT\SQLEXPRESS (16.0 RTM) | SIFAT\sifat (71) | WAREHOUSE\_MANAGEMENT | 00:00:00 | 9 rows".

## Table: Orders

WareHouse\_Values\_Insertion.sql - SIFAT\SQLEXPRESS.WAREHOUSE\_MANAGEMENT (SIFAT\sifat (71)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

AREHOUSE\_MANAGEMENT Execute New Query

Object Explorer Connect WareHouse\_Tables.sql... (SIFAT\sifat (59)) WareHouse\_Queries.s... (SIFAT\sifat (73)) WareHouse\_Values\_In... (SIFAT\sifat (71))

```
(4, 4, 5, 4, 250, '2024-09-18'),
(5, 5, 1, 5, 300, '2024-09-19');
select * from Transfers
INSERT INTO Orders (order_id, product_id, warehouse_id, order_quantity, order_date, delivery_date, customer_name, customer_contact) VALUES
(1, 1, 1, 50, '2024-09-20', '2024-09-25', 'Alice Johnson', 'alice@example.com'),
(2, 2, 2, 100, '2024-09-21', '2024-09-26', 'Bob Smith', 'bob@example.com'),
(3, 3, 3, 150, '2024-09-22', '2024-09-27', 'Charlie Brown', 'charlie@example.com'),
(4, 4, 4, 200, '2024-09-23', '2024-09-28', 'David Green', 'david@example.com'),
(5, 5, 5, 250, '2024-09-24', '2024-09-29', 'Eva White', 'eva@example.com');
select * from Orders
```

Results Messages

order_id	product_id	warehouse_id	order_quantity	order_date	delivery_date	customer_name	customer_contact
1	1	1	50	2024-09-20	2024-09-25	Alice Johnson	alice@example.com
2	2	2	100	2024-09-21	2024-09-26	Bob Smith	bob@example.com
3	3	3	150	2024-09-22	2024-09-27	Charlie Brown	charlie@example.com
4	4	4	200	2024-09-23	2024-09-28	David Green	david@example.com
5	5	5	250	2024-09-24	2024-09-29	Eva White	eva@example.com

Query executed successfully.

SIFAT\SQLEXPRESS (16.0 RTM) | SIFAT\sifat (71) | WAREHOUSE\_MANAGEMENT | 00:00:00 | 5 rows

Ready Ln 108 Col 1 Ch 1 INS

## Table: Supply Orders:

WareHouse\_Values\_Insertion.sql - SIFAT\SQLEXPRESS.WAREHOUSE\_MANAGEMENT (SIFAT\sifat (71)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

AREHOUSE\_MANAGEMENT Execute New Query

Object Explorer Connect WareHouse\_Tables.sql... (SIFAT\sifat (59)) WareHouse\_Queries.s... (SIFAT\sifat (73)) WareHouse\_Values\_In... (SIFAT\sifat (71))

```
INSERT INTO Supply_Orders (supply_order_id, supplier_id, product_id, warehouse_id, supply_quantity, supply_date) VALUES
(1, 1, 1, 1, 500, '2024-08-30'),
(2, 2, 2, 2, 900, '2024-08-31'),
(3, 3, 3, 3, 600, '2024-09-01'),
(4, 4, 4, 4, 500, '2024-09-02'),
(5, 5, 5, 5, 600, '2024-09-03'),
(6, 6, 6, 2, 750, '2024-09-04'),
(7, 9, 17, 1, 350, '2024-09-05');
select * from Supply_Orders
INSERT INTO Transfers (transfer_id, source_warehouse_id, destination_warehouse_id, product_id, transfer_quantity, transfer_date) VALUES
(1, 1, 2, 1, 100, '2024-09-15'),
(2, 2, 3, 2, 150, '2024-09-16'),
(3, 3, 4, 3, 200, '2024-09-17'),
(4, 4, 5, 4, 250, '2024-09-18');
```

Results Messages

supply_order_id	supplier_id	product_id	warehouse_id	supply_quantity	supply_date
1	1	1	1	500	2024-08-30
2	2	2	2	900	2024-08-31
3	3	3	3	600	2024-09-01
4	4	4	4	500	2024-09-02
5	5	5	5	600	2024-09-03
6	6	6	2	750	2024-09-04
7	9	17	1	350	2024-09-05

Query executed successfully.

SIFAT\SQLEXPRESS (16.0 RTM) | SIFAT\sifat (71) | WAREHOUSE\_MANAGEMENT | 00:00:00 | 7 rows

Ready Ln 95 Col 1 Ch 1 INS

## Running Query:

### 1. Find A Specific Product Name, Quantity, WareHouse Name & Locations.

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure. The main window contains a query editor with the following code:

```
USE WAREHOUSE_MANAGEMENT;

--Find A Specific Product Name, Quantity, WareHouse Name & Locations
SELECT w.name AS WareHouse_Name, p.name AS Product_Name, i.quantity AS Quantity , w.location AS Location
FROM Inventory i
JOIN Warehouses w ON i.warehouse_id = w.warehouse_id
JOIN Products p ON i.product_id = p.product_id
WHERE p.product_id = 1;

-- Deduct Product Quantity From a Specific WareHouse

UPDATE Inventory
SET quantity = quantity - 100, last_updated = GETDATE()
WHERE warehouse_id = 1 AND product_id = 1;
```

The results pane displays a table with the following data:

	WareHouse_Name	Product_Name	Quantity	Location
1	Warehouse Kawla	Samsung Galaxy S23	400	Kawla Airport
2	Warehouse Airport	Samsung Galaxy S23	900	Dhaka Airport
3	Warehouse Narayanganj	Samsung Galaxy S23	450	Narayanganj
4	Warehouse Fatulla	Samsung Galaxy S23	300	Dhaka Fatulla
5	Warehouse Uttra	Samsung Galaxy S23	850	Uttra

At the bottom, a message indicates "Query executed successfully."

### 2. Deduct & Add Product Quantity From a Specific WareHouse

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure. The main window contains a query editor with the following code:

```
WHERE p.product_id = 1;

-- Deduct Product Quantity From a Specific WareHouse

UPDATE Inventory
SET quantity = quantity - 100, last_updated = GETDATE()
WHERE warehouse_id = 1 AND product_id = 1;

--Add Product Quantity to a specific WareHouse

UPDATE Inventory
SET quantity = quantity + 100, last_updated = GETDATE()
WHERE warehouse_id = 2 AND product_id = 1;

--Check stock availability of a specific product in a specific warehouse
```

The results pane displays the messages from the execution:

```
(1 row affected)
(1 row affected)
Completion time: 2024-09-30T03:21:57.1522994+06:00
```

At the bottom, a message indicates "Query executed successfully."

### 3. Check stock availability of a specific product in a specific warehouse

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure under 'SIFAT\SQLEXPRESS (SQL Server)'. The central pane displays a query window titled 'WareHouse\_Questions.s... (SIFAT\sifat (73))' containing the following T-SQL code:

```
--Add Product Quantity to a specific Warehouse
UPDATE Inventory
SET quantity = quantity + 100, last_updated = GETDATE()
WHERE warehouse_id = 2 AND product_id = 1;

--Check stock availability of a specific product in a specific warehouse
SELECT quantity
FROM Inventory
WHERE warehouse_id = 1 AND product_id = 1;

--Deduct Quantity if Stock available
UPDATE Inventory
SET quantity = quantity - 50, last_updated = GETDATE()
```

The 'Results' tab shows the output of the second query:

quantity
300

The status bar at the bottom indicates 'Query executed successfully.' and other session details.

### 4. Stock Level Report.

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure under 'SIFAT\SQLEXPRESS (SQL Server)'. The central pane displays a query window titled 'WareHouse\_Questions.s... (SIFAT\sifat (73))' containing the following T-SQL code:

```
--Stock Levels Report
SELECT w.name AS warehouse_name, p.name AS product_name, i.quantity
FROM Inventory i
JOIN Warehouses w ON i.warehouse_id = w.warehouse_id
JOIN Products p ON i.product_id = p.product_id;
```

The 'Results' tab shows the output of the query, listing 20 rows of stock levels:

	warehouse_name	product_name	quantity
1	Warehouse Kawa	Samsung Galaxy S23	300
2	Warehouse Kawa	Apple iPhone 14	300
3	Warehouse Kawa	Sony WH-1000XM4 Headphones	250
4	Warehouse Kawa	Instant Pot Duo 7-in-1	600
5	Warehouse Kawa	Dyson V11 Torque Drive	700
6	Warehouse Kawa	IKEA Malm Bed Frame	150
7	Warehouse Kawa	West Elm Sofa	400
8	Warehouse Airport	Samsung Galaxy S23	1000
9	Warehouse Airport	Apple iPhone 14	900
10	Warehouse Airport	Sony WH-1000XM4 Headphones	300
11	Warehouse Airport	Instant Pot Duo 7-in-1	350
12	Warehouse Airport	Dyson V11 Torque Drive	200
13	Warehouse Airport	IKEA Malm Bed Frame	650
14	Warehouse Airport	West Elm Sofa	500
15	Warehouse Nara	Samsung Galaxy S23	450
16	Warehouse Nara	Apple iPhone 14	550
17	Warehouse Nara	Sony WH-1000XM4 Headphones	600
18	Warehouse Nara	Instant Pot Duo 7-in-1	250
19	Warehouse Nara	Dyson V11 Torque Drive	750
20	Warehouse Nara	IKEA Malm Bed Frame	350

The status bar at the bottom indicates 'Query executed successfully.' and other session details.

## 5. Product Transformation History & Fulfillment Report.

The screenshot shows the Microsoft SQL Server Management Studio interface with two queries running in separate tabs:

- WareHouse\_Tables.sql... (SIFAT\sisfat (59))**: Contains the first query for Product Transfer History.
- WareHouse\_Questions... (SIFAT\sisfat (73))**: Contains the second query for Order Fulfillment Report.

**--Product Transfer History**

```
SELECT t.transfer_date, ws.name AS source_warehouse, wd.name AS destination_warehouse, t.transfer_quantity
FROM Transfers t
JOIN Warehouses ws ON t.source_warehouse_id = ws.warehouse_id
JOIN Warehouses wd ON t.destination_warehouse_id = wd.warehouse_id
WHERE t.product_id = 1;
```

**--Order Fulfillment Report**

```
SELECT o.order_id, p.name AS product_name, o.order_quantity, o.order_date, o.delivery_date, o.customer_name
FROM Orders o
JOIN Products p ON o.product_id = p.product_id
WHERE o.warehouse_id = 1;
```

**Results** tab (WareHouse\_Tables.sql... (SIFAT\sisfat (59))):

	transfer_date	source_warehouse	destination_warehouse	transfer_quantity
1	2024-09-15	Warehouse Kawla	Warehouse Airport	100

**Results** tab (WareHouse\_Questions... (SIFAT\sisfat (73))):

	order_id	product_name	order_quantity	order_date	delivery_date	customer_name
1	1	Samsung Galaxy S23	50	2024-09-20	2024-09-25	Alice Johnson

Message bar: Query executed successfully.

## 7. Find suppliers for a specific product & Find products supplied to a specific warehouse by a specific supplier.

The screenshot shows the Microsoft SQL Server Management Studio interface with two queries running in separate tabs:

- WareHouse\_Tables.sql... (SIFAT\sisfat (59))**: Contains the first query for finding suppliers for a specific product.
- WareHouse\_Questions... (SIFAT\sisfat (73))**: Contains the second query for finding products supplied to a specific warehouse by a specific supplier.

**--Find suppliers for a specific product**

```
SELECT s.supplier_name, s.contact_info, p.name AS Product_Name
FROM Suppliers s
JOIN Supply_Orders so ON s.supplier_id = so.supplier_id
JOIN Products p ON so.product_id = p.product_id
WHERE p.product_id = 1;
```

**--Find products supplied to a specific warehouse by a specific supplier**

```
SELECT p.name AS Product_Name, w.name AS Warehouse_Name, so.supply_quantity, so.supply_date
FROM Supply_Orders so
JOIN Products p ON so.product_id = p.product_id
JOIN Warehouses w ON so.warehouse_id = w.warehouse_id
WHERE so.supplier_id = 1 AND so.warehouse_id = 1;
```

**Results** tab (WareHouse\_Tables.sql... (SIFAT\sisfat (59))):

	supplier_name	contact_info	Product_Name
1	Samsung Electronics	samsung@example.com	Samsung Galaxy S23

**Results** tab (WareHouse\_Questions... (SIFAT\sisfat (73))):

	Product_Name	Warehouse_Name	supply_quantity	supply_date
1	Samsung Galaxy S23	Warehouse Kawla	500	2024-08-30

Message bar: Query executed successfully.

## 8. Report products with low stock in any warehouse & Find top-selling products based on order quantity

The screenshot shows the Microsoft SQL Server Management Studio interface with two queries in the query editor.

**Query 1: Products with low stock in any warehouse**

```
--Report products with low stock in any warehouse
SELECT p.name AS Product_Name, w.name AS Warehouse_Name, i.quantity
FROM Inventory i
JOIN Products p ON i.product_id = p.product_id
JOIN Warehouses w ON i.warehouse_id = w.warehouse_id
WHERE i.quantity < 100;
```

**Query 2: Find top-selling products based on order quantity**

```
--Find top-selling products based on order quantity
SELECT p.name AS Product_Name, SUM(o.order_quantity) AS Total_Sold
FROM Orders o
JOIN Products p ON o.product_id = p.product_id
GROUP BY p.name
ORDER BY Total_Sold DESC;
```

The results pane displays the output of the second query:

Product_Name	Total_Sold
Dyson V11 Torque Drive	250
Instant Pot Duo 7-in-1	200
Sony WH-1000XM4 Headphones	150
Apple iPhone 14	100
Samsung Galaxy S23	50

Message bar: Query executed successfully.

## 9. Find product transfer history between specific dates

The screenshot shows the Microsoft SQL Server Management Studio interface with a single query for product transfer history.

**Query: Find product transfer history between specific dates**

```
--Find product transfer history between specific dates
SELECT t.transfer_date, ws.name AS Source_Warehouse, wd.name AS Destination_Warehouse, t.transfer_quantity, p.name AS Product_Name
FROM Transfers t
JOIN Warehouses ws ON t.source_warehouse_id = ws.warehouse_id
JOIN Warehouses wd ON t.destination_warehouse_id = wd.warehouse_id
JOIN Products p ON t.product_id = p.product_id
WHERE t.transfer_date BETWEEN '2024-09-01' AND '2024-09-30';
```

**Query: Track order history for products supplied by each supplier**

```
--Track order history for products supplied by each supplier
SELECT s.supplier_name, p.name AS Product_Name, so.supply_quantity, so.supply_date
FROM Supply_Orders so
```

The results pane displays the output of the first part of the query:

transfer_date	Source_Warehouse	Destination_Warehouse	transfer_quantity	Product_Name
2024-09-15	Warehouse Kawla	Warehouse Airport	100	Samsung Galaxy S23
2024-09-16	Warehouse Airport	Warehouse Narayanganj	150	Apple iPhone 14
2024-09-17	Warehouse Narayanganj	Warehouse Fatulla	200	Sony WH-1000XM4 Headphones
2024-09-18	Warehouse Fatulla	Warehouse Uttra	250	Instant Pot Duo 7-in-1
2024-09-19	Warehouse Uttra	Warehouse Kawla	300	Dyson V11 Torque Drive

Message bar: Query executed successfully.

## 10. Track order history for products supplied by each supplier

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure. The main pane displays a T-SQL script for tracking order history. The results pane shows a table with columns: supplier\_name, Product\_Name, supply\_quantity, and supply\_date. The data includes entries for various suppliers like Maybelline, Adidas AG, Nike Inc., Dyson Ltd, Sony Corp, Apple Inc., and Samsung Electronics, with their respective product names, quantities supplied, and dates.

```
FROM Transfers t
JOIN Warehouses ws ON t.source_warehouse_id = ws.warehouse_id
JOIN Warehouses wd ON t.destination_warehouse_id = wd.warehouse_id
JOIN Products p ON t.product_id = p.product_id
WHERE t.transfer_date BETWEEN '2024-09-01' AND '2024-09-30';

--Track order history for products supplied by each supplier
SELECT s.supplier_name, p.name AS Product_Name, so.supply_quantity, so.supply_date
FROM Supply_Orders so
JOIN Suppliers s ON so.supplier_id = s.supplier_id
JOIN Products p ON so.product_id = p.product_id
ORDER BY so.supply_date DESC;

--Summary report for each warehouse with total products and quantities
SELECT w.name AS Warehouse_Name, COUNT(i.product_id) AS Total_Products, SUM(i.quantity) AS Total_Quantity
```

supplier_name	Product_Name	supply_quantity	supply_date
1 Maybelline	Maybelline Mascara	350	2024-09-05
2 Adidas AG	Adidas Ultraboost 21	750	2024-09-04
3 Nike Inc.	Nike Air Max 270	600	2024-09-03
4 Dyson Ltd	Dyson V11 Torque Drive	500	2024-09-02
5 Sony Corp	Sony WH-1000XM4 Headphones	600	2024-09-01
6 Apple Inc.	Apple iPhone 14	900	2024-08-31
7 Samsung Electronics	Samsung Galaxy S23	500	2024-08-30

## 11. Summary report for each warehouse with total products and quantities & Calculate total value of products in each warehouse

The screenshot shows the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure. The main pane displays a T-SQL script for generating a summary report. The results pane shows two tables. The first table, 'Warehouse\_Report', lists warehouses with their names, total products, and total quantity. The second table, 'Warehouse\_Value\_Report', lists warehouses with their names and total inventory value. Both reports group by warehouse name.

```
ORDER BY so.supply_date DESC;

--Summary report for each warehouse with total products and quantities
SELECT w.name AS Warehouse_Name, COUNT(i.product_id) AS Total_Products, SUM(i.quantity) AS Total_Quantity
FROM Inventory i
JOIN Warehouses w ON i.warehouse_id = w.warehouse_id
GROUP BY w.name;

--Calculate total value of products in each warehouse
SELECT w.name AS Warehouse_Name, SUM(i.quantity * p.price) AS Total_Inventory_Value
FROM Inventory i
JOIN Products p ON i.product_id = p.product_id
JOIN Warehouses w ON i.warehouse_id = w.warehouse_id
GROUP BY w.name;
```

Warehouse_Name	Total_Products	Total_Quantity
1 Warehouse Airport	7	3900
2 Warehouse Fatulla	6	2450
3 Warehouse Kawla	7	2700
4 Warehouse Narayanganj	7	3750
5 Warehouse Uttra	7	4350

Warehouse_Name	Total_Inventory_Value
1 Warehouse Airport	3155220.50
2 Warehouse Fatulla	2021583.00
3 Warehouse Kawla	1770827.00
4 Warehouse Narayanganj	2915922.00
5 Warehouse Uttra	3373666.50