

ASSIGNMENT-2

**DATABASE MANAGEMENT SYSTEM
CSA0593**

**SATHISH.R
192324204**

Database Design for Managing Suppliers, Products, Shipments, and Inventory

This system is designed to manage the core components of supply chain operations, including suppliers, products, shipments, and inventory. The database ensures smooth management of supplier relationships, product tracking, inventory levels, and shipment processing.

Database Structure

1. Tables:

Suppliers: This table stores information about the suppliers providing products.

Column Name	Data Type	Description
supplier_id	INT	Primary key, unique supplier ID
name	VARCHAR	Name of the supplier
contact_name	VARCHAR	Contact person's name
phone	VARCHAR	Supplier's contact number
email	VARCHAR	Supplier's email address
address	VARCHAR	Supplier's address

Products: This table holds information about the products supplied by the suppliers.

Column Name	Data Type	Description
product_id	INT	Primary key, unique product ID
name	VARCHAR	Product name
description	TEXT	Description of the product
price	DECIMAL	Price of the product
supplier_id	INT	Foreign key, references Suppliers table
stock_quantity	INT	Quantity of product available in inventory

Shipments: This table tracks the shipment details of products ordered from suppliers.

Column Name	Data Type	Description
shipment_id	INT	Primary key, unique shipment ID
supplier_id	INT	Foreign key, references Suppliers table
shipment_date	TIMESTAMP	Date when the shipment was dispatched
estimated_arrival	TIMESTAMP	Estimated arrival date of the shipment

Column Name	Data Type	Description
shipment_status	VARCHAR	Current status of the shipment (e.g., pending, shipped, delivered)

Inventory: This table manages the current inventory levels of products.

Column Name	Data Type	Description
inventory_id	INT	Primary key, unique inventory ID
product_id	INT	Foreign key, references Products table
quantity	INT	Quantity of the product in inventory
last_updated	TIMESTAMP	Date and time the inventory was last updated

Stored Procedures:

Place Order: This procedure places an order with a supplier for a specific quantity of a product.

```
sql

CREATE PROCEDURE PlaceOrder(IN supplier_id INT, IN product_id INT, IN order_quantity INT)BEGIN
  DECLARE product_price DECIMAL(10, 2);
  DECLARE total_cost DECIMAL(10, 2);

  -- Get product price from Products table
  SELECT price INTO product_price FROM Products WHERE product_id = product_id;

  -- Calculate total cost of the order
  SET total_cost = product_price * order_quantity;

  -- Create shipment record
  INSERT INTO Shipments(supplier_id, shipment_date, estimated_arrival, shipment_status)
  VALUES(supplier_id, NOW(), DATE_ADD(NOW(), INTERVAL 7 DAY), 'Pending');

  -- Insert into inventory (assuming the shipment is not received yet)
  INSERT INTO Inventory(product_id, quantity, last_updated)
  VALUES(product_id, order_quantity, NOW());END;
```

Track Shipment: This procedure tracks the shipment status for a specific shipment ID.

```
Sql

CREATE PROCEDURE TrackShipment(IN shipment_id INT)BEGIN
  SELECT shipment_status, estimated_arrival FROM Shipments
  WHERE shipment_id = shipment_id;END;
```

Triggers:

Update Inventory on Shipment Arrival: This trigger updates the inventory levels when a shipment is marked as delivered.

```
sql

CREATE TRIGGER UpdateInventoryOnShipmentArrival
```

```
AFTER UPDATE ON ShipmentsFOR EACH ROWBEGIN
IF NEW.shipment_status = 'Delivered' THEN
    UPDATE Inventory SET quantity = quantity +
    (SELECT quantity FROM Shipments WHERE shipment_id = NEW.shipment_id)
    WHERE product_id = NEW.product_id;
END IF;END;
```

Update Shipment Status on Delivery: This trigger updates the shipment status when the estimated arrival date has passed.

Sql

```
CREATE TRIGGER UpdateShipmentStatus
AFTER UPDATE ON ShipmentsFOR EACH ROWBEGIN
IF NEW.estimated_arrival <= NOW() THEN
    UPDATE Shipments SET shipment_status = 'Delivered'
    WHERE shipment_id = NEW.shipment_id;
END IF;END;
```

SQL Queries for Analysis:

Inventory Turnover: This query calculates the inventory turnover ratio, which shows how quickly inventory is sold and replaced.

Sql

```
SELECT p.name,
    SUM(i.quantity) AS total_sold,
    AVG(i.quantity) AS average_inventory,
    (SUM(i.quantity) / AVG(i.quantity)) AS inventory_turnoverFROM Inventory iJOIN Products p ON i.product_id =
p.product_idGROUP BY p.product_id;
```

Supplier Performance: This query analyzes supplier performance based on the number of shipments and the average time it takes for delivery.

Sql

```
SELECT s.name AS supplier_name,
    COUNT(sh.shipment_id) AS num_shipments,
    AVG(DATEDIFF(sh.estimated_arrival, sh.shipment_date)) AS avg_delivery_timeFROM Shipments shJOIN Suppliers s ON
sh.supplier_id = s.supplier_idGROUP BY s.supplier_id;
```

Conclusion:

The Supplier, Product, Shipment, and Inventory Management System streamlines supply chain operations by tracking and managing products, shipments, and inventory levels. With the use of stored procedures for placing orders and tracking shipments, as well as triggers to update inventory and shipment statuses in real time, the system ensures efficiency and accuracy. SQL queries for inventory turnover and supplier performance analysis provide valuable insights for optimizing operations. This database design supports informed decision-making and enhances overall supply chain management.