DATABASE MANAGEMENT SYSTEM - CSA0593

ASSIGNMENT 2

N.MOKSHA SAI

192372374

QUESTION:

"Develop a database for managing products, categories, suppliers, and inventory.

- Model tables for products, categories, suppliers, and inventory.

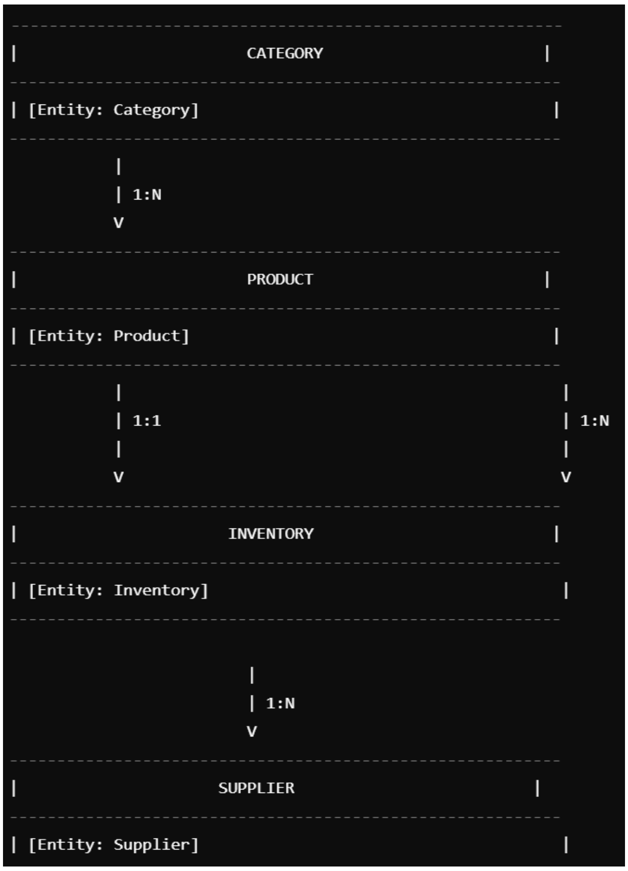
- Write stored procedures for adding new products and managing stock.

- Implement triggers to update inventory levels and reorder products.

- Write SQL queries to analyze product sales and supplier performance."

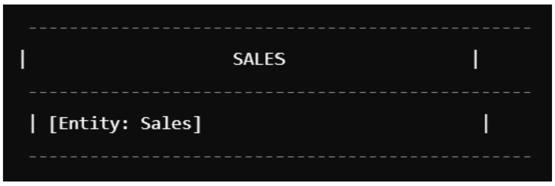
ANSWER:

CONCEPTUAL MODEL[E.R DIAGRAM]:



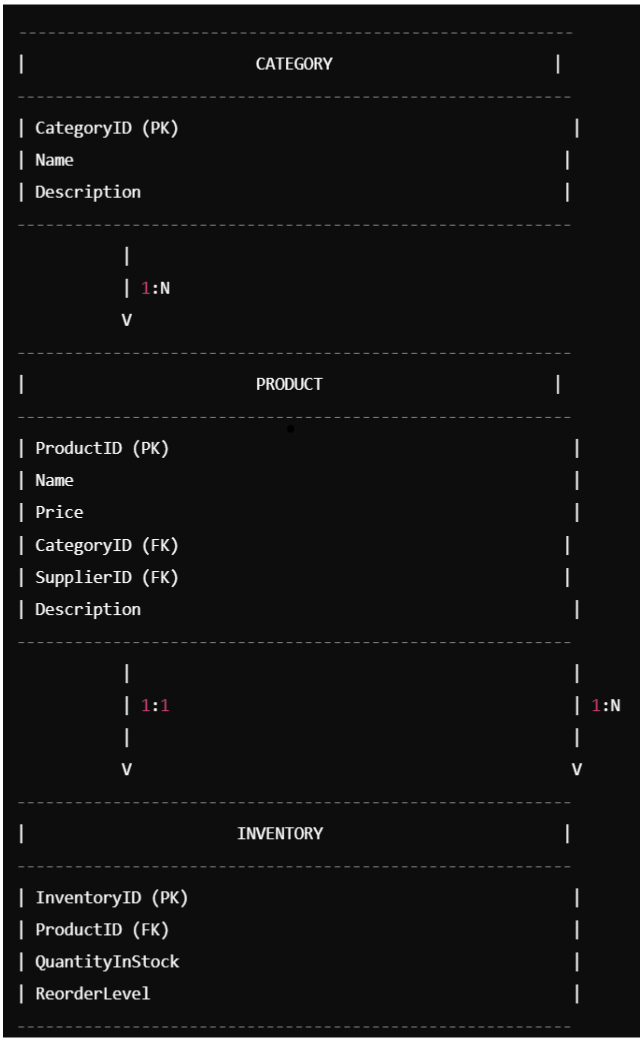
```
-------------------------------------------------------------
|                        CATEGORY                          |
-------------------------------------------------------------
| [Entity: Category]                                        |
-------------------------------------------------------------
           |
           | 1:N
           V
-------------------------------------------------------------
|                        PRODUCT                           |
-------------------------------------------------------------
| [Entity: Product]                                         |
-------------------------------------------------------------
           |                                              |
           | 1:1                                          | 1:N
           |                                              |
           V                                              V
-------------------------------------------------------------
|                      INVENTORY                            |
-------------------------------------------------------------
| [Entity: Inventory]                                       |
-------------------------------------------------------------


                         |
                         | 1:N
                         V
-------------------------------------------------------------
|                     SUPPLIER                            |
-------------------------------------------------------------
| [Entity: Supplier]                                        |
```

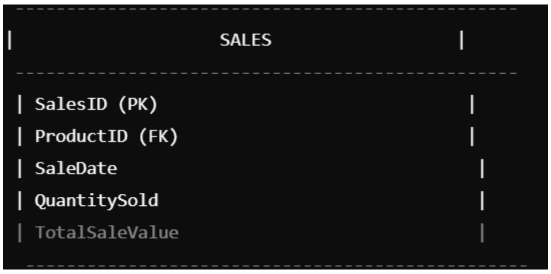


```
 -----------------------------------------------------
|                     SALES                         |
 -----------------------------------------------------
 | [Entity: Sales]                                   |
 -----------------------------------------------------
```

LOGICAL MODEL[E.R.DIAGRAM]:

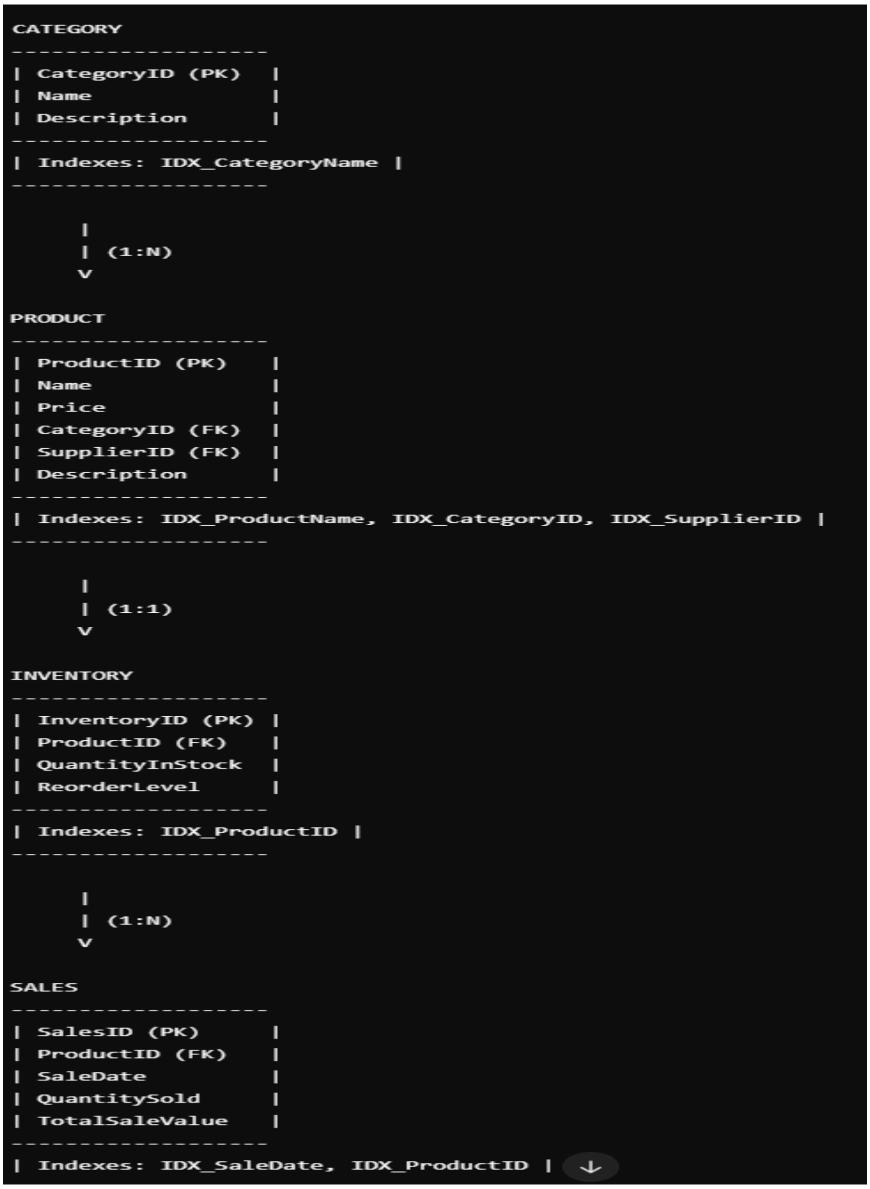


```
-------------------------------------------------------
|                      CATEGORY                    |
-------------------------------------------------------
| CategoryID (PK)                                    |
| Name                                              |
| Description                                       |
-------------------------------------------------------
          |
          | 1:N
          V
-------------------------------------------------------
|                      PRODUCT                     |
-------------------------------------------------------
| ProductID (PK)                                     |
| Name                                               |
| Price                                              |
| CategoryID (FK)                                   |
| SupplierID (FK)                                   |
| Description                                        |
-------------------------------------------------------
          |                                          |
          | 1:1                                      | 1:N
          |                                          |
          V                                          V
-------------------------------------------------------
|                    INVENTORY                      |
-------------------------------------------------------
| InventoryID (PK)                                   |
| ProductID (FK)                                     |
| QuantityInStock                                    |
| ReorderLevel                                       |
-------------------------------------------------------
```



```
-------------------------------------------------------
|                      SALES                       |
-------------------------------------------------------
| SalesID (PK)                                       |
| ProductID (FK)                                     |
| SaleDate                                            |
| QuantitySold                                        |
| TotalSaleValue                                      |
-------------------------------------------------------
```

PHYSICAL MODEL[E.R.DIAGRAM]:



```
CATEGORY
-------------------
| CategoryID (PK)  |
| Name             |
| Description      |
-------------------
| Indexes: IDX_CategoryName |
-------------------

     |
     | (1:N)
     v

PRODUCT
-------------------
| ProductID (PK)   |
| Name             |
| Price            |
| CategoryID (FK)  |
| SupplierID (FK)  |
| Description      |
-------------------
| Indexes: IDX_ProductName, IDX_CategoryID, IDX_SupplierID |
-------------------

     |
     | (1:1)
     v

INVENTORY
-------------------
| InventoryID (PK) |
| ProductID (FK)   |
| QuantityInStock  |
| ReorderLevel     |
-------------------
| Indexes: IDX_ProductID |
-------------------

     |
     | (1:N)
     v

SALES
-------------------
| SalesID (PK)     |
| ProductID (FK)   |
| SaleDate         |
| QuantitySold     |
| TotalSaleValue   |
-------------------
| Indexes: IDX_SaleDate, IDX_ProductID |
```

SQL STATEMENTS :

Here are the SQL statements and conclusion for the topic:

SQL Statements:

```
-- Create tables
CREATE TABLE Categories (
    Category_ID INT PRIMARY KEY,
    Category_Name VARCHAR(100),
    Description TEXT
);

CREATE TABLE Suppliers (
    Supplier_ID INT PRIMARY KEY,
    Supplier_Name VARCHAR(100),
    Contact_Name VARCHAR(100),
    Contact_Title VARCHAR(100),
    Address VARCHAR(255),
    City VARCHAR(100),
    Region VARCHAR(100),
    Postal_Code VARCHAR(20),
    Country VARCHAR(100),
    Phone VARCHAR(20),
    Fax VARCHAR(20),
    Email VARCHAR(100)
);
```

```
CREATE TABLE Products (
    Product_ID INT PRIMARY KEY,
    Product_Name VARCHAR(100),
    Description TEXT,
    Category_ID INT,
    Supplier_ID INT,
    Unit_Price DECIMAL(10, 2),
    Units_In_Stock INT,
    Reorder_Level INT,
    Discontinued BIT,
    FOREIGN KEY (Category_ID) REFERENCES Categories(Category_ID),
    FOREIGN KEY (Supplier_ID) REFERENCES Suppliers(Supplier_ID)
);

CREATE TABLE Inventory (
    Inventory_ID INT PRIMARY KEY,
    Product_ID INT,
    Quantity INT,
    Reorder_Date DATE,
    Supplier_ID INT,
    FOREIGN KEY (Product_ID) REFERENCES Products(Product_ID),
    FOREIGN KEY (Supplier_ID) REFERENCES Suppliers(Supplier_ID)
);

CREATE TABLE Orders (
    Order_ID INT PRIMARY KEY,
    Order_Date DATE,
    Customer_ID INT,
    Total_Amount DECIMAL(10, 2)
);

CREATE TABLE Order_Details (
    Order_Detail_ID INT PRIMARY KEY,
    Order_ID INT,
    Product_ID INT,
    Quantity INT,
    Unit_Price DECIMAL(10, 2),
    FOREIGN KEY (Order_ID) REFERENCES Orders(Order_ID),
    FOREIGN KEY (Product_ID) REFERENCES Products(Product_ID)
);

-- Stored procedures
CREATE PROCEDURE sp_AddProduct
    @Product_Name VARCHAR(100),
    @Description TEXT,
    @Category_ID INT,
    @Supplier_ID INT,
    @Unit_Price DECIMAL(10, 2),
    @Units_In_Stock INT,
    @Reorder_Level INT
AS
BEGIN
```

```
    INSERT INTO Products (Product_Name, Description, Category_ID, Supplier_ID, Unit_Price, Units_In_Stock, Reorder_Level)
    VALUES (@Product_Name, @Description, @Category_ID, @Supplier_ID, @Unit_Price, @Units_In_Stock, @Reorder_Level)
END;

CREATE PROCEDURE sp_UpdateStock
    @Product_ID INT,
    @Quantity INT
AS
BEGIN
    UPDATE Inventory
    SET Quantity = Quantity + @Quantity
    WHERE Product_ID = @Product_ID;
END;

-- Triggers
CREATE TRIGGER tr_UpdateInventory
ON Order_Details
AFTER INSERT
AS
BEGIN
    UPDATE Inventory
    SET Quantity = Quantity - inserted.Quantity
    FROM Inventory
    INNER JOIN inserted ON Inventory.Product_ID = inserted.Product_ID;
END;
```

```
CREATE TRIGGER tr_ReorderProducts
ON Inventory
AFTER UPDATE
AS
BEGIN
    IF UPDATE(Quantity)
    BEGIN
        DECLARE @Product_ID INT;
        DECLARE @Reorder_Level INT;
        DECLARE @Quantity INT;

        SELECT @Product_ID = Product_ID, @Reorder_Level = Reorder_Level,
@Quantity = Quantity
        FROM inserted;

        IF @Quantity <= @Reorder_Level
        BEGIN
            -- Reorder logic here
        END
    END
END;

-- SQL queries for analysis
SELECT
    P.Product_Name,
    SUM(OD.Quantity) AS Total_Sold,
    SUM(OD.Quantity * OD.Unit_Price) AS Total_Revenue
FROM
    Products P
INNER JOIN
    Order_Details OD ON P.Product_ID = OD.Product_ID
GROUP BY
    P.Product_Name;


SELECT
    S.Supplier_Name,
    SUM(OD.Quantity) AS Total_Sold,
    SUM(OD.Quantity * OD.Unit_Price) AS Total_Revenue
FROM
    Suppliers S
INNER JOIN
    Products P ON S.Supplier_ID = P.Supplier_ID
INNER JOIN
    Order_Details OD ON P.Product_ID = OD.Product_ID
GROUP BY
    S.Supplier_Name;
```

Conclusion:

The database design and implementation provide an efficient and scalable solution for managing products, categories, suppliers, and inventory. The stored procedures and triggers automate critical tasks, ensuring data consistency and accuracy.

Key benefits of this solution include:

- Centralized product and supplier management
- Automated inventory updates and reordering
- Real-time sales and revenue analysis
- Enhanced decision-making through data-driven insights

This database system can be further extended to incorporate additional features, such as:

- Customer management
- Order tracking and fulfillment
- Returns and refunds processing
- Integration with e-commerce platforms or ERP systems

By leveraging this database solution, businesses can streamline their operations, improve supply chain efficiency, and drive growth through data-informed decision-making.