Q1: Design a star schema to analyze sales data for a retail company. Include dimension tables for Time, Products, Customers, and Stores. Create a fact table to record sales transactions, including measures such as sales amount, quantity sold, and discount.

STAR

```
-- Step 2: Create Dimension Tables
CREATE TABLE Dim_Time (
    Time_ID INT PRIMARY KEY,
    Date DATE,
    Day VARCHAR(10),
    Month VARCHAR(10),
    Year INT,
    Quarter VARCHAR(10)
);

CREATE TABLE Dim_Products (
    Product_ID INT PRIMARY KEY,
    Product_Name VARCHAR(50),
    Category VARCHAR(50),
    Price DECIMAL(10, 2)
);

CREATE TABLE Dim_Customers (
    Customer_ID INT PRIMARY KEY,
    Customer_Name VARCHAR(50),
    Gender VARCHAR(10),
    Age INT,
    City VARCHAR(50)
);

CREATE TABLE Dim_Stores (
    Store_ID INT PRIMARY KEY,
    Store_Name VARCHAR(50),
    Location VARCHAR(50),
    Manager_Name VARCHAR(50)
);

-- Step 3: Create Fact Table
CREATE TABLE Fact_Sales (
    Transaction_ID INT PRIMARY KEY,
    Time_ID INT,
    Product_ID INT,
    Customer_ID INT,
    Store_ID INT,
    Sales_Amount DECIMAL(10, 2),
    Quantity_Sold INT,
    Discount DECIMAL(10, 2),
    FOREIGN KEY (Time_ID) REFERENCES Dim_Time(Time_ID),
    FOREIGN KEY (Product_ID) REFERENCES Dim_Products(Product_ID),
    FOREIGN KEY (Customer_ID) REFERENCES Dim_Customers(Customer_ID),
    FOREIGN KEY (Store_ID) REFERENCES Dim_Stores(Store_ID)
);

-- Step 4: Insert Data into Dimension Tables
INSERT INTO Dim_Time VALUES (1, '2024-11-18', 'Monday', 'November', 2024, 'Q4');
INSERT INTO Dim_Time VALUES (2, '2024-11-19', 'Tuesday', 'November', 2024, 'Q4');
INSERT INTO Dim_Time VALUES (3, '2024-11-20', 'Wednesday', 'November', 2024, 'Q4');
INSERT INTO Dim_Time VALUES (4, '2024-11-21', 'Thursday', 'November', 2024, 'Q4');
INSERT INTO Dim_Time VALUES (5, '2024-11-22', 'Friday', 'November', 2024, 'Q4');

INSERT INTO Dim_Products VALUES (1, 'Laptop', 'Electronics', 1000.00);
INSERT INTO Dim_Products VALUES (2, 'Headphones', 'Electronics', 100.00);
INSERT INTO Dim_Products VALUES (3, 'Chair', 'Furniture', 50.00);
INSERT INTO Dim_Products VALUES (4, 'Table', 'Furniture', 150.00);
INSERT INTO Dim_Products VALUES (5, 'Smartphone', 'Electronics', 700.00);

INSERT INTO Dim_Customers VALUES (1, 'Alice', 'Female', 30, 'New York');
INSERT INTO Dim_Customers VALUES (2, 'Bob', 'Male', 25, 'Los Angeles');
INSERT INTO Dim_Customers VALUES (3, 'Charlie', 'Male', 35, 'Chicago');
INSERT INTO Dim_Customers VALUES (4, 'Diana', 'Female', 40, 'Houston');
INSERT INTO Dim_Customers VALUES (5, 'Eve', 'Female', 28, 'Seattle');

INSERT INTO Dim_Stores VALUES (1, 'Downtown Store', 'New York', 'John Smith');
INSERT INTO Dim_Stores VALUES (2, 'Mall Outlet', 'Los Angeles', 'Jane Doe');
INSERT INTO Dim_Stores VALUES (3, 'Suburban Branch', 'Chicago', 'Mike Johnson');
INSERT INTO Dim_Stores VALUES (4, 'City Center', 'Houston', 'Emma Brown');
INSERT INTO Dim_Stores VALUES (5, 'Plaza Store', 'Seattle', 'Luke Wilson');

-- Step 5: Insert Data into Fact Table
INSERT INTO Fact_Sales VALUES (1, 1, 1, 1, 1, 900.00, 1, 100.00);
INSERT INTO Fact_Sales VALUES (2, 2, 2, 2, 2, 90.00, 1, 10.00);
INSERT INTO Fact_Sales VALUES (3, 3, 3, 3, 3, 45.00, 1, 5.00);
INSERT INTO Fact_Sales VALUES (4, 4, 4, 4, 4, 135.00, 1, 15.00);
INSERT INTO Fact_Sales VALUES (5, 5, 5, 5, 5, 630.00, 1, 70.00);
INSERT INTO Fact_Sales VALUES (6, 1, 2, 1, 1, 180.00, 2, 20.00);
INSERT INTO Fact_Sales VALUES (7, 2, 3, 2, 2, 90.00, 2, 10.00);
INSERT INTO Fact_Sales VALUES (8, 3, 5, 3, 3, 1400.00, 2, 100.00);
```

```
-- Step 6: Queries

-- Query 1: Total sales amount for each product
SELECT
    P.Product_Name,
    SUM(F.Sales_Amount) AS Total_Sales
FROM
    Fact_Sales F
JOIN
    Dim_Products P ON F.Product_ID = P.Product_ID
GROUP BY
    P.Product_Name;

-- Query 2: Total quantity sold by store
SELECT
    S.Store_Name,
    SUM(F.Quantity_Sold) AS Total_Quantity
FROM
    Fact_Sales F
JOIN
    Dim_Stores S ON F.Store_ID = S.Store_ID
GROUP BY
    S.Store_Name;

-- Query 3: Total discount given per customer
SELECT
    C.Customer_Name,
    SUM(F.Discount) AS Total_Discount
FROM
    Fact_Sales F
JOIN
    Dim_Customers C ON F.Customer_ID = C.Customer_ID
GROUP BY
    C.Customer_Name;

-- Query 4: Sales amount by date
SELECT
    T.Date,
    SUM(F.Sales_Amount) AS Total_Sales
FROM
    Fact_Sales F
JOIN
    Dim_Time T ON F.Time_ID = T.Time_ID
```

```
GROUP BY
   T.Date;

-- Query 5: Total sales by product category
SELECT
   P.Category,
   SUM(F.Sales_Amount) AS Total_Sales
FROM
   Fact_Sales F
JOIN
   Dim_Products P ON F.Product_ID = P.Product_ID
GROUP BY
   P.Category;
```

GALAXY

```
-- Create Dimension Tables
CREATE TABLE Time_Dimension (
   TimeID INTEGER PRIMARY KEY,
   Date DATE NOT NULL,
   Year INT NOT NULL,
   Month INT NOT NULL,
   Day INT NOT NULL
);

CREATE TABLE Product_Dimension (
   ProductID INTEGER PRIMARY KEY,
   ProductName VARCHAR(50),
   Category VARCHAR(50),
   Price DECIMAL
);

CREATE TABLE Customer_Dimension (
   CustomerID INTEGER PRIMARY KEY,
   CustomerName VARCHAR(50),
   Gender VARCHAR(10),
   City VARCHAR(50)
);

CREATE TABLE Store_Dimension (
   StoreID INTEGER PRIMARY KEY,
   StoreName VARCHAR(50),
   Location VARCHAR(50)
);

-- Create Fact Tables
CREATE TABLE Sales_Fact (
   SalesID INTEGER PRIMARY KEY,
   TimeID INT REFERENCES Time_Dimension(TimeID),
   ProductID INT REFERENCES Product_Dimension(ProductID),
   CustomerID INT REFERENCES Customer_Dimension(CustomerID),
   StoreID INT REFERENCES Store_Dimension(StoreID),
   SalesAmount DECIMAL,
   QuantitySold INT,
   Discount DECIMAL
);

CREATE TABLE Inventory_Fact (
   InventoryID INTEGER PRIMARY KEY,
   ProductID INT REFERENCES Product_Dimension(ProductID),
   StoreID INT REFERENCES Store_Dimension(StoreID),
   QuantityInStock INT,
   RestockDate DATE
);

-- Insert Data into Dimension Tables
INSERT INTO Time_Dimension (TimeID, Date, Year, Month, Day) VALUES
(1, '2024-11-01', 2024, 11, 1),
(2, '2024-11-02', 2024, 11, 2),
(3, '2024-11-03', 2024, 11, 3),
(4, '2024-11-04', 2024, 11, 4),
(5, '2024-11-05', 2024, 11, 5);

INSERT INTO Product_Dimension (ProductID, ProductName, Category, Price) VALUES
(1, 'Laptop', 'Electronics', 1500),
(2, 'Smartphone', 'Electronics', 800),
(3, 'Shoes', 'Apparel', 120),
(4, 'Bag', 'Accessories', 60),
(5, 'Watch', 'Accessories', 200);

INSERT INTO Customer_Dimension (CustomerID, CustomerName, Gender, City) VALUES
(1, 'John Doe', 'Male', 'New York'),
(2, 'Jane Smith', 'Female', 'Los Angeles'),
(3, 'Bob Brown', 'Male', 'Chicago'),
(4, 'Alice White', 'Female', 'Houston'),
(5, 'Chris Black', 'Male', 'Phoenix');
```

```sql
INSERT INTO Store_Dimension (StoreID, StoreName, Location) VALUES
(1, 'Store A', 'Downtown'),
(2, 'Store B', 'Uptown'),
(3, 'Store C', 'Suburb'),
(4, 'Store D', 'Mall'),
(5, 'Store E', 'City Center');

-- Insert Data into Fact Tables
INSERT INTO Sales_Fact (SalesID, TimeID, ProductID, CustomerID, StoreID, SalesAmount,
QuantitySold, Discount) VALUES
(1, 1, 1, 1, 1, 1500, 1, 50),
(2, 2, 2, 2, 2, 800, 1, 30),
(3, 3, 3, 3, 3, 240, 2, 10),
(4, 4, 4, 4, 4, 120, 2, 5),
(5, 5, 5, 5, 5, 200, 1, 20);

INSERT INTO Inventory_Fact (InventoryID, ProductID, StoreID, QuantityInStock, RestockDate)
VALUES
(1, 1, 1, 50, '2024-11-01'),
(2, 2, 2, 100, '2024-11-02'),
(3, 3, 3, 200, '2024-11-03'),
(4, 4, 4, 150, '2024-11-04'),
(5, 5, 5, 300, '2024-11-05');

-- Queries
-- 1. Total Sales by Product
SELECT ProductName, SUM(SalesAmount) AS TotalSales
FROM Sales_Fact
JOIN Product_Dimension ON Sales_Fact.ProductID = Product_Dimension.ProductID
GROUP BY ProductName;

-- 2. Inventory Levels for Products
SELECT ProductName, StoreName, QuantityInStock
FROM Inventory_Fact
JOIN Product_Dimension ON Inventory_Fact.ProductID = Product_Dimension.ProductID
JOIN Store_Dimension ON Inventory_Fact.StoreID = Store_Dimension.StoreID;

-- 3. Total Quantity Sold by Store
SELECT StoreName, SUM(QuantitySold) AS TotalQuantity
FROM Sales_Fact
JOIN Store_Dimension ON Sales_Fact.StoreID = Store_Dimension.StoreID
GROUP BY StoreName;

-- 4. Top Customer by Sales
```

```
SELECT CustomerName, SUM(SalesAmount) AS TotalSpent
FROM Sales_Fact
JOIN Customer_Dimension ON Sales_Fact.CustomerID = Customer_Dimension.CustomerID
GROUP BY CustomerName
ORDER BY TotalSpent DESC
LIMIT 1;

-- 5. Total Discount Given
SELECT SUM(Discount) AS TotalDiscount
FROM Sales_Fact;
```

SNOWFLAKE

```
-- Drop tables if they exist for a fresh start
DROP TABLE IF EXISTS Sales_Fact;
DROP TABLE IF EXISTS Time_Dimension;
DROP TABLE IF EXISTS Products_Dimension;
DROP TABLE IF EXISTS Customers_Dimension;
DROP TABLE IF EXISTS Stores_Dimension;

-- Create Dimension Tables
CREATE TABLE Time_Dimension (
    Time_ID INT PRIMARY KEY,
    Year INT,
    Quarter VARCHAR(10),
    Month VARCHAR(20),
    Day INT
);

CREATE TABLE Products_Dimension (
    Product_ID INT PRIMARY KEY,
    Product_Name VARCHAR(100),
    Category VARCHAR(50),
    Subcategory VARCHAR(50),
    Brand VARCHAR(50)
);

CREATE TABLE Customers_Dimension (
    Customer_ID INT PRIMARY KEY,
    Customer_Name VARCHAR(100),
    Gender VARCHAR(10),
    Age_Group VARCHAR(20),
    Location VARCHAR(100)
);
```

```
CREATE TABLE Stores_Dimension (
    Store_ID INT PRIMARY KEY,
    Store_Name VARCHAR(100),
    Region VARCHAR(50),
    Store_Type VARCHAR(50)
);

-- Create Fact Table
CREATE TABLE Sales_Fact (
    Sales_ID INT PRIMARY KEY,
    Time_ID INT,
    Product_ID INT,
    Customer_ID INT,
    Store_ID INT,
    Sales_Amount DECIMAL(10, 2),
    Quantity_Sold INT,
    Discount DECIMAL(10, 2),
    FOREIGN KEY (Time_ID) REFERENCES Time_Dimension(Time_ID),
    FOREIGN KEY (Product_ID) REFERENCES Products_Dimension(Product_ID),
    FOREIGN KEY (Customer_ID) REFERENCES Customers_Dimension(Customer_ID),
    FOREIGN KEY (Store_ID) REFERENCES Stores_Dimension(Store_ID)
);

-- Insert Data into Dimension Tables
INSERT INTO Time_Dimension VALUES
(1, 2024, 'Q1', 'January', 15),
(2, 2024, 'Q1', 'February', 20),
(3, 2024, 'Q1', 'March', 5),
(4, 2024, 'Q2', 'April', 10),
(5, 2024, 'Q2', 'May', 18);

INSERT INTO Products_Dimension VALUES
(1, 'Laptop', 'Electronics', 'Computers', 'Brand A'),
(2, 'Smartphone', 'Electronics', 'Phones', 'Brand B'),
(3, 'Shoes', 'Apparel', 'Footwear', 'Brand C'),
(4, 'Bag', 'Apparel', 'Accessories', 'Brand D'),
(5, 'Watch', 'Accessories', 'Wristwear', 'Brand E');

INSERT INTO Customers_Dimension VALUES
(1, 'Alice', 'Female', '25-34', 'New York'),
(2, 'Bob', 'Male', '35-44', 'Los Angeles'),
(3, 'Charlie', 'Male', '18-24', 'Chicago'),
(4, 'Diana', 'Female', '45-54', 'Houston'),
(5, 'Eva', 'Female', '25-34', 'Miami');

INSERT INTO Stores_Dimension VALUES
(1, 'Downtown Store', 'East', 'Retail'),
(2, 'Suburban Store', 'West', 'Retail'),
(3, 'Mall Outlet', 'North', 'Retail'),
(4, 'Airport Store', 'South', 'Specialty'),
(5, 'Online Store', 'Global', 'Online');

-- Insert Data into Fact Table
INSERT INTO Sales_Fact VALUES
(1, 1, 1, 1, 1, 1500.00, 1, 50.00),
(2, 2, 2, 2, 2, 799.99, 1, 30.00),
(3, 3, 3, 3, 3, 59.99, 2, 10.00),
(4, 4, 4, 4, 4, 120.00, 1, 20.00),
(5, 5, 5, 5, 5, 300.00, 1, 25.00);

-- Example Queries

-- Query 1: Total sales amount by product category
SELECT
    p.Category,
    SUM(s.Sales_Amount) AS Total_Sales
FROM Sales_Fact s
JOIN Products_Dimension p ON s.Product_ID = p.Product_ID
GROUP BY p.Category;

-- Query 2: Total sales by region
SELECT
    st.Region,
    SUM(s.Sales_Amount) AS Total_Sales
FROM Sales_Fact s
JOIN Stores_Dimension st ON s.Store_ID = st.Store_ID
GROUP BY st.Region;

-- Query 3: Total quantity sold for each product
SELECT
    p.Product_Name,
    SUM(s.Quantity_Sold) AS Total_Quantity
FROM Sales_Fact s
JOIN Products_Dimension p ON s.Product_ID = p.Product_ID
GROUP BY p.Product_Name;

-- Query 4: Sales amount for each store type
```

```
SELECT
   st.Store_Type,
   SUM(s.Sales_Amount) AS Total_Sales
FROM Sales_Fact s
JOIN Stores_Dimension st ON s.Store_ID = st.Store_ID
GROUP BY st.Store_Type;

-- Query 5: Average discount given to customers in different age groups
SELECT
   c.Age_Group,
   AVG(s.Discount) AS Average_Discount
FROM Sales_Fact s
JOIN Customers_Dimension c ON s.Customer_ID = c.Customer_ID
GROUP BY c.Age_Group;
```

Q2:Consider an order management operational database that tracks order numbers, dates, the requested ship dates, customers and their shipping and billing addresses, products and their quantity and gross dollar amount, sales representatives that take and process orders, the deals (promotions) and discounts proposed/offered to customers. You have to design a data warehouse that will be updated from the above operational database and should support decision making by helping to answer analytical questions about the net order dollar amounts per customer, products, promotions or deals, and the performance of their sales representatives or agents.
Analysis of requested ship dates is important for analysis as well. It is also important to allow for performing order amount analysis in various currencies: dollars, dirhams, euros.
1) Draw the star schema(s) showing the main attributes, including primary keys, foreign keys. and facts.
2) Insert appropriate values in the database. Write one SQL statement that runs on your schema and returns the net order dollar amount per customer, products, promotions, and performance of sales representatives.
3) Make necessary assumptions to compute an approximate size (in MB) of your DW over a period of 5 years.

STAR:

```
-- Customers Dimension Table
CREATE TABLE Dim_Customers (
   Customer_ID INT PRIMARY KEY,
   Customer_Name VARCHAR(100),
   Billing_Address VARCHAR(255),
   Shipping_Address VARCHAR(255)
```

```
);

-- Products Dimension Table
CREATE TABLE Dim_Products (
    Product_ID INT PRIMARY KEY,
    Product_Name VARCHAR(100),
    Product_Category VARCHAR(50)
);

-- Sales Representatives Dimension Table
CREATE TABLE Dim_Sales_Representatives (
    Sales_Rep_ID INT PRIMARY KEY,
    Sales_Rep_Name VARCHAR(100)
);

-- Promotions Dimension Table
CREATE TABLE Dim_Promotions (
    Promotion_ID INT PRIMARY KEY,
    Promotion_Name VARCHAR(100),
    Discount_Percentage DECIMAL(5, 2)
);

-- Time Dimension Table
CREATE TABLE Dim_Time (
    Time_ID INT PRIMARY KEY,
    Date DATE,
    Day VARCHAR(10),
    Month VARCHAR(10),
    Year INT,
    Quarter VARCHAR(10)
);

-- Step 3: Create Fact Table
CREATE TABLE Fact_Orders (
    Order_ID INT PRIMARY KEY,
    Customer_ID INT,
    Product_ID INT,
    Sales_Rep_ID INT,
    Promotion_ID INT,
    Order_Date_ID INT,
    Ship_Date_ID INT,
    Order_Amount DECIMAL(10, 2),
    Quantity INT,
    Currency VARCHAR(3),
    Discount DECIMAL(10, 2),
    FOREIGN KEY (Customer_ID) REFERENCES Dim_Customers(Customer_ID),
    FOREIGN KEY (Product_ID) REFERENCES Dim_Products(Product_ID),
    FOREIGN KEY (Sales_Rep_ID) REFERENCES
Dim_Sales_Representatives(Sales_Rep_ID),
    FOREIGN KEY (Promotion_ID) REFERENCES Dim_Promotions(Promotion_ID),
    FOREIGN KEY (Order_Date_ID) REFERENCES Dim_Time(Time_ID),
    FOREIGN KEY (Ship_Date_ID) REFERENCES Dim_Time(Time_ID)
);

-- Step 4: Insert Data into Dimension Tables

-- Dim_Customers
INSERT INTO Dim_Customers VALUES (1, 'John Doe', '123 Main St, NY', '456 Oak Rd, NY');
INSERT INTO Dim_Customers VALUES (2, 'Jane Smith', '789 Elm St, LA', '101 Pine St, LA');
INSERT INTO Dim_Customers VALUES (3, 'Alice Johnson', '202 Maple St, SF', '303 Birch Rd,
SF');
INSERT INTO Dim_Customers VALUES (4, 'Bob Brown', '404 Cedar St, Miami', '505 Pine Ave,
Miami');
INSERT INTO Dim_Customers VALUES (5, 'Charlie White', '606 Willow St, Chicago', '707 Ash
St, Chicago');

-- Dim_Products
INSERT INTO Dim_Products VALUES (1, 'Laptop', 'Electronics');
INSERT INTO Dim_Products VALUES (2, 'Smartphone', 'Electronics');
INSERT INTO Dim_Products VALUES (3, 'Tablet', 'Electronics');
INSERT INTO Dim_Products VALUES (4, 'Desk', 'Furniture');
INSERT INTO Dim_Products VALUES (5, 'Chair', 'Furniture');

-- Dim_Sales_Representatives
INSERT INTO Dim_Sales_Representatives VALUES (1, 'Sarah Lee');
INSERT INTO Dim_Sales_Representatives VALUES (2, 'Michael Scott');
INSERT INTO Dim_Sales_Representatives VALUES (3, 'David Kim');
INSERT INTO Dim_Sales_Representatives VALUES (4, 'Emma Watson');
INSERT INTO Dim_Sales_Representatives VALUES (5, 'Luke Turner');

-- Dim_Promotions
INSERT INTO Dim_Promotions VALUES (1, 'Summer Sale', 10.00);
INSERT INTO Dim_Promotions VALUES (2, 'Black Friday Deal', 20.00);
INSERT INTO Dim_Promotions VALUES (3, 'Clearance Discount', 15.00);
INSERT INTO Dim_Promotions VALUES (4, 'Holiday Special', 25.00);
INSERT INTO Dim_Promotions VALUES (5, 'Flash Sale', 5.00);

-- Dim_Time (using example dates)
```

```sql
INSERT INTO Dim_Time VALUES (1, '2024-01-01', 'Monday', 'January', 2024, 'Q1');
INSERT INTO Dim_Time VALUES (2, '2024-02-01', 'Thursday', 'February', 2024, 'Q1');
INSERT INTO Dim_Time VALUES (3, '2024-03-01', 'Friday', 'March', 2024, 'Q1');
INSERT INTO Dim_Time VALUES (4, '2024-04-01', 'Monday', 'April', 2024, 'Q2');
INSERT INTO Dim_Time VALUES (5, '2024-05-01', 'Tuesday', 'May', 2024, 'Q2');

-- Step 5: Insert Data into Fact Table
INSERT INTO Fact_Orders VALUES (1, 1, 1, 1, 1, 1, 1, 900.00, 1, 'USD', 50.00);
INSERT INTO Fact_Orders VALUES (2, 2, 2, 2, 2, 2, 2, 400.00, 2, 'USD', 40.00);
INSERT INTO Fact_Orders VALUES (3, 3, 3, 3, 3, 3, 3, 500.00, 1, 'EUR', 30.00);
INSERT INTO Fact_Orders VALUES (4, 4, 4, 4, 4, 4, 4, 300.00, 1, 'AED', 20.00);
INSERT INTO Fact_Orders VALUES (5, 5, 5, 5, 5, 5, 5, 600.00, 3, 'USD', 25.00);
INSERT INTO Fact_Orders VALUES (6, 1, 2, 1, 2, 1, 2, 350.00, 2, 'EUR', 15.00);
INSERT INTO Fact_Orders VALUES (7, 2, 1, 2, 3, 2, 3, 450.00, 1, 'USD', 5.00);
INSERT INTO Fact_Orders VALUES (8, 3, 4, 3, 4, 3, 4, 200.00, 1, 'AED', 10.00);

-- Step 6: Perform Queries

-- Query 1: Net Order Dollar Amount per Customer
SELECT
   C.Customer_Name,
   SUM(F.Order_Amount - F.Discount) AS Net_Order_Amount
FROM
   Fact_Orders F
JOIN
   Dim_Customers C ON F.Customer_ID = C.Customer_ID
GROUP BY
   C.Customer_Name;

-- Query 2: Net Order Dollar Amount per Product
SELECT
   P.Product_Name,
   SUM(F.Order_Amount - F.Discount) AS Net_Order_Amount
FROM
   Fact_Orders F
JOIN
   Dim_Products P ON F.Product_ID = P.Product_ID
GROUP BY
   P.Product_Name;

-- Query 3: Net Order Dollar Amount per Promotion
SELECT
   PR.Promotion_Name,
   SUM(F.Order_Amount - F.Discount) AS Net_Order_Amount
```

```
FROM
   Fact_Orders F
JOIN
   Dim_Promotions PR ON F.Promotion_ID = PR.Promotion_ID
GROUP BY
   PR.Promotion_Name;

-- Query 4: Net Order Dollar Amount per Sales Representative
SELECT
   SR.Sales_Rep_Name,
   SUM(F.Order_Amount - F.Discount) AS Net_Order_Amount
FROM
   Fact_Orders F
JOIN
   Dim_Sales_Representatives SR ON F.Sales_Rep_ID = SR.Sales_Rep_ID
GROUP BY
   SR.Sales_Rep_Name;

-- Query 5: Net Order Dollar Amount per Currency (USD, EUR, AED)
SELECT
   F.Currency,
   SUM(F.Order_Amount - F.Discount) AS Net_Order_Amount
FROM
   Fact_Orders F
GROUP BY
   F.Currency;
```

SNOWFLAKE

```
-- Snowflake Model: Table Creation
CREATE TABLE DimCustomer (
   customer_id INT PRIMARY KEY,
   customer_name VARCHAR(100),
   shipping_address VARCHAR(255),
   billing_address VARCHAR(255)
);

CREATE TABLE DimProduct (
   product_id INT PRIMARY KEY,
   product_name VARCHAR(100),
   category VARCHAR(100),
   price DECIMAL(10, 2)
);
```

```
CREATE TABLE DimPromotion (
    promo_id INT PRIMARY KEY,
    promo_name VARCHAR(100),
    discount_percentage DECIMAL(5, 2)
);

CREATE TABLE DimSalesRep (
    sales_rep_id INT PRIMARY KEY,
    sales_rep_name VARCHAR(100),
    region VARCHAR(100)
);

CREATE TABLE FactOrder (
    order_id INT PRIMARY KEY,
    customer_id INT,
    product_id INT,
    promo_id INT,
    sales_rep_id INT,
    gross_amount DECIMAL(10, 2),
    discount_amount DECIMAL(10, 2),
    net_amount DECIMAL(10, 2),
    currency VARCHAR(10),
    requested_ship_date DATE,
    FOREIGN KEY (customer_id) REFERENCES DimCustomer(customer_id),
    FOREIGN KEY (product_id) REFERENCES DimProduct(product_id),
    FOREIGN KEY (promo_id) REFERENCES DimPromotion(promo_id),
    FOREIGN KEY (sales_rep_id) REFERENCES DimSalesRep(sales_rep_id)
);

-- Snowflake Model: Data Insertion
INSERT INTO DimCustomer VALUES
(1, 'Alice', '123 Elm St, NY', '789 Oak St, NY'),
(2, 'Bob', '456 Maple St, CA', '321 Pine St, CA'),
(3, 'Charlie', '789 Birch St, TX', '654 Cedar St, TX');

INSERT INTO DimProduct VALUES
(1, 'Laptop', 'Electronics', 1000.00),
(2, 'Smartphone', 'Electronics', 800.00),
(3, 'Tablet', 'Electronics', 600.00);

INSERT INTO DimPromotion VALUES
(1, 'Holiday Sale', 10.00),
(2, 'Clearance Sale', 20.00),
(3, 'Weekend Deal', 15.00);

INSERT INTO DimSalesRep VALUES
(1, 'John', 'East'),
(2, 'Jane', 'West'),
(3, 'Jack', 'South');

INSERT INTO FactOrder VALUES
(1, 1, 1, 1, 1, 1000.00, 100.00, 900.00, 'USD', '2024-11-15'),
(2, 2, 2, 2, 2, 800.00, 160.00, 640.00, 'USD', '2024-11-16'),
(3, 3, 3, 3, 3, 600.00, 90.00, 510.00, 'USD', '2024-11-17'),
(4, 1, 2, 1, 2, 800.00, 80.00, 720.00, 'USD', '2024-11-18'),
(5, 2, 1, 3, 1, 1000.00, 150.00, 850.00, 'USD', '2024-11-19');

-- Snowflake Model: Queries
-- Query 1: Total Net Order Amount per Customer
SELECT
    c.customer_name,
    SUM(f.net_amount) AS total_net_amount
FROM FactOrder f
JOIN DimCustomer c ON f.customer_id = c.customer_id
GROUP BY c.customer_name;

-- Query 2: Total Gross and Discount Amounts per Promotion
SELECT
    promo.promo_name,
    SUM(f.gross_amount) AS total_gross_amount,
    SUM(f.discount_amount) AS total_discount_amount
FROM FactOrder f
JOIN DimPromotion promo ON f.promo_id = promo.promo_id
GROUP BY promo.promo_name;

-- Query 3: Performance of Sales Representatives
SELECT
    sr.sales_rep_name,
    COUNT(f.order_id) AS total_orders_handled,
    SUM(f.net_amount) AS total_sales
FROM FactOrder f
JOIN DimSalesRep sr ON f.sales_rep_id = sr.sales_rep_id
GROUP BY sr.sales_rep_name;

-- Query 4: Sales by Product Category
SELECT
    p.category,
    SUM(f.net_amount) AS total_net_sales
FROM FactOrder f
JOIN DimProduct p ON f.product_id = p.product_id
GROUP BY p.category;

-- Query 5: Total Sales Amount by Currency
SELECT
    f.currency,
    SUM(f.net_amount) AS total_net_sales
FROM FactOrder f
GROUP BY f.currency;
```

GALAXY

```
-- Galaxy Model: Table Creation
CREATE TABLE FactOrder (
    order_id INT PRIMARY KEY,
    customer_id INT,
    product_id INT,
    promo_id INT,
    sales_rep_id INT,
    gross_amount DECIMAL(10, 2),
    discount_amount DECIMAL(10, 2),
    net_amount DECIMAL(10, 2),
    currency VARCHAR(10),
    requested_ship_date DATE
);

CREATE TABLE DimCustomer (
    customer_id INT PRIMARY KEY,
    customer_name VARCHAR(100),
    shipping_address VARCHAR(255),
    billing_address VARCHAR(255)
);

CREATE TABLE DimProduct (
    product_id INT PRIMARY KEY,
    product_name VARCHAR(100),
    category VARCHAR(100),
    price DECIMAL(10, 2)
);

CREATE TABLE DimPromotion (
    promo_id INT PRIMARY KEY,
    promo_name VARCHAR(100),
    discount_percentage DECIMAL(5, 2)
);

CREATE TABLE DimSalesRep (
    sales_rep_id INT PRIMARY KEY,
    sales_rep_name VARCHAR(100),
    region VARCHAR(100)
);

-- Galaxy Model: Data Insertion
INSERT INTO DimCustomer VALUES
(1, 'Alice', '123 Elm St, NY', '789 Oak St, NY'),
(2, 'Bob', '456 Maple St, CA', '321 Pine St, CA'),
(3, 'Charlie', '789 Birch St, TX', '654 Cedar St, TX');

INSERT INTO DimProduct VALUES
(1, 'Laptop', 'Electronics', 1000.00),
(2, 'Smartphone', 'Electronics', 800.00),
(3, 'Tablet', 'Electronics', 600.00);

INSERT INTO DimPromotion VALUES
(1, 'Holiday Sale', 10.00),
(2, 'Clearance Sale', 20.00),
(3, 'Weekend Deal', 15.00);

INSERT INTO DimSalesRep VALUES
(1, 'John', 'East'),
(2, 'Jane', 'West'),
(3, 'Jack', 'South');

INSERT INTO FactOrder VALUES
(1, 1, 1, 1, 1, 1000.00, 100.00, 900.00, 'USD', '2024-11-15'),
(2, 2, 2, 2, 2, 800.00, 160.00, 640.00, 'USD', '2024-11-16'),
(3, 3, 3, 3, 3, 600.00, 90.00, 510.00, 'USD', '2024-11-17'),
(4, 1, 2, 1, 2, 800.00, 80.00, 720.00, 'USD', '2024-11-18'),
(5, 2, 1, 3, 1, 1000.00, 150.00, 850.00, 'USD', '2024-11-19');

-- Galaxy Model: Queries
-- Query 1: Total Orders and Net Sales per Customer
SELECT
    c.customer_name,
    COUNT(f.order_id) AS total_orders,
    SUM(f.net_amount) AS total_net_sales
```

```
FROM FactOrder f
JOIN DimCustomer c ON f.customer_id = c.customer_id
GROUP BY c.customer_name;

-- Query 2: Gross Sales by Product
SELECT
    p.product_name,
    SUM(f.gross_amount) AS total_gross_sales
FROM FactOrder f
JOIN DimProduct p ON f.product_id = p.product_id
GROUP BY p.product_name;

-- Query 3: Discount Utilization by Promotion
SELECT
    promo.promo_name,
    COUNT(f.order_id) AS total_orders_with_promo,
    SUM(f.discount_amount) AS total_discounts_given
FROM FactOrder f
JOIN DimPromotion promo ON f.promo_id = promo.promo_id
GROUP BY promo.promo_name;

-- Query 4: Total Orders by Requested Ship Date
SELECT
    f.requested_ship_date,
    COUNT(f.order_id) AS total_orders
FROM FactOrder f
GROUP BY f.requested_ship_date
ORDER BY f.requested_ship_date;

-- Query 5: Sales Performance of Sales Representatives by Region
SELECT
    sr.region,
    sr.sales_rep_name,
    SUM(f.net_amount) AS total_sales
FROM FactOrder f
JOIN DimSalesRep sr ON f.sales_rep_id = sr.sales_rep_id
GROUP BY sr.region, sr.sales_rep_name
ORDER BY sr.region, total_sales DESC;
```

Q3:Consider a book management operational database that tracks different book types, cost, quantity, profit, location, authors and their age and country, publication name and

their country and year. You have to design a data warehouse that will be updated from the above operational database and should support decision making by helping to answer analytical questions about the quantity and profit made per book type, location, author, and publication.

1) Draw the star schema(s) showing the main attributes, including primary keys, foreign keys, and facts.

2) Insert appropriate values in the database. Write SQL statements that runs on your schema and returns the quantity and profit made per book type, location, author, and publication.

STAR** **USED cause a Star Schema is designed to handle analytical queries effectively by organizing facts (measurements) around central fact tables, with surrounding dimension tables. In this case, we have entities like book types, cost, quantity, profit, location, authors, and publication that can be categorized as dimensions. The fact is related to sales or performance (profit, quantity), which is connected to the dimensions.

```
-- Book Types Dimension Table
CREATE TABLE Dim_Book_Types (
   Book_Type_ID INT PRIMARY KEY,
   Book_Type_Name VARCHAR(100)
);

-- Locations Dimension Table
CREATE TABLE Dim_Locations (
   Location_ID INT PRIMARY KEY,
   Location_Name VARCHAR(100)
);

-- Authors Dimension Table
CREATE TABLE Dim_Authors (
   Author_ID INT PRIMARY KEY,
   Author_Name VARCHAR(100),
   Author_Age INT,
   Author_Country VARCHAR(100)
);

-- Publications Dimension Table
CREATE TABLE Dim_Publications (
   Publication_ID INT PRIMARY KEY,
   Publication_Name VARCHAR(100),
   Publication_Country VARCHAR(100),
   Publication_Year INT
);

-- Step 3: Create Fact Table
CREATE TABLE Fact_Books (
   Fact_ID INT PRIMARY KEY,
   Book_Type_ID INT,
   Location_ID INT,
   Author_ID INT,
   Publication_ID INT,
   Quantity_Sold INT,
   Profit DECIMAL(10, 2),
   Book_Cost DECIMAL(10, 2),
   FOREIGN KEY (Book_Type_ID) REFERENCES Dim_Book_Types(Book_Type_ID),
   FOREIGN KEY (Location_ID) REFERENCES Dim_Locations(Location_ID),
   FOREIGN KEY (Author_ID) REFERENCES Dim_Authors(Author_ID),
   FOREIGN KEY (Publication_ID) REFERENCES Dim_Publications(Publication_ID)
);

-- Step 4: Insert Data into Dimension Tables

-- Dim_Book_Types
INSERT INTO Dim_Book_Types VALUES (1, 'Fiction');
INSERT INTO Dim_Book_Types VALUES (2, 'Non-Fiction');
INSERT INTO Dim_Book_Types VALUES (3, 'Science');
INSERT INTO Dim_Book_Types VALUES (4, 'History');
INSERT INTO Dim_Book_Types VALUES (5, 'Fantasy');

-- Dim_Locations
INSERT INTO Dim_Locations VALUES (1, 'New York');
INSERT INTO Dim_Locations VALUES (2, 'London');
INSERT INTO Dim_Locations VALUES (3, 'Paris');
INSERT INTO Dim_Locations VALUES (4, 'Berlin');
INSERT INTO Dim_Locations VALUES (5, 'Tokyo');

-- Dim_Authors
INSERT INTO Dim_Authors VALUES (1, 'J.K. Rowling', 58, 'United Kingdom');
INSERT INTO Dim_Authors VALUES (2, 'George Orwell', 46, 'United Kingdom');
INSERT INTO Dim_Authors VALUES (3, 'Albert Einstein', 76, 'Germany');
INSERT INTO Dim_Authors VALUES (4, 'Isaac Newton', 84, 'United Kingdom');
INSERT INTO Dim_Authors VALUES (5, 'Stephen Hawking', 76, 'United Kingdom');

-- Dim_Publications
INSERT INTO Dim_Publications VALUES (1, 'Bloomsbury', 'United Kingdom', 1997);
INSERT INTO Dim_Publications VALUES (2, 'Penguin Books', 'United Kingdom', 1949);
INSERT INTO Dim_Publications VALUES (3, 'Princeton University Press', 'USA', 1921);
INSERT INTO Dim_Publications VALUES (4, 'Cambridge University Press', 'UK', 1687);
INSERT INTO Dim_Publications VALUES (5, 'HarperCollins', 'USA', 1988);

-- Step 5: Insert Data into Fact Table
INSERT INTO Fact_Books VALUES (1, 1, 1, 1, 1, 1500, 20000.00, 10.00);
INSERT INTO Fact_Books VALUES (2, 2, 2, 2, 2, 1000, 12000.00, 8.00);
INSERT INTO Fact_Books VALUES (3, 3, 3, 3, 3, 500, 5000.00, 15.00);
INSERT INTO Fact_Books VALUES (4, 4, 4, 4, 4, 700, 7000.00, 12.00);
INSERT INTO Fact_Books VALUES (5, 5, 5, 5, 5, 800, 8000.00, 18.00);
INSERT INTO Fact_Books VALUES (6, 1, 2, 1, 2, 1200, 18000.00, 9.00);
INSERT INTO Fact_Books VALUES (7, 2, 3, 2, 3, 950, 11000.00, 7.50);
INSERT INTO Fact_Books VALUES (8, 3, 4, 3, 4, 300, 4000.00, 14.00);

-- Step 6: Perform Queries

-- Query 1: Quantity Sold and Profit per Book Type
SELECT
   B.Book_Type_Name,
   SUM(F.Quantity_Sold) AS Total_Quantity_Sold,
   SUM(F.Profit) AS Total_Profit
FROM
   Fact_Books F
JOIN
   Dim_Book_Types B ON F.Book_Type_ID = B.Book_Type_ID
GROUP BY
   B.Book_Type_Name;

-- Query 2: Quantity Sold and Profit per Location
SELECT
   L.Location_Name,
   SUM(F.Quantity_Sold) AS Total_Quantity_Sold,
   SUM(F.Profit) AS Total_Profit
FROM
   Fact_Books F
JOIN
   Dim_Locations L ON F.Location_ID = L.Location_ID
GROUP BY
   L.Location_Name;

-- Query 3: Quantity Sold and Profit per Author
SELECT
   A.Author_Name,
   SUM(F.Quantity_Sold) AS Total_Quantity_Sold,
   SUM(F.Profit) AS Total_Profit
FROM
   Fact_Books F
JOIN
   Dim_Authors A ON F.Author_ID = A.Author_ID
GROUP BY
   A.Author_Name;

-- Query 4: Quantity Sold and Profit per Publication
SELECT
   P.Publication_Name,
   SUM(F.Quantity_Sold) AS Total_Quantity_Sold,
   SUM(F.Profit) AS Total_Profit
FROM
   Fact_Books F
JOIN
   Dim_Publications P ON F.Publication_ID = P.Publication_ID
GROUP BY
   P.Publication_Name;

-- Query 5: Quantity Sold and Profit per Book Type and Location
SELECT
   B.Book_Type_Name,
   L.Location_Name,
   SUM(F.Quantity_Sold) AS Total_Quantity_Sold,
   SUM(F.Profit) AS Total_Profit
FROM
   Fact_Books F
JOIN
   Dim_Book_Types B ON F.Book_Type_ID = B.Book_Type_ID
JOIN
   Dim_Locations L ON F.Location_ID = L.Location_ID
GROUP BY
   B.Book_Type_Name, L.Location_Name;
```

SNOWLAKE:

```
-- Dim_Book_Type_Category (Normalized Book Category)
CREATE TABLE Dim_Book_Type_Category (
   Category_ID INT PRIMARY KEY,
   Category_Name VARCHAR(50)
);

-- Dim_Book_Type (Book Types normalized into Category)
```

```
CREATE TABLE Dim_Book_Type (
   Book_Type_ID INT PRIMARY KEY,
   Category_ID INT,
   Book_Type_Name VARCHAR(50),
   FOREIGN KEY (Category_ID) REFERENCES Dim_Book_Type_Category(Category_ID)
);

-- Dim_Location (Location details)
CREATE TABLE Dim_Location (
   Location_ID INT PRIMARY KEY,
   Location_Name VARCHAR(100)
);

-- Dim_Author_Country (Country of Author)
CREATE TABLE Dim_Author_Country (
   Author_Country_ID INT PRIMARY KEY,
   Author_Country VARCHAR(50)
);

-- Dim_Author_Age (Age of Author)
CREATE TABLE Dim_Author_Age (
   Author_Age_ID INT PRIMARY KEY,
   Author_Age INT
);

-- Dim_Author (Author details, references country and age)
CREATE TABLE Dim_Author (
   Author_ID INT PRIMARY KEY,
   Author_Name VARCHAR(100),
   Author_Country_ID INT,
   Author_Age_ID INT,
   FOREIGN KEY (Author_Country_ID) REFERENCES
Dim_Author_Country(Author_Country_ID),
   FOREIGN KEY (Author_Age_ID) REFERENCES Dim_Author_Age(Author_Age_ID)
);

-- Dim_Publication_Country (Country of Publication)
CREATE TABLE Dim_Publication_Country (
   Publication_Country_ID INT PRIMARY KEY,
   Publication_Country VARCHAR(50)
);

-- Dim_Publication_Year (Year of Publication)
CREATE TABLE Dim_Publication_Year (
   Publication_Year_ID INT PRIMARY KEY,
   Publication_Year INT
);

-- Dim_Publication (Publication details, references country and year)
CREATE TABLE Dim_Publication (
   Publication_ID INT PRIMARY KEY,
   Publication_Name VARCHAR(100),
   Publication_Country_ID INT,
   Publication_Year_ID INT,
   FOREIGN KEY (Publication_Country_ID) REFERENCES
Dim_Publication_Country(Publication_Country_ID),
   FOREIGN KEY (Publication_Year_ID) REFERENCES
Dim_Publication_Year(Publication_Year_ID)
);

-- Step 3: Create Fact Table (Sales Data)
CREATE TABLE Fact_Book_Sales (
   Fact_Sales_ID INT PRIMARY KEY,
   Book_Type_ID INT,
   Location_ID INT,
   Author_ID INT,
   Publication_ID INT,
   Quantity_Sold INT,
   Profit DECIMAL(15, 2),
   FOREIGN KEY (Book_Type_ID) REFERENCES Dim_Book_Type(Book_Type_ID),
   FOREIGN KEY (Location_ID) REFERENCES Dim_Location(Location_ID),
   FOREIGN KEY (Author_ID) REFERENCES Dim_Author(Author_ID),
   FOREIGN KEY (Publication_ID) REFERENCES Dim_Publication(Publication_ID)
);

-- Step 4: Insert Sample Data

-- Dim_Book_Type_Category (Book Categories)
INSERT INTO Dim_Book_Type_Category VALUES (1, 'Fiction');
INSERT INTO Dim_Book_Type_Category VALUES (2, 'Non-Fiction');
INSERT INTO Dim_Book_Type_Category VALUES (3, 'Educational');
INSERT INTO Dim_Book_Type_Category VALUES (4, 'Science');
INSERT INTO Dim_Book_Type_Category VALUES (5, 'Biography');

-- Dim_Book_Type (Books under each Category)
INSERT INTO Dim_Book_Type VALUES (1, 1, 'Novel');
INSERT INTO Dim_Book_Type VALUES (2, 1, 'Short Story');
INSERT INTO Dim_Book_Type VALUES (3, 2, 'Self Help');
INSERT INTO Dim_Book_Type VALUES (4, 3, 'Mathematics');
INSERT INTO Dim_Book_Type VALUES (5, 4, 'Physics');

-- Dim_Location (Bookstore Locations)
INSERT INTO Dim_Location VALUES (1, 'New York');
INSERT INTO Dim_Location VALUES (2, 'Los Angeles');
INSERT INTO Dim_Location VALUES (3, 'Chicago');
INSERT INTO Dim_Location VALUES (4, 'Houston');
INSERT INTO Dim_Location VALUES (5, 'San Francisco');

-- Dim_Author_Country (Countries of Authors)
INSERT INTO Dim_Author_Country VALUES (1, 'United Kingdom');
INSERT INTO Dim_Author_Country VALUES (2, 'United States');
INSERT INTO Dim_Author_Country VALUES (3, 'Germany');
INSERT INTO Dim_Author_Country VALUES (4, 'France');
INSERT INTO Dim_Author_Country VALUES (5, 'India');

-- Dim_Author_Age (Age of Authors)
INSERT INTO Dim_Author_Age VALUES (1, 55);
INSERT INTO Dim_Author_Age VALUES (2, 60);
INSERT INTO Dim_Author_Age VALUES (3, 80);
INSERT INTO Dim_Author_Age VALUES (4, 45);
INSERT INTO Dim_Author_Age VALUES (5, 50);

-- Dim_Author (Authors details)
INSERT INTO Dim_Author VALUES (1, 'J.K. Rowling', 1, 1);
INSERT INTO Dim_Author VALUES (2, 'George R.R. Martin', 2, 2);
INSERT INTO Dim_Author VALUES (3, 'Isaac Newton', 3, 3);
INSERT INTO Dim_Author VALUES (4, 'Albert Einstein', 4, 4);
INSERT INTO Dim_Author VALUES (5, 'Stephen Hawking', 5, 5);

-- Dim_Publication_Country (Countries of Publications)
INSERT INTO Dim_Publication_Country VALUES (1, 'United Kingdom');
INSERT INTO Dim_Publication_Country VALUES (2, 'United States');
INSERT INTO Dim_Publication_Country VALUES (3, 'Germany');
INSERT INTO Dim_Publication_Country VALUES (4, 'France');
INSERT INTO Dim_Publication_Country VALUES (5, 'India');

-- Dim_Publication_Year (Publication Year)
INSERT INTO Dim_Publication_Year VALUES (1, 1995);
INSERT INTO Dim_Publication_Year VALUES (2, 2000);
INSERT INTO Dim_Publication_Year VALUES (3, 2010);
INSERT INTO Dim_Publication_Year VALUES (4, 2015);
INSERT INTO Dim_Publication_Year VALUES (5, 2020);
```

```
-- Dim_Publication (Publication details)
INSERT INTO Dim_Publication VALUES (1, 'Penguin Books', 1, 1);
INSERT INTO Dim_Publication VALUES (2, 'HarperCollins', 2, 2);
INSERT INTO Dim_Publication VALUES (3, 'Oxford University Press', 3, 3);
INSERT INTO Dim_Publication VALUES (4, 'Springer', 4, 4);
INSERT INTO Dim_Publication VALUES (5, 'Macmillan', 5, 5);

-- Fact_Book_Sales (Sales Data)
INSERT INTO Fact_Book_Sales VALUES (1, 1, 1, 1, 1, 500, 2500.00);
INSERT INTO Fact_Book_Sales VALUES (2, 2, 2, 2, 2, 300, 1500.00);
INSERT INTO Fact_Book_Sales VALUES (3, 3, 3, 3, 3, 400, 2000.00);
INSERT INTO Fact_Book_Sales VALUES (4, 4, 4, 4, 4, 600, 3500.00);
INSERT INTO Fact_Book_Sales VALUES (5, 5, 5, 5, 5, 200, 1200.00);

-- Step 5: Execute Queries

-- Query 1: Quantity and Profit per Book Type
SELECT BT.Book_Type_Name, SUM(FBS.Quantity_Sold) AS Total_Quantity_Sold,
SUM(FBS.Profit) AS Total_Profit
FROM Fact_Book_Sales FBS
JOIN Dim_Book_Type BT ON FBS.Book_Type_ID = BT.Book_Type_ID
GROUP BY BT.Book_Type_Name;

-- Query 2: Quantity and Profit per Location
SELECT L.Location_Name, SUM(FBS.Quantity_Sold) AS Total_Quantity_Sold,
SUM(FBS.Profit) AS Total_Profit
FROM Fact_Book_Sales FBS
JOIN Dim_Location L ON FBS.Location_ID = L.Location_ID
GROUP BY L.Location_Name;

-- Query 3: Quantity and Profit per Author
SELECT A.Author_Name, SUM(FBS.Quantity_Sold) AS Total_Quantity_Sold, SUM(FBS.Profit)
AS Total_Profit
FROM Fact_Book_Sales FBS
JOIN Dim_Author A ON FBS.Author_ID = A.Author_ID
GROUP BY A.Author_Name;

-- Query 4: Quantity and Profit per Publication
SELECT P.Publication_Name, SUM(FBS.Quantity_Sold) AS Total_Quantity_Sold,
SUM(FBS.Profit) AS Total_Profit
FROM Fact_Book_Sales FBS
JOIN Dim_Publication P ON FBS.Publication_ID = P.Publication_ID
GROUP BY P.Publication_Name;
```

```
-- Query 5: Quantity and Profit by Book Type and Author
SELECT BT.Book_Type_Name, A.Author_Name, SUM(FBS.Quantity_Sold) AS
Total_Quantity_Sold, SUM(FBS.Profit) AS Total_Profit
FROM Fact_Book_Sales FBS
JOIN Dim_Book_Type BT ON FBS.Book_Type_ID = BT.Book_Type_ID
JOIN Dim_Author A ON FBS.Author_ID = A.Author_ID
GROUP BY BT.Book_Type_Name, A.Author_Name;
```

GALAXY

```
-- Dim_Book_Type (Book Type details)
CREATE TABLE Dim_Book_Type (
    Book_Type_ID INT PRIMARY KEY,
    Book_Type_Name VARCHAR(50)
);

-- Dim_Location (Bookstore Locations)
CREATE TABLE Dim_Location (
    Location_ID INT PRIMARY KEY,
    Location_Name VARCHAR(100)
);

-- Dim_Author (Author details)
CREATE TABLE Dim_Author (
    Author_ID INT PRIMARY KEY,
    Author_Name VARCHAR(100)
);

-- Dim_Publication (Publication details)
CREATE TABLE Dim_Publication (
    Publication_ID INT PRIMARY KEY,
    Publication_Name VARCHAR(100)
);

-- Step 3: Create Fact Tables

-- Fact_Book_Sales (Sales data)
CREATE TABLE Fact_Book_Sales (
    Fact_Sales_ID INT PRIMARY KEY,
    Book_Type_ID INT,
    Location_ID INT,
    Author_ID INT,
    Publication_ID INT,
    Quantity_Sold INT,
    Profit DECIMAL(15, 2),
    FOREIGN KEY (Book_Type_ID) REFERENCES Dim_Book_Type(Book_Type_ID),
    FOREIGN KEY (Location_ID) REFERENCES Dim_Location(Location_ID),
    FOREIGN KEY (Author_ID) REFERENCES Dim_Author(Author_ID),
    FOREIGN KEY (Publication_ID) REFERENCES Dim_Publication(Publication_ID)
);

-- Fact_Book_Inventory (Inventory data)
CREATE TABLE Fact_Book_Inventory (
    Fact_Inventory_ID INT PRIMARY KEY,
    Book_Type_ID INT,
    Location_ID INT,
    Quantity_In_Stock INT,
    FOREIGN KEY (Book_Type_ID) REFERENCES Dim_Book_Type(Book_Type_ID),
    FOREIGN KEY (Location_ID) REFERENCES Dim_Location(Location_ID)
);

-- Fact_Author_Profit (Profit data for authors)
CREATE TABLE Fact_Author_Profit (
    Fact_Author_Profit_ID INT PRIMARY KEY,
    Author_ID INT,
    Profit DECIMAL(15, 2),
    FOREIGN KEY (Author_ID) REFERENCES Dim_Author(Author_ID)
);

-- Step 4: Insert Sample Data

-- Dim_Book_Type (Book Types)
INSERT INTO Dim_Book_Type VALUES (1, 'Fiction');
INSERT INTO Dim_Book_Type VALUES (2, 'Non-Fiction');
INSERT INTO Dim_Book_Type VALUES (3, 'Educational');
INSERT INTO Dim_Book_Type VALUES (4, 'Science');
INSERT INTO Dim_Book_Type VALUES (5, 'Biography');

-- Dim_Location (Bookstore Locations)
INSERT INTO Dim_Location VALUES (1, 'New York');
INSERT INTO Dim_Location VALUES (2, 'Los Angeles');
INSERT INTO Dim_Location VALUES (3, 'Chicago');
INSERT INTO Dim_Location VALUES (4, 'Houston');
INSERT INTO Dim_Location VALUES (5, 'San Francisco');
```

```
-- Dim_Author (Authors details)
INSERT INTO Dim_Author VALUES (1, 'J.K. Rowling');
INSERT INTO Dim_Author VALUES (2, 'George R.R. Martin');
INSERT INTO Dim_Author VALUES (3, 'Isaac Newton');
INSERT INTO Dim_Author VALUES (4, 'Albert Einstein');
INSERT INTO Dim_Author VALUES (5, 'Stephen Hawking');

-- Dim_Publication (Publication details)
INSERT INTO Dim_Publication VALUES (1, 'Penguin Books');
INSERT INTO Dim_Publication VALUES (2, 'HarperCollins');
INSERT INTO Dim_Publication VALUES (3, 'Oxford University Press');
INSERT INTO Dim_Publication VALUES (4, 'Springer');
INSERT INTO Dim_Publication VALUES (5, 'Macmillan');

-- Fact_Book_Sales (Sales Data)
INSERT INTO Fact_Book_Sales VALUES (1, 1, 1, 1, 1, 500, 2500.00);
INSERT INTO Fact_Book_Sales VALUES (2, 2, 2, 2, 2, 300, 1500.00);
INSERT INTO Fact_Book_Sales VALUES (3, 3, 3, 3, 3, 400, 2000.00);
INSERT INTO Fact_Book_Sales VALUES (4, 4, 4, 4, 4, 600, 3500.00);
INSERT INTO Fact_Book_Sales VALUES (5, 5, 5, 5, 5, 200, 1200.00);

-- Fact_Book_Inventory (Inventory Data)
INSERT INTO Fact_Book_Inventory VALUES (1, 1, 1, 100);
INSERT INTO Fact_Book_Inventory VALUES (2, 2, 2, 200);
INSERT INTO Fact_Book_Inventory VALUES (3, 3, 3, 150);
INSERT INTO Fact_Book_Inventory VALUES (4, 4, 4, 50);
INSERT INTO Fact_Book_Inventory VALUES (5, 5, 5, 120);

-- Fact_Author_Profit (Author Profit Data)
INSERT INTO Fact_Author_Profit VALUES (1, 1, 2500.00);
INSERT INTO Fact_Author_Profit VALUES (2, 2, 3000.00);
INSERT INTO Fact_Author_Profit VALUES (3, 3, 2000.00);
INSERT INTO Fact_Author_Profit VALUES (4, 4, 3500.00);
INSERT INTO Fact_Author_Profit VALUES (5, 5, 1500.00);

-- Step 5: Execute Queries

-- Query 1: Sales and profit per Book Type
SELECT BT.Book_Type_Name, SUM(FBS.Quantity_Sold) AS Total_Quantity_Sold,
SUM(FBS.Profit) AS Total_Profit
FROM Fact_Book_Sales FBS
JOIN Dim_Book_Type BT ON FBS.Book_Type_ID = BT.Book_Type_ID
GROUP BY BT.Book_Type_Name;
```

```
-- Query 2: Sales and Profit per Location
SELECT L.Location_Name, SUM(FBS.Quantity_Sold) AS Total_Quantity_Sold,
SUM(FBS.Profit) AS Total_Profit
FROM Fact_Book_Sales FBS
JOIN Dim_Location L ON FBS.Location_ID = L.Location_ID
GROUP BY L.Location_Name;

-- Query 3: Sales and Profit per Author
SELECT A.Author_Name, SUM(FBS.Quantity_Sold) AS Total_Quantity_Sold, SUM(FBS.Profit)
AS Total_Profit
FROM Fact_Book_Sales FBS
JOIN Dim_Author A ON FBS.Author_ID = A.Author_ID
GROUP BY A.Author_Name;

-- Query 4: Inventory per Book Type and Location
SELECT BT.Book_Type_Name, L.Location_Name, SUM(FBI.Quantity_In_Stock) AS
Total_Inventory
FROM Fact_Book_Inventory FBI
JOIN Dim_Book_Type BT ON FBI.Book_Type_ID = BT.Book_Type_ID
JOIN Dim_Location L ON FBI.Location_ID = L.Location_ID
GROUP BY BT.Book_Type_Name, L.Location_Name;

-- Query 5: Author's Profit
SELECT A.Author_Name, SUM(FAP.Profit) AS Total_Author_Profit
FROM Fact_Author_Profit FAP
JOIN Dim_Author A ON FAP.Author_ID = A.Author_ID
GROUP BY A.Author_Name;
```

Q4:Gather Business Requirements for Banking enterprise and design it using any multi-dimensional data model namely Star, Snowflake, or Galaxy schema.

STAR SCHEMA:

```
-- Step 1: Drop existing tables to avoid errors
DROP TABLE IF EXISTS Fact_Transactions;
DROP TABLE IF EXISTS Dim_Customers;
DROP TABLE IF EXISTS Dim_Accounts;
```

```sql
DROP TABLE IF EXISTS Dim_Transactions;
DROP TABLE IF EXISTS Dim_Branches;
DROP TABLE IF EXISTS Dim_Date;

-- Step 2: Create Dimension Tables

-- Customers Dimension Table
CREATE TABLE Dim_Customers (
    Customer_ID INT PRIMARY KEY,
    Customer_Name VARCHAR(100),
    Address VARCHAR(255),
    City VARCHAR(50),
    Country VARCHAR(50)
);

-- Accounts Dimension Table
CREATE TABLE Dim_Accounts (
    Account_ID INT PRIMARY KEY,
    Account_Type VARCHAR(50),  -- Savings, Checking, etc.
    Balance DECIMAL(15, 2),
    Account_Status VARCHAR(50) -- Active, Closed, Dormant
);

-- Transactions Dimension Table
CREATE TABLE Dim_Transactions (
    Transaction_Type_ID INT PRIMARY KEY,
    Transaction_Type VARCHAR(50)  -- Deposit, Withdrawal, Transfer, etc.
);

-- Branches Dimension Table
CREATE TABLE Dim_Branches (
    Branch_ID INT PRIMARY KEY,
    Branch_Name VARCHAR(100),
    Branch_Location VARCHAR(100)
);

-- Date Dimension Table
CREATE TABLE Dim_Date (
    Date_ID INT PRIMARY KEY,
    Date DATE,
    Year INT,
    Month INT,
    Day INT,
    Quarter INT
);

-- Step 3: Create Fact Table
CREATE TABLE Fact_Transactions (
    Fact_ID INT PRIMARY KEY,
    Customer_ID INT,
    Account_ID INT,
    Transaction_Type_ID INT,
    Branch_ID INT,
    Date_ID INT,
    Transaction_Amount DECIMAL(15, 2),
    FOREIGN KEY (Customer_ID) REFERENCES Dim_Customers(Customer_ID),
    FOREIGN KEY (Account_ID) REFERENCES Dim_Accounts(Account_ID),
    FOREIGN KEY (Transaction_Type_ID) REFERENCES
Dim_Transactions(Transaction_Type_ID),
    FOREIGN KEY (Branch_ID) REFERENCES Dim_Branches(Branch_ID),
    FOREIGN KEY (Date_ID) REFERENCES Dim_Date(Date_ID)
);

-- Step 4: Insert Data into Dimension Tables

-- Dim_Customers
INSERT INTO Dim_Customers VALUES (1, 'John Doe', '123 Elm St', 'New York', 'USA');
INSERT INTO Dim_Customers VALUES (2, 'Jane Smith', '456 Oak Rd', 'Los Angeles', 'USA');
INSERT INTO Dim_Customers VALUES (3, 'Mike Johnson', '789 Pine Ave', 'Chicago', 'USA');
INSERT INTO Dim_Customers VALUES (4, 'Emily Davis', '101 Maple Dr', 'Houston', 'USA');
INSERT INTO Dim_Customers VALUES (5, 'Chris Lee', '202 Birch Blvd', 'San Francisco',
'USA');

-- Dim_Accounts
INSERT INTO Dim_Accounts VALUES (1, 'Savings', 10000.00, 'Active');
INSERT INTO Dim_Accounts VALUES (2, 'Checking', 5000.00, 'Active');
INSERT INTO Dim_Accounts VALUES (3, 'Savings', 20000.00, 'Closed');
INSERT INTO Dim_Accounts VALUES (4, 'Checking', 3000.00, 'Active');
INSERT INTO Dim_Accounts VALUES (5, 'Savings', 15000.00, 'Dormant');

-- Dim_Transactions
INSERT INTO Dim_Transactions VALUES (1, 'Deposit');
INSERT INTO Dim_Transactions VALUES (2, 'Withdrawal');
INSERT INTO Dim_Transactions VALUES (3, 'Transfer');
INSERT INTO Dim_Transactions VALUES (4, 'Loan Repayment');
INSERT INTO Dim_Transactions VALUES (5, 'Interest Payment');

-- Dim_Branches
```

```
INSERT INTO Dim_Branches VALUES (1, 'Downtown Branch', 'New York');
INSERT INTO Dim_Branches VALUES (2, 'West End Branch', 'Los Angeles');
INSERT INTO Dim_Branches VALUES (3, 'Midtown Branch', 'Chicago');
INSERT INTO Dim_Branches VALUES (4, 'East Side Branch', 'Houston');
INSERT INTO Dim_Branches VALUES (5, 'South Bay Branch', 'San Francisco');

-- Dim_Date
INSERT INTO Dim_Date VALUES (1, '2024-01-01', 2024, 1, 1, 1);
INSERT INTO Dim_Date VALUES (2, '2024-01-02', 2024, 1, 2, 1);
INSERT INTO Dim_Date VALUES (3, '2024-02-15', 2024, 2, 15, 1);
INSERT INTO Dim_Date VALUES (4, '2024-03-10', 2024, 3, 10, 1);
INSERT INTO Dim_Date VALUES (5, '2024-04-20', 2024, 4, 20, 2);

-- Step 5: Insert Data into Fact Table
INSERT INTO Fact_Transactions VALUES (1, 1, 1, 1, 1, 1, 5000.00);
INSERT INTO Fact_Transactions VALUES (2, 2, 2, 2, 2, 2, 2000.00);
INSERT INTO Fact_Transactions VALUES (3, 3, 3, 3, 3, 3, 3000.00);
INSERT INTO Fact_Transactions VALUES (4, 4, 4, 4, 4, 4, 1500.00);
INSERT INTO Fact_Transactions VALUES (5, 5, 5, 5, 5, 5, 1000.00);

-- Step 6: Perform Queries

-- Query 1: Total Transactions and Amount per Customer
SELECT
    C.Customer_Name,
    SUM(F.Transaction_Amount) AS Total_Transaction_Amount
FROM
    Fact_Transactions F
JOIN
    Dim_Customers C ON F.Customer_ID = C.Customer_ID
GROUP BY
    C.Customer_Name;

-- Query 2: Total Transactions and Amount per Account
SELECT
    A.Account_Type,
    SUM(F.Transaction_Amount) AS Total_Transaction_Amount
FROM
    Fact_Transactions F
JOIN
    Dim_Accounts A ON F.Account_ID = A.Account_ID
GROUP BY
    A.Account_Type;
```

```
-- Query 3: Total Transactions and Amount per Branch
SELECT
   B.Branch_Name,
   SUM(F.Transaction_Amount) AS Total_Transaction_Amount
FROM
   Fact_Transactions F
JOIN
   Dim_Branches B ON F.Branch_ID = B.Branch_ID
GROUP BY
   B.Branch_Name;

-- Query 4: Total Transactions and Amount per Transaction Type
SELECT
   T.Transaction_Type,
   SUM(F.Transaction_Amount) AS Total_Transaction_Amount
FROM
   Fact_Transactions F
JOIN
   Dim_Transactions T ON F.Transaction_Type_ID = T.Transaction_Type_ID
GROUP BY
   T.Transaction_Type;

-- Query 5: Total Transactions and Amount per Month
SELECT
   D.Month,
   SUM(F.Transaction_Amount) AS Total_Transaction_Amount
FROM
   Fact_Transactions F
JOIN
   Dim_Date D ON F.Date_ID = D.Date_ID
GROUP BY
   D.Month;
```

SNOWFLAKE:

```
-- Customers Dimension Table
CREATE TABLE Dim_Customers (
   Customer_ID INT PRIMARY KEY,
   Customer_Name VARCHAR(100)
);

-- Customer Address Table
CREATE TABLE Dim_Customer_Address (
   Address_ID INT PRIMARY KEY,
    Customer_ID INT,
    Address VARCHAR(255),
    City VARCHAR(50),
    Country VARCHAR(50),
    FOREIGN KEY (Customer_ID) REFERENCES Dim_Customers(Customer_ID)
);

-- Account Types Dimension Table
CREATE TABLE Dim_Account_Types (
    Account_Type_ID INT PRIMARY KEY,
    Account_Type VARCHAR(50)  -- Savings, Checking, etc.
);

-- Transactions Dimension Table
CREATE TABLE Dim_Transactions (
    Transaction_Type_ID INT PRIMARY KEY,
    Transaction_Type VARCHAR(50)  -- Deposit, Withdrawal, etc.
);

-- Branches Dimension Table
CREATE TABLE Dim_Branches (
    Branch_ID INT PRIMARY KEY,
    Branch_Name VARCHAR(100),
    Branch_Location VARCHAR(100)
);

-- Date Dimension Table
CREATE TABLE Dim_Date (
    Date_ID INT PRIMARY KEY,
    Date DATE,
    Year INT,
    Month INT,
    Day INT,
    Quarter INT
);

-- Step 3: Create Fact Table
CREATE TABLE Fact_Transactions (
    Fact_ID INT PRIMARY KEY,
    Customer_ID INT,
    Account_Type_ID INT,
    Transaction_Type_ID INT,
    Branch_ID INT,
    Date_ID INT,
```

```sql
    Transaction_Amount DECIMAL(15, 2),
    FOREIGN KEY (Customer_ID) REFERENCES Dim_Customers(Customer_ID),
    FOREIGN KEY (Account_Type_ID) REFERENCES Dim_Account_Types(Account_Type_ID),
    FOREIGN KEY (Transaction_Type_ID) REFERENCES
Dim_Transactions(Transaction_Type_ID),
    FOREIGN KEY (Branch_ID) REFERENCES Dim_Branches(Branch_ID),
    FOREIGN KEY (Date_ID) REFERENCES Dim_Date(Date_ID)
);

-- Step 4: Insert Data into Dimension Tables

-- Dim_Customers
INSERT INTO Dim_Customers VALUES (1, 'John Doe');
INSERT INTO Dim_Customers VALUES (2, 'Jane Smith');
INSERT INTO Dim_Customers VALUES (3, 'Mike Johnson');
INSERT INTO Dim_Customers VALUES (4, 'Emily Davis');
INSERT INTO Dim_Customers VALUES (5, 'Chris Lee');

-- Dim_Customer_Address
INSERT INTO Dim_Customer_Address VALUES (1, 1, '123 Elm St', 'New York', 'USA');
INSERT INTO Dim_Customer_Address VALUES (2, 2, '456 Oak Rd', 'Los Angeles', 'USA');
INSERT INTO Dim_Customer_Address VALUES (3, 3, '789 Pine Ave', 'Chicago', 'USA');
INSERT INTO Dim_Customer_Address VALUES (4, 4, '101 Maple Dr', 'Houston', 'USA');
INSERT INTO Dim_Customer_Address VALUES (5, 5, '202 Birch Blvd', 'San Francisco', 'USA');

-- Dim_Account_Types
INSERT INTO Dim_Account_Types VALUES (1, 'Savings');
INSERT INTO Dim_Account_Types VALUES (2, 'Checking');
INSERT INTO Dim_Account_Types VALUES (3, 'Loan');
INSERT INTO Dim_Account_Types VALUES (4, 'Credit');
INSERT INTO Dim_Account_Types VALUES (5, 'Investment');

-- Dim_Transactions
INSERT INTO Dim_Transactions VALUES (1, 'Deposit');
INSERT INTO Dim_Transactions VALUES (2, 'Withdrawal');
INSERT INTO Dim_Transactions VALUES (3, 'Transfer');
INSERT INTO Dim_Transactions VALUES (4, 'Loan Repayment');
INSERT INTO Dim_Transactions VALUES (5, 'Interest Payment');

-- Dim_Branches
INSERT INTO Dim_Branches VALUES (1, 'Downtown Branch', 'New York');
INSERT INTO Dim_Branches VALUES (2, 'West End Branch', 'Los Angeles');
INSERT INTO Dim_Branches VALUES (3, 'Midtown Branch', 'Chicago');
INSERT INTO Dim_Branches VALUES (4, 'East Side Branch', 'Houston');
INSERT INTO Dim_Branches VALUES (5, 'South Bay Branch', 'San Francisco');

-- Dim_Date
INSERT INTO Dim_Date VALUES (1, '2024-01-01', 2024, 1, 1, 1);
INSERT INTO Dim_Date VALUES (2, '2024-01-02', 2024, 1, 2, 1);
INSERT INTO Dim_Date VALUES (3, '2024-02-15', 2024, 2, 15, 1);
INSERT INTO Dim_Date VALUES (4, '2024-03-10', 2024, 3, 10, 1);
INSERT INTO Dim_Date VALUES (5, '2024-04-20', 2024, 4, 20, 2);

-- Step 5: Insert Data into Fact Table
INSERT INTO Fact_Transactions VALUES (1, 1, 1, 1, 1, 1, 5000.00);
INSERT INTO Fact_Transactions VALUES (2, 2, 2, 2, 2, 2, 2000.00);
INSERT INTO Fact_Transactions VALUES (3, 3, 3, 3, 3, 3, 3000.00);
INSERT INTO Fact_Transactions VALUES (4, 4, 4, 4, 4, 4, 1500.00);
INSERT INTO Fact_Transactions VALUES (5, 5, 5, 5, 5, 5, 1000.00);

-- Step 6: Perform Queries

-- Query 1: Total Transactions and Amount per Customer
SELECT
    C.Customer_Name,
    SUM(F.Transaction_Amount) AS Total_Transaction_Amount
FROM
    Fact_Transactions F
JOIN
    Dim_Customers C ON F.Customer_ID = C.Customer_ID
GROUP BY
    C.Customer_Name;

-- Query 2: Total Transactions and Amount per Account Type
SELECT
    A.Account_Type,
    SUM(F.Transaction_Amount) AS Total_Transaction_Amount
FROM
    Fact_Transactions F
JOIN
    Dim_Account_Types A ON F.Account_Type_ID = A.Account_Type_ID
GROUP BY
    A.Account_Type;

-- Query 3: Total Transactions and Amount per Branch
SELECT
    B.Branch_Name,
    SUM(F.Transaction_Amount) AS Total_Transaction_Amount
```

```
FROM
   Fact_Transactions F
JOIN
   Dim_Branches B ON F.Branch_ID = B.Branch_ID
GROUP BY
   B.Branch_Name;

-- Query 4: Total Transactions and Amount per Transaction Type
SELECT
   T.Transaction_Type,
   SUM(F.Transaction_Amount) AS Total_Transaction_Amount
FROM
   Fact_Transactions F
JOIN
   Dim_Transactions T ON F.Transaction_Type_ID = T.Transaction_Type_ID
GROUP BY
   T.Transaction_Type;

-- Query 5: Total Transactions and Amount per Date
SELECT
   D.Month,
   SUM(F.Transaction_Amount) AS Total_Transaction_Amount
FROM
   Fact_Transactions F
JOIN
   Dim_Date D ON F.Date_ID = D.Date_ID
GROUP BY
   D.Month;
```

GALAXY:

```
-- Dim_Customers (Customer Details)
CREATE TABLE Dim_Customers (
   Customer_ID INT PRIMARY KEY,
   Customer_Name VARCHAR(100)
);

-- Dim_Accounts (Account Type Details)
CREATE TABLE Dim_Accounts (
   Account_ID INT PRIMARY KEY,
   Account_Type VARCHAR(50) -- e.g., Savings, Checking, Loan, Credit, etc.
);
```

```
-- Dim_Transactions (Transaction Type Details)
CREATE TABLE Dim_Transactions (
   Transaction_Type_ID INT PRIMARY KEY,
   Transaction_Type VARCHAR(50) -- e.g., Deposit, Withdrawal, Transfer, Loan Repayment,
etc.
);

-- Dim_Branches (Branch Details)
CREATE TABLE Dim_Branches (
   Branch_ID INT PRIMARY KEY,
   Branch_Name VARCHAR(100),
   Branch_Location VARCHAR(100)
);

-- Dim_Date (Date Details)
CREATE TABLE Dim_Date (
   Date_ID INT PRIMARY KEY,
   Date DATE,
   Year INT,
   Month INT,
   Day INT,
   Quarter INT
);

-- Step 3: Create Fact Tables

-- Fact_Transactions (Transaction Data)
CREATE TABLE Fact_Transactions (
   Fact_ID INT PRIMARY KEY,
   Customer_ID INT,
   Account_ID INT,
   Transaction_Type_ID INT,
   Branch_ID INT,
   Date_ID INT,
   Transaction_Amount DECIMAL(15, 2),
   FOREIGN KEY (Customer_ID) REFERENCES Dim_Customers(Customer_ID),
   FOREIGN KEY (Account_ID) REFERENCES Dim_Accounts(Account_ID),
   FOREIGN KEY (Transaction_Type_ID) REFERENCES
Dim_Transactions(Transaction_Type_ID),
   FOREIGN KEY (Branch_ID) REFERENCES Dim_Branches(Branch_ID),
   FOREIGN KEY (Date_ID) REFERENCES Dim_Date(Date_ID)
);
```

```
-- Fact_Loans (Loan and Repayment Data)
CREATE TABLE Fact_Loans (
    Loan_Fact_ID INT PRIMARY KEY,
    Customer_ID INT,
    Account_ID INT,
    Branch_ID INT,
    Date_ID INT,
    Loan_Amount DECIMAL(15, 2),
    Repayment_Amount DECIMAL(15, 2),
    FOREIGN KEY (Customer_ID) REFERENCES Dim_Customers(Customer_ID),
    FOREIGN KEY (Account_ID) REFERENCES Dim_Accounts(Account_ID),
    FOREIGN KEY (Branch_ID) REFERENCES Dim_Branches(Branch_ID),
    FOREIGN KEY (Date_ID) REFERENCES Dim_Date(Date_ID)
);

-- Step 4: Insert Sample Data into Dimension Tables

-- Insert Customers
INSERT INTO Dim_Customers VALUES (1, 'John Doe');
INSERT INTO Dim_Customers VALUES (2, 'Jane Smith');
INSERT INTO Dim_Customers VALUES (3, 'Mike Johnson');
INSERT INTO Dim_Customers VALUES (4, 'Emily Davis');
INSERT INTO Dim_Customers VALUES (5, 'Chris Lee');

-- Insert Accounts (Account Types)
INSERT INTO Dim_Accounts VALUES (1, 'Savings');
INSERT INTO Dim_Accounts VALUES (2, 'Checking');
INSERT INTO Dim_Accounts VALUES (3, 'Loan');
INSERT INTO Dim_Accounts VALUES (4, 'Credit');
INSERT INTO Dim_Accounts VALUES (5, 'Investment');

-- Insert Transaction Types
INSERT INTO Dim_Transactions VALUES (1, 'Deposit');
INSERT INTO Dim_Transactions VALUES (2, 'Withdrawal');
INSERT INTO Dim_Transactions VALUES (3, 'Transfer');
INSERT INTO Dim_Transactions VALUES (4, 'Loan Repayment');
INSERT INTO Dim_Transactions VALUES (5, 'Interest Payment');

-- Insert Branches
INSERT INTO Dim_Branches VALUES (1, 'Downtown Branch', 'New York');
INSERT INTO Dim_Branches VALUES (2, 'West End Branch', 'Los Angeles');
INSERT INTO Dim_Branches VALUES (3, 'Midtown Branch', 'Chicago');
INSERT INTO Dim_Branches VALUES (4, 'East Side Branch', 'Houston');
INSERT INTO Dim_Branches VALUES (5, 'South Bay Branch', 'San Francisco');
```

```
-- Insert Date Details
INSERT INTO Dim_Date VALUES (1, '2024-01-01', 2024, 1, 1, 1);
INSERT INTO Dim_Date VALUES (2, '2024-01-02', 2024, 1, 2, 1);
INSERT INTO Dim_Date VALUES (3, '2024-02-15', 2024, 2, 15, 1);
INSERT INTO Dim_Date VALUES (4, '2024-03-10', 2024, 3, 10, 1);
INSERT INTO Dim_Date VALUES (5, '2024-04-20', 2024, 4, 20, 2);

-- Step 5: Insert Data into Fact Tables

-- Insert Sample Data into Fact_Transactions
INSERT INTO Fact_Transactions VALUES (1, 1, 1, 1, 1, 1, 5000.00);
INSERT INTO Fact_Transactions VALUES (2, 2, 2, 2, 2, 2, 2000.00);
INSERT INTO Fact_Transactions VALUES (3, 3, 3, 3, 3, 3, 3000.00);
INSERT INTO Fact_Transactions VALUES (4, 4, 4, 4, 4, 4, 1500.00);
INSERT INTO Fact_Transactions VALUES (5, 5, 5, 5, 5, 5, 1000.00);

-- Insert Sample Data into Fact_Loans
INSERT INTO Fact_Loans VALUES (1, 1, 3, 1, 1, 20000.00, 5000.00);
INSERT INTO Fact_Loans VALUES (2, 2, 4, 2, 2, 15000.00, 3000.00);
INSERT INTO Fact_Loans VALUES (3, 3, 2, 3, 3, 25000.00, 7000.00);
INSERT INTO Fact_Loans VALUES (4, 4, 5, 4, 4, 10000.00, 2000.00);
INSERT INTO Fact_Loans VALUES (5, 5, 1, 5, 5, 5000.00, 1000.00);

-- Step 6: Perform Queries

-- Query 1: Total Transactions and Amount per Customer
SELECT
   C.Customer_Name,
   SUM(F.Transaction_Amount) AS Total_Transaction_Amount
FROM
   Fact_Transactions F
JOIN
   Dim_Customers C ON F.Customer_ID = C.Customer_ID
GROUP BY
   C.Customer_Name;

-- Query 2: Total Transactions and Amount per Account Type
SELECT
   A.Account_Type,
   SUM(F.Transaction_Amount) AS Total_Transaction_Amount
FROM
   Fact_Transactions F
JOIN
```

```
    Dim_Accounts A ON F.Account_ID = A.Account_ID
GROUP BY
    A.Account_Type;

-- Query 3: Total Loans and Repayments per Branch
SELECT
    B.Branch_Name,
    SUM(L.Loan_Amount) AS Total_Loan_Amount,
    SUM(L.Repayment_Amount) AS Total_Repayment_Amount
FROM
    Fact_Loans L
JOIN
    Dim_Branches B ON L.Branch_ID = B.Branch_ID
GROUP BY
    B.Branch_Name;

-- Query 4: Total Transactions per Transaction Type
SELECT
    T.Transaction_Type,
    SUM(F.Transaction_Amount) AS Total_Transaction_Amount
FROM
    Fact_Transactions F
JOIN
    Dim_Transactions T ON F.Transaction_Type_ID = T.Transaction_Type_ID
GROUP BY
    T.Transaction_Type;

-- Query 5: Total Loan Amount and Repayment per Date
SELECT
    D.Month,
    SUM(L.Loan_Amount) AS Total_Loan_Amount,
    SUM(L.Repayment_Amount) AS Total_Repayment_Amount
FROM
    Fact_Loans L
JOIN
    Dim_Date D ON L.Date_ID = D.Date_ID
GROUP BY
    D.Month;
```

Q5:Gather Business Requirements for Banking enterprise and design it using any multi-dimensional data model namely Star, Snowflake, or Galaxy schema.

STAR:

```
-- Dim_Customers (Customer details)
CREATE TABLE Dim_Customers (
    Customer_ID INT PRIMARY KEY,
    Customer_Name VARCHAR(100)
);

-- Dim_Accounts (Account Type details)
CREATE TABLE Dim_Accounts (
    Account_ID INT PRIMARY KEY,
    Account_Type VARCHAR(50) -- e.g., Savings, Checking
);

-- Dim_Transactions (Transaction Types)
CREATE TABLE Dim_Transactions (
    Transaction_Type_ID INT PRIMARY KEY,
    Transaction_Type VARCHAR(50) -- e.g., Deposit, Withdrawal, Loan Repayment
);

-- Dim_Branches (Branch details)
CREATE TABLE Dim_Branches (
    Branch_ID INT PRIMARY KEY,
    Branch_Name VARCHAR(100),
    Branch_Location VARCHAR(100)
);

-- Dim_Date (Date details)
CREATE TABLE Dim_Date (
    Date_ID INT PRIMARY KEY,
    Date DATE,
    Year INT,
    Month INT,
    Day INT,
    Quarter INT
);

-- Step 3: Create Fact Tables

-- Fact_Transactions (Transaction Data)
CREATE TABLE Fact_Transactions (
    Fact_ID INT PRIMARY KEY,
    Customer_ID INT,
    Account_ID INT,
    Transaction_Type_ID INT,
    Branch_ID INT,
    Date_ID INT,
    Transaction_Amount DECIMAL(15, 2),
    FOREIGN KEY (Customer_ID) REFERENCES Dim_Customers(Customer_ID),
    FOREIGN KEY (Account_ID) REFERENCES Dim_Accounts(Account_ID),
    FOREIGN KEY (Transaction_Type_ID) REFERENCES
Dim_Transactions(Transaction_Type_ID),
    FOREIGN KEY (Branch_ID) REFERENCES Dim_Branches(Branch_ID),
    FOREIGN KEY (Date_ID) REFERENCES Dim_Date(Date_ID)
);

-- Fact_Loans (Loan Data)
CREATE TABLE Fact_Loans (
    Loan_Fact_ID INT PRIMARY KEY,
    Customer_ID INT,
    Account_ID INT,
    Branch_ID INT,
    Date_ID INT,
    Loan_Amount DECIMAL(15, 2),
    Repayment_Amount DECIMAL(15, 2),
    FOREIGN KEY (Customer_ID) REFERENCES Dim_Customers(Customer_ID),
    FOREIGN KEY (Account_ID) REFERENCES Dim_Accounts(Account_ID),
    FOREIGN KEY (Branch_ID) REFERENCES Dim_Branches(Branch_ID),
    FOREIGN KEY (Date_ID) REFERENCES Dim_Date(Date_ID)
);

-- Step 4: Insert Sample Data

-- Insert Customers
INSERT INTO Dim_Customers VALUES (1, 'John Doe');
INSERT INTO Dim_Customers VALUES (2, 'Jane Smith');
INSERT INTO Dim_Customers VALUES (3, 'Mike Johnson');
INSERT INTO Dim_Customers VALUES (4, 'Emily Davis');
INSERT INTO Dim_Customers VALUES (5, 'Chris Lee');

-- Insert Accounts
INSERT INTO Dim_Accounts VALUES (1, 'Savings');
INSERT INTO Dim_Accounts VALUES (2, 'Checking');
INSERT INTO Dim_Accounts VALUES (3, 'Loan');
INSERT INTO Dim_Accounts VALUES (4, 'Credit');
```

```sql
INSERT INTO Dim_Accounts VALUES (5, 'Investment');

-- Insert Transaction Types
INSERT INTO Dim_Transactions VALUES (1, 'Deposit');
INSERT INTO Dim_Transactions VALUES (2, 'Withdrawal');
INSERT INTO Dim_Transactions VALUES (3, 'Loan Repayment');
INSERT INTO Dim_Transactions VALUES (4, 'Transfer');
INSERT INTO Dim_Transactions VALUES (5, 'Interest Payment');

-- Insert Branches
INSERT INTO Dim_Branches VALUES (1, 'Downtown Branch', 'New York');
INSERT INTO Dim_Branches VALUES (2, 'West End Branch', 'Los Angeles');
INSERT INTO Dim_Branches VALUES (3, 'Midtown Branch', 'Chicago');
INSERT INTO Dim_Branches VALUES (4, 'East Side Branch', 'Houston');
INSERT INTO Dim_Branches VALUES (5, 'South Bay Branch', 'San Francisco');

-- Insert Date Details
INSERT INTO Dim_Date VALUES (1, '2024-01-01', 2024, 1, 1, 1);
INSERT INTO Dim_Date VALUES (2, '2024-01-02', 2024, 1, 2, 1);
INSERT INTO Dim_Date VALUES (3, '2024-02-15', 2024, 2, 15, 1);
INSERT INTO Dim_Date VALUES (4, '2024-03-10', 2024, 3, 10, 1);
INSERT INTO Dim_Date VALUES (5, '2024-04-20', 2024, 4, 20, 2);

-- Insert Data into Fact Tables
INSERT INTO Fact_Transactions VALUES (1, 1, 1, 1, 1, 1, 5000.00);
INSERT INTO Fact_Transactions VALUES (2, 2, 2, 2, 2, 2, 2000.00);
INSERT INTO Fact_Transactions VALUES (3, 3, 3, 3, 3, 3, 3000.00);
INSERT INTO Fact_Transactions VALUES (4, 4, 4, 4, 4, 4, 1500.00);
INSERT INTO Fact_Transactions VALUES (5, 5, 5, 5, 5, 5, 1000.00);

INSERT INTO Fact_Loans VALUES (1, 1, 3, 1, 1, 20000.00, 5000.00);
INSERT INTO Fact_Loans VALUES (2, 2, 4, 2, 2, 15000.00, 3000.00);
INSERT INTO Fact_Loans VALUES (3, 3, 2, 3, 3, 25000.00, 7000.00);
INSERT INTO Fact_Loans VALUES (4, 4, 5, 4, 4, 10000.00, 2000.00);
INSERT INTO Fact_Loans VALUES (5, 5, 1, 5, 5, 5000.00, 1000.00);

-- Step 5: Execute Queries

-- Query 1: Total Transaction Amount per Customer
SELECT C.Customer_Name, SUM(F.Transaction_Amount) AS Total_Transaction_Amount
FROM Fact_Transactions F
JOIN Dim_Customers C ON F.Customer_ID = C.Customer_ID
GROUP BY C.Customer_Name;
```

```
-- Query 2: Total Transaction Amount per Account Type
SELECT A.Account_Type, SUM(F.Transaction_Amount) AS Total_Transaction_Amount
FROM Fact_Transactions F
JOIN Dim_Accounts A ON F.Account_ID = A.Account_ID
GROUP BY A.Account_Type;

-- Query 3: Total Loan Amount and Repayment per Branch
SELECT B.Branch_Name, SUM(L.Loan_Amount) AS Total_Loan_Amount,
SUM(L.Repayment_Amount) AS Total_Repayment_Amount
FROM Fact_Loans L
JOIN Dim_Branches B ON L.Branch_ID = B.Branch_ID
GROUP BY B.Branch_Name;

-- Query 4: Total Transactions per Transaction Type
SELECT T.Transaction_Type, SUM(F.Transaction_Amount) AS Total_Transaction_Amount
FROM Fact_Transactions F
JOIN Dim_Transactions T ON F.Transaction_Type_ID = T.Transaction_Type_ID
GROUP BY T.Transaction_Type;

-- Query 5: Loan Amount and Repayment per Date
SELECT D.Month, SUM(L.Loan_Amount) AS Total_Loan_Amount,
SUM(L.Repayment_Amount) AS Total_Repayment_Amount
FROM Fact_Loans L
JOIN Dim_Date D ON L.Date_ID = D.Date_ID
GROUP BY D.Month;
```

SNOWFLAKE

```
-- Dim_Customers (Customer details)
CREATE TABLE Dim_Customers (
   Customer_ID INT PRIMARY KEY,
   Customer_Name VARCHAR(100)
);

-- Dim_Accounts (Account Type details)
CREATE TABLE Dim_Accounts (
   Account_ID INT PRIMARY KEY,
   Account_Type VARCHAR(50)
);

-- Dim_Transactions (Transaction Types)
CREATE TABLE Dim_Transactions (
   Transaction_Type_ID INT PRIMARY KEY,
   Transaction_Type VARCHAR(50)
);

-- Dim_Branch_Location (Branch Location details)
CREATE TABLE Dim_Branch_Location (
   Location_ID INT PRIMARY KEY,
   Branch_Location VARCHAR(100)
);

-- Dim_Branches (Branch details, now referencing Dim_Branch_Location)
CREATE TABLE Dim_Branches (
   Branch_ID INT PRIMARY KEY,
   Branch_Name VARCHAR(100),
   Location_ID INT,
   FOREIGN KEY (Location_ID) REFERENCES Dim_Branch_Location(Location_ID)
);

-- Dim_Date (Date details)
CREATE TABLE Dim_Date (
   Date_ID INT PRIMARY KEY,
   Date DATE,
   Year INT,
   Month INT,
   Day INT,
   Quarter INT
);

-- Step 3: Create Fact Tables

-- Fact_Transactions (Transaction Data)
CREATE TABLE Fact_Transactions (
   Fact_ID INT PRIMARY KEY,
   Customer_ID INT,
   Account_ID INT,
   Transaction_Type_ID INT,
   Branch_ID INT,
   Date_ID INT,
   Transaction_Amount DECIMAL(15, 2),
   FOREIGN KEY (Customer_ID) REFERENCES Dim_Customers(Customer_ID),
   FOREIGN KEY (Account_ID) REFERENCES Dim_Accounts(Account_ID),
   FOREIGN KEY (Transaction_Type_ID) REFERENCES
Dim_Transactions(Transaction_Type_ID),
   FOREIGN KEY (Branch_ID) REFERENCES Dim_Branches(Branch_ID),
   FOREIGN KEY (Date_ID) REFERENCES Dim_Date(Date_ID)
);

-- Fact_Loans (Loan Data)
CREATE TABLE Fact_Loans (
   Loan_Fact_ID INT PRIMARY KEY,
   Customer_ID INT,
   Account_ID INT,
   Branch_ID INT,
   Date_ID INT,
   Loan_Amount DECIMAL(15, 2),
   Repayment_Amount DECIMAL(15, 2),
   FOREIGN KEY (Customer_ID) REFERENCES Dim_Customers(Customer_ID),
   FOREIGN KEY (Account_ID) REFERENCES Dim_Accounts(Account_ID),
   FOREIGN KEY (Branch_ID) REFERENCES Dim_Branches(Branch_ID),
   FOREIGN KEY (Date_ID) REFERENCES Dim_Date(Date_ID)
);

-- Step 4: Insert Sample Data

-- Insert Customers
INSERT INTO Dim_Customers VALUES (1, 'John Doe');
INSERT INTO Dim_Customers VALUES (2, 'Jane Smith');
INSERT INTO Dim_Customers VALUES (3, 'Mike Johnson');
INSERT INTO Dim_Customers VALUES (4, 'Emily Davis');
INSERT INTO Dim_Customers VALUES (5, 'Chris Lee');

-- Insert Accounts
INSERT INTO Dim_Accounts VALUES (1, 'Savings');
INSERT INTO Dim_Accounts VALUES (2, 'Checking');
INSERT INTO Dim_Accounts VALUES (3, 'Loan');
INSERT INTO Dim_Accounts VALUES (4, 'Credit');
INSERT INTO Dim_Accounts VALUES (5, 'Investment');

-- Insert Transaction Types
INSERT INTO Dim_Transactions VALUES (1, 'Deposit');
INSERT INTO Dim_Transactions VALUES (2, 'Withdrawal');
INSERT INTO Dim_Transactions VALUES (3, 'Loan Repayment');
INSERT INTO Dim_Transactions VALUES (4, 'Transfer');
INSERT INTO Dim_Transactions VALUES (5, 'Interest Payment');

-- Insert Branch Locations
INSERT INTO Dim_Branch_Location VALUES (1, 'New York');
INSERT INTO Dim_Branch_Location VALUES (2, 'Los Angeles');
INSERT INTO Dim_Branch_Location VALUES (3, 'Chicago');
INSERT INTO Dim_Branch_Location VALUES (4, 'Houston');
INSERT INTO Dim_Branch_Location VALUES (5, 'San Francisco');

-- Insert Branches
INSERT INTO Dim_Branches VALUES (1, 'Downtown Branch', 1);
INSERT INTO Dim_Branches VALUES (2, 'West End Branch', 2);
INSERT INTO Dim_Branches VALUES (3, 'Midtown Branch', 3);
INSERT INTO Dim_Branches VALUES (4, 'East Side Branch', 4);
INSERT INTO Dim_Branches VALUES (5, 'South Bay Branch', 5);

-- Insert Date Details
INSERT INTO Dim_Date VALUES (1, '2024-01-01', 2024, 1, 1, 1);
INSERT INTO Dim_Date VALUES (2, '2024-01-02', 2024, 1, 2, 1);
INSERT INTO Dim_Date VALUES (3, '2024-02-15', 2024, 2, 15, 1);
INSERT INTO Dim_Date VALUES (4, '2024-03-10', 2024, 3, 10, 1);
INSERT INTO Dim_Date VALUES (5, '2024-04-20', 2024, 4, 20, 2);

-- Insert Data into Fact Tables
INSERT INTO Fact_Transactions VALUES (1, 1, 1, 1, 1, 1, 5000.00);
INSERT INTO Fact_Transactions VALUES (2, 2, 2, 2, 2, 2, 2000.00);
INSERT INTO Fact_Transactions VALUES (3, 3, 3, 3, 3, 3, 3000.00);
INSERT INTO Fact_Transactions VALUES (4, 4, 4, 4, 4, 4, 1500.00);
INSERT INTO Fact_Transactions VALUES (5, 5, 5, 5, 5, 5, 1000.00);

INSERT INTO Fact_Loans VALUES (1, 1, 3, 1, 1, 20000.00, 5000.00);
INSERT INTO Fact_Loans VALUES (2, 2, 4, 2, 2, 15000.00, 3000.00);
INSERT INTO Fact_Loans VALUES (3, 3, 2, 3, 3, 25000.00, 7000.00);
INSERT INTO Fact_Loans VALUES (4, 4, 5, 4, 4, 10000.00, 2000.00);
INSERT INTO Fact_Loans VALUES (5, 5, 1, 5, 5, 5000.00, 1000.00);

-- Step 5: Execute Queries

-- Query 1: Total Transaction Amount per Customer
SELECT C.Customer_Name, SUM(F.Transaction_Amount) AS Total_Transaction_Amount
FROM Fact_Transactions F
JOIN Dim_Customers C ON F.Customer_ID = C.Customer_ID
GROUP BY C.Customer_Name;

-- Query 2: Total Transaction Amount per Account Type
SELECT A.Account_Type, SUM(F.Transaction_Amount) AS Total_Transaction_Amount
FROM Fact_Transactions F
JOIN Dim_Accounts A ON F.Account_ID = A.Account_ID
GROUP BY A.Account_Type;
```

```
-- Query 3: Total Loan Amount and Repayment per Branch
SELECT B.Branch_Name, SUM(L.Loan_Amount) AS Total_Loan_Amount,
SUM(L.Repayment_Amount) AS Total_Repayment_Amount
FROM Fact_Loans L
JOIN Dim_Branches B ON L.Branch_ID = B.Branch_ID
GROUP BY B.Branch_Name;

-- Query 4: Total Transactions per Transaction Type
SELECT T.Transaction_Type, SUM(F.Transaction_Amount) AS Total_Transaction_Amount
FROM Fact_Transactions F
JOIN Dim_Transactions T ON F.Transaction_Type_ID = T.Transaction_Type_ID
GROUP BY T.Transaction_Type;

-- Query 5: Loan Amount and Repayment per Date
SELECT D.Month, SUM(L.Loan_Amount) AS Total_Loan_Amount,
SUM(L.Repayment_Amount) AS Total_Repayment_Amount
FROM Fact_Loans L
JOIN Dim_Date D ON L.Date_ID = D.Date_ID
GROUP BY D.Month;
```

GALAXY

```
-- Step 1: Drop existing tables to avoid errors
DROP TABLE IF EXISTS Fact_Transactions;
DROP TABLE IF EXISTS Fact_Loans;
DROP TABLE IF EXISTS Dim_Customers;
DROP TABLE IF EXISTS Dim_Accounts;
DROP TABLE IF EXISTS Dim_Transactions;
DROP TABLE IF EXISTS Dim_Branches;
DROP TABLE IF EXISTS Dim_Branch_Location;
DROP TABLE IF EXISTS Dim_Date;

-- Step 2: Create Dimension Tables (same as Snowflake Schema)
CREATE TABLE Dim_Customers (
   Customer_ID INT PRIMARY KEY,
   Customer_Name VARCHAR(100)
);

CREATE TABLE Dim_Accounts (
```

```sql
    Account_ID INT PRIMARY KEY,
    Account_Type VARCHAR(50)
);

CREATE TABLE Dim_Transactions (
    Transaction_Type_ID INT PRIMARY KEY,
    Transaction_Type VARCHAR(50)
);

CREATE TABLE Dim_Branch_Location (
    Location_ID INT PRIMARY KEY,
    Branch_Location VARCHAR(100)
);

CREATE TABLE Dim_Branches (
    Branch_ID INT PRIMARY KEY,
    Branch_Name VARCHAR(100),
    Location_ID INT,
    FOREIGN KEY (Location_ID) REFERENCES Dim_Branch_Location(Location_ID)
);

CREATE TABLE Dim_Date (
    Date_ID INT PRIMARY KEY,
    Date DATE,
    Year INT,
    Month INT,
    Day INT,
    Quarter INT
);

-- Step 3: Create Fact Tables (Multiple Fact Tables in Galaxy Schema)

-- Fact_Transactions (Transaction Data)
CREATE TABLE Fact_Transactions (
    Fact_ID INT PRIMARY KEY,
    Customer_ID INT,
    Account_ID INT,
    Transaction_Type_ID INT,
    Branch_ID INT,
    Date_ID INT,
    Transaction_Amount DECIMAL(15, 2),
    FOREIGN KEY (Customer_ID) REFERENCES Dim_Customers(Customer_ID),
    FOREIGN KEY (Account_ID) REFERENCES Dim_Accounts(Account_ID),
    FOREIGN KEY (Transaction_Type_ID) REFERENCES
Dim_Transactions(Transaction_Type_ID),
    FOREIGN KEY (Branch_ID) REFERENCES Dim_Branches(Branch_ID),
    FOREIGN KEY (Date_ID) REFERENCES Dim_Date(Date_ID)
);

-- Fact_Loans (Loan Data)
CREATE TABLE Fact_Loans (
    Loan_Fact_ID INT PRIMARY KEY,
    Customer_ID INT,
    Account_ID INT,
    Branch_ID INT,
    Date_ID INT,
    Loan_Amount DECIMAL(15, 2),
    Repayment_Amount DECIMAL(15, 2),
    FOREIGN KEY (Customer_ID) REFERENCES Dim_Customers(Customer_ID),
    FOREIGN KEY (Account_ID) REFERENCES Dim_Accounts(Account_ID),
    FOREIGN KEY (Branch_ID) REFERENCES Dim_Branches(Branch_ID),
    FOREIGN KEY (Date_ID) REFERENCES Dim_Date(Date_ID)
);

-- Step 4: Insert Sample Data

-- Insert Customers
INSERT INTO Dim_Customers VALUES (1, 'John Doe');
INSERT INTO Dim_Customers VALUES (2, 'Jane Smith');
INSERT INTO Dim_Customers VALUES (3, 'Mike Johnson');
INSERT INTO Dim_Customers VALUES (4, 'Emily Davis');
INSERT INTO Dim_Customers VALUES (5, 'Chris Lee');

-- Insert Accounts
INSERT INTO Dim_Accounts VALUES (1, 'Savings');
INSERT INTO Dim_Accounts VALUES (2, 'Checking');
INSERT INTO Dim_Accounts VALUES (3, 'Loan');
INSERT INTO Dim_Accounts VALUES (4, 'Credit');
INSERT INTO Dim_Accounts VALUES (5, 'Investment');

-- Insert Transaction Types
INSERT INTO Dim_Transactions VALUES (1, 'Deposit');
INSERT INTO Dim_Transactions VALUES (2, 'Withdrawal');
INSERT INTO Dim_Transactions VALUES (3, 'Loan Repayment');
INSERT INTO Dim_Transactions VALUES (4, 'Transfer');
INSERT INTO Dim_Transactions VALUES (5, 'Interest Payment');
```

```
-- Insert Branch Locations
INSERT INTO Dim_Branch_Location VALUES (1, 'New York');
INSERT INTO Dim_Branch_Location VALUES (2, 'Los Angeles');
INSERT INTO Dim_Branch_Location VALUES (3, 'Chicago');
INSERT INTO Dim_Branch_Location VALUES (4, 'Houston');
INSERT INTO Dim_Branch_Location VALUES (5, 'San Francisco');

-- Insert Branches
INSERT INTO Dim_Branches VALUES (1, 'Downtown Branch', 1);
INSERT INTO Dim_Branches VALUES (2, 'West End Branch', 2);
INSERT INTO Dim_Branches VALUES (3, 'Midtown Branch', 3);
INSERT INTO Dim_Branches VALUES (4, 'East Side Branch', 4);
INSERT INTO Dim_Branches VALUES (5, 'South Bay Branch', 5);

-- Insert Date Details
INSERT INTO Dim_Date VALUES (1, '2024-01-01', 2024, 1, 1, 1);
INSERT INTO Dim_Date VALUES (2, '2024-01-02', 2024, 1, 2, 1);
INSERT INTO Dim_Date VALUES (3, '2024-02-15', 2024, 2, 15, 1);
INSERT INTO Dim_Date VALUES (4, '2024-03-10', 2024, 3, 10, 1);
INSERT INTO Dim_Date VALUES (5, '2024-04-20', 2024, 4, 20, 2);

-- Insert Data into Fact Tables
INSERT INTO Fact_Transactions VALUES (1, 1, 1, 1, 1, 1, 5000.00);
INSERT INTO Fact_Transactions VALUES (2, 2, 2, 2, 2, 2, 2000.00);
INSERT INTO Fact_Transactions VALUES (3, 3, 3, 3, 3, 3, 3000.00);
INSERT INTO Fact_Transactions VALUES (4, 4, 4, 4, 4, 4, 1500.00);
INSERT INTO Fact_Transactions VALUES (5, 5, 5, 5, 5, 5, 1000.00);

INSERT INTO Fact_Loans VALUES (1, 1, 3, 1, 1, 20000.00, 5000.00);
INSERT INTO Fact_Loans VALUES (2, 2, 4, 2, 2, 15000.00, 3000.00);
INSERT INTO Fact_Loans VALUES (3, 3, 2, 3, 3, 25000.00, 7000.00);
INSERT INTO Fact_Loans VALUES (4, 4, 5, 4, 4, 10000.00, 2000.00);
INSERT INTO Fact_Loans VALUES (5, 5, 1, 5, 5, 5000.00, 1000.00);

-- Step 5: Execute Queries

-- Query 1: Total Transaction Amount per Customer
SELECT C.Customer_Name, SUM(F.Transaction_Amount) AS Total_Transaction_Amount
FROM Fact_Transactions F
JOIN Dim_Customers C ON F.Customer_ID = C.Customer_ID
GROUP BY C.Customer_Name;

-- Query 2: Total Transaction Amount per Account Type
SELECT A.Account_Type, SUM(F.Transaction_Amount) AS Total_Transaction_Amount
```

```
FROM Fact_Transactions F
JOIN Dim_Accounts A ON F.Account_ID = A.Account_ID
GROUP BY A.Account_Type;

-- Query 3: Total Loan Amount and Repayment per Branch
SELECT B.Branch_Name, SUM(L.Loan_Amount) AS Total_Loan_Amount,
SUM(L.Repayment_Amount) AS Total_Repayment_Amount
FROM Fact_Loans L
JOIN Dim_Branches B ON L.Branch_ID = B.Branch_ID
GROUP BY B.Branch_Name;

-- Query 4: Total Transactions per Transaction Type
SELECT T.Transaction_Type, SUM(F.Transaction_Amount) AS Total_Transaction_Amount
FROM Fact_Transactions F
JOIN Dim_Transactions T ON F.Transaction_Type_ID = T.Transaction_Type_ID
GROUP BY T.Transaction_Type;

-- Query 5: Loan Amount and Repayment per Date
SELECT D.Month, SUM(L.Loan_Amount) AS Total_Loan_Amount,
SUM(L.Repayment_Amount) AS Total_Repayment_Amount
FROM Fact_Loans L
JOIN Dim_Date D ON L.Date_ID = D.Date_ID
GROUP BY D.Month;
```