Name : Bhoomi Mangesh Naik
Class : D15C
Roll No. : 60
Practical No. : 1

Aim : Design a Star and Snowflake Schema for the given system.

Theory :

1. Data Warehousing

- A **Data Warehouse (DW)** is a central repository of integrated data from multiple sources.
- It supports **querying, reporting, analysis, and decision-making**.

2. Schemas in Data Warehousing

Schemas are logical structures that define how data is organized in a data warehouse.

Star Schema

- The **central fact table** contains quantitative data (measurable values).
- **Dimension tables** surround it, storing descriptive attributes.
- The structure looks like a **star**, hence the name.
- **Advantages**: Simple design, fast query performance.
- **Disadvantages**: Data redundancy in dimensions.

Snowflake Schema

- Extension of star schema where **dimension tables are normalized**.
- Each dimension can be split into sub-dimensions.
- Structure resembles a **snowflake**.
- **Advantages**: Reduces redundancy, better storage efficiency.
- **Disadvantages**: More complex queries, slightly slower performance.

Syntax (SQL for Schema Creation)

Example: Star Schema (Sales Data Warehouse)

```
-- Fact Table

CREATE TABLE FactSales (

    SaleID INT PRIMARY KEY,
```

```
    DateID INT,
    ProductID INT,
    CustomerID INT,
    StoreID INT,
    SalesAmount DECIMAL(10,2),
    Quantity INT
);
-- Dimension Tables
CREATE TABLE DimDate (
    DateID INT PRIMARY KEY,
    Date DATE,
    Month VARCHAR(20),
    Quarter VARCHAR(10),
    Year INT
);
CREATE TABLE DimProduct (
    ProductID INT PRIMARY KEY,
    ProductName VARCHAR(100),
    Category VARCHAR(50),
    Brand VARCHAR(50)
);
CREATE TABLE DimCustomer (
    CustomerID INT PRIMARY KEY,
    CustomerName VARCHAR(100),
    Region VARCHAR(50),
```

```
    AgeGroup VARCHAR(20)
);
CREATE TABLE DimStore (
    StoreID INT PRIMARY KEY,
    StoreName VARCHAR(100),
    Location VARCHAR(100)
);
```

Example: Snowflake Schema (Normalized Dimensions)

```
-- Split product dimension
CREATE TABLE DimCategory (
    CategoryID INT PRIMARY KEY,
    CategoryName VARCHAR(50)
);
CREATE TABLE DimBrand (
    BrandID INT PRIMARY KEY,
    BrandName VARCHAR(50)
);
CREATE TABLE DimProduct (
    ProductID INT PRIMARY KEY,
    ProductName VARCHAR(100),
    CategoryID INT,
    BrandID INT,
    FOREIGN KEY (CategoryID) REFERENCES DimCategory(CategoryID),
    FOREIGN KEY (BrandID) REFERENCES DimBrand(BrandID)
);
```

Widgets and Properties (Applicable in Visualization / Design Tools)

When you design schemas in tools like **Power BI, Tableau, SSMS ERD Designer, or MySQL Workbench**, you use **widgets** (entities, relationships, connectors).

Common Widgets

1. **Entity (Table Widget)**
    - Represents **Fact or Dimension tables**.
    - Properties:
        - Table Name
        - Attributes (columns, datatype, keys)
          Primary/Foreign Keys

2. **Relationship / Connector Widget**
    - Represents **links between Fact and Dimension tables**.
    - Properties:
        - Cardinality (1:1, 1:N)
        - Join type (Inner, Left)

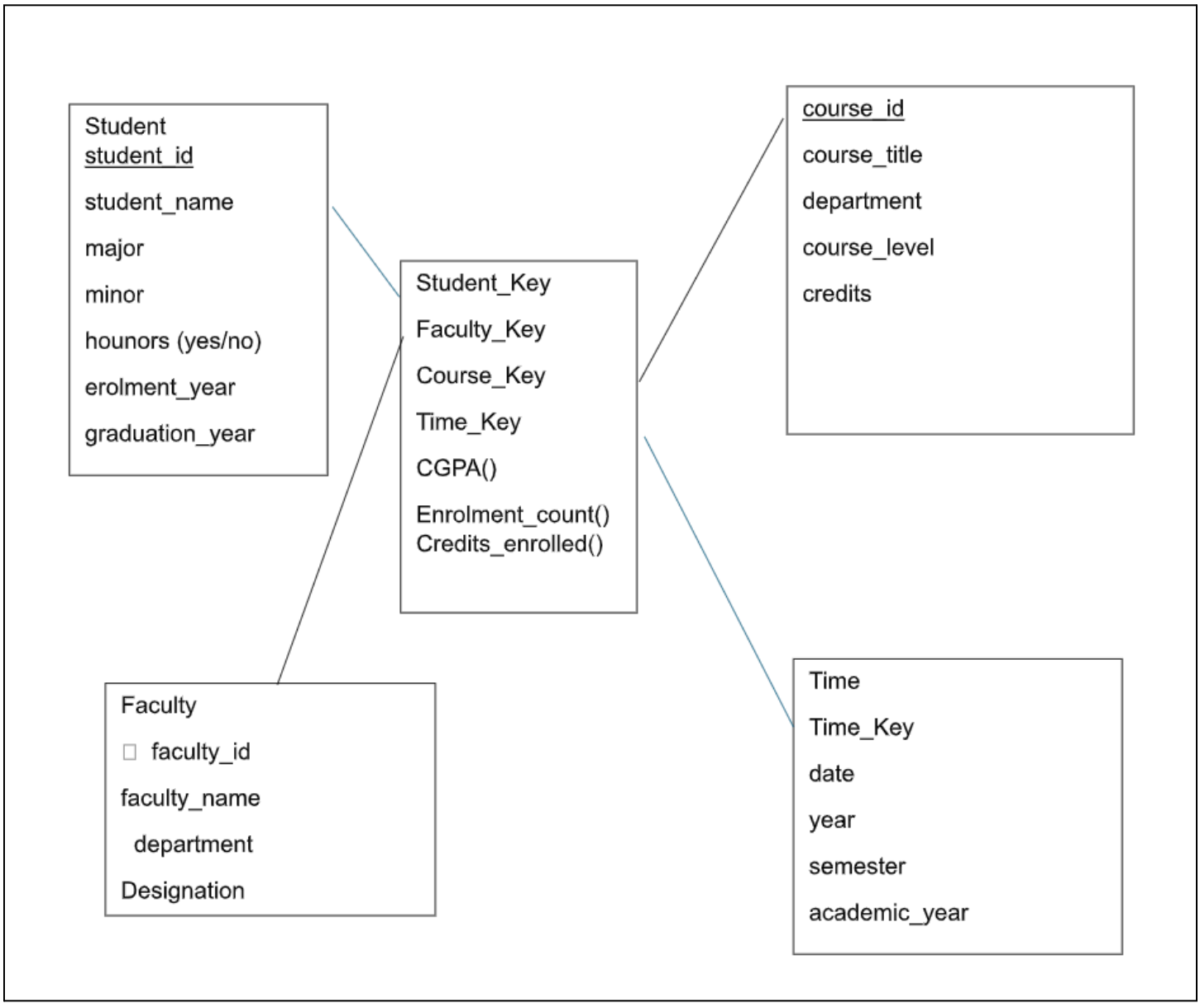
3. **Attribute Properties**
    - Name
    - Data Type (INT, VARCHAR, DATE, DECIMAL)
    - Constraint (PK, FK, NOT NULL, UNIQUE)

4. **Schema View Widget**
    - Allows you to toggle between **Star Schema View** and **Snowflake Schema View**.

Output :

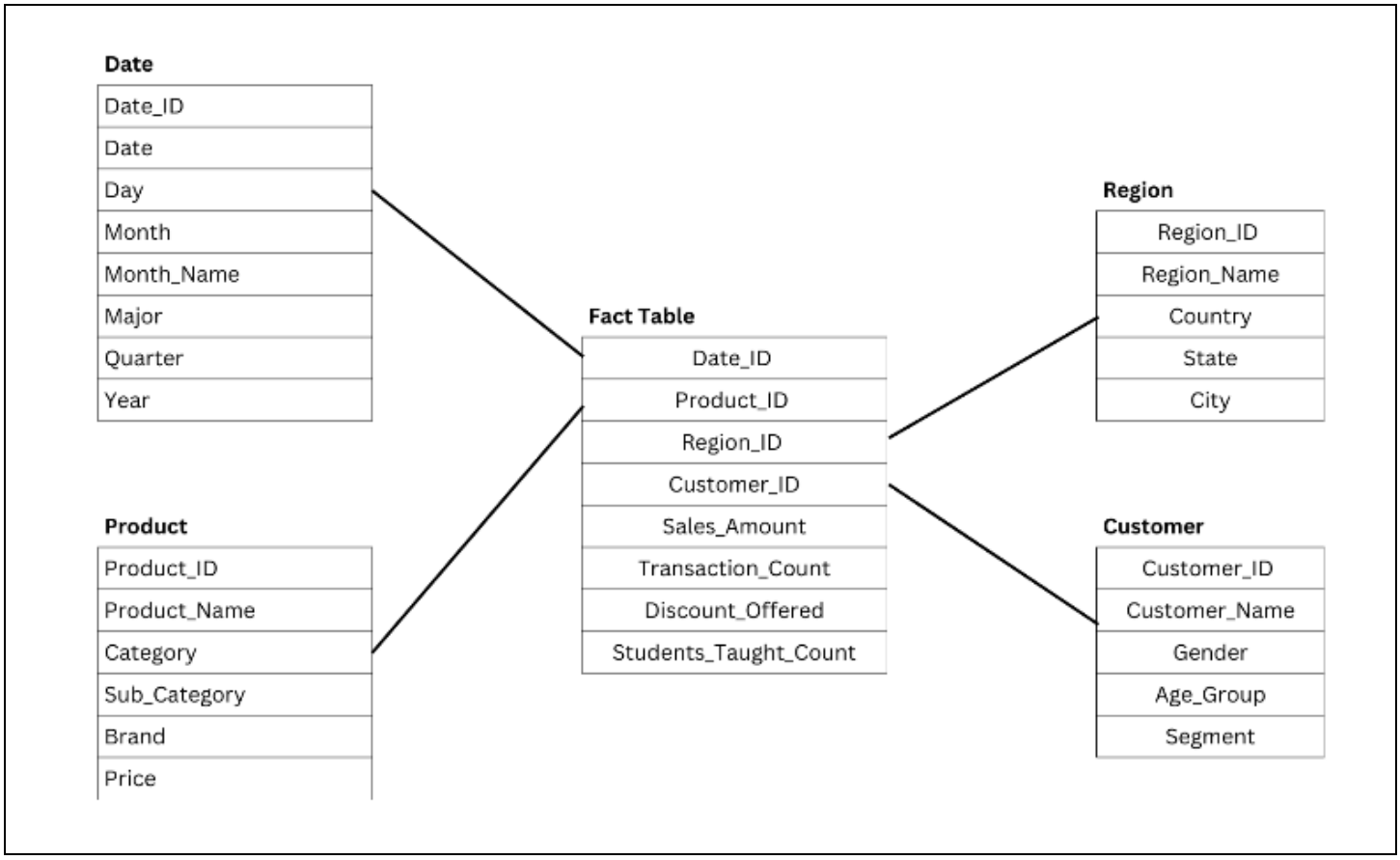
1. **A university wants to design a data warehouse to analyze student performance, course enrollments, and faculty workload. The university's operational database is highly normalized, making it difficult to perform analytical queries.**

Star schema :



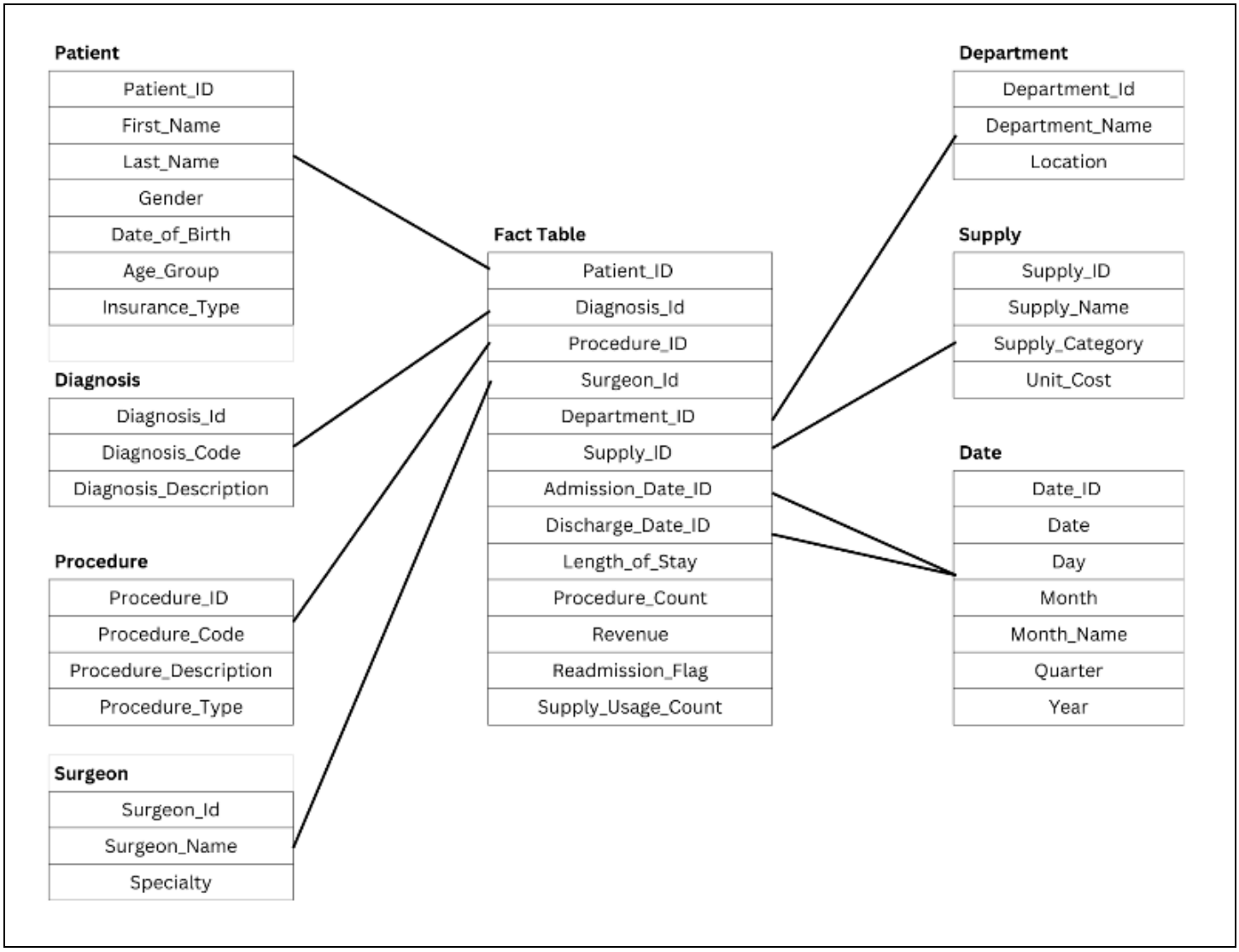
2. A retail company wants to analyze its sales performance across different regions, time periods, products, and customer segments. The company wants to track total sales, number of transactions, and discounts offered.

Star schema :



3. A hospital management wants to create a data warehouse to analyze patient admissions, procedures, and billing information. The goal is to improve operational efficiency and patient care by answering questions such as:

Star Schema :



1. **What is the average length of stay for patients with a specific diagnosis?**

Ans. :

```
SELECT
   d.Diagnosis_Description,
   AVG(f.Length_of_Stay) AS Avg_Length_of_Stay
FROM Fact_Patient_Admissions f
JOIN Dim_Diagnosis d ON f.Diagnosis_ID = d.Diagnosis_ID
WHERE d.Diagnosis_Description = 'Pneumonia'
GROUP BY d.Diagnosis_Description;
```

2. **How many surgical procedures were performed by each surgeon last month?**

Ans. :

```
SELECT
   s.Surgeon_Name,
   SUM(f.Procedure_Count) AS Total_Surgical_Procedures
```

```
FROM Fact_Patient_Admissions f
JOIN Dim_Procedure p ON f.Procedure_ID = p.Procedure_ID
JOIN Dim_Surgeon s ON f.Surgeon_ID = s.Surgeon_ID
JOIN Dim_Date dt ON f.Admission_Date_ID = dt.Date_ID
WHERE p.Procedure_Type = 'Surgical'
  AND dt.Month = 7
  AND dt.Year = 2025
GROUP BY s.Surgeon_Name
ORDER BY Total_Surgical_Procedures DESC;
```

3. **What is the total revenue generated by a particular department (e.g., Cardiology, Orthopedics) per quarter?**

Ans. :

```
SELECT
    dept.Department_Name,
    dt.Quarter,
    dt.Year,
    SUM(f.Revenue) AS Total_Revenue
FROM Fact_Patient_Admissions f
JOIN Dim_Department dept ON f.Department_ID = dept.Department_ID
JOIN Dim_Date dt ON f.Admission_Date_ID = dt.Date_ID
GROUP BY dept.Department_Name, dt.Quarter, dt.Year
ORDER BY dt.Year, dt.Quarter, Total_Revenue DESC;
```

4. **Which medical supplies are most frequently used in the emergency department?**

Ans. :

```
SELECT
    sup.Supply_Name,
    SUM(f.Supply_Usage_Count) AS Total_Usage
FROM Fact_Patient_Admissions f
JOIN Dim_Department dept ON f.Department_ID = dept.Department_ID
JOIN Dim_Supply sup ON f.Supply_ID = sup.Supply_ID
WHERE dept.Department_Name = 'Emergency'
GROUP BY sup.Supply_Name
ORDER BY Total_Usage DESC;
```

5. **What is the readmission rate for patients who had a certain procedure?**

Ans. :

```
SELECT
    p.Procedure_Description,
```

```
    (SUM(f.Readmission_Flag) * 100.0 / COUNT(*)) AS Readmission_Rate_Percent
FROM Fact_Patient_Admissions f
JOIN Dim_Procedure p ON f.Procedure_ID = p.Procedure_ID
WHERE p.Procedure_Description = 'Heart Bypass Surgery'
GROUP BY p.Procedure_Description;
```

Conclusion :

The star schema designed for the given system provides a simplified yet efficient structure for analytical querying and decision-making. By separating business measures into a central fact table and descriptive attributes into surrounding dimension tables, the schema ensures faster query performance, easier data navigation, and improved reporting capabilities. This design supports the organization’s analytical needs by enabling multidimensional analysis, reducing query complexity, and enhancing data accessibility for strategic and operational insights.