Name : Krish Gupta
Div : D15C
Batch : A
Roll No : 20

DMBI Experiment 01

Aim: Design a Star Schema for the given problem statement.

Theory:

Dimensional Data Modeling is one of the data modeling techniques used in data warehouse design. The concept of Dimensional Modeling was developed by Ralph Kimball which is comprised of facts and dimension tables. Since the main goal of this modeling is to improve the data retrieval so it is optimized for SELECT OPERATION. The advantage of using this model is that we can store data in such a way that it is easier to store and retrieve the data once stored in a data warehouse. The dimensional model is the data model used by many OLAP systems.

A **star schema** is a type of data modeling technique used in data warehousing to represent data in a structured and intuitive way. In a star schema, data is organized into a central fact table that contains the measures of interest, surrounded by dimension tables that describe the attributes of the measures.

The **fact table** in a star schema contains the measures or metrics that are of interest to the user or organization. For example, in a sales data warehouse, the fact table might contain sales revenue, units sold, and profit margins. Each record in the fact table represents a specific event or transaction, such as a sale or order.

The **dimension tables** in a star schema contain the descriptive attributes of the measures in the fact table. These attributes are used to slice and dice the data in the fact table, allowing users to analyze the data from different perspectives. For example, in a sales data warehouse, the dimension tables might include product, customer, time, and location.

Each dimension table is joined to the fact table through a foreign key relationship. This allows users to query the data in the fact table using attributes from the dimension tables. For example, a user might want to see sales revenue by product category, or by region and time period.

Features of Star Schema

1.Central fact table: The star schema revolves around a central fact table that contains the numerical data being analyzed. This table contains foreign keys to link to dimension tables.

2.Dimension tables: Dimension tables are tables that contain descriptive attributes about the data being analyzed. These attributes provide context to the numerical data in the fact table. Each dimension table is linked to the fact table through a foreign key.

3.Denormalized structure: A star schema is denormalized, which means that redundancy is allowed in the schema design to improve query performance. This is because it is easier and faster to join a small number of tables than a large number of tables.

A **snowflake schema** is a type of data model where the fact table links to normalized dimension tables split into multiple related tables. It’s a more detailed version of the star schema and is used to handle complex data structures. The snowflake effect applies only to dimension tables, not the fact table.

Features of the Snowflake Schema

1.Normalization: Snowflake schema uses normalized tables to reduce redundancy and improve consistency.

2.Hierarchical Structure: Built around a central **fact table** with connected **dimension tables**.

3.Multiple Levels: Dimensions can be split into multiple levels, allowing detailed **drill-down** analysis.

Fact Constellation in Data Warehouse modeling is a schema design that integrates multiple fact tables sharing common dimensions, often referred to as a "Galaxy schema." This approach allows businesses to conduct multi-dimensional analysis across complex datasets.

Fact Constellation Schema, also known as the **Galaxy Schema**, is an advanced data modeling technique used in designing data warehouses. Unlike simpler models like the Star Schema and Snowflake Schema, the

Fact Constellation Schema consists of multiple fact tables that share common dimensional tables.

Benefits of Fact Constellation

1.Enhanced Query Performance: Fact Constellation improves query performance by organizing multiple fact tables around shared dimensions, reducing query complexity and minimizing the need for large joins, thus speeding up data retrieval.

2.Flexibility in Reporting and Analysis: It offers flexibility in reporting by connecting multiple fact tables to common dimensions, allowing dynamic and customizable aggregation of data across different business metrics.

3.Improved Scalability: Fact Constellation efficiently handles large datasets by distributing data across fact tables and shared dimensions, ensuring scalability as data volumes grow.

Challenges in Implementing Fact Constellation Schema

1.Increased Complexity in Design: Designing a Fact Constellation schema is more intricate due to multiple fact tables sharing common dimensions, requiring careful planning.

2.Performance Issues with Complex Queries: Complex queries involving multiple fact tables and joins can cause performance bottlenecks, especially with large datasets.

3.Difficulty in Maintaining Consistency: Changes in shared dimension tables can impact all related fact tables, making it challenging to maintain data consistency.

Problem Statement:

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:

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

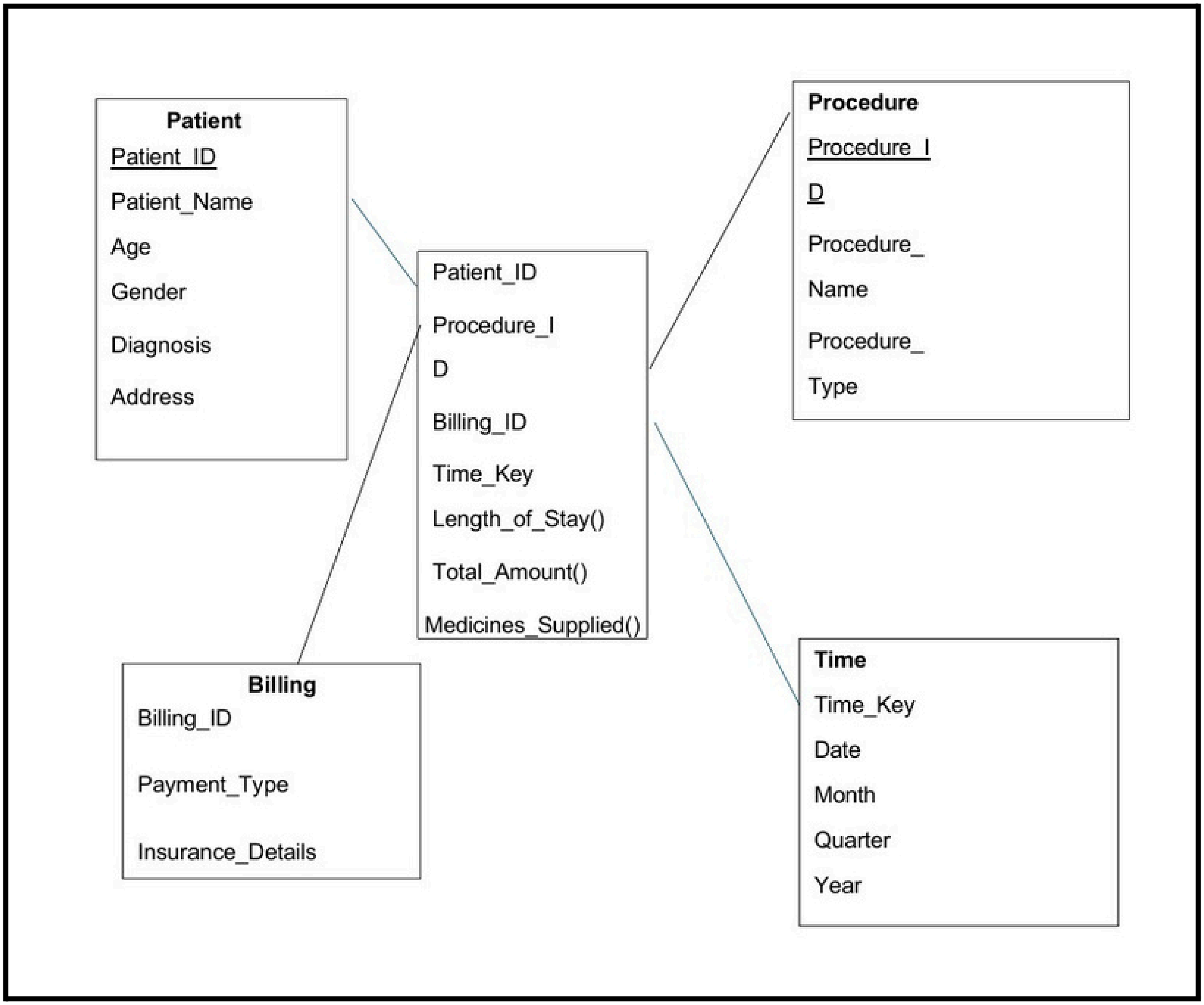
2.How many surgical procedures were performed by each surgeon last month?

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

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

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

Star Schema:



Queries:

1.SELECT Diagnosis, AVG(Length_of_Stay) FROM FactTable JOIN Patient FactTable.Patient_ID = Patient.Patient_ID WHERE Diagnosis = 'specific diagnosis' GROUP BY Diagnosis;

2.SELECT Surgeon.Name, COUNT(Procedure_ID) FROM FactTable JOIN Surgeon ON FactTable.Surgeon_ID = Surgeon.Surgeon_ID JOIN Time ON FactTable.Admission_Date_Key = Time.Time_Key WHERE Procedure_Type = 'Surgical' AND Time.Month = 'LastMonth' GROUP BY Surgeon.Name;

3.SELECT Department.Department_Name, Time.Quarter, SUM(Total_Amou FROM FactTable JOIN Department ON FactTable.Department_ID = Department.Department_ID JOIN Time ON FactTable.Admission_Date_Key = Time.Time_Key WHERE Department.Department_Name = 'Cardiology' GROUP BY Department.Department_Name, Time.Quarter;

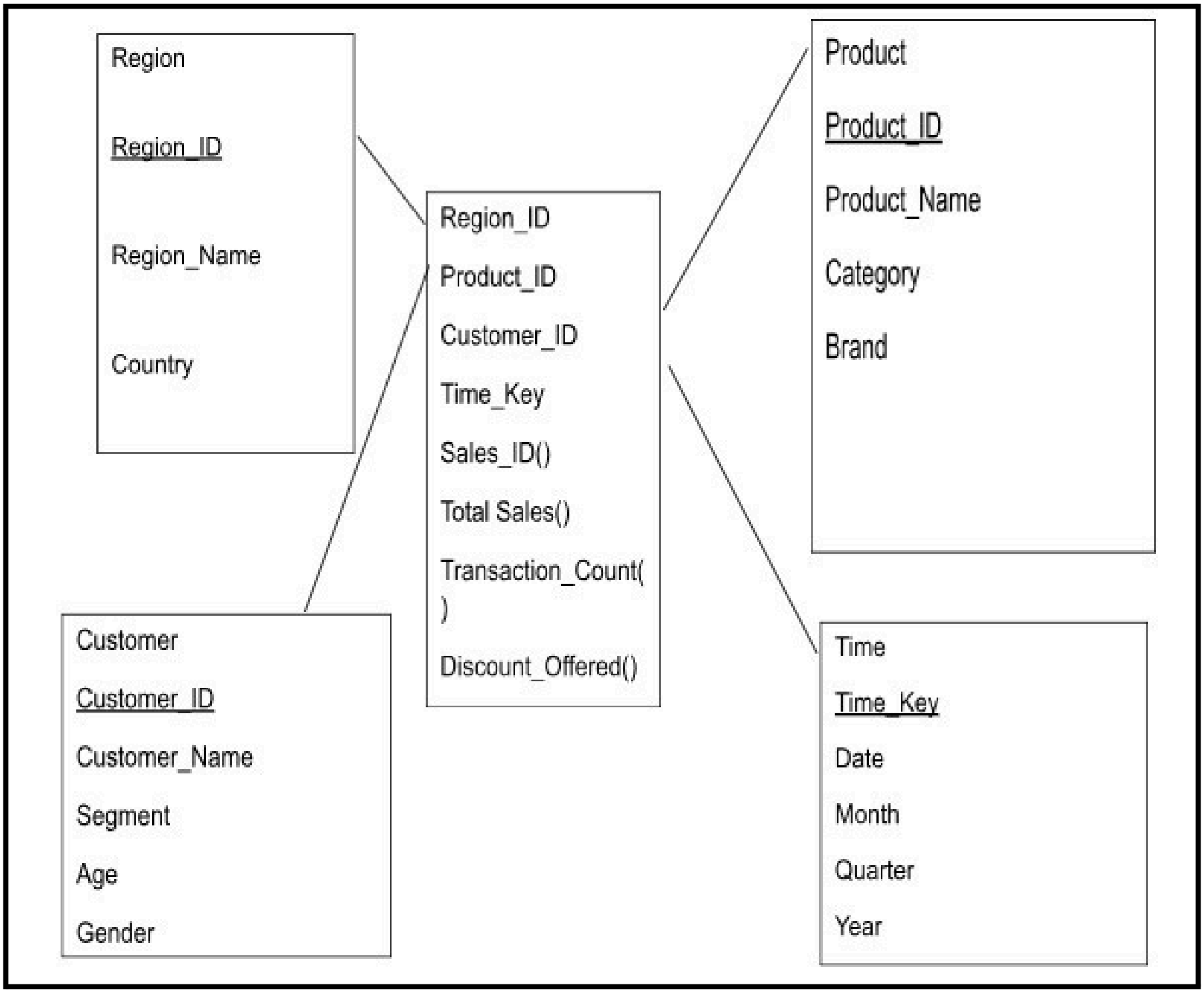
4.SELECT Medical_Supply.Supply_Name, SUM(Quantity_Supplied) FROM FactTable JOIN Medical_Supply ON FactTable.Medical_Supply_ID = Medical_Supply.Medical_Supply_ID JOIN Department ON FactTable.Department_ID = Department.Department_ID WHERE Department.Department_Name = 'Emergency' GROUP BY Medical_Supply.Supply_Name ORDER BY SUM(Quantity_Supplied) DESC;

5.SELECT Procedure.Procedure_Name, (SUM(CASE WHEN Readmission_ = 'Yes' THEN 1 ELSE 0 END)/COUNT(*)) * 100 AS Readmission_Rate FROM FactTable JOIN Procedure ON FactTable.Procedure_ID = Procedure.Procedure_ID WHERE Procedure.Procedure_Name = 'certain procedure' GROUP BY Procedure.Procedure_Name;

Problem Statement:

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 discount offered.

Star Schema:



Conclusion:

In this experiment, I studied and analyzed the concept, structure, and application of the Star Schema in data warehousing. The Star Schema, with its central fact table connected to multiple dimension tables, provides an efficient, intuitive, and performance-oriented design for analytical queries. Its denormalized nature simplifies query logic, enhances retrieval speed, and supports OLAP operations for business intelligence and reporting.

By organizing data into measurable facts and descriptive dimensions, it enables users to slice and dice information from multiple perspectives, making decision-making more data-driven. Although it has limitations in handling complex many-to-many relationships and enforcing data integrity compared to normalized schemas, its simplicity and efficiency make it one of the most widely adopted designs in data warehousing.