DMBI: EXPERIMENT - 1

Aim:

Design a Star Schema for the given system.

Theory:

Dimensional modelling is a data modelling technique designed specifically for data warehousing and business intelligence systems. It focuses on optimizing the structure of data for easy retrieval, fast query performance, and user-friendly analysis. Unlike traditional Entity-Relationship (ER) modelling, which emphasizes data normalization for transactional processing, dimensional modelling is centered on denormalization to improve query efficiency and reporting speed.
Developed and popularized by Ralph Kimball, dimensional modelling is the foundation of the Kimball methodology for data warehouse design. It simplifies complex business data into a structure that is intuitive for end users, particularly for non-technical analysts, enabling them to slice, dice, drill down, and roll up data easily.

Dimensional modelling revolves around two main components: fact tables and dimension tables.

- **Fact Tables:**
  These tables store measurable, quantitative business data such as sales revenue, profit, or quantity sold. Facts are typically numerical and additive (e.g., total sales can be summed over time or regions). Fact tables also store foreign keys that link to the relevant dimension tables.
- **Dimension Tables:**
  These store descriptive attributes (also called textual or categorical data) related to the facts. Examples include product names, customer demographics, dates, or geographic locations. Dimensions provide context to facts, enabling meaningful business analysis.

The relationship between facts and dimensions typically forms a star schema or a snowflake schema.

Dimensional modelling has several schema types, each designed for specific business and technical needs.

1. **Star Schema**
   The star schema is the simplest and most common form of dimensional model.
   a. Structure: A central fact table is surrounded by dimension tables, forming a star-like shape.
   b. Advantages: Simple design, high query performance, and easy navigation for end-users.
   c. Example: A sales fact table linked to dimension tables for Date, Product, Customer, and Region.
   d. Use Case: Best suited for smaller to medium-sized data warehouses or marts where simplicity and performance are priorities.
2. **Snowflake Schema**
   The snowflake schema is a variant of the star schema where dimension tables are further normalized into multiple related tables.
   a. Structure: Dimension tables are split into sub-dimensions to remove redundancy.
   b. Advantages: Saves storage space and maintains data integrity.
   c. Disadvantages: Slightly more complex queries due to additional joins.
   d. Example: In a sales model, the Product dimension may be split into Product, Category, and Supplier tables.
   e. Use Case: Useful when dimensions have large hierarchies or when storage efficiency is more important than query simplicity.

3. **Galaxy Schema (or Fact Constellation)**
   The galaxy schema is used when multiple fact tables share dimension tables.
   a. Structure: Multiple fact tables, possibly representing different business processes, are linked to a set of shared dimension tables.
   b. Advantages: Integrates data from different business processes in one model.
   c. Example: A Sales fact table and an Inventory fact table sharing dimensions like Date, Product, and Store.
   d. Use Case: Suitable for enterprise-level data warehouses that consolidate data from various departments.

Benefits of Dimensional Modelling:

- User Friendliness: The design is intuitive and easily understood by non-technical users.
- High Performance: Denormalization reduces the number of joins, improving query speed.
- Scalability: Can handle large volumes of data over time.
- Flexibility: Supports various analytical operations, including time-series analysis, trend monitoring, and predictive modelling.

Limitations:

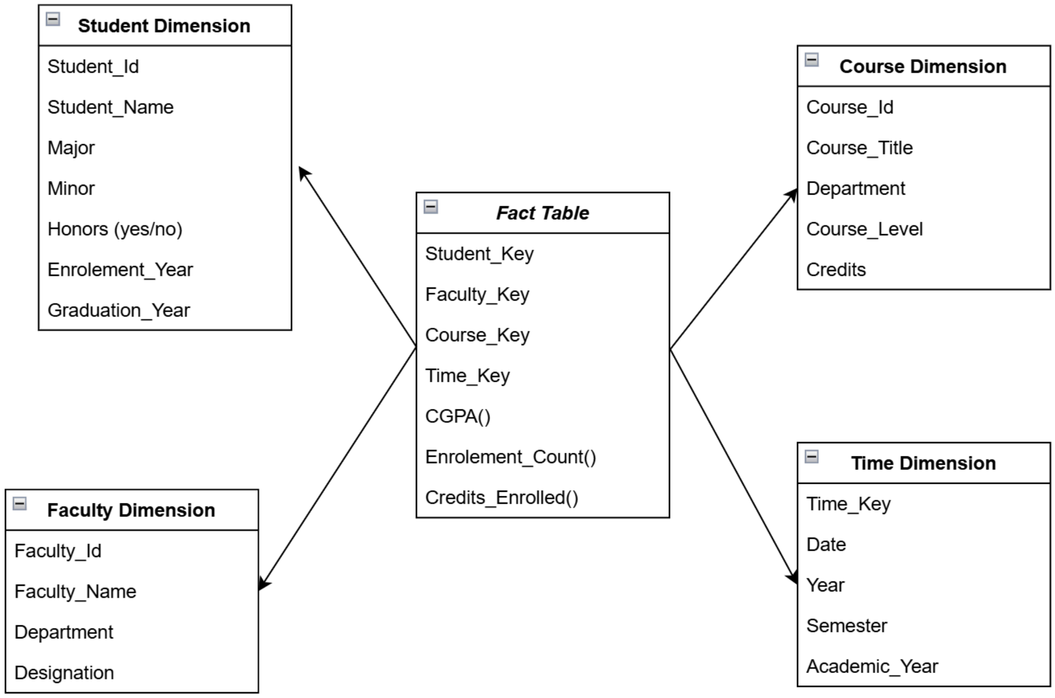
- Data Redundancy: Denormalization may lead to repeated storage of descriptive data.
- Maintenance Overhead: Updates to dimension attributes must be carefully managed.
- Storage Usage: Larger storage requirements compared to normalized models.

Conclusion:

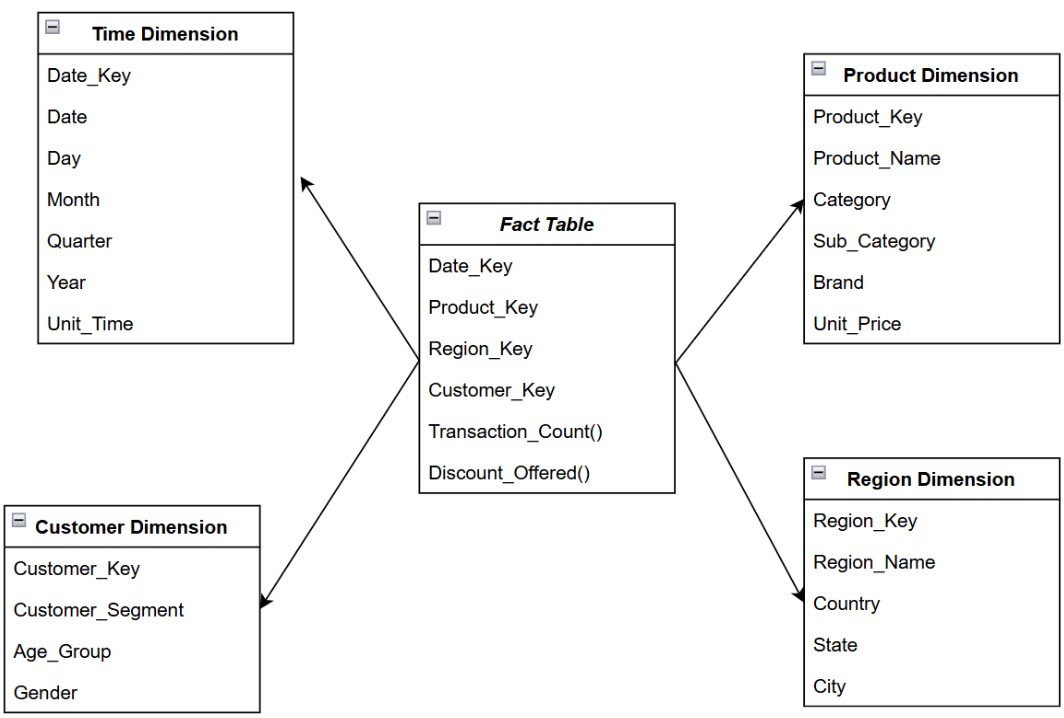
In this experiment, a star schema was designed to demonstrate the application of dimensional modelling for real-world analytical needs. The schema helped restructure complex, normalized operational data into a simplified format that supports faster queries and more intuitive analysis. Case studies across three domains—universities, retail, and hospitals—showed how separating facts and dimensions enables organizations to track performance, identify trends, and make informed decisions with greater ease.

Challenges such as redundancy and increased storage were acknowledged, but the trade-off was justified by the improvements in accessibility and reporting efficiency. Overall, the experiment highlighted how the star schema provides a practical and scalable foundation for data warehouses, making business intelligence systems both user-friendly and effective in supporting long-term analytical goals.

Q1) 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.



Q2) 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.



Q3) 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.