Dimensional Modeling Schemas

Star Schema

The **Star Schema** is a simple and commonly used data warehouse model. It consists of a central **fact table** connected to various **dimension tables** like a star shape. Each dimension table contains descriptive data (e.g., product details, geography, time) related to facts, which are numeric values (e.g., sales, revenue) in the fact table.

Key Points:

1. **Fact Table:** Stores measurable data (e.g., sales, profits) and foreign keys that link to dimension tables.
2. **Dimension Tables:** Store descriptive attributes (e.g., product names, store locations).

Example:

Suppose we have a retail data warehouse tracking store sales.

- **Fact Table:**
  - Columns: `Transaction_ID`, `Date_ID`, `Store_ID`, `Product_ID`, `Sales_Amount`
- **Dimension Tables:**
  - **Product Dimension:** `Product_ID`, `Product_Name`, `Category`, `Brand`
  - **Store Dimension:** `Store_ID`, `Store_Name`, `City`, `Region`
  - **Date Dimension:** `Date_ID`, `Day`, `Month`, `Year`

Query Example:
“How much did we sell in raincoats (product category) during January 2023 in stores located in Boston?”

You would:

- Filter the **Product Dimension** by `Category = 'Raincoats'`.
- Filter the **Store Dimension** by `City = 'Boston'`.
- Filter the **Date Dimension** by `Month = 'January'` and `Year = 2023`.
- Join these dimensions with the **Fact Table** to sum the sales amount.

Snowflake Schema

The **Snowflake Schema** is an extension of the Star Schema but has a more normalized structure. It splits dimension tables into sub-dimension tables. This leads to a snowflake-like appearance due to multiple layers of dimension tables, improving data storage efficiency but increasing the complexity of queries.

Key Points:

1. **Normalization:** Dimension tables are normalized (i.e., further broken down into sub-tables) to remove redundancy.
2. **Fact Table:** Remains central but connects to more normalized (split) dimension tables.

Example:

Using the same retail sales data as above, in a Snowflake Schema:

- **Fact Table:** Remains the same.
- **Product Dimension (Normalized):**
  - **Product Table:** `Product_ID`, `Product_Name`, `Category_ID`
  - **Category Table:** `Category_ID`, `Category_Name`, `Brand_ID`
  - **Brand Table:** `Brand_ID`, `Brand_Name`
- **Store Dimension (Normalized):**
  - **Store Table:** `Store_ID`, `Store_Name`, `City_ID`
  - **City Table:** `City_ID`, `City_Name`, `Region_ID`
  - **Region Table:** `Region_ID`, `Region_Name`

Query Example:
The same question about raincoat sales in Boston would require joining multiple levels of the **Product** and **Store** dimensions, making the query more complex but efficient in terms of data storage.

Key Differences Between Star and Snowflake Schema:

Aspect	Star Schema	Snowflake Schema
Structure	Denormalized, simpler, fewer tables	Normalized, more complex, more tables
Performance	Faster queries, but more storage	Slower queries, less storage
Use Case	Best for smaller datasets or quick queries	Best for complex data models with large datasets
Example Query	Direct joins between fact and dimension tables	Requires multiple joins between fact and dimension sub-tables

Star Schema is often used for simplicity and performance, while Snowflake Schema is used for better storage optimization in large-scale data warehouses.

Consider the following three dimensions and a base fact table for the next Questions:

Semester (SemID, SemDescription, AcademicYearID, AcademicYearDescription)
Course (CourseCode, CourseDescription, OfferingSchoolID, OfferingSchoolDescription)
Student (StudentID, StudentDescription, BatchID, BatchDescription)
Registration (SemID, CourseID, StudentID, GPA, LetterGrade, RegistrationCount (always=1)

Following queries are made most frequently:

Query 1. Average GPA and total number of registered students by semester by offering school by batch.
Query 2. Average GPA by semester by batch.

Assume: 12 semesters, 6 academic years, 400 courses, 10 schools, 5000 students, and 5 batches.

Q2. *(10 points)* Draw a star schema that includes registration base fact table and aggregate fact tables for the above requirements. Take appropriate assumption, if required. Show the primary keys, foreign keys and all the relationships between the dimensions and fact tables. Note: Draw only a single diagram that includes base fact table as well as aggregate fact tables.