Vidyavardhini's College of Engineering and Technology
Department of Artificial Intelligence & Data Science

Name:	Pratik Sanjay Avhad
Roll No:	01
Class/Sem:	TE/V
Experiment No.:	1
Title:	Data Warehouse Construction – Star schema and Snowflake schema
Date of Performance:	
Date of Submission:	
Marks:	
Sign of Faculty:	

Aim: To Build a Data Warehouse – Star Schema and Snowflake Schema

Objective: A data warehouse is a large store of data collected from multiple sources within a business. The objective of the data warehouse system is to provide consolidated, flexible, meaningful data storage to the end user for reporting and analysis.

Theory:

In general, the warehouse design process consists of the following steps:

1. Choose a business process to model (e.g., orders, invoices, shipments, inventory, account administration, sales, or the general ledger). If the business process is organizational and involves multiple complex object collections, a data warehouse model should be followed. However, if the process is departmental and focuses on the analysis of one kind of business process, a data mart model should be chosen.
2. Choose the business process grain, which is the fundamental, atomic level of data to be represented in the fact table for this process (e.g., individual transactions, individual daily snapshots, and so on).
3. Choose the dimensions that will apply to each fact table record. Typical dimensions are time, item, customer, supplier, warehouse, transaction type, and status.
4. Choose the measures that will populate each fact table record. Typical measures are numeric additive quantities like dollars sold and units sold.

Star Schema:

The most common modeling paradigm is the star schema, in which the data warehouse contains:

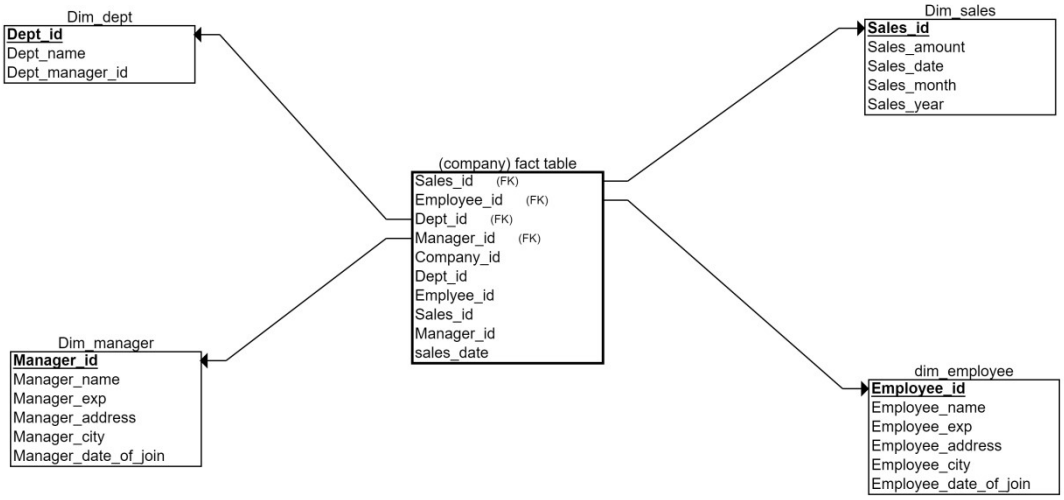
- a large central table (fact table) containing the bulk of the data, with no redundancy, and
- a set of smaller attendant tables (dimension tables), one for each dimension.

Snowflake Schema:

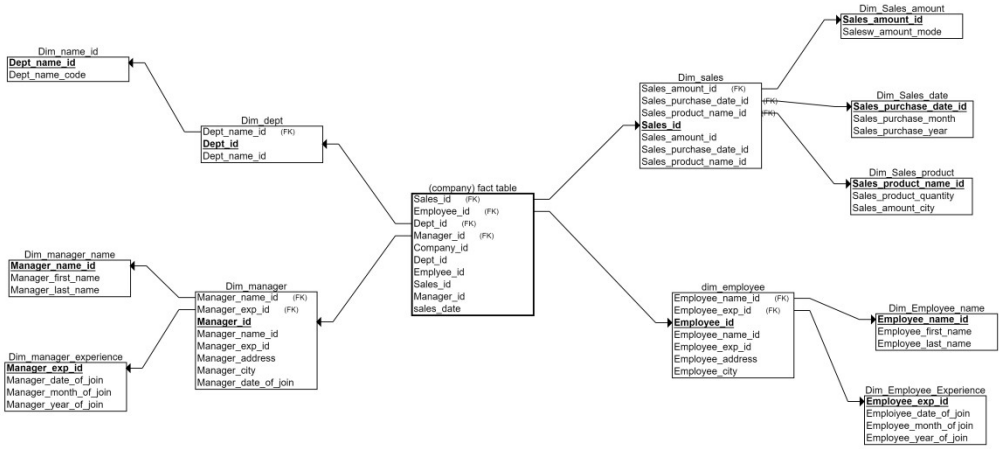
- The snowflake schema is a variant of the star schema model, where some dimension tables are normalized, thereby further splitting the data into additional tables.
- The resulting schema graph forms a shape similar to a snowflake.
- The major difference between the snowflake and star schema models is that the dimension tables of the snowflake model may be kept in normalized form to reduce redundancies.
- Such a table is easy to maintain and saves storage space.
- However, this space savings is negligible in comparison to the typical magnitude of the fact table.

Construction of Star schema and Snowflake schema:

Star schema



Snowflake schema



Conclusion:

What are the main differences between the Star Schema and Snowflake Schema in terms of structure and design?

Ans. The Star Schema has a straightforward structure where a central fact table connects directly to several denormalized dimension tables, forming a star-like appearance. This denormalization means dimension tables contain redundant data, simplifying the schema and making it easier to understand and query, as fewer joins are needed. In contrast, the Snowflake Schema has a more complex design due to its normalized structure. Dimension tables are further broken down into smaller related tables, resembling a snowflake. This normalization reduces data redundancy and storage requirements but makes the schema harder to manage and understand, as it requires more joins between tables for querying. While the Star Schema prioritizes simplicity and query performance, the Snowflake Schema focuses on minimizing data redundancy and optimizing storage, albeit with increased design complexity.

Which schema did you find easier to design and implement? Why?

Ans. The Star Schema is generally easier to design and implement due to its simple structure, with a central fact table directly connected to denormalized dimension tables. This straightforward setup makes it intuitive, reduces complexity, and simplifies maintenance, querying, and debugging. In contrast, the Snowflake Schem**a** involves normalizing dimension tables into multiple related tables, adding complexity with more joins required in queries, which can slow performance and complicate the design process. For ease of implementation and faster performance, the Star Schema is often preferred, especially in scenarios where simplicity and speed are key.