Star Schema vs. Snowflake Schema Explained

Star schema and snowflake schema are two methods for arranging data in a warehouse. Learn the key differences and which is right for your organization.

as organizations continue to dish out reams of data, a system needs to be in place to collect it all and later use it for analytics. The type of data storage a company chooses to utilize is critical to how effectively they can take advantage of the data they collect.

There are two common methods: star schema and snowflake. Star schema is the simplest method. It involves a fact table at the center connecting to dimension tables around it, looking like a star. Snowflake is a method of storing data in which fact tables, dimension tables and sub-dimension tables are connected through foreign keys. It's more complex but contains more depth for analysis.

Star Schema vs. Snowflake Schema Defined

- **Star Schema:** Star schema is the simplest method for arranging data in a data warehouse. It contains a fact table at the center connected to dimension tables around it. Star schema is most effective for quick and simple data query execution.
- **Snowflake Schema:** Snowflake schema is a more complex method of storing data in which fact tables, dimension tables and sub-dimension tables are connected through foreign keys. Snowflake is most effective for in-depth data query analyses.

Data architects need to factor in parameters like speed, cost, security, dependability, and more when considering the schema they will be using for the storage.

This article throws light on star and snowflake schema, exploring their characteristics and key differences. It also lays down which one works best in a given scenario.

What Is a Star Schema?

Star schema is the simplest method for arranging data in the data warehouse. It consists of the fact table at the center connecting to the dimension tables around. The fact table stores information about metrics, while the dimension tables hold information about descriptive attributes.

The schema distinguishes between the fact data related to an organization and the descriptive data attached to it. Star schema contains denormalized data, which refers to the process of adding redundant data to a relational database to improve read performance at the cost of write performance.

Examples of the fact table could be employee names, sales price, sales quantity, and distance measures. Dimension tables associated with such tables will include names of particular employees (John, Bob, etc.), and numerical figures for the other parameters in the fact table.

Advantages and Disadvantages of Star Schema

Star schema suits applications requiring simplicity and performance as it has fewer tables and joins.

The main demerit of the star schema is its denormalization and redundancy. This can lead to wide dimension tables that take up more storage space and rake up the risk of data inconsistency.

As the schema flattens the dimension hierarchies into a single table, their depth and breadth gets limited. This makes the star schema unsuitable for data warehouses having complex and dynamic dimensions that require accuracy and high data normalization.

What Is a Snowflake Schema?

Snowflake schema consists of three types of tables: Fact tables, dimension tables and sub-dimension tables.

The fact table is the central table in the schema. The dimension table stores details about the facts. Dimension tables of the snowflake schema are further normalized into sub-dimension tables.

For example, suppose a school has built a database for enrollment of its students and grades. The data architect might make “Enrollment” the central fact table. Dimension tables connected to the fact table might be the “Students” table, containing data on the students, the “Courses” table storing data regarding the courses on the anvil, and the 'Teachers' table with the data about the teachers.

The “Students” dimension table might have sub-dimension tables like “Parental education,” “Family background,” and “Career objectives.” The “Courses” dimension might have sub-dimension tables such as “Language,” “Science,” and “Commerce.” The “Teachers” table might have sub-dimension tables like “Doctorate,” “Sports,” or “Physical education.”

The three tables are inter-connected through foreign keys. A foreign key is a column, or columns, in a table whose values need to essentially match values of a column in the related table. For instance, suppose there’s a restaurant database with an orders table and a customers table. If the algorithm creates a column orders.customer_id referencing the customers.id primary key, any value updated or inserted in orders.customer_id must precisely match a value in customers.id.

An objective of snowflake schema is to normalize the star schema’s denormalized data. Normalization refers to the process of organizing data. Until full normalization occurs, the dimension tables are segregated across multiple sub-tables.

Ultimately, the snowflake schema is an extension of a star schema. The dimensions are highly structured, intricate and have multiple connections with each other, thus giving its name, snowflake.

Advantages and Disadvantages of Snowflake Schema

Snowflake schema can accommodate complex and dynamic dimensions and hierarchies. This is because the architecture is designed for quick query performance, separating computing and storage. Normalization also helps in making query handling more efficient. Snowflake also helps in preventing data repetition.

Its main downside is its complexity and performance resulting in many small dimension tables that convolute the data warehouse design and maintenance. Increase in the number of disk accesses and joins slows down the queries.

For those reasons, snowflake schema is a fit for users who want to consume data in a drill down fashion. For instance, users who might want to dig into a duration and desire to look at data generated over a decade and compare it to another data state.

Let us understand this with an example where a sales data warehouse has four dimensions — user, account, lead and location. In this scenario, a user, account or lead, all might have their own location. Rather than repeating location in each of these, a better approach would be to create a foreign key from each dimension to the location dimension. An entity relationship diagram (ERD) model demonstrating this relationship will resemble a snowflake.

Key Differences Between Star Schema and Snowflake Schema

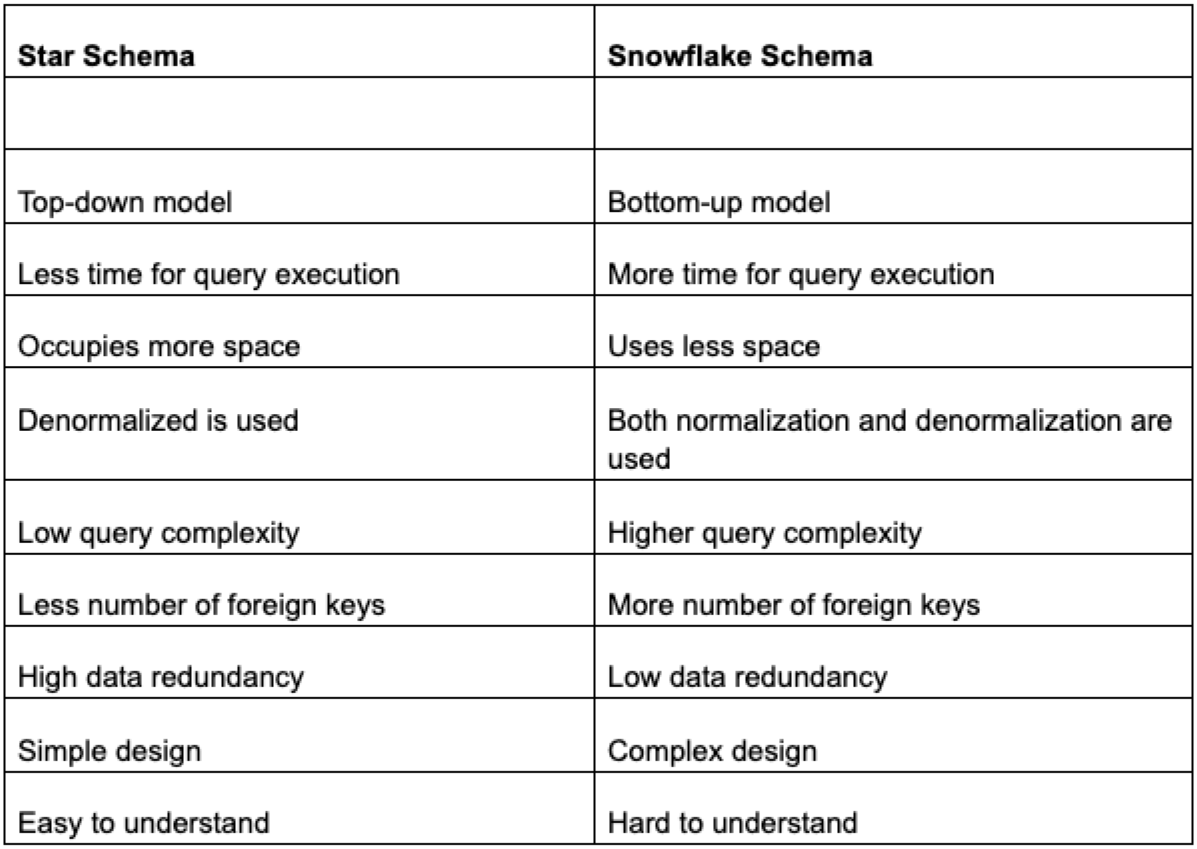
Here are the major differences between star schema and snowflake schema:

Star Schema

- Top down model.
- Less time for query execution.
- Uses more space. Since star schema uses redundant data it can take up a lot of storage space.
- Denormalized data is used.
- Low query complexity.
- Fewer foreign keys.
- High data redundancy.
- Simple design.
- Easy to understand.

Snowflake Schema

- Bottom-up model.
- More time for query execution.
- Uses less space.
- Both normalization and denormalization are used.
- Higher query complexity
- More foreign keys.
- Low data redundancy.
- Complex design.
- Hard to understand.



Star Schema	Snowflake Schema
Top-down model	Bottom-up model
Less time for query execution	More time for query execution
Occupies more space	Uses less space
Denormalized is used	Both normalization and denormalization are used
Low query complexity	Higher query complexity
Less number of foreign keys	More number of foreign keys
High data redundancy	Low data redundancy
Simple design	Complex design
Easy to understand	Hard to understand

Star Schema vs. Snowflake Schema: Which One Is Best for You?

Selecting the right schema depends on the needs of your organization If you need a simple cloud data warehouse solution that offers quick query execution, a star schema might work best. For applications having a few dimensions with limited levels of hierarchy and low cardinality, star schema is suitable. For example, the star schema will be suitable for the sales data warehouse of a small enterprise. Sales might be the central table while time, product, time and employees constitute the dimension tables.

However, if you’re looking for flexibility of use cases while retaining data integrity, a snowflake schema will better serve your needs. Data integrity refers to accuracy, consistency, completeness and validity of an organization's data. For example, a snowflake schema would be effective for a large organization that has a huge amount of normalized data, such as situations when data has been segregated into a string of columns.

Your choice of schema will hinge on factors such as data characteristics, storage constraints, query needs and performance expectations. If you have a few dimensions with limited levels of hierarchy and low cardinality, but you require fast query execution, a star schema is the right choice. However, if you have several dimensions with multiple levels of hierarchy and high cardinality, a snowflake schema will be a better scheme.

For more visualization watch
What is STAR schema | Star vs Snowflake Schema | Fact vs Dimension Table