What is a star schema

A star schema is a database organizational structure optimized for use in a data warehouse or business intelligence that uses a single large fact table to store transactional or measured data, and one or more smaller dimensional tables that store attributes about the data. It is called a star schema because the fact table sits at the center of the logical diagram, and the small dimensional tables branch off to form the points of the star.

A fact table sits at the center of a star schema database, and each star schema database only has a single fact table. The fact table contains the specific measurable (or quantifiable) primary data to be analyzed, such as sales records, logged performance data or financial data. It may be transactional -- in that rows are added as events happen -- or it may be a snapshot of historical data up to a point in time.

How a star schema works

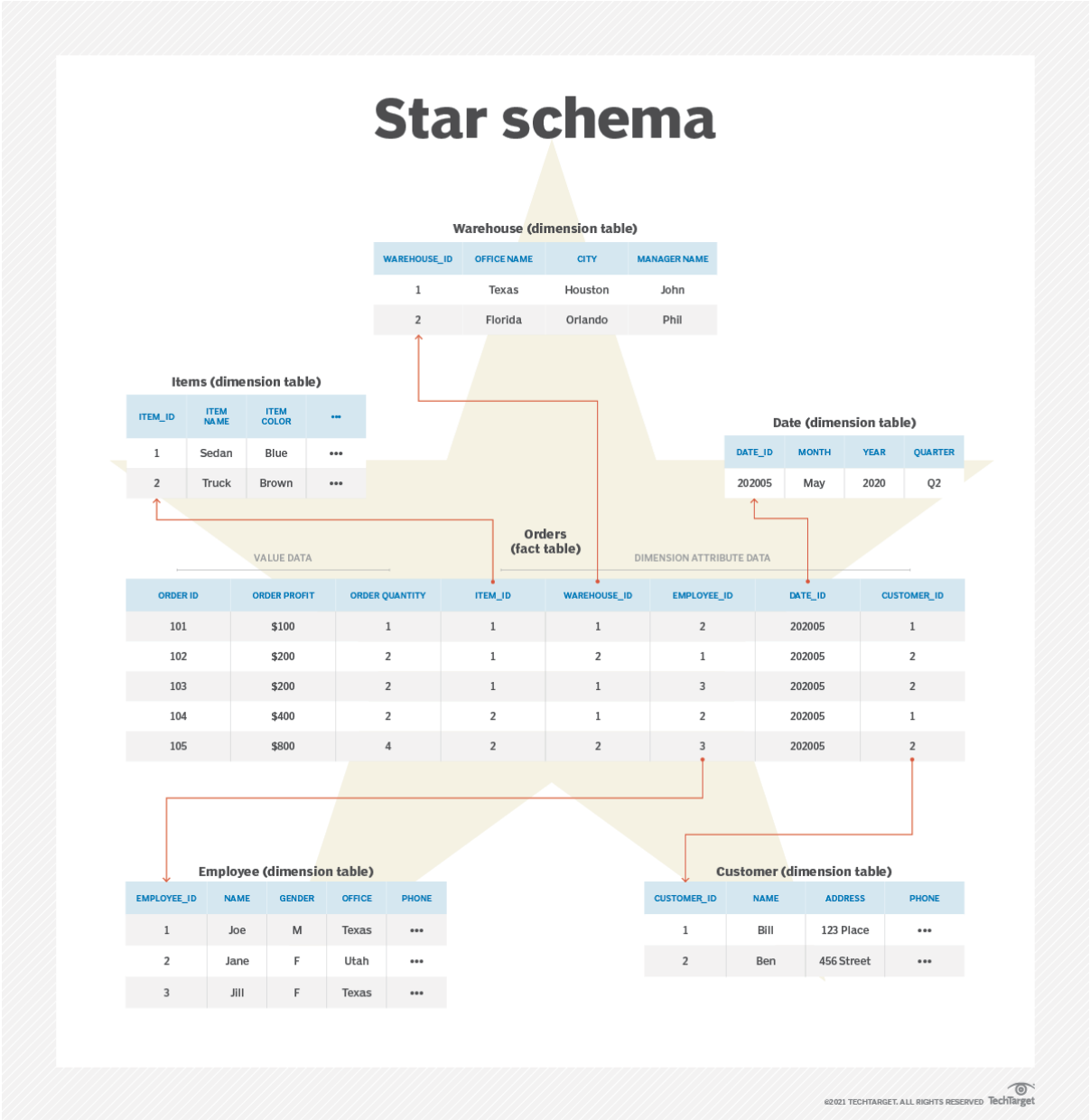
The fact table stores two types of information: numeric values and dimension attribute values. Using a sales database as an example:

- **Numeric value cells** are unique to each row or data point and do not correlate or relate to data stored in other rows. These might be facts about a transaction, such as an order ID, total amount, net profit, order quantity or exact time.
- **The dimension attribute values** do not directly store data, but they store the foreign key value for a row in a related dimensional table. Many rows in the fact table will reference this type of information. So, for example, it might store the sales employee ID, a date value, a product ID or a branch office ID.

Dimension tables store supporting information to the fact table. Each star schema database has at least one dimension table, but will often have many. Each dimension table will relate to a column in the fact table with a dimension value, and will store additional information about that value.

For example:

- The **employee dimension table** may use the employee ID as a key value and can contain information such as the employee's name, gender, address or phone number.
- A **product dimension table** may store information such as the product name, manufacture cost, color or first date on market.



Star schema vs. snowflake schema

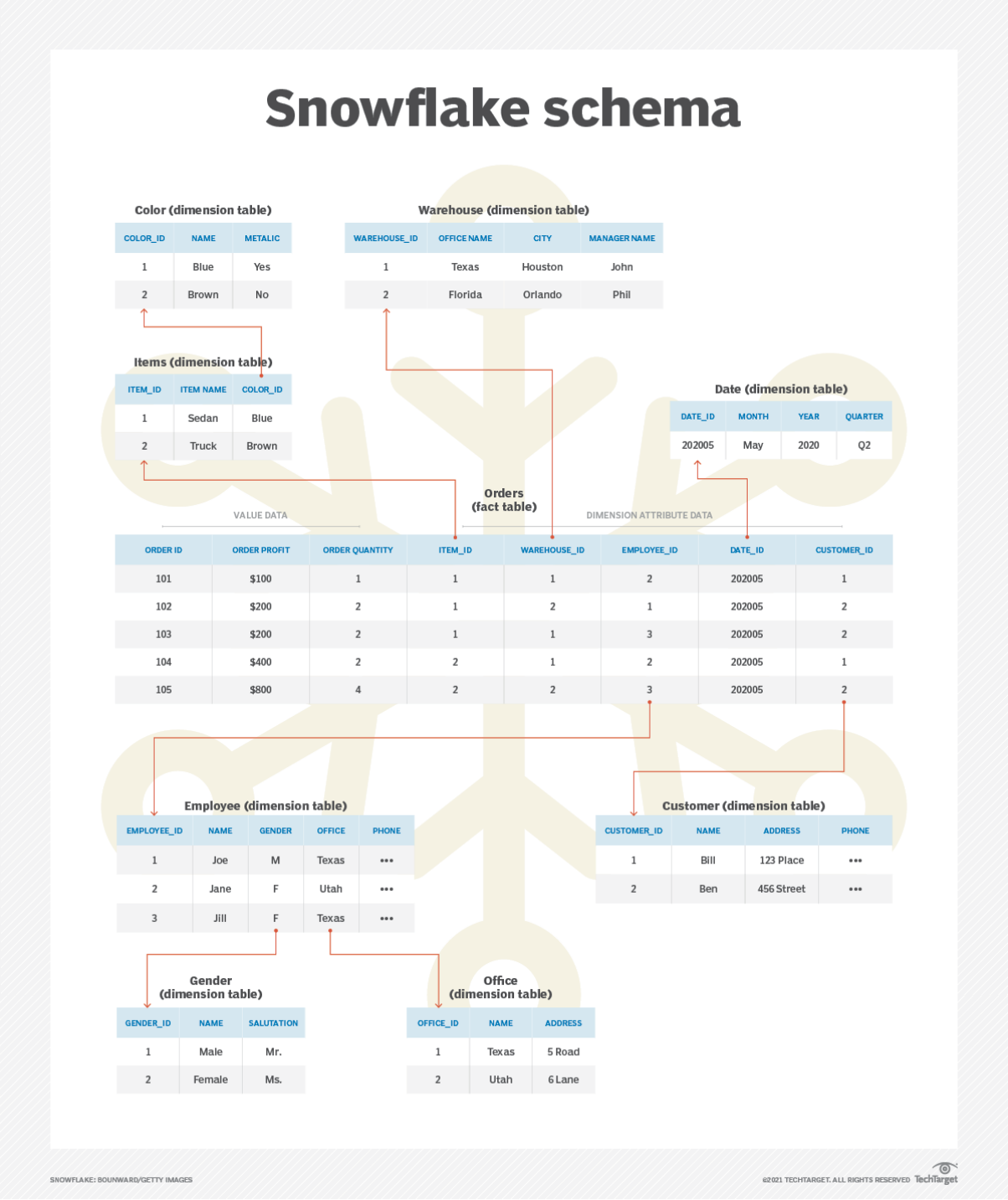
Star schema's dimension tables do not contain any foreign keys. That is, the dimension tables do not reference any other tables, nor do they have any "sub-dimension tables." They are generally denormalized because some information may be duplicated in the dimension tables. This allows star schema databases to be optimized for read and query performance along specific dimensions.

A snowflake schema database is similar to a star schema in that it has a single fact table and many dimension tables. However, for a snowflake schema, each dimension table might have foreign keys that relate to other dimension tables.

So, in a star schema there is no further branching from each dimension table. But in a snowflake schema each branch might have further branches -- like a snowflake with each branch having successively smaller branches coming out of a central core in a fractal pattern. A snowflake schema is also more normalized than a star schema, though not necessarily fully normalized.

As a simple example, the sales record in the fact table contains an employee ID. This employee ID relates to an employee dimension table that contains information such as the first name, last name, gender and branch office.

- In a **star schema**, the record would contain information such as "male" and "Texas Office." This would be duplicated data also in other rows for employees with the same gender or branch office.
- In a **snowflake schema**, the gender or branch office would contain a foreign key value to a gender dimension table and a branch office dimension table. These could contain information such as gender, name, salutation (Mr. or Ms.), branch office name, branch office address or branch manager.



Star schema pros and cons

There are several pluses and minus to using star schema.

Star schema pros

- simple design;
- fast read and queries;

- easy data aggregation; and
- easy integration with OLAP systems and data cubes.

Star schema cons

- redundant data makes for larger storage on disk;
- potential for data abnormalities, errors and inconsistencies;
- slower queries;
- limited flexibility on non-dimensional data.

Star schema use cases

Star Schema databases are best used for historical data.

This makes them work most optimally for data warehouses, data marts, BI use and OLAP. Primarily read optimized, star schemas will deliver good performance over large data sets. Organizations can also tailor them to provide their best performance along the specific criteria considered the most important or most used to query against. Data can be added transactionally as it comes in, or it can be batch imported then checked and properly denormalized at that time.

Star schema database structures are generally not a good fit for live data, such as in online transaction processing. Their denormalized nature imposes restrictions that a fully normalized database does not. For example, slow writes to a customer order database could cause a slowdown or overload during high customer activity. This potential for data abnormalities could be disastrous in a live order fulfillment system.