Star schema, Snowflake schema & Fact constellation

Dr. Shalini Gambhir

Difference Between Normalized and Denormalized Form

- In normalized form, data is stored in multiple tables, reducing data redundancy and inconsistency, thus achieving data integrity. In the denormalized form, data is stored in a limited number of tables (maybe a single table) to reduce querying time.

  Both of them contain joined tables, but the key difference between them is the degree of normalization. As the degree of normalization increases, the complexity of the model increases, and as the complexity of the model increases, the time to retrieve data also increases.

Dimensional Modeling

- The data model used to store data in the denormalized form is called Dimensional Modeling. It is the technique of storing data in a Data Warehouse in such a way that enables fast query performance and easy access to its business users. It involves creating a set of dimensional tables that are designed to support business intelligence and reporting needs.
- The goal of dimensional modeling is to provide a simple and intuitive way to access and analyze data, making it easy for business users to understand and use it. It aims at making simple data models. When the data models are as simple as possible, they can be understood easily, allowing the software to navigate and deliver results quickly and efficiently.

Basic idea of what Facts and Dimensions are.

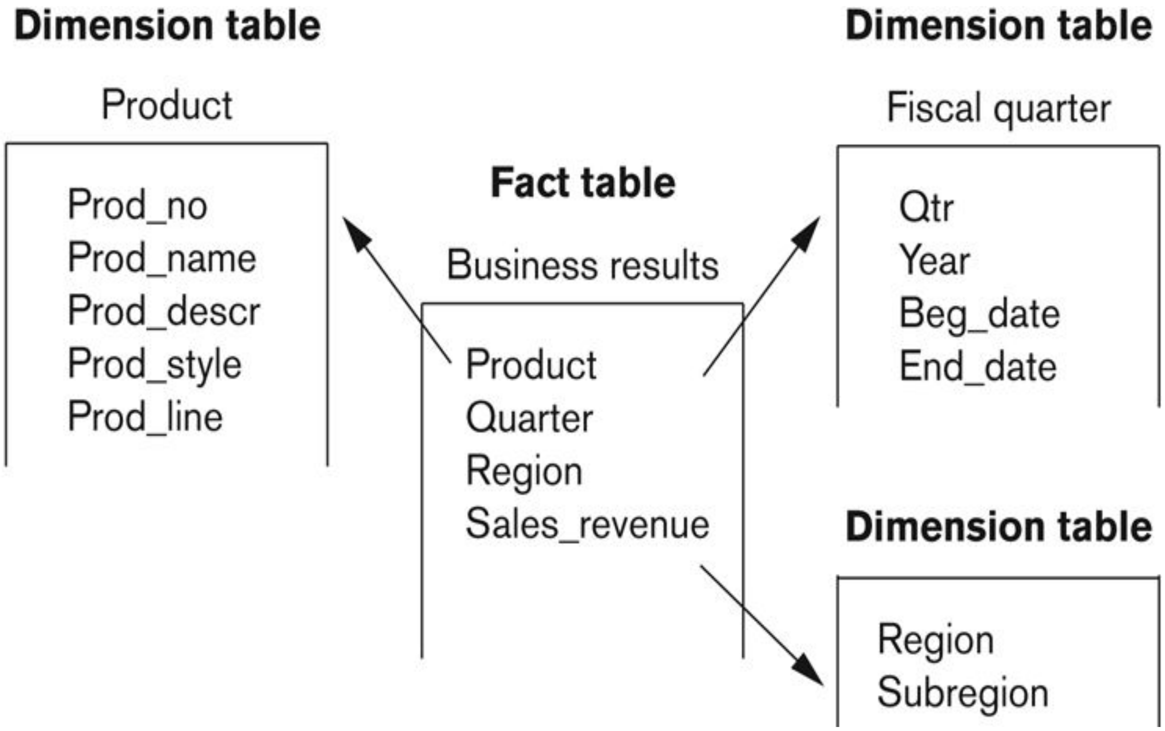
- Fact tables contain measures or numerical data associated with a business process, like the number of products sold. In contrast, dimensional tables store the description or textual information related to the business process, like who bought the products.
- A dimensional model represents the different business processes of an organization. A fact table with its dimension table is a single business process.
- Each dimensional model consists of many fact tables, with each fact table joined with corresponding dimension tables. A fact table is connected to another fact table via a common dimensional table between them; this common dimensional table is called a bridge table.

Multi-dimensional Schemas

- Two common multi-dimensional schemas are
  - **Star schema:**
    - Consists of a fact table with a single table for each dimension
  - **Snowflake Schema:**
    - It is a variation of star schema, in which the dimensional tables from a star schema are organized into a hierarchy by normalizing them.

Multi-dimensional Schemas

- **Star schema**:
  - Consists of a fact table with a single table for each dimension.



Dimension table

Product

Prod_no
Prod_name
Prod_descr
Prod_style
Prod_line

Fact table

Business results

Product
Quarter
Region
Sales_revenue

Dimension table

Fiscal quarter

Qtr
Year
Beg_date
End_date

Dimension table

Region
Subregion

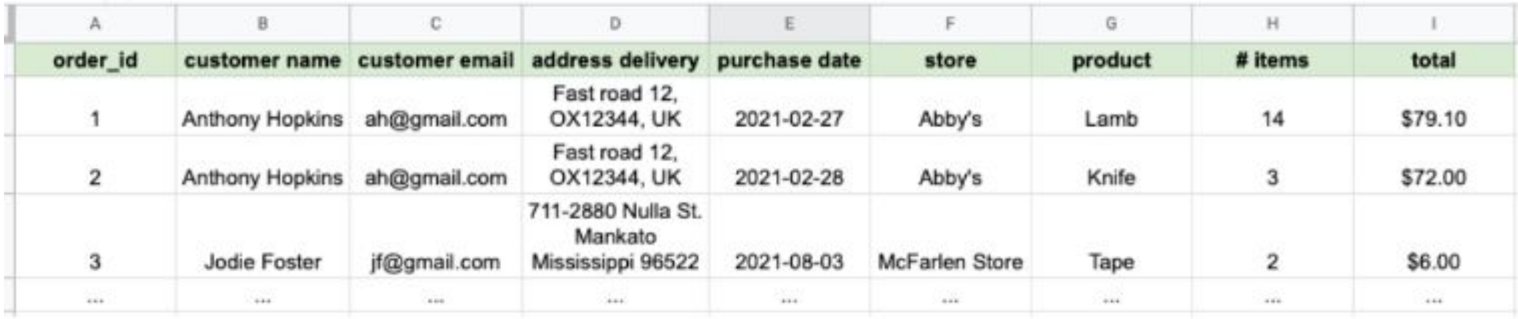
Figure 29.7
A star schema with fact and dimensional tables.

Star Schema vs Snowflake Schema

- Star schemas and snowflake schemas are the two predominant types of data warehouse schemas.
- A data warehouse schema refers to the shape your data takes - how you structure your tables and their mutual relationships within a database or data warehouse.

What is a star schema?

- To understand the data modeling behind a star schema, let us look at a retail example. Imagine you are running an international shopping brand and you want to analyze purchases across your physical locations. You pull out data from your database as an Excel file:

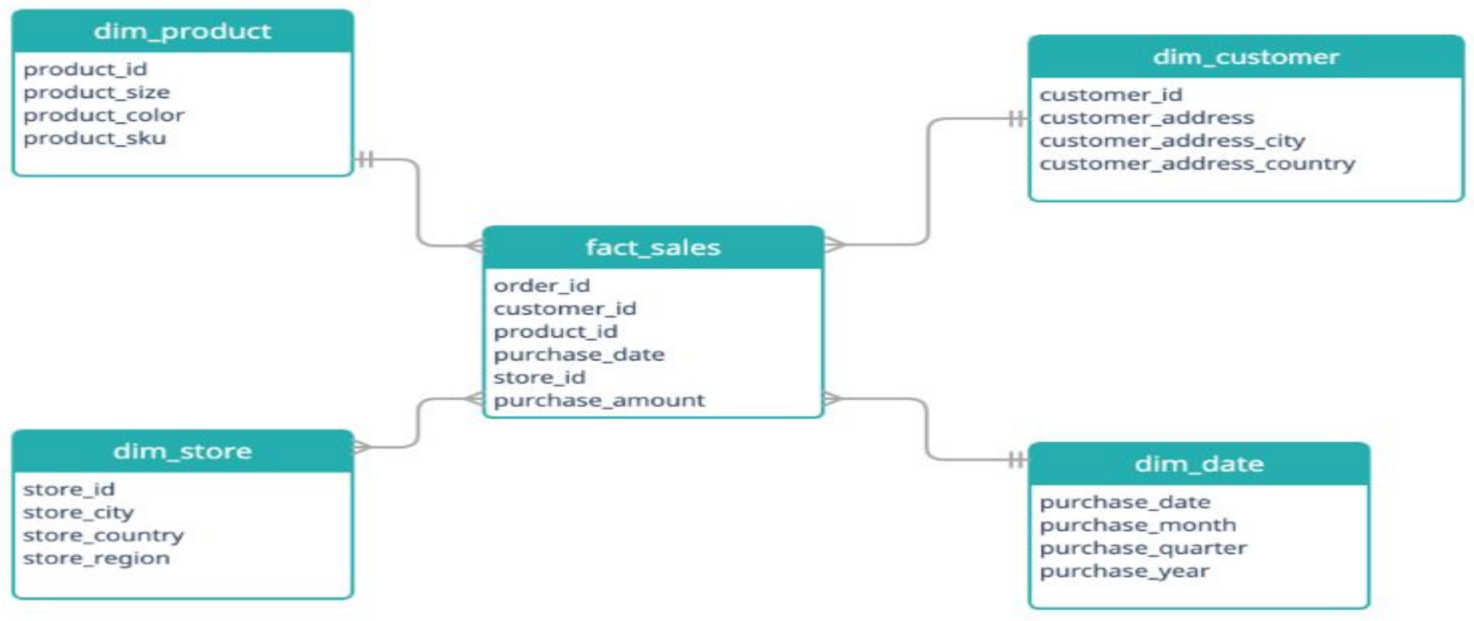


A	B	C	D	E	F	G	H	I
order_id	customer name	customer email	address delivery	purchase date	store	product	# items	total
1	Anthony Hopkins	ah@gmail.com	Fast road 12, OX12344, UK	2021-02-27	Abby's	Lamb	14	$79.10
2	Anthony Hopkins	ah@gmail.com	Fast road 12, OX12344, UK	2021-02-28	Abby's	Knife	3	$72.00
3	Jodie Foster	jf@gmail.com	711-2880 Nulla St. Mankato Mississippi 96522	2021-08-03	McFarlen Store	Tape	2	$6.00
...	...	...	...	...	...	...	...	...

- But you soon realize there are too many rows, and the data needs to be cleaned before you can analyze it.

- You decide to turn the data into a star schema.
- A star schema is a data model that stores information in multiple table types: a single fact table and multiple dimensional tables.

- In contrast to the classical database design of normalizing tables, star schemas connect dimensional data with fact data in a shape resembling a star (hence the name), as can be seen from the following diagram:

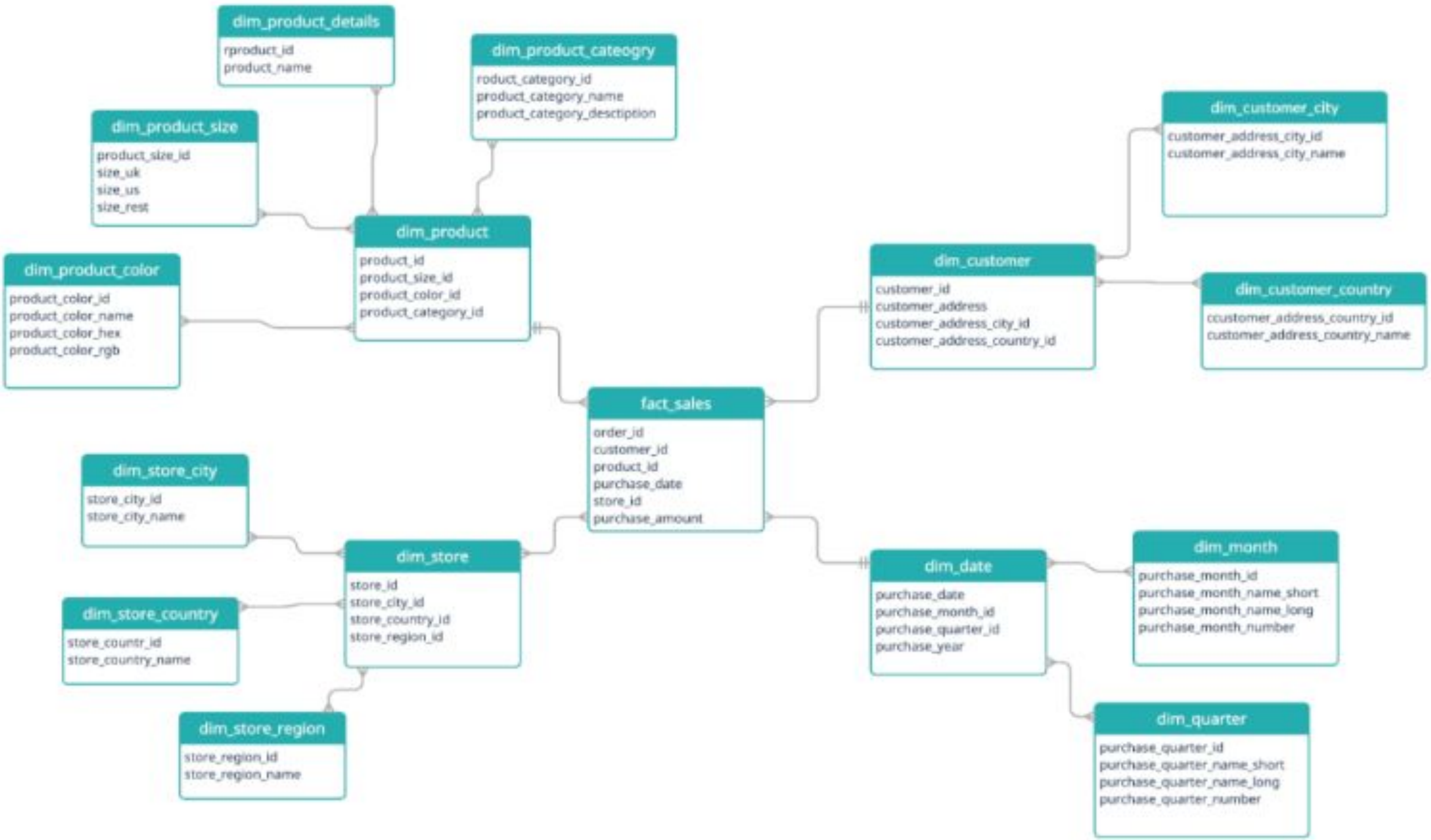


- In the diagram, we see a central fact table (holding all the facts of the sales) and four dimension tables - a separate table describing the customer, date (of purchase), store where the purchase happened, and product purchased.
- The fact table is linked via a foreign key relationship to the primary key of each dimension (aka, the id in each dimension table, for example, the customer_id links the customer from the dim_customer table to the fact_sales table).
- This type of data modeling allows us to query data faster and with simpler queries than the normalized database design.

What is a snowflake schema?

- A snowflake schema is very similar to the simple star schema above. The main difference is that snowflake schemas split dimensional tables into further dimensional tables (also called lookup tables).
- Each dimension is split until it is normalized - aka, there is no redundancy in the dimensional table, no repetition of values (except for identifier values, such as id’s).

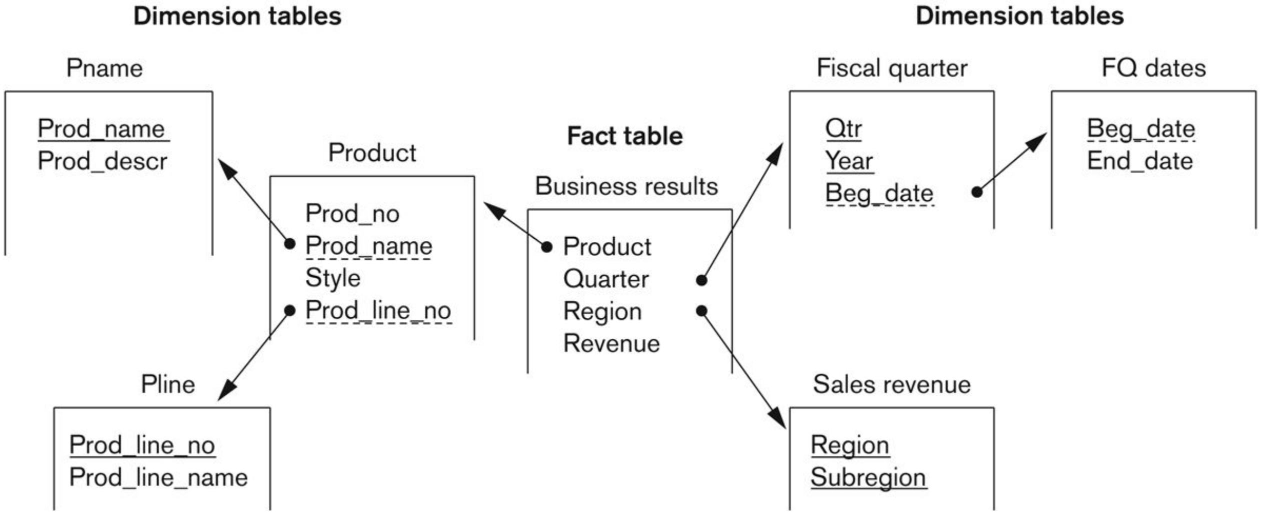
For example, the above diagram would show the customer_country field being split into further dimensional tables:



Multi-dimensional Schemas

- **Snowflake Schema:**
  - It is a variation of star schema, in which the dimensional tables from a star schema are organized into a hierarchy by normalizing them.

Figure 29.8
A snowflake schema.



The 7 critical differences between a star schema and a snowflake schema

1. Normalization of dimension tables

- The snowflake schema is a fully normalized data structure. Dimensional hierarchies (such as city > country > region) are stored in separate dimensional tables.
- On the other hand, star schema dimensions are denormalized. Denormalization refers to the repeating of the same values within a table.

- 2. Data redundancy
- Star schema stores redundant data in dimension tables, while snowflake schema fully normalizes dimension tables and avoids data redundancy.

- For example, a star schema would repeat the values in field *customer_address_country* for each order from the same country.

- The redundancy, or duplicated entries, occurs because of the denormalization vs normalization schema design.

- 3. Query complexity
- A simple star schema leads to simple query writing. Because the fact table is joined to only one level of dimensional tables, analysts do not need to write multiple joins.

- On the other hand, snowflake schemas require a more complex query design. Because of complex relationships between the fact table and its dimensional tables, more joins are needed to link the additional tables. This causes an additional overhead when writing analytical queries.

- 4. Query performance
- The query execution time is faster in star schemas. Because they require a single join between a fact and its set of attributes in dimensional tables, a star schema acts almost as a single table for query lookups.
- In contrast, snowflake schemas require complex joins of dimensional tables with their own sub-dimensional or supra-dimensional tables. This slows down query processing and can affect other OLAP products such as cube processing.

- 5. Disk space
- Star schemas might run queries faster, but they require more storage space than snowflake schemas because of their data redundancy.

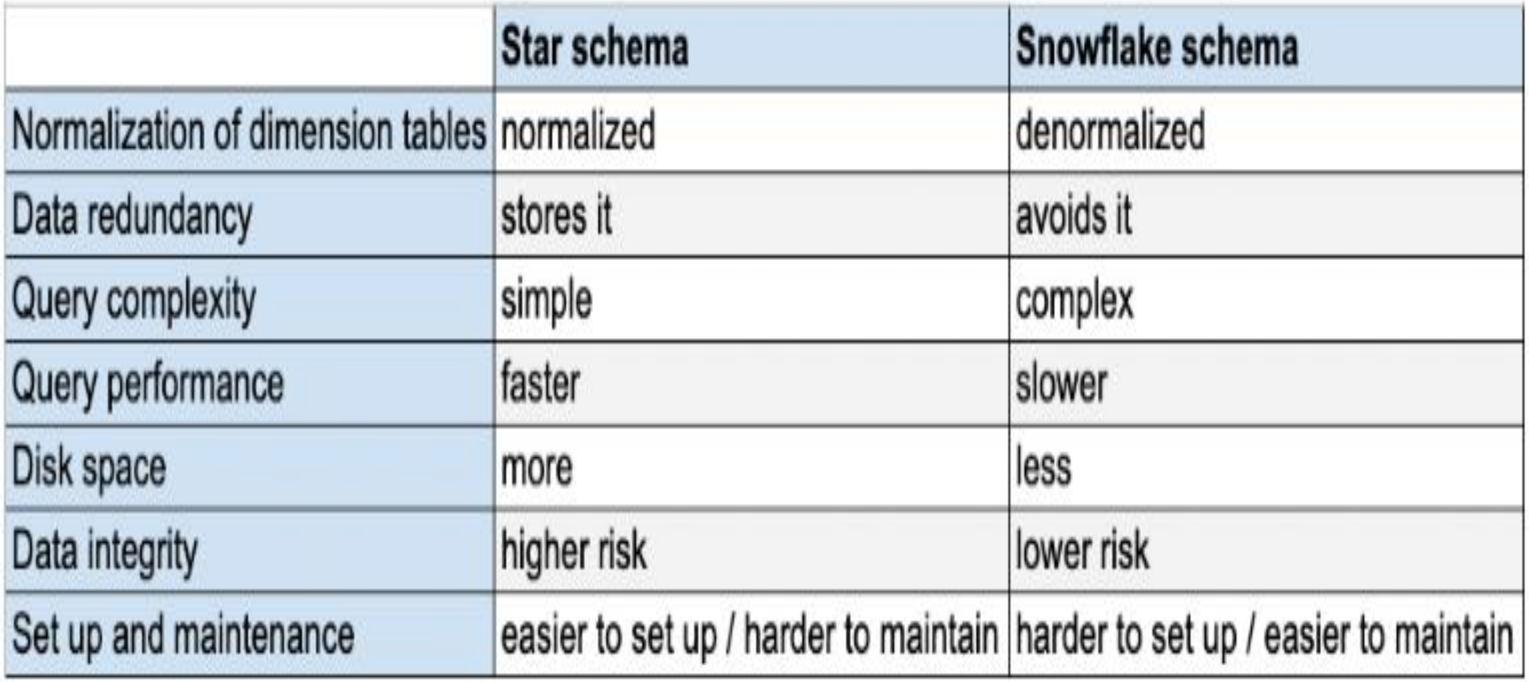
- 6. Data integrity
- Data integrity is more at risk in star schemas than snowflake schemas. Because data is stored redundantly, multiple copies of the same data exist in the star schema’s dimensional tables. This means new inserts, updates, or deletes can compromise the integrity of data.

- In contrast, the snowflake schema is less prone to data integrity issues, because it fully normalizes dimensional tables, storing dimension data only once in the appropriate table.

- Set up and maintenance
- Star schemas are easier to design and set up. Because they are represented by simple relationships, it is easy for a data engineer or data architect to set up an appropriate star schema.

- On the other hand, star schemas are harder to maintain than snowflake schemas. As new data is ingested into the data warehouse, star schemas become harder to maintain and check against data integrity violations.

Which one should I pick?



	Star schema	Snowflake schema
Normalization of dimension tables	normalized	denormalized
Data redundancy	stores it	avoids it
Query complexity	simple	complex
Query performance	faster	slower
Disk space	more	less
Data integrity	higher risk	lower risk
Set up and maintenance	easier to set up / harder to maintain	harder to set up / easier to maintain

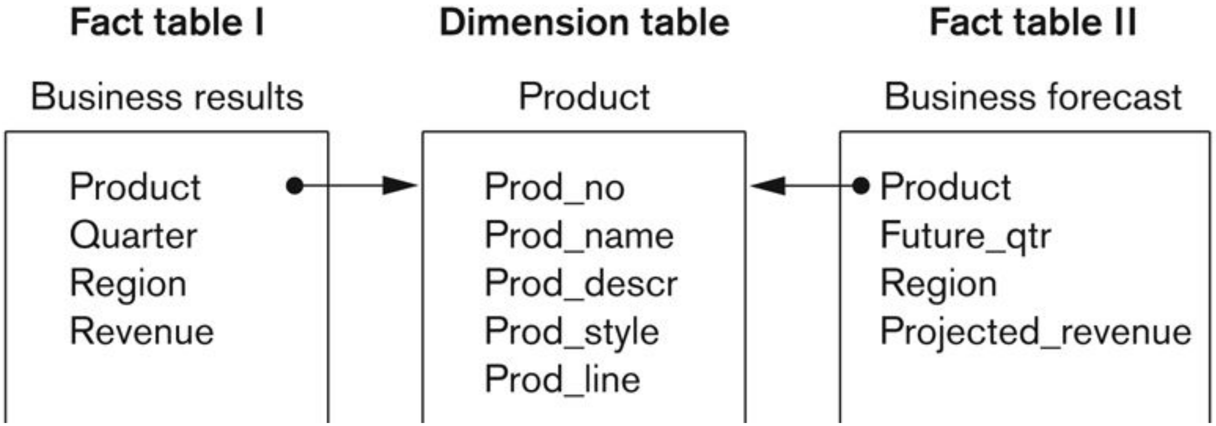
On one hand, star schemas are simpler, run queries faster, and are easier to set up.

On the other hand, snowflake schemas are less prone to data integrity issues, are easier to maintain, and utilize less space.

Based on the tradeoffs above, it depends on which advantage (or disadvantage) best suits your business use cases.

Multi-dimensional Schemas

- **Fact Constellation**
  - Fact constellation is a set of tables that share some dimension tables. However, fact constellations limit the possible queries for the warehouse.



Fact table I

Business results

Product
Quarter
Region
Revenue

Dimension table

Product

Prod_no
Prod_name
Prod_descr
Prod_style
Prod_line

Fact table II

Business forecast

Product
Future_qtr
Region
Projected_revenue

Figure 29.9
A fact constellation.