Data

- Piece of information
- Data is the new oil
- Data is very important

Advantages of semi-structured data over structured data

- More flexible
- Schema is not fixed
- More efficient in terms of space: null values are seldom used in semi-structured data thus saving that memory

Problems with using dat, txt, and other file formats used in the file system

- It doesn’t support large amounts of storage
- It takes a long time to read through the data
- There is high data redundancy

Data Mart Benefits

- Localization
- Ex.: customer file, transaction file, .. in a Banking System

Keys

- Primary key
- Foreign key
- Candidate key
- Composite key
- Super key
- Unique keys

Primary Key

- It must be unique and not null

Unique Key

- It is any column that has unique values without being assigned as such
- This can be null
- Example: account number in a customer file for a banking system. CID is the primary key

Super key

- Combination of 2 or more keys that uniquely identify a column

Composite key

- A composite key can also be made by the combination of more than one candidate key.
- A composite key cannot be null
- The difference between composite keys and super keys is that composite keys are minimal super keys. If even 1 column is removed from the set of keys in a composite key, it is no longer unique

Why can't we have all the records in one table

- A lot of data redundancy
- Normalization is the way to help with data redundancy

Data Driven Decision

- Takes care of storage and processing

Partial dependency

- One column is dependent on the other and the other column happens to be a candidate key

*** Normalization types: watch You Tube

Anomalies in DBMS

- Insertion anomaly
- Updating anomaly
- Deletion anomaly

Requirement:

- ER model for Book store

- Book, author, customer, order, publisher - entities
- Book – bid, bname, auth_id, edition, pub_id, cost
- Author - auth_id, auth_name, phone_num
- Customer – cust_id, cust_name, phone_num
- Order – order_id, cust_id, book_id, qty, amount
- Publisher – pub_id, pub_name, phone_num

First step in working with any requirements

- Analyze the requirement
- Come up with the ER model

1NF

- No multi-valued attributes should be present

2NF

- Should be in 1NF
- Non-key attributes should not depend on the partial of the primary key
- Composite key -> id + course
- Age is dependent on the partial of this composite key: id, name, age are present in the same table
- Id and course are present in another table

3NF

- Should be in 2NF
- Non-key attributes should not have any dependency among them

BCNF

- Should be in 3NF
- Identify the primary key
- Identify the functional dependencies
- Check if the columns are non-key attributes
- If they are not key attributes, then break the table and put them in another table

Common Terminologies

- Key-value attributes: unique identifier
- Non-key attribute: non unique identifiers

Dimension Modelling

Dimension Table:

- Contains all the columns

Fact Table

- Contains all the columns that are quantitatively measured

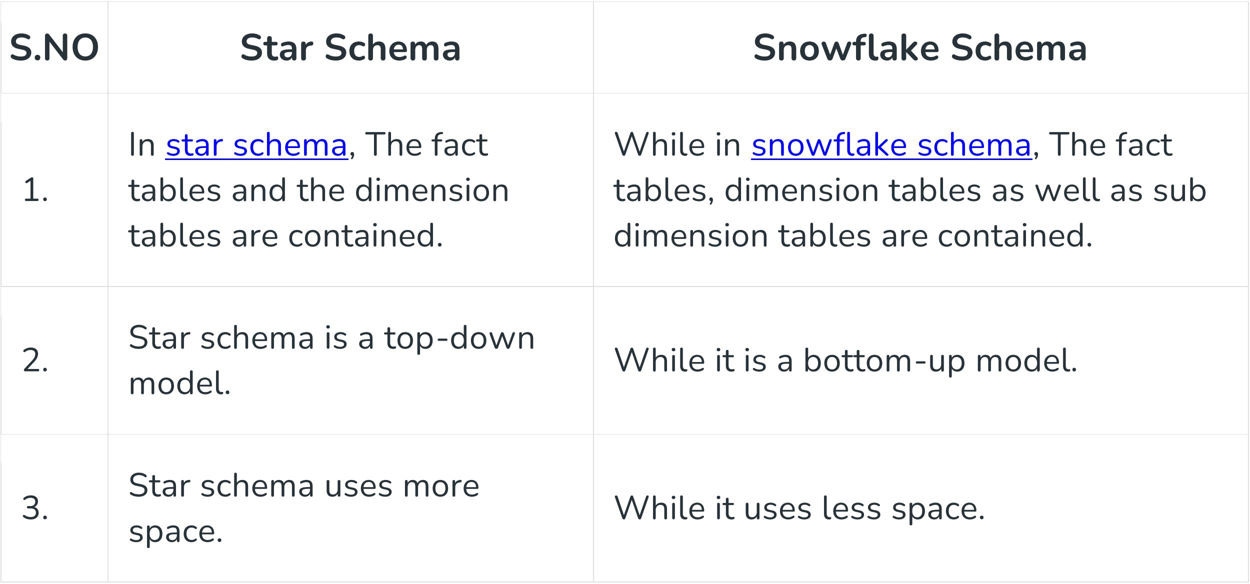
Star Schema

- Fact table surrounded by a dimension table

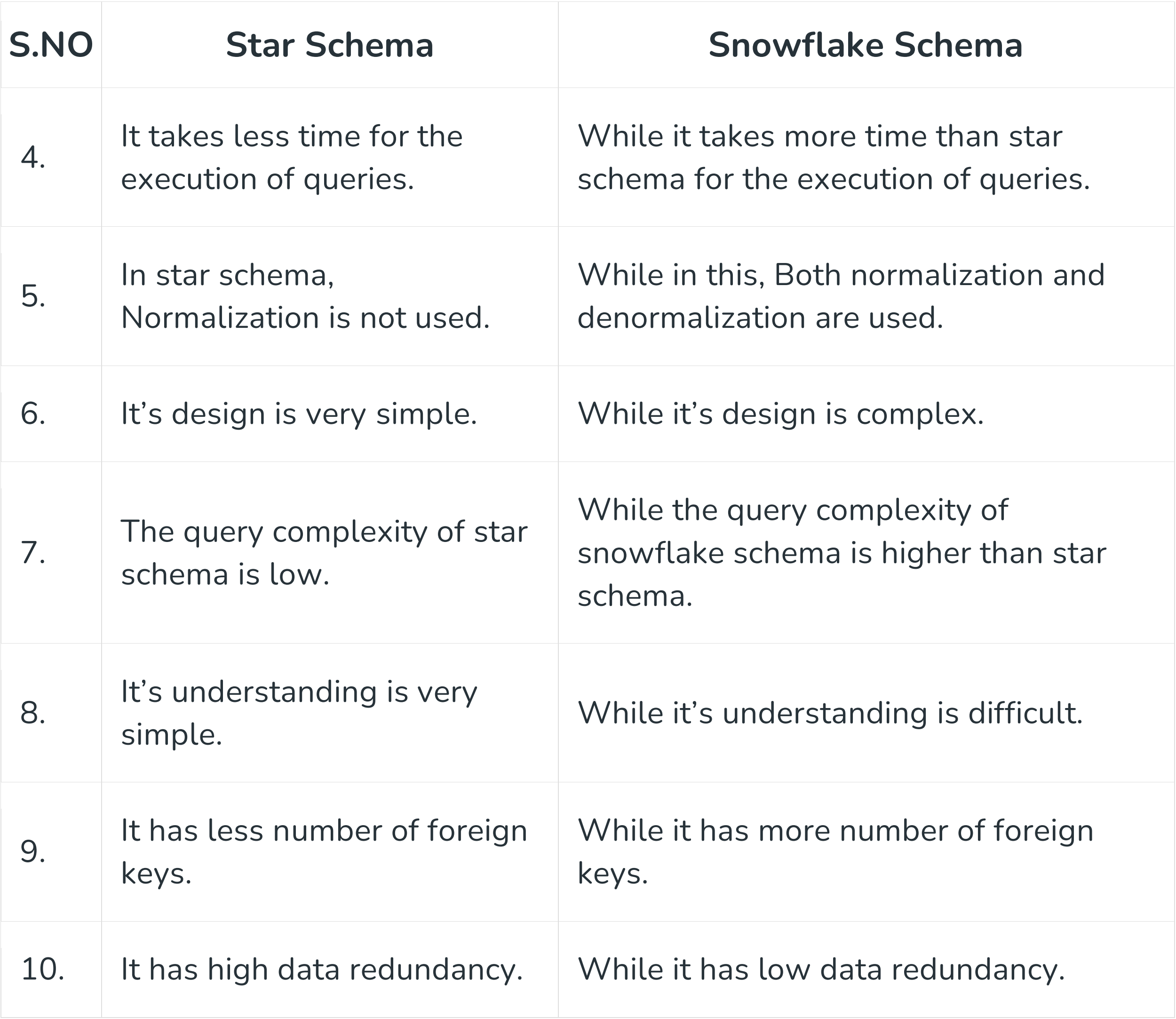
Snowflake schema

- Fact table is surrounded by a dimension table
- Each dimension table is also further surrounded by dimensions table

Difference between Star and Snowflake Schema



S.NO	Star Schema	Snowflake Schema
1.	In star schema, The fact tables and the dimension tables are contained.	While in snowflake schema, The fact tables, dimension tables as well as sub dimension tables are contained.
2.	Star schema is a top-down model.	While it is a bottom-up model.
3.	Star schema uses more space.	While it uses less space.



S.NO	Star Schema	Snowflake Schema
4.	It takes less time for the execution of queries.	While it takes more time than star schema for the execution of queries.
5.	In star schema, Normalization is not used.	While in this, Both normalization and denormalization are used.
6.	It's design is very simple.	While it's design is complex.
7.	The query complexity of star schema is low.	While the query complexity of snowflake schema is higher than star schema.
8.	It's understanding is very simple.	While it's understanding is difficult.
9.	It has less number of foreign keys.	While it has more number of foreign keys.
10.	It has high data redundancy.	While it has low data redundancy.

All the data must be reported.

Table List

- ⇨ Product
- ⇨ Date
- ⇨ Location
- ⇨ Customer
- ⇨ Salesperson

- ➔ The above table will use a Star Schema

Slowly changing dimension (SCD):

A Slowly Changing Dimension (SCD) is a dimension that stores and manages both current and historical data over time in a data warehouse. There are 3 types of SCD

1. SCD1: we just update the data in the table if there are changes in the data
2. SCD2: we add 3 extra columns (effect from date, effect to date, current flag) to update the previous data
3. SCD3: we add the new column according to the valued changed.