**Surrogate key**: A column that is not generated from the data in the database is known as a surrogate key. DBMS generates a unique identifier for you. In database tables, surrogate keys are frequently utilized as primary keys.

**Features of the Surrogate Key:**

* It is automatically generated by the system.
* It holds an anonymous integer.
* It contains a unique value for all records of the table.
* The value can never be modified by the user or application.
* The surrogate key is called the factless key as it is added just for our ease of identification of unique values and contains no relevant fact(or information) that is useful for the table.

**Fact Tables**

Fact tables store the core transactional or quantitative data in a data warehouse. These are the tables that record measurable events or facts that are associated with business operations.

Key Characteristics of Fact Tables:

* Quantitative Data: The fact table generally stores numeric data like revenue, quantity sold, sales amount, order total, etc.
* Foreign Keys: Fact tables contain foreign keys that link to dimension tables. These foreign keys represent the context in which the facts occurred (e.g., customer, product, time).
* Measures or Metrics: These are the numeric values that the fact table contains, such as sales totals, quantities, profits, etc.
* High Cardinality: Fact tables often have a large number of rows, as they store every transaction or event.
* Low Cardinality Columns: Fact tables typically have fewer columns, which include the foreign keys and the measures.

**Transactional Fact Tables:** these are designed to capture individual business events or transactions. These tables are particularly useful for analyzing customer behaviour, sales patterns and operational efficiency

**Periodic Snapshot Fact Tables:** these tables provide a summerized view of metrics over regular time intervals. These tables are particularly useful for monitoring performance, identifying trends, and measuring progress

**Accumulating Snapshot Fact Tables:** these tables are designed to track the stages of a business process or workflow. These tables provide valuable insights into process efficiency, identifying bottlenecks, optimizing operations

**Dimension Tables**

Dimension tables contain descriptive, textual, or categorical data that provide the context for the facts in the fact table. They help to provide more meaningful insights when analyzing the data.

Key Characteristics of Dimension Tables:

* Descriptive Data: They contain the descriptive attributes related to the entities of the business. For example, a Product dimension will store information like the product name, category, manufacturer, etc.
* Few Columns, Many Rows: Dimension tables typically have fewer columns but many rows, representing different categories or entities.
* Primary Keys: The primary key of a dimension table is often used as a foreign key in the fact table.
* Attributes: Dimension tables contain attributes that provide detail and context for the facts.

**Slowly Changing Dimension**: These are the dimension tables that change slowly over time

**Conformed Dimension**: A dimension that can be used by multiple facts and has the same meaning across the model is called a Confirmed Dimension.

**Degenerate Dimension**:When a “fact” table stores dimensional values (not foreign keys to Dimensional tables) then we call it a degenerate dimension.

**Junk Dimension:**If the model has too many small dimensions then all of them can be put into one dimension though each of those small dimensions is unrelated.

**Role Playing Dimension**:A role-playing dimension consists of values that can be associated with multiple facts. For example, Sale location, Product Location, Person Location can all be related to a single Place Dimension but in changes in meaning based on the context.

**Static Dimension**:A static dimension usually never changes. Most often this would also mean that it is not driven by any of the source tables

**Shrunken Dimension**:If a dimension can be further divided into a smaller dimension (Snowflake) then that is called a shrunken dimension.

**Copying files from source to destination:**

To copy all files from one directory to another on your local machine using Python, you can use the **shutil** module

def copy\_files(source\_dir, destination\_dir):

# Ensure the destination directory exists

os.makedirs(destination\_dir, exist\_ok=True)

# Loop through all files in the source directory

for file\_name in os.listdir(source\_dir):

# Create full file paths

source\_file = os.path.join(source\_dir, file\_name)

destination\_file = os.path.join(destination\_dir, file\_name)

# Check if it's a file and copy it

if os.path.isfile(source\_file):

shutil.copy(source\_file, destination\_file)