**OLTP:** The term "On-Line Transaction Processing (OLTP) System" refers to a system for managing transaction-oriented applications. These systems are intended to enable online transactions and process queries fast via the Internet.  
For example: The POS (point of sale) system in any supermarket is an OLTP system.  
Every industry in today's world uses an OLTP system to capture transaction data. The primary concern of OLTP systems is to enter, store, and retrieve data. They cover all of an organization's day-to-day activities, including purchasing, manufacturing, payroll, accounting, and so on. Such systems have a huge number of users who complete brief transactions. It provides minimal database queries, resulting in incredibly rapid response times for any user operation.

Data collected by an OLTP system is kept in a commercial RDBMS, which may then be accessed by an OLAP system for data analytics and other business intelligence activities. Other types of OLTP systems include order entry, retail sales, and financial transaction systems.

**OLAP:** Online analytical processing, is a technique that enables high-speed complicated queries or multidimensional analysis on large amounts of data in a data warehouse, data lake, or other data repository. OLAP is used for business intelligence, decision support, and a wide range of forecasting and reporting applications.   
Most business data have numerous dimensions categories into which the data is classified for display, monitoring, or analysis. For example, sales numbers may include characteristics such as place (region, nation, state, shop), time (year, month, week, day), product (clothing, men/women/children, brand, kind), and others.

However, in a data warehouse or data lake, data sets are stored in tables, which can only arrange data into two of these dimensions at a time. OLAP collects data from numerous relational data sets and reorganizes it into a multidimensional structure, allowing for quick processing and smart analysis.

**DIFFERENCE BETWEEN OLAP AND OLTP:**

| **Category** | **OLAP (Online Analytical Processing)** | **OLTP (Online Transaction Processing)** |
| --- | --- | --- |
| Definition | It is well-known as an online database query management system. | It is well-known as an online database modifying system. |
| Data source | Consists of historical data from various Databases. | Consists of only operational current data. |
| Method used | It makes use of a data warehouse. | It makes use of a standard database management system (DBMS). |
| Application | It is subject-oriented. Used for Data Mining, Analytics, Decisions making, etc. | It is application-oriented. Used for business tasks. |
| Normalized | In an OLAP database, tables are not normalized. | In an OLTP database, tables are [normalized (3NF)](https://www.geeksforgeeks.org/third-normal-form-3nf/). |
| Usage of data | The data is used in planning, problem-solving, and decision-making. | The data is used to perform day-to-day fundamental operations. |
| Task | It provides a multi-dimensional view of different business tasks. | It reveals a snapshot of present business tasks. |
| Purpose | It serves the purpose to extract information for analysis and decision-making. | It serves the purpose to Insert, Update, and Delete information from the database. |
| Volume of data | A large amount of data is stored typically in TB, PB | The size of the data is relatively small as the historical data is archived in MB, and GB. |

**DATABASE NORMAL FORMS:**

**FIRST NORMAL FORM(1NF):** This is the simplest level of normalization, known as First Normal Form (1NF). In 1NF, every table cell must have a single value, and every column must have a distinct name. The first normal form makes searches simpler and helps get rid of redundant data.

**SECOND NORMAL FORM(2NF):** Every non-key attribute in 2NF must depend on the primary key in order to remove unnecessary data. This implies that every column rather than relating to other columns should have a direct relationship with the main key.

**THIRD NORMAL FORM(3NF):** All non-key qualities must be independent of one another for 3NF to build on 2NF. This implies that every column and not any other column in the same table should have a direct relationship to the main key.

**BOYCE-CODD NORMAL FORM (BCNF):** BCNF is a tougher version of 3NF that assures that every determinant in a table is a candidate key. In other words, BCNF guarantees that each non-key property is only dependent on the candidate key.

**FOURTH NORMAL FORM(4NF):** 4NF, a further refinement of BCNF, ensures that a table does not include any multi-valued dependencies.

**FIFTH NORMAL FORM(5NF):** The greatest level of normalization is 5NF, which entails breaking down a database into smaller tables to reduce data redundancy and improve data integrity.

**DIFFERENCE BETWEEN FACT TABLE AND DIMENSION TABLE:**

| **S.NO** | **Fact Table** | **Dimension Table** |
| --- | --- | --- |
| 1. | Fact table contains the measuring of the attributes of a dimension table. | Dimension table contains the attributes on that truth table calculates the metric. |
| 2. | In fact table, There is less attributes than dimension table. | While in dimension table, there is more attributes than fact table. |
| 3. | In fact table, there is more records than dimension table. | While in dimension table, there is less records than fact table. |
| 4. | Fact table forms a vertical table. | While dimension table forms a horizontal table. |
| 5. | The attribute format of fact table is in numerical format and text format. | While the attribute format of dimension table is in text format. |
| 6. | It comes after dimension table. | While it comes before fact table. |
| 7. | The number of fact table is less than dimension table in a schema. | While the number of dimensions is more than fact table in a schema. |
| 8. | It is used for analysis purpose and decision making. | While the main task of dimension table is to store the information about a business and its process. |

**TYPES OF DIMENSIONS:**

A framework for labeling the measurements and facts in order to respond to business inquiries. People, Date/Time, Products/Items, and Region/Area are dimensions that are frequently utilized.   
**Slowly Changing Dimensions:** (SCD) are dimensions that vary gradually over time as opposed to periodically or on a set basis. There are numerous strategies for managing SCD. The most well-liked ones are:   
**Type 0:** Passive Approach When a dimensional change occurs in this approach, no particular action is taken. Certain dimension data may be erased, while others may stay the same as when it was originally inserted.

**Type 1:** Overwriting the previous values.

**Type 2:** Creating a new additional record.

**Type 3:**  Adding a new column.

**Rapid/fast changing dimensions:** The Rapidly changing dimension (RCD) is a dimension which has attributes where values will be getting changed often.

**Junk dimensions:** A junk dimension is a small, auxiliary dimension in a data warehouse that combines low-cardinality attributes to reduce complexity.

**Degenerate dimension**: A degenerate dimension is a dimension table that consists of attributes already present in fact tables, eliminating the need for a separate dimension table.

**Role play dimension:** A role-playing dimension is a single dimension table used multiple times in a fact table, each time representing a different perspective or role.

**Shrunken dimension**: A shrunken dimension is a reduced form of a dimension table that eliminates infrequently used properties.

**Static dimension:** Static dimensions are dimension tables that do not frequently change and contain fixed, unchanging data over time.

# **SNOWFLAKE Vs STAR SCHEMA:**

|  |  |  |
| --- | --- | --- |
| Aspect | Star Schema | Snowflake Schema |
| Hierarchies | Hierarchies stored in dimensional table | Hierarchies divided into separate tables |
| Structure | Fact table surrounded by dimension tables | One fact table surrounded by dimension tables, which are further surrounded by dimension tables |
| Joins | Single join for fact table and dimensions | Many joins required for fetching data |
| Design Complexity | Simple design | Very complex design |
| Data Structure | Denormalized | Normalized |
| Data Redundancy | High | Very low |
| Aggregated Data | Single dimension table contains aggregated data | Data split into different dimension tables |
| Query Performance | Faster | May be slower due to complex joins |
| Cube Processing | Faster | May be slower due to complex joins |
| Join Optimization | Star Join Query Optimization | - |
| Table Connections | Tables connected with multiple dimensions | Centralized fact table unlikely connected with multiple dimensions |