**Data Warehouse Concepts**

**Difference between OLAP and OLTP**

**Online Analytical Processing and Online Transactional Processing:** Are the data processing systems that help you store and analyze business data. You can collect and store data from multiple sources—such as websites, applications, smart meters, and internal systems. OLAP **combines and groups** the data so you can analyze it from different points of view. Conversely, OLTP **stores and updates transactional data** reliably and efficiently in high volumes. OLTP databases can be one among several data sources for an OLAP system.

|  |  |  |
| --- | --- | --- |
|  | **Online Analytical Processing** | **Online Transactional Processing** |
| Purpose | OLAP helps you analyze large volumes of data to support decision-making. | OLTP helps you manage and process real-time transactions. |
| Data source | OLAP uses historical and aggregated data from multiple sources. | OLTP uses real-time and transactional data from a single source. |
| Data structure | OLAP uses multidimensional (cubes) or relational databases. | OLTP uses relational databases. |
| Data model | OLAP uses star schema, snowflake schema, or other analytical models. | OLTP uses normalized or denormalized models. |
| Volume of data | OLAP has large storage requirements. Think terabytes (TB) and petabytes (PB). | OLTP has comparatively smaller storage requirements. Think gigabytes (GB). |
| Response time | OLAP has longer response times, typically in seconds or minutes. | OLTP has shorter response times, typically in milliseconds |
| Example applications | OLAP is good for analyzing trends, predicting customer behavior, and identifying profitability. | OLTP is good for processing payments, customer data management, and order processing. |

**How Normalizations differ from OLTP and OLAP?**

**Normalization:** It is a process of organizing data in relational databases by reducing the redundancies, aiming to remove any data anomalies and improving the data integrity.

**1NF:** **Atomicity –** Remove duplicate columns from table (Only unique columns)

**2NF: No Partial dependency –** Place subset of data in different table and create a relation with primary key

**3NF: No Transitive dependency –** Removing all the columns that are not dependent on primary key

**4NF:** No attribute is dependent on combination on attributes which is not a candidate key

**Boyce NF: Every attribute is a candidate key**

**Normalized** data structures are typically used in **OLTP** systems, **denormalized** or multidimensional data structures are more common in **OLAP** systems to improve query performance. These concepts are often complementary and serve different needs within an organization.

**Fact Table and Dimension Table Relationship:**

In a data warehouse, the fact table serves as the repository for key performance indicators (KPIs) or measures. It holds summary data regarding business processes or events. Dimension tables, on the other hand, provide descriptive information about the objects or entities associated with the measures stored in the fact table. The relationship between the fact table and dimension table is established through primary and foreign keys.

**Schema Types:**

* **Star Schema:** In a star schema, the fact table is at the center, surrounded by dimension tables. This design simplifies querying and improves performance due to denormalization.
* **Snowflake Schema:** Snowflake schema represents normalized data with micro granularity. Here, the fact table is surrounded by dimension tables, and these dimension tables may further be connected to additional dimension tables.

**Slowly Changing Dimensions (SCD):**

* **SCD Type 1:** In SCD Type 1, no history of data changes is maintained. It's suitable for scenarios where historical data isn't critical, such as with debit card or credit card details.
* **SCD Type 2:** SCD Type 2 maintains all historical data changes using techniques like versioning, flags, and date tracking. This ensures a comprehensive historical record by storing previous and current versions of data.
* **SCD Type 3:** SCD Type 3 maintains current and previous historical data, offering partial history storage compared to Type 2.

**Surrogate Keys and Data Loading:**

Surrogate keys are system-generated keys used to substitute natural primary keys in dimension tables. These keys provide a unique identifier for each record in the table.

Data loading into the data warehouse can be categorized into two phases:

* **Initial Load:** The initial load involves loading data into the warehouse for the first time, populating the tables with the initial set of data.
* **Batch or Incremental Load:** After the initial load, subsequent updates or additions to the data are performed incrementally using batch or incremental loading techniques. This involves updating records based on attributes such as source update date, current date, and end date.

**Factless Fact Table and Types of Facts:**

* **Factless Fact Table:** Factless fact tables capture many-to-many relationships between dimensions without containing any measurable numeric or textual facts. They are useful for tracking events or occurrences where no measures are involved, such as student registrations.
* **Types of Facts:**
  + **Additive:** Additive facts allow for the summation of measures across all dimensions. Examples include quantity sold or dollars sold.
  + **Semi-Additive:** Semi-additive facts allow for the summation of measures across some dimensions but not all. An example is inventory levels.
  + **Non-Additive:** Non-additive facts do not allow for the summation of measures across any dimensions.

**Types of Dimensions:**

* **Conformed Dimension:** Conformed dimensions are dimension tables that are created or built once and used across multiple instances or processes. An example is a date dimension.
* **Junk Dimension:** Junk dimensions are collections of random transaction codes, flags, or text attributes. They help in simplifying and organizing data storage and analysis.
* **Degenerative Dimension:** Degenerative dimensions are characterized by having a foreign key present in the fact table without an associated primary key. They represent a measurable value in the fact table without a corresponding dimension table.