1. **Difference between OLAP and OLTP?**

|  |  |
| --- | --- |
| **OLAP (Online Analytical Processing)** | **OLTP (Online Transaction Processing)** |
| Online Database query Management system. | Online Database modifying system. |
| Historical Data. | Current Data. |
| Subject Oriented. | Application Oriented. |
| Tables are not Normalized. | Tables are Normalized. |
| Data Stored in TB/PB. | Data Stored in MB/GB. |
| Query operation is slow. | Query operation is fast. |
| Mutli- dimensional. | Two Dimensional. |
| DWH is Bulit on OLAP. | DBs are built on OLTP. |
| Process is focused on Customer. | Process is focused on market. |
| Designed with Subject. | Designed with Application. |
| Improves the efficiency of business analysts. | Enhances the user’s productivity. |
| Ready Only but rare write operations. | Both read and write operations. |

1. **Database Normalization and 5 normal forms?**

* Normalization is a database design technique used to organize data in a relational database efficiently and reduce data redundancy.
* It involves breaking down large tables into smaller, related tables and defining relationships between them.
* The goal of normalization is to eliminate data anomalies, such as insertion, update, and deletion anomalies, and ensure data integrity.
* There are several normal forms in normalization, each building upon the previous one.
* First Normal Form (1NF):
  + Ensures that each column in a table contains atomic values (indivisible values).
  + Eliminates repeating groups by creating separate tables for related data and using primary keys and foreign keys to establish relationships.
* Second Normal Form (2NF):
* Builds on 1NF by ensuring that all non-key attributes are fully functionally dependent on the primary key.
* Involves breaking down tables with composite primary keys and moving attributes that depend on only part of the primary key to separate tables.
* Third Normal Form (3NF):
  + Builds on 2NF by removing transitive dependencies, where non-key attributes depend on other non-key attributes.
  + Involves creating additional tables to store data that is functionally dependent on non-key attributes, but not on the primary key.
* There are also higher normal forms such as Boyce-Codd Normal Form (BCNF), Fourth Normal Form (4NF), and Fifth Normal Form (5NF), which further refine data organization and eliminate more complex dependencies.
* Normalization helps improve database performance, reduce storage requirements, prevent data anomalies, and simplify data maintenance.
* It's important to strike a balance between normalization and denormalization based on specific database requirements and performance considerations.

**3.Dimension Vs Fact Table and Types of Dimensions?**

* Dimension (Descriptive information)
* The tables that describe the dimensions involved are called Dimension tables.
* Dividing a Data Warehouse project into dimension provides structured information for analysis and reporting.

E-Commers ----- Subject

Customer Product Date ------ Dimension

------------------------------------------------- Attribute

**Facts & Measures**

* Fact is a measure that can be summed, averaged or manipulated.
* Fact table contains 2 Kinds of data. – Dimension key and Measure.
* Every Dimension table is linked to a fact table.

(The FK here which will be referencing to the primary key on our dimension tables.)

Product\_ID Dimension Key

Product

No of Units Sold Measure

(Calculation or Arithmetic value)

**Types of Dimensions**:

Slowly Changing Dimensions (SCDs): These dimensions capture historical changes in data over time.

* There are several types of SCDs:
  + **SCD Type 1**: Only Current will be there in Data Warehouse.
  + **SCD Type 2**: Stores all data (Flag, Versioning, Dates)
  + **SCD Type 3**: Stores Current and previous Data.
  + **SCD Type 4**: Uses separate tables to store historical changes, maintaining a current version of the dimension.
  + **Junk Dimensions**: These dimensions combine multiple flags or indicators into a single dimension to reduce the number of dimension tables and simplify the data model.
  + **Role-Playing Dimensions**: A dimension table can be used multiple times in a fact table with different roles. For example, a Date dimension can be used in the same fact table.
  + **Degenerate Dimensions**: These dimensions consist of attributes that are derived from the fact table itself, such as invoice numbers, order numbers, or transaction IDs that do not warrant a separate dimension table.
  + **Conformed Dimensions**: Dimensions that are shared and consistent across multiple fact tables in a data warehouse or across different data marts within an organization. Conformed dimensions ensure data consistency and enable integration of data across various analytical processes.

4.**Star schema Vs Snowflake Schema?**

|  |  |
| --- | --- |
| **Star schema** | **Snowflake Schema** |
| Stored in the dimensional table. | Divided into separate tables |
| Fact table surrounded by dimension tables. | One fact table surrounded by dimension table which are in turn surrounded by dimension table |
| Single join creates the relationship between the fact table and any dimension tables. | Requires many joins to fetch the data. |
| Simple Database design. | Very Complex Database Design. |
| Denormalized. | Normalized. |
| High Level of data redundancy. | Very Low Level of data redundancy. |
| Processing is high. | Processing is low. |
| Granularity level is low | Granularity level is high. |
| Every Dimension table will be link to fact table. | Every Dimension table is unlikely connected with multiple dimensions. |

**Star Schema:**

**Dimension Table Dimension Table**

**Fact table**

**Dimension Table Dimension Table**

**Snowflake Schema:**

**DT (Dimension Table)**

**FT (Fact Table)**

**DT**

**DT DT**

**DT FT DT**

**DT DT**