1. Write about OLTP and OLAP.

Nowadays, almost all businesses rely on data to run their operations. A business's ability to extract value from its data is essential to offering customers experiences and goods and services that maintain their importance and effectiveness.

Systems for processing data can be approached in two ways: one concentrates on operations, while the other focuses on analytics for business intelligence. Both are necessary to fully utilize the potential of data.

These two systems are OLTP stands for online transaction processing; and OLAP stands for online analytical processing.

Online transaction processing (OLTP) captures, stores, and processes data from transactions in real-time. Online analytical processing (OLAP) uses complex queries to analyze aggregated historical data from OLTP systems.

ABOUT OLTP (ONLINE TRANSACTION PROCESSING):

Large numbers of individuals can execute a high amount of database transactions in real-time by using online transaction processing or OLTP. OLTP systems are employed in several daily transactions, including those involving ATMs, online banking, e-commerce, text messaging, and account modifications.

They use a relational database or SQL database to handle extensive volumes of simple transactions. An OLTP system captures and maintains transaction data in a database. Each transaction involves individual database records made up of multiple fields or columns. This process can be challenging without the right tools.

Because OLTP databases are read, written, and updated often, speed is emphasized in OLTP. Data integrity is guaranteed by internal system logic if a transaction fails.  
Since OLTP and OLAP systems collaborate to maximize the value of data, OLTP systems can receive data from their OLAP systems.

Example for OLTP:

There are many examples of OLTP. One of the examples of OLTP is in the context of an e-commerce website. Imagine a popular online bookstore where customers can browse, search for books, add them to their cart, and proceed to checkout. The database system behind this website would operate as an OLTP system, efficiently handling numerous transactions in real time.

ABOUT OLAP:

OLAP systems are designed for complex analysis and reporting tasks. Data analysts and data engineers use online analytical processing (OLAP) for data mining, analytics, and business intelligence. OLAP is used to process multidimensional analysis on large volumes of data at very high speeds (milliseconds). An OLTP system often processes and stores data in repositories, which OLAP then sources for analysis. Many businesses use OLAP for financial analysis, forecasting, budgeting, reporting, marketing and sales optimization, and decision-making.

Example for OLAP:

Imagine a chain of retail stores that sells electronic gadgets and appliances. The company maintains a data warehouse containing historical sales data from its stores. They use OLAP for in-depth analysis of sales performance across different dimensions.

2) Difference between OLTP and OLAP?

|  |  |  |
| --- | --- | --- |
|  | **OLTP** | **OLAP** |
| **Characteristics** | Handles many small transactions | Handles large volumes of data with complex queries |
| **Query types** | Simple standardized queries | Complex queries |
| **Operations** | Based on INSERT, UPDATE, and DELETE commands | Based on SELECT commands to aggregate data for reporting |
| **Response time** | Milliseconds | Seconds, minutes, or hours depending on the amount of data to process |
| **Design** | Industry-specific, such as retail, manufacturing, or banking | Subject-specific, such as sales, inventory, or marketing |
| **Source** | Transactions | Aggregated data from transactions |
| **Purpose** | Control and run essential business operations in real-time | Plan, solve problems, support decisions, discover hidden insights |
| **Data updates** | Short, fast updates initiated by the user | Data periodically refreshed with scheduled, long-running batch jobs |
| **Space requirements** | Generally small if historical data is archived | Generally large due to aggregating large datasets |
| **Backup and recovery** | Regular backups are required to ensure business continuity and meet legal and governance requirements | Lost data can be reloaded from the OLTP database as needed instead of regular backups |
| **Productivity** | Increases productivity of end users | Increases productivity of business managers, data analysts, and executives |
| **Data view** | Lists day-to-day business transactions | Multi-dimensional view of enterprise data |
| **User examples** | Customer-facing personnel, clerks, online shoppers | Knowledge workers such as data analysts, business analysts, and executives |
| **Database design** | Normalized databases for efficiency | Denormalized databases for analysis |

3) Database normal forms?

Definition:

Normalization is the process of creating a set schema to store non-redundant and consistent data. It is used to reduce data redundancy and inconsistency. It is used in the OLTP system, where the emphasis is on making the insert, deletes, and updates anomalies faster and storing the quality data.

1. First normal form: It is used to remove multivalued attributes

2. Second normal form: It is used to remove partial dependencies

3. Third normal form: It is used to remove transitive dependencies

4. Boyce-Codd normal form: It is used to remove remaining anomalies resulting from functional dependencies

5. Fourth normal form: It is used to remove multivalued dependencies

6. Fifth normal form: It is used to remove remaining anomalies.

4) Dimensions vs fact table?

The Dimension table is a partner to the fact table and contains descriptive qualities that can be used as query constraints. The fact table includes measurements, metrics, or facts about business operations.

A star and snowflake schema's core is where the table containing it is located, while its edges are where the dimension table is situated.

A fact table should be lengthy, descriptive, full, and of guaranteed quality, while a fact table is characterized by its grain or even most atomic level.

While the Dimension database includes extensive information, the Fact table is used to hold report labels.

In contrast to the Dimension table, which incorporates hierarchies, the Table does not.

5) Types of dimensions?

1. Conformed Dimensions:

Conformed dimensions are dimensions that are shared and consistent across multiple data marts or data warehouse systems. They provide a consistent view of data across the organization, ensuring that the same dimension attributes are used consistently in different parts of the data warehouse. An example would be Item, region, location, and Datetime dimensions, i.e. both Inventory and sales departments can use such dimensions for their reporting purpose.

1. Junk Dimensions:

A dimension is considered junk if it has several low-cardinality flags or indicators that don't belong in any one category. To decrease the number of dimension tables and simplify the data model, they are made by integrating multiple minor dimensions or characteristics into a single-dimension table. Status indicators, binary flags, and categorical features with a finite number of possible values are commonly stored in junk dimensions.

1. Degenerate Dimensions:

Degenerate dimensions are those that aren't kept in different dimension tables; instead, they are contained within fact tables. They stand for dimension attributes that belong in a transactional data set but don't need their dimension table. When storing transactional IDs or keys that are important for analysis but unrelated to any dimension, degenerate dimensions are frequently utilized.

1. Role-playing Dimensions:

Role-playing dimensions are those that are applied to a single data model but in different roles or circumstances. Depending on the context, they can be understood differently even though they reflect the same set of properties. Within the same data model, a date dimension, for instance, can function as a role-playing dimension for the order, shipping, and delivery dates.

1. Slowly Changing Dimensions (SCDs):

Dimensions that change gradually over time are known as slowly changing dimensions, and to conduct analysis, their previous values must be retained. Depending on how they respond to modifications to dimension attributes, they are divided into three categories (Type 1, Type 2, and Type 3). Type 2 SCDs produce new records to track historical changes, Type 3 SCDs save both current and past values to keep limited history, and Type 1 SCDs overwrite existing dimension attributes with new values.

1. Static dimensions:

Static Dimensions are developed inside the data warehouse; they are not taken from the original data source. A static dimension, like a date or time dimension, can be created automatically by a process or explicitly loaded, such as in the case of status codes.

6. Snowflake vs Star Schema?

|  |  |
| --- | --- |
| Star Schema | Snowflake Schema |
| The star schema is the simplest data warehouse scheme. | Snowflake schema is a more complex data warehouse model than a star schema. |
| In a star schema, each of the dimensions is represented in a single table. It should not have any hierarchies between dims. | In the Snowflake schema, at least one hierarchy should exist between dimension tables. |
| It contains a fact table surrounded by dimensions that are de-normalizes, we say it is a star schema design. | It contains a fact table surrounded by dimension tables. If the dimensions table is normalized, we can it is a snowflake. |
| In a star schema, only one join establishes the relationship between the fact table and any one of the dimension tables. | In snowflake schema, since there is a relationship between the dimensions tables it must do many joins to fetch the data |
| A star schema optimizes performance by keeping queries simple and providing fast response time. All information is stored in one row. | Snowflake schemes normalize dimensions to eliminate redundancy. The result is more complex queries and reduced query performance |