AWS \_CLASS 1

1. **What is OLTP and OLAP?**

**OLTP (Online Transaction Processing): -**

* OLTP systems are designed for managing and processing day-to-day transactional data generated by operational applications.
* These systems are optimized for handling high volumes of short, rapid transactions, such as recording sales, processing orders, or updating customer information.
* OLTP databases typically have normalized schemas to minimize data redundancy and ensure data integrity.
* They prioritize fast response times and concurrent access, focusing on transaction throughput and ensuring data consistency.
* Examples of OLTP systems include retail point-of-sale systems, banking transaction systems, airline reservation systems, and online booking systems.

**OLAP (Online Analytical Processing): -**

* OLAP systems are designed for analyzing and querying large volumes of historical or aggregated data to gain insights and support decision-making processes.
* These systems are optimized for complex analytical queries, such as multidimensional analysis, data mining, and trend analysis.
* OLAP databases often use denormalized or star schema structures to optimize query performance by pre-aggregating data and reducing the number of joins.
* They prioritize query performance, providing fast responses to complex analytical queries over large datasets.
* Examples of OLAP systems include data warehouses, business intelligence platforms, and reporting tools used for strategic decision-making and data analysis.

**2) differences between OLTP and OLAP:**

|  |  |  |
| --- | --- | --- |
|  | **OLTP** | **OLAP** |
| Purpose | OLTP systems are designed for managing and processing day-to-day transactional data generated by operational applications. | OLAP systems are designed for analyzing and querying large volumes of historical or aggregated data to gain insights and support decision-making processes. |
| Type of Data | OLTP systems handle current, operational data generated by routine transactions, such as sales, orders, and customer interactions. | OLAP systems deal with historical or aggregated data used for analytical querying and decision support, such as sales trends, customer behavior, and market analysis. |
| **Database Structure** | OLTP databases typically have normalized schemas to minimize data redundancy and ensure data integrity. | OLAP databases often use denormalized or star schema structures to optimize query performance by pre-aggregating data and reducing the number of joins. |
| **Query Complexity** | OLTP systems prioritize fast response times and concurrent access for short, rapid transactions, focusing on transaction throughput and data consistency. | OLAP systems prioritize query performance, providing fast responses to complex analytical queries over large datasets. |
| Usage | OLTP systems are used in operational environments for routine transactional processing, such as retail sales, banking transactions, and online bookings. | OLAP systems are used in analytical environments for strategic decision-making, data analysis, and business intelligence, supporting tasks such as trend analysis, forecasting, and performance monitoring. |

**3)database normal forms:**

Normalization is the process of minimizing redundancy from a relation or set of relations. Redundancy in relation may cause insertion, deletion, and update anomalies. So, it helps to minimize the redundancy in relations. Normal forms are used to eliminate or reduce redundancy in database tables.

* **First Normal Form (1NF):**
  + Ensures that each column in a table contains atomic (indivisible) values and there are no repeating groups of columns.
  + Each attribute must have a single value for each tuple (row) in the table.
* **Second Normal Form (2NF):**
  + Builds on 1NF and eliminates partial dependencies by ensuring that all non-key attributes are fully functionally dependent on the primary key.
  + It means that no non-key attribute is dependent on only a portion of the primary key.
* **Third Normal Form (3NF):**
  + Builds on 2NF and eliminates transitive dependencies by ensuring that all non-key attributes are non-transitively dependent on the primary key.
  + It means that no non-key attribute depends on another non-key attribute.
* **Boyce-Codd Normal Form (BCNF):**
  + A stronger version of 3NF, where every determinant is a candidate key.
  + It addresses certain types of anomalies that may arise in tables that are in 3NF but still have overlapping candidate keys.
* **Fourth Normal Form (4NF):**
  + Addresses multi-valued dependencies, where a non-key attribute is functionally dependent on a combination of attributes that are not a candidate key.
  + It ensures that a relation is free of a particular type of redundancy known as a multi-valued dependency.
* **Fifth Normal Form (5NF):**
  + Also known as Project-Join Normal Form (PJ/NF), it addresses join dependencies, which occur when data is redundantly stored in multiple tables.
  + It eliminates redundancy by decomposing tables and using foreign keys to re-establish relationships.
* **Domain-Key Normal Form (DK/NF):**
  + Ensures that all constraints are domain constraints, meaning they are logical assertions about the types of values that are permitted in columns.
  + It ensures that every constraint on the table is a logical consequence of the definition of its attributes.

**4)dimension vs Fact table:**

|  |  |
| --- | --- |
| **Dimension Table** | **Fact Table** |
| Contains descriptive attributes (metadata) that provide context and categorization for the data in a data warehouse. | Contains quantitative measures or metrics (facts) that represent business transactions or events. |
| Typically, smaller in size compared to fact tables. | Typically, larger in size compared to dimension tables. |
| Used for filtering, grouping, and analyzing data in analytical queries. | Used for aggregating and analyzing numerical data using various dimensions. |
| Often referenced by fact tables through foreign key relationships. | Often includes foreign key references to dimension tables to establish relationships. |
| Examples include customer dimension (containing attributes like customer ID, name, address), product dimension (containing attributes like product ID, name, category), time dimension (containing attributes like date, month, year). | Examples include sales fact table (containing measures like revenue, quantity sold, discount), inventory fact table (containing measures like stock level, reorder quantity). |

**types of dimensions:**

**Slowly Changing Dimension (SCD):**

* + Represents entities with attributes that change over time but at a slow rate.
  + Types of SCDs include:
  + Type 1 SCD: Overwrites existing data with new values, losing historical information.
  + Type 2 SCD: Adds new rows for each change, preserving historical data and creating a versioning history.
  + Type 3 SCD: Adds new columns to the dimension table to track changes, providing limited historical information.

**Conformed Dimension:**

* + Dimension that has the same meaning and content when referred from different fact tables.
  + Ensures consistency and allows for integration and comparison of data across multiple data marts or data warehouses.

**Junk Dimension:**

* + Consolidates multiple low-cardinality flags or indicators into a single dimension table to reduce the number of joins in queries.
  + Contains non-meaningful or irrelevant attributes that don't fit into existing dimensions.

**Role-Playing Dimension:**

* + A single dimension table that is referenced multiple times in a fact table, each time representing a different role or context.
  + Example includes a date dimension table referenced multiple times for order date, ship date, delivery date, etc.

**Degenerate Dimension:**

* + A dimension that is derived from fact table attributes rather than being stored as a separate dimension table.
  + Typically consists of transactional or transaction-related attributes that are not part of any existing dimension.Top of Form**Bottom of Form**

**5) snowflake vs star schema:**

**Snowflake Schema:**

* Snowflake schema is a database modeling technique where dimension tables are normalized, resulting in reduced redundancy and improved data integrity.
* Named after its resemblance to a snowflake due to the normalized structure, which may include sub-dimension tables.
* Suitable for data warehousing environments prioritizing data integrity and space optimization.

**Star Schema:**

* Star schema is a database modeling technique where dimension tables are denormalized and directly connected to the fact table, forming a star-like structure.
* Offers a simpler and intuitive schema design, with better query performance compared to snowflake schema.
* Ideal for scenarios prioritizing query performance and where schema complexity can be managed effectively**.**
  + **Top of Form**

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  | |
|  | |  | |
|  | |  | |
|  | |  | |
|  | |  | |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | |  |

* + **Bottom of Form**

|  |  |  |
| --- | --- | --- |
|  | **Snowflake Schema:** | **Star Schema:** |
| **Structure** | Consists of a centralized fact table surrounded by multiple dimension tables, where each dimension table may be further normalized into sub-dimension tables. | Consists of a centralized fact table surrounded by denormalized dimension tables, forming a star-like structure. |
| **Characteristics** | Offers higher levels of normalization compared to the star schema, leading to reduced redundancy and better data integrity. | Offers simpler and more intuitive schema design compared to snowflake schema, making it easier to understand and query. |
| **Advantages** | Improved data integrity: Normalization reduces redundancy and helps maintain consistency in the data. | Simplified design: Denormalized dimension tables result in a simpler and more intuitive schema structure. |
| **Disadvantages** | Increased complexity: The normalization of dimension tables can lead to a more complex schema design, making it harder to understand and manage. | Simplified design: Denormalized dimension tables result in a simpler and more intuitive schema structure. |