**NAME: PRAKHAR MISHRA**

**DATE: 18/01/2024**

**QUESTIONS**

1. **What do you mean by RDBMS?**

* **Relational Data Model:** RDBMS, which stands for Relational Database Management System, is a type of database management system that is based on the relational data model. In this model, data is organized into tables with rows and columns, allowing for easy representation and retrieval of information.
* **Structured Query Language (SQL):** RDBMS uses SQL as its standard language for interacting with the database. SQL enables users to define, manipulate, and query the data stored in the relational database. Common SQL operations include SELECT (retrieve data), INSERT (add new data), UPDATE (modify existing data), and DELETE (remove data).
* **Data Integrity:** RDBMS ensures data integrity by enforcing constraints on the data stored in tables. This includes primary key constraints, which ensure each row in a table is uniquely identified, and foreign key constraints, which establish relationships between tables. These constraints maintain the accuracy and consistency of the data.
* **ACID Properties:** RDBMS adheres to the ACID properties (Atomicity, Consistency, Isolation, and Durability) to ensure reliable and robust transaction processing. These properties guarantee that database transactions are processed reliably even in the event of system failures, ensuring the overall integrity of the data.
* **Normalization:** RDBMS supports normalization, a process of organizing data to reduce redundancy and dependency within a database. This is achieved by breaking down large tables into smaller, related tables and establishing relationships between them. Normalization enhances data integrity, reduces data duplication, and makes the database more flexible and efficient.

1. **What do you mean by OLAP and OLTP?**

* **OLAP (Online Analytical Processing):** OLAP is a category of software tools that allows users to interactively analyze and explore multidimensional data from different perspectives. It is optimized for complex queries and reporting, enabling users to gain insights from historical and aggregated data for decision-making purposes.
* **OLTP (Online Transaction Processing):** OLTP is a type of database processing focused on managing and facilitating real-time transaction-oriented applications. It handles day-to-day operations, such as inserting, updating, and deleting records in a database. OLTP databases are designed for high transaction throughput and support concurrent access by multiple users.
* **Data Structure:** OLAP databases typically use a multidimensional data structure, organized into data cubes, where data is pre-aggregated for quick analysis. In contrast, OLTP databases use a normalized relational data structure, minimizing redundancy and ensuring efficient transaction processing.
* **Query Complexity:** OLAP involves complex queries that aggregate and analyze large volumes of historical data. OLAP queries often involve operations like slicing, dicing, drilling down, and pivoting to explore data from various dimensions. OLTP queries, on the other hand, are simpler and focused on individual transactions, such as inserting or updating a single record.
* **Performance and Workload:** OLAP systems are optimized for read-heavy workloads and complex analytical queries, providing fast response times for decision support. OLTP systems are optimized for write-heavy workloads, emphasizing quick and efficient transaction processing for day-to-day business operations.

1. **What are the features of Data Warehousing? Explain them**.

There are four basic features of Data Warehousing:

1. **Subject-Oriented**: Data are organized according to the subject instead of application.

**Example**: In a retail data warehouse, data is organized around subjects like sales, inventory, and customer demographics rather than being separated by individual applications. This makes it easier for analysts to focus on specific business areas when extracting insights.

1. **Time-Invariant**: The time duration for data warehouse is longer than that of operational systems.

**Example**: A financial institution tracks daily account balances, and this historical data is stored in the data warehouse. This enables analysts to assess trends, monitor account fluctuations over time, and identify patterns in customer transactions.

1. **Integrated**: Constructed by integrating multiple, heterogeneous data sources like relational databases, flat files, on-line transaction records.

Ensure consistency in naming conventions, encoding structures, attribute measures, etc. among different data sources.

**Example:** A manufacturing company integrates data from production, supply chain, and sales systems into a data warehouse. This integration ensures that information about inventory levels, production schedules, and sales performance is consistent and can be analyzed together.

1. **Non-Volatile**: No updates are allowed. Once the data entered into the data warehouse, they are never removed.

The data in warehouse represent company’s history, the operational data representing near term history are always added to it.

**Example:** A healthcare organization stores patient records in a data warehouse. Once a patient's data is entered, it remains unchanged in the data warehouse, providing a reliable historical record for medical research and compliance purposes.

1. **Explain SQL and its features**.

**Structured Query Language (SQL):**

**Definition:** SQL (Structured Query Language) is a standardized programming language used for managing and manipulating relational databases. It provides a set of commands for defining, querying, updating, and managing relational databases.

**Data Querying and Retrieval:** SQL allows users to retrieve data from a database using the SELECT statement. This statement enables users to specify the columns they want to retrieve, apply filters, and sort the results.

Example: **SELECT column1, column2 FROM table WHERE condition;**

**Data Modification:** SQL supports data manipulation operations like INSERT, UPDATE, and DELETE for adding, modifying, and removing records in a database.

Example: **INSERT INTO table (column1, column2) VALUES (value1, value2);**

**UPDATE table SET column1 = new\_value WHERE condition;**

**DELETE FROM table WHERE condition;**

**Data Definition:** SQL provides statements for defining the structure of a database, including creating tables (CREATE TABLE), altering table structures (ALTER TABLE), and dropping tables (DROP TABLE).

Example:

**CREATE TABLE table\_name (**

**column1 datatype,**

**column2 datatype,**

**PRIMARY KEY (column1)**

**);**

**Data Integrity and Constraints**: SQL supports the definition of constraints to maintain data integrity, including primary key constraints, foreign key constraints, unique constraints, and check constraints.

Example:

**ALTER TABLE table\_name ADD CONSTRAINT constraint\_name PRIMARY KEY (column1);**

**Data Aggregation and Grouping:** SQL includes functions like SUM, AVG, COUNT, MIN, and MAX for aggregating data. The GROUP BY clause allows grouping data based on specified columns.

Example:

**SELECT column1, SUM(column2) FROM table GROUP BY column1;**

**Joins and Relationships:** SQL supports joining tables based on common columns using JOIN operations. This enables the retrieval of related data from multiple tables.

Example:

**SELECT \* FROM table1**

**INNER JOIN table2 ON table1.column = table2.column;**

**SQL's versatility and comprehensive set of features make it a powerful tool for managing and interacting with relational databases**.