**Interview Questions**

**Topic: DBMS & RDBMS**

1. What is Database?

A database is an organized collection of data that is stored and accessed electronically. It can include anything from a simple list of contacts to complex data sets used in large enterprises.

1. What is DBMS?

A Database Management System (DBMS) is software that enables users to create, manage, and organize databases in a secure and efficient manner. It provides a way to store and retrieve data, as well as tools for managing access and maintaining the integrity of the data.

1. What is RDBMS?

RDBMS stands for Relational Database Management System. RDBMS store the data into the collection of tables, which is related by common fields between the columns of the table. It also provides relational operators to manipulate the data stored into the tables.

1. What is Data?

Data is information that is stored and processed in a computer system. It can include anything from simple text to complex sets of information used to make decisions and inform analysis. Data is often raw and meaningless without further processing or interpretation.

1. What is SQL?

SQL stands for Structured Query Language. It is a programming language used to manage and manipulate data in a relational database. With SQL, users can create, update, and delete data, as well as retrieve information for analysis and reporting purposes.

1. What are the different types of RDBMS?

The different types of RDBMS have unique features and capabilities that distinguish them from each other.

MySQL: known for its speed and ease of use, and is often used for web-based applications.

Oracle: known for its scalability, reliability, and advanced features, and is often used in large enterprises for mission-critical applications.

Microsoft SQL Server: often preferred in Windows-based environments, and offers advanced features such as business intelligence and data warehousing.

PostgreSQL: known for its powerful features and support for complex data types, and is often used for large-scale and complex applications.

SQLite: a lightweight, file-based option that is commonly used in mobile and embedded applications, and is often used for simple or small-scale applications.

1. What is Data Model?

A data model is a conceptual representation of data structures and relationships within a database system. It defines how data is organized, stored, and accessed in a system, and provides a way to describe the data in a way that can be easily understood by users and developers.

1. What is E-R Model?

The E-R (Entity-Relationship) model is a type of data model used to describe the relationships between entities (objects) in a database. It shows how different entities are related to each other and how they can be accessed and manipulated. The E-R model is commonly used to design and develop relational databases.

1. What are the types of Relationships?

One-to-one relationship: A relationship between two tables where one record in the first table corresponds to one and only one record in the second table, and vice versa.

One-to-many relationship: A relationship between two tables where one record in the first table corresponds to one or more records in the second table, but each record in the second table corresponds to only one record in the first table.

Many-to-many relationship: A relationship between two tables where one record in the first table can correspond to many records in the second table, and vice versa. This type of relationship requires a third table, known as a junction or bridge table, to link the two tables together.

1. What is ACID?

Atomicity: Ensures that a transaction is treated as a single, indivisible unit of work.

Consistency: Ensures that the database remains in a valid state before and after the transaction.

Isolation: Ensures that concurrent transactions do not interfere with each other.

Durability: Ensures that the results of a committed transaction are permanent and can survive system failures.

1. What is a Table?

In a database, a table is a collection of related data organized into rows and columns. Each row represents a unique record, and each column represents a specific attribute of that record. Tables are the basic structure used to store and manage data in a relational database system.

1. What is a Column?

In a database table, a column (also known as a field or attribute) is a vertical set of data values that belong to a specific type of data. Each column is identified by a unique name and represents a specific type of information or attribute of the data entity represented by the table. For example, in a customer database, columns might include customer name, address, email, and phone number.

1. What is a Row?

In a database table, a row (also known as a record or tuple) is a horizontal set of data values that represent a single instance of the entity represented by the table. Each row contains data values for each column of the table, which describe a specific aspect of the entity being represented. For example, in a customer database, a row might contain the customer's name, address, email, and phone number.

1. What is a Key? What are its types?

In a database, a key is a field or set of fields used to uniquely identify each record in a table. It is used to establish relationships between different tables in a database and to ensure data integrity. There are several types of keys in a database, including:

**Primary key:** A primary key is a unique identifier for each record in a table, and it cannot contain null values.

**Foreign key:** A foreign key is a field in a table that refers to the primary key of another table and is used to establish relationships between tables.

**Candidate key:** A candidate key is a set of fields that can uniquely identify a record in a table but is not the primary key.

**Alternate key:** An alternate key is a candidate key that is not chosen as the primary key.

**Composite key:** A composite key is a key that is made up of two or more fields, and together they uniquely identify a record in a table.

1. What is Constraint? Why is it required?

In a database, constraints are rules or restrictions applied to columns or tables to ensure data integrity and accuracy. Constraints are used to enforce business rules and maintain the quality of the data stored in a database.

Constraints are required for several reasons, including:

**Ensuring data accuracy and consistency:** Constraints ensure that only valid data is entered into a database and help prevent errors and inconsistencies.

**Maintaining data integrity:** Constraints help maintain the relationships between tables and ensure that data is consistent across the database.

**Preventing data loss:** Constraints help prevent data loss by ensuring that only valid and complete data is entered into the database.

**Enforcing business rules:** Constraints help enforce business rules and ensure that the data stored in a database meets the requirements of the organization or application that uses it.

1. What are the different types of Constraints?

**Primary key constraint:** A primary key constraint is used to uniquely identify each record in a table and prevents duplicate entries.

**Foreign key constraint:** A foreign key constraint is used to establish relationships between tables and ensures that the data in the foreign key column corresponds to data in the primary key column of another table.

**Unique constraint:** A unique constraint ensures that each value in a column is unique and prevents duplicate entries.

**Check constraint:** A check constraint is used to ensure that data in a column meets certain criteria or conditions.

**Not null constraint:** A not null constraint ensures that a column cannot contain null values.

**Null constraint:** A null constraint specifies whether a column can contain null values or not. If a column has a not null constraint, then it cannot contain null values.

**Date constraints:** Date constraints are used to restrict the range of dates that can be entered into a column. For example, a column may be constrained to only allow dates within a certain range, or to only allow dates in a specific format.

1. What is Data type?

In a database, data types define the type of data that can be stored in a column, such as integer, string, or date. Data types are important for maintaining the accuracy and consistency of the data stored in a database, as they ensure that each column contains only the appropriate type of data.

**Types of Data types:**

**Numeric data types:** Used to store numeric values such as integers or floating-point numbers.

**Character data types:** Used to store textual data such as strings of characters or words.

**Date/time data types:** Used to store date and time values.

**Boolean data types:** Used to store true/false or yes/no values.

**Binary data types:** Used to store binary data such as images or audio files.

**Array data types:** Used to store arrays or lists of values.

**Date:** Used to store a date value, such as '2023-03-11'.

**Time:** Used to store a time value, such as '13:30:00'.

**DateTime:** Used to store a combination of date and time values, such as '2023-03-11 13:30:00'.

**Timestamp:** Used to store a timestamp value, which is a unique identifier for a specific moment in time.

1. What is Normalization?

Normalization is a process of organizing data in a database to eliminate redundancy and improve data integrity.

There are several types of normalization, including first normal form (1NF), second normal form (2NF), and third normal form (3NF). These levels of normalization progressively remove redundancies and dependencies between attributes.

1. What is Denormalization?

Denormalization is the process of intentionally adding redundant data to a database in order to improve performance by reducing the number of table joins required for queries. It can increase query speed but may also make data updates more difficult and increase the risk of inconsistencies. Denormalization is often used in high-transaction-volume systems where query performance is critical.