**1. What do you understand By Database?**

A.The Database is an essential part of our life. As we encounter several activities that involve our interaction with databases, for example in the bank, in the railway station, in school, in a grocery store, etc. These are the instances where we need to store a large amount of data in one place and fetch these data easily.

Data : statistics it is raw and unprocessed . ex- name , class , marks etc.

information = when data is processed . ” record is also information “. example – pass or fail table etc.

Database : an organized collection of data and information or interrelated data collected at one place.

**2.What is Normalization**?

A. Normalization is the process of reducing data redundancy in a table and improving data integrity.

Normalization involves organizing the columns and tables in the database to ensure that their dependencies are correctly implemented using database constraints.

Normalization is the process of organizing data in a proper manner. It is used to minimize the duplication of various relationships in the database. It is also used to troubleshoot exceptions such as inserts, deletes, and updates in the table. It helps to split a large table into several small normalized tables. Relational links and links are used to reduce redundancy.

Normalization, also known as database normalization or data normalization, is an important part of relational database design because it helps to improve the speed, accuracy, and efficiency of the database.

**3.What is Difference between DBMS and RDBMS?**

**A**. Database Management System (DBMS) is a software that is used to define, create and maintain a database and provides controlled access to the data.

Relational Database Management System (RDBMS) is an advanced version of a DBMS.

DBMS stores data as a file whereas in RDBMS, data is stored in the form of tables.

DBMS supports single users, while RDBMS supports multiple users.

DBMS does not support client-server architecture but RDBMS supports client-server architecture.

DBMS has low software and hardware requirements whereas RDBMS has higher hardware and software requirements.

In DBMS, data redundancy is common while in RDBMS, keys and indexes do not allow data redundancy

4.What is MF Cod Rule of RDBMS Systems?

A. Codd’s 12 Rules for Relational Database Management

Edgar F. Codd wrote a paper in 1985 defining rules for Relational Database Management Systems (RDBMS), which revolutionized the IT industry. In 1993, Codd and colleagues worked up these 12 rules for defining OLAP (Online Analytical Processing), an industry of software and data processing which allows consolidation and analysis of data in a multidimensional space. Codd’s 12 rules are:

Multidimensional conceptual view

User-analysts would view an enterprise as being multidimensional in nature – for example, profits could be viewed by region, product, time period, or scenario (such as actual, budget, or forecast). Multi-dimensional data models enable more straightforward and intuitive manipulation of data by users, including “slicing and dicing“.

Transparency

When OLAP forms part of the users’ customary spreadsheet or graphics package, this should be transparent to the user. OLAP should be part of an open systems architecture which can be embedded in any place desired by the user without adversely affecting the functionality of the host tool. The user should not be exposed to the source of the data supplied to the OLAP tool, which may be homogeneous or heterogeneous.

Accessibility

The OLAP tool should be capable of applying its own logical structure to access heterogeneous sources of data and perform any conversions necessary to present a coherent view to the user. The tool (and not the user) should be concerned with where the physical data comes from.

Consistent reporting performance

Performance of the OLAP tool should not suffer significantly as the number of dimensions is increased.

Client/server architecture

The server component of OLAP tools should be sufficiently intelligent that the various clients can be attached with minimum effort. The server should be capable of mapping and consolidating data between disparate databases.

Generic Dimensionality

Every data dimension should be equivalent in its structure and operational capabilities.

Dynamic sparse matrix handling

The OLAP server’s physical structure should have optimal sparse matrix handling.

Multi-user support

OLAP tools must provide concurrent retrieval and update access, integrity and security.

Unrestricted cross-dimensional operations

Computational facilities must allow calculation and data manipulation across any number of data dimensions, and must not restrict any relationship between data cells.

Intuitive data manipulation

Data manipulation inherent in the consolidation path, such as drilling down or zooming out, should be accomplished via direct action on the analytical model’s cells, and not require use of a menu or multiple trips across the user interface.

Flexible reporting

Reporting facilities should present information in any way the user wants to view it.

Unlimited Dimensions and aggregation levels.

The number of data dimensions supported should, to all intents and purposes, be unlimited. Each generic dimensions should enable an essentially unlimited number of user-defined aggregation levels within any given consolidation path.

**5.What do you understand By Data Redundancy?**

A. Data redundancy occurs when the same piece of data exists in multiple places, whereas data inconsistency is when the same data exists in different formats in multiple tables. Unfortunately, data redundancy can cause data inconsistency, which can provide a company with unreliable and/or meaningless information.

**6.What is DDL Interpreter?**

A.A data definition language (DDL) is a computer language used to create and modify the structure of database objects in a database. These database objects include views, schemas, tables, indexes, etc.

This term is also known as data description language in some contexts, as it describes the fields and records in a database table.

**7.What is DML Compiler in SQL?**

A. DML Compiler: It processes the DML statements into low level instruction (machine language), so that they can be executed.

DML stands for Data Manipulation Language.DML compiler translates the DML statements which are there in a query language into the low-level instructions which the query evaluation engine understands easily.

**8. What is SQL Key Constraints writing an Example of SQL Key Constraints**

A. SQL constraints are used to specify rules for the data in a table.

Constraints are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the table. If there is any violation between the constraint and the data action, the action is aborted.

Constraints can be column level or table level. Column level constraints apply to a column, and table level constraints apply to the whole table.

The following constraints are commonly used in SQL:

NOT NULL - Ensures that a column cannot have a NULL value

UNIQUE - Ensures that all values in a column are different

PRIMARY KEY - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table

FOREIGN KEY - Prevents actions that would destroy links between tables

CHECK - Ensures that the values in a column satisfies a specific condition

DEFAULT - Sets a default value for a column if no value is specified

CREATE INDEX - Used to create and retrieve data from the database very quickly

**9.What is save Point? How to create a save Point write a Query?**

A. Savepoint is a command in SQL that is used with the rollback command.

It is a command in Transaction Control Language that is used to mark the transaction in a table.

Consider you are making a very long table, and you want to roll back only to a certain position in a table then; this can be achieved using the savepoint.

If you made a transaction in a table, you could mark the transaction as a certain name, and later on, if you want to roll back to that point, you can do it easily by using the transaction's name.

Savepoint is helpful when we want to roll back only a small part of a table and not the whole table. In simple words, we can say savepoint is a bookmark in SQL.

Let us see the practical examples to understand this concept more clearly. We will use the MySQL database for writing all the queries.

To create a table in the database, first, we need to select the database in which we want to create a table.

mysql> USE dbs;

Then we will write a query to create a table named student in the selected database 'dbs'.

mysql> CREATE TABLE student(ID INT, Name VARCHAR(20), Percentage INT, Location VARCHAR(20)

**10. What is trigger and how to create a Trigger in SQL?**

A. A trigger is a special type of stored procedure that automatically runs when an event occurs in the database server. DML triggers run when a user tries to modify data through a data manipulation language (DML) event. DML events are INSERT, UPDATE, or DELETE statements on a table or view.

For example, a trigger can be invoked when a row is inserted into a specified table or when certain table columns are being updated. Explanation of syntax: create trigger [trigger\_name]: Creates or replaces an existing trigger with the trigger\_name. [before | after]: This specifies when the trigger will be executed.

**1. Get First\_Name from employee table using Tom name “Employee Name”**

A. SELECT First\_name AS "Employee Name" FROM employee;

**2.Get FIRST\_NAME, Joining Date, and Salary from employee table.**

A. SELECT First\_name,joining\_date ,salary FROM employee;

**3. Get all employee details from the employee table order by First Name Ascending**

**And Salary descending?**

SELECT \* FROM employee ORDER BY First\_name ASC , Salary DESC;

**4.Get employee details from employee table whose first name contains ‘J’**

SELECT \* FROM employee WHERE First\_name like "%J";

5. **Get department wise maximum salary from employee table order by salary**

**ascending?**

SELECT MAX(Salary) FROM employee;

**6.Select first\_name, incentive amount from employee and incentives table for**

**those employees who have incentives and incentive amount greater than 3000**

A. SELECT First\_name,Incentive\_amount FROM employee LEFT JOIN incentive ON employee.First\_name = incentive.Incentive\_amount;

**1 All orders for more than $1000**

A. SELECT \* FROM customer WHERE (sno > 1000);

**2 Names and cities of all salespeople in London with commission above 0.12**

SELECT sname,city,comm FROM salseperson WHERE (city = "london") AND (comm > ".10");

**3.All salespeople either in Barcelona or in London**

A. SELECT \* FROM salseperson WHERE (city = "barcelona") OR (city = "london");

**4. All salespeople with commission between 0.10 and 0.12. (Boundary values**

**should be excluded).**

A.SELECT \* FROM salseperson WHERE comm BETWEEN ".10" AND ".12";

5. All customers excluding those with rating <= 100 unless they are located in

Rome

A. SELECT \* FROM customer WHERE (rating > 100);