1. **What do you understand by database?**

A database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system (DBMS)

1. **What is Normalization?**

**Normalization** is a database design technique that reduces data redundancy and eliminates undesirable characteristics like Insertion, Update and Deletion Anomalies. Normalization rules divides larger tables into smaller tables and links them using relationships. The purpose of Normalisation in SQL is to eliminate redundant (repetitive) data and ensure data is stored logically.

**Database Normal Forms**

Here is a list of Normal Forms in SQL:

* 1NF (First Normal Form)
* 2NF (Second Normal Form)
* 3NF (Third Normal Form)
* BCNF (Boyce-Codd Normal Form)
* 4NF (Fourth Normal Form)
* 5NF (Fifth Normal Form)
* 6NF (Sixth Normal Form)

1. **What is the difference between DBMS and RDBMS?**

|  |  |
| --- | --- |
| **RDBMS** | **DBMS** |
| Data stored is in table format | Data stored is in the file format |
| Multiple data elements are accessible together | Individual access of data elements |
| Data in the form of a table are linked together | No connection between data |
| Normalisation is not achievable | There is normalisation |
| Support distributed database | No support for distributed database |
| Data is stored in a large amount | Data stored is a small quantity |
| Here, redundancy of data is reduced with the help of key and indexes in RDBMS | Data redundancy is common |
| RDBMS supports multiple users | DBMS supports a single user |
| It features multiple layers of security while handling data | There is only low security while handling data |
| The software and hardware requirements are higher | The software and hardware requirements are low |
| Oracle, SQL Server. | XML, Microsoft Access. |

1. **What is EF codd rule of RDBMS?**

EF Codd was a computer scientist who invented RDMS system.

Codd proposed 13 rules popularly known as **Codd's 12 rules** to test DBMS's concept against his relational model. Codd's rule actualy define what quality a DBMS requires in order to become a Relational Database Management System(RDBMS)

**Rule 0 − Foundation rule**

Any relational database management system that is propounded to be RDBMS or advocated to be a RDBMS should be able to manage the stored data in its entirety through its relational capabilities.

**Rule 1 − Rule of Information**

Relational Databases should store the data in the form of relations. Tables are relations in Relational Database Management Systems. Be it any user defined data or meta-data, it is important to store the value as an entity in the table cells.

**Rule 2 − Rule of Guaranteed Access**

The use of pointers to access data logically is strictly forbidden. Every data entity which is atomic in nature should be accessed logically by using a right combination of the name of table, primary key represented by a specific row value and column name represented by attribute value.

**Rule 3 − Rule of Systematic Null Value Support**

Null values are completely supported in relational databases. They should be uniformly considered as ‘missing information’. Null values are independent of any data type. They should not be mistaken for blanks or zeroes or empty strings. Null values can also be interpreted as ‘inapplicable data’ or ‘unknown information.’

**Rule 4 − Rule of Active and online relational Catalog**

In the Database Management Systems lexicon, ‘metadata’ is the data about the database or the data about the data. The active online catalog that stores the metadata is called ‘Data dictionary’. The so called data dictionary is accessible only by authored users who have the required privileges and the query languages used for accessing the database should be used for accessing the data of data dictionary.

**Rule 5 − Rule of Comprehensive Data Sub-language**

A single robust language should be able to define integrity constraints, views, data manipulations, transactions and authorizations. If the database allows access to the aforementioned ones, it is violating this rule.

**Rule 6 − Rule of Updating Views**

Views should reflect the updates of their respective base tables and vice versa. A view is a logical table which shows restricted data. Views generally make the data readable but not modifiable. Views help in data abstraction.

**Rule 7 − Rule of Set level insertion, update and deletion**

A single operation should be sufficient to retrieve, insert, update and delete the data.

**Rule 8 − Rule of Physical Data Independence**

Batch and end user operations are logically separated from physical storage and respective access methods.

**Rule 9 − Rule of Logical Data Independence**

Batch and end users can change the database schema without having to recreate it or recreate the applications built upon it.

**Rule 10 − Rule of Integrity Independence**

Integrity constraints should be available and stored as metadata in data dictionary and not in the application programs.

**Rule 11 − Rule of Distribution Independence**

The Data Manipulation Language of the relational system should not be concerned about the physical data storage and no alterations should be required if the physical data is centralized or distributed.

**Rule 12 − Rule of Non-Subversion**

Any row should obey the security and integrity constraints imposed. No special privileges are applicable.

1. **What do you understand by data redundancy?**

**In DBMS, when the same data is stored in different tables, it causes data redundancy.**

**Sometimes, it is done on purpose for recovery or backup of data, faster access of data, or updating data easily. Redundant data costs extra money, demands higher storage capacity, and requires extra effort to keep all the files up to date.**

**Sometimes, unintentional duplicity of data causes a problem for the database to work properly, or it may become harder for the end user to access data. Redundant data unnecessarily occupy space in the database to save identical copies, which leads to space constraints, which is one of the major problems.**

1. **What is DDL Interpreter?**

**In DBMS, DDL Interpreter is a components of Query Processor**

**DDL Interpreter:**It processes the DDL statements into a set of table containing meta data (data about data).

1. **What is DML compiler in SQL?**

**DML Compiler is also a components of Query Processor**

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

1. **What is SQL Key Constraints? Write an Example of SQL Key Constraints**

In a database table, we can add rules to a column known as **constraints**. These rules control the data that can be stored in a column.

There are two types of key constraints in SQL DBMS.

**Primary Key:** he PRIMARY KEY constraint is simply a combination of NOT NULL and UNIQUE constraints. It means that the column value is used to uniquely identify the row. For example,

CREATE TABLE Colleges (

college\_id INT PRIMARY KEY,

college\_code VARCHAR (20) NOT NULL,

college\_name VARCHAR (50)

);

**Foreign Key:** The FOREIGN KEY (REFERENCES in some databases) constraint in a column is used to reference a record that exists in another table. For example,

CREATE TABLE Orders (

order\_id INT PRIMARY KEY,

customer\_id int REFERENCES Customers(id)

);

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

A SAVEPOINT is a point in a transaction when you can roll the transaction back to a certain point without rolling back the entire transaction.

The syntax for a SAVEPOINT command is as shown below.

SAVEPOINT Savepoint\_Name;

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

**Trigger:** A trigger is a stored procedure in database which automatically invokes whenever a special event in the database occurs. For example, a trigger can be invoked when a row is inserted into a specified table or when certain table columns are being updated.

Syntax:

create trigger [trigger\_name]

[before | after]

{insert | update | delete}

on [table\_name]

[for each row]

[trigger\_body]