

1) What do you understand By Database

==> A database is an electronically stored, systematic collection of data. It can contain any type of data, including words, numbers, images, videos, and files. You can use software called a database management system (DBMS) to store, retrieve, and edit data.

2) What is Normalization?

==> Normalization is the process of organizing the data in the database. Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate undesirable characteristics like Insertion, Update, and Deletion Anomalies. Normalization divides the larger table into smaller and links them using relationships. The normal form is used to reduce redundancy from the database table.

3) What is Difference between DBMS and RDBMS?

==> The significant points of difference between a DBMS and an RDBMS are as follows:

- DBMS stores the information in files, but RDBMS stores them in tables. Thus, there exists no relationship or link between the data in DBMS.
- DBMS can have repeated data, but RDBMS prevents it by indexing.
- DBMS does not allow normalisation, but RDBMS does.
- DBMS offers low security and speed. In contrast, RDBMS has a higher speed and is more secure.
- DBMSs require less software and hardware. In an RDBMS, hardware and software requirements are comparatively higher.
- DBMS allows only one user, while RDBMS allows multiple users.

- DBMS has less storage capacity.
- Usually, small organisations with fewer data and single users prefer DBMS, but RDBMS is preferable in other situations.
- DBMS handles navigational or hierarchical format, while RDBMS uses primary identifiers and tables for data handling.
- Common examples of DBMS are Windows registry, Microsoft Access, and XML. General examples of RDBMS are SQL, MySQL, Postgres, and Oracle.

4) What is MF Cod Rule of RDBMS Systems?

==> Here is brief note on E.F Codd's Twelve rules:

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.

Almost all full scale DBMSs are RDMSs. Oracle implements 11+ rules and so does Sybase. SQL Server also implements 11+ rules while FoxPro implements 7+ rules.

5) What do you understand By Data Redundancy?

==> 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?

==> It interprets the DDL (Data Definition Language) Instructions and stores the record in a data dictionary (in a table containing meta-data) Query Optimizer: It executes the

DML Instructions and picks the lowest cost evaluation plan out of all the alternatives present.

7) What is DML Compiler in SQL?

==> Data Manipulation Language or DML is a subset of operations used to insert, delete, and update data in a database. A DML is often a sublanguage of a more extensive language like SQL; DML comprises some of the operators in the language.

8) What is SQL Key Constraints writing an Example of SQL Key Constraints

==> The PRIMARY KEY constraint uniquely identifies each record in a table. Primary keys must contain UNIQUE values, and cannot contain NULL values. A table can have only ONE primary key; and in the table, this primary key can consist of single or multiple columns (fields).

The FOREIGN KEY constraint is used to prevent actions that would destroy links between tables. A FOREIGN KEY is a field (or collection of fields) in one table, that refers to the PRIMARY KEY in another table. The table with the foreign key is called the child table, and the table with the primary key is called the referenced or parent table.

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

==> SAVEPOINT creates points within the groups of transactions in which to ROLLBACK. A SAVEPOINT is a point in a transaction in which you can roll the transaction back to a certain point without rolling back the entire transaction.

Example;

```
SAVEPOINT SP1;
```

```
//Savepoint created.
```

```
DELETE FROM Student WHERE AGE = 20;
```

```
//deleted
```

```
SAVEPOINT SP2;
```

```
//Savepoint created.
```

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

==> A trigger is a stored procedure in a database that 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 specific table columns are updated. In simple words, a trigger is a collection of SQL statements with particular names that are stored in system memory. It belongs to a specific class of stored procedures that are automatically invoked in response to database server events. Every trigger has a table attached to it.

Example;

```
create trigger [trigger_name]
```

```
[before | after]
```

```
{insert | update | delete}
```

```
on [table_name]
```

```
[for each row]
```

```
[trigger_body]
```