Q:1-What do you understand By Database

ANS: - Database Management Systems (DBMS) are software systems used to store, retrieve, and run queries on data. A DBMS serves as an interface between an end-user and a database, allowing users to create, read, update, and delete data in the database.

DBMS manage the data, the database engine, and the database schema, allowing for data to be manipulated or extracted by users and other programs. This helps provide data security, data integrity, concurrency, and uniform data administration procedures.

Database management systems can be classified based on a variety of criteria such as the data model, the database distribution, or user numbers. The most widely used types of DBMS software are relational, distributed, hierarchical, object-oriented, and network.

Q-2-What is Normalization?

ANS- The basic objective of normalization is to reduce redundancy which means that information is to be stored only once. Storing information several times leads to wastage of storage space and increase in the total size of the data stored.

Relations are normalized so that when relations in a database are to be altered during the life time of the database, we do not lose information or introduce inconsistencies. The type of alterations normally needed for relations are: o Insertion of new data values to a relation. This should be possible without being forced to leave blank fields for some attributes.

Q-3-What is Difference between DBMS and RDBMS?

ANS-The Difference between DBMS AND RDBMS:

|  |  |
| --- | --- |
| RDBMS | DBMS |
| Data stored is in table format | Data stored is in the files 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 no Normalisation |
| Data is stored in large amount | Data is stored in small quantity |
| Here, redundancy of data is reduced with the help of indexes in RDBMS | Data redundancy is common |
| RDBMS support multiple users | DBMS support single user |
| It features multiple security while handling data | There is only low security while handling data |
| ORACLE, SQL SERVER | XML, MICROSOFT ACCESSS |

Q-4-What is MF Cod Rule of RDBMS Systems?

ANS-Dr Edgar F. Codd, after his extensive research on the Relational Model of database systems, came up with twelve rules of his own, which according to him, a database must obey in order to be regarded as a true relational database.

These rules can be applied on any database system that manages stored data using only its relational capabilities. This is a foundation rule, which acts as a base for all the other rules.

Rule 1: Information Rule

The data stored in a database, may it be user data or metadata, must be a value of some table cell. Everything in a database must be stored in a table format.

Rule 2: Guaranteed Access Rule

Every single data element (value) is guaranteed to be accessible logically with a combination of table-name, primary-key (row value), and attribute-name (column value). No other means, such as pointers, can be used to access data.

Rule 3: Systematic Treatment of NULL Values

The NULL values in a database must be given a systematic and uniform treatment. This is a very important rule because a NULL can be interpreted as one the following − data is missing, data is not known, or data is not applicable.

Rule 4: Active Online Catalog

The structure description of the entire database must be stored in an online catalog, known as data dictionary, which can be accessed by authorized users. Users can use the same query language to access the catalog which they use to access the database itself.

Rule 5: Comprehensive Data Sub-Language Rule

A database can only be accessed using a language having linear syntax that supports data definition, data manipulation, and transaction management operations. This language can be used directly or by means of some application. If the database allows access to data without any help of this language, then it is considered as a violation.

Rule 6: View Updating Rule

All the views of a database, which can theoretically be updated, must also be updatable by the system.

Rule 7: High-Level Insert, Update, and Delete Rule

A database must support high-level insertion, updating, and deletion. This must not be limited to a single row, that is, it must also support union, intersection and minus operations to yield sets of data records.

Rule 8: Physical Data Independence

The data stored in a database must be independent of the applications that access the database. Any change in the physical structure of a database must not have any impact on how the data is being accessed by external applications.

Rule 9: Logical Data Independence

The logical data in a database must be independent of its user’s view (application). Any change in logical data must not affect the applications using it. For example, if two tables are merged or one is split into two different tables, there should be no impact or change on the user application. This is one of the most difficult rule to apply.

Rule 10: Integrity Independence

A database must be independent of the application that uses it. All its integrity constraints can be independently modified without the need of any change in the application. This rule makes a database independent of the front-end application and its interface.

Rule 11: Distribution Independence

The end-user must not be able to see that the data is distributed over various locations. Users should always get the impression that the data is located at one site only. This rule has been regarded as the foundation of distributed database systems.

Rule 12: Non-Subversion Rule

If a system has an interface that provides access to low-level records, then the interface must not be able to subvert the system and bypass security and integrity constraints.

Q-5-What do you understand By Data Redundancy?

ANS-Data redundancy refers to the practice of keeping data in two or more places within a database or data storage system. Data redundancy ensures an organization can provide continued operations or services in the event something happens to its data -- for example, in the case of data corruption or data loss.

Q-6-What is DDL Interpreter?

ANS-DDL Interpreter: It processes the DDL statements into a set of tables containing meta data (data about data).

Q-7-What is DML Compiler in SQL?

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

Q-8-What is SQL Key Constraints writing an Example of SQL Key Constraints?

Ans-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.

The following constraints are commonly used in SQL:

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

EXAMPLE: -

|  |
| --- |
| CREATE TABLE NIRMAL  (ROLL \_NO INT **NOT NULL**, FIRST\_NAME VARCHAR(30) **NOT NULL**) |

* UNIQE - Ensures that all values in a column are different

|  |
| --- |
| CREATE TABLE NIRMAL  (ROLL \_NO INT UNIQUE, FIRST\_NAME VARCHAR(30) **NOT NULL**) |

* PRIMARY KEY - A combination of a NOT NULL and UNIQUE. Uniquely identifies each

row in a table

|  |
| --- |
| CREATE TABLE NIRMAL  (ROLL \_NO INT PRIMARY KEY, FIRST\_NAME VARCHAR(30) **NOT NULL**) |

* FOREIGN KEY - Prevents 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](https://www.w3schools.com/sql/sql_primarykey.asp) 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.

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

Ans- SAVE POINT: -

A save point is a point in a transaction when u can roll the transaction back at the certain point without rolling back the entire transaction.

The syntax for a saves point is SAVEPOINT SAVEPOINT\_NAME;

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

ANS- 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.

Syntax:

create trigger [trigger name] [before | after] {insert | update | delete} on [table name] [for each row] [trigger body]