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

→ It is a collection of interrelated data and management system is a set of programs to store and retrieves those 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.

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

| **DBMS** | **RDBMS** |
| --- | --- |
| DBMS stores data as a file. | RDBMS stores data in tabular form. |
| Data elements need to be accessed individually. | Multiple data elements can be accessed at the same time. |
| No relationship between data. | Data is stored in the form of tables which are related to each other. |
| Normalization is not present. | Normalization is present. |
| DBMS does not support distributed databases. | RDBMS supports distributed databases. |
| It stores data in either a navigational or hierarchical form. | It uses a tabular structure where the headers are the column names, and the rows contain corresponding values. |
| It deals with a small quantity of data. | It deals with large amounts of data. |
| Data redundancy is common in this model. | Keys and indexes do not allow Data redundancy. |
| It is used for small organizations and deals with small data. | It is used to handle large amounts of data. |
| It supports single users. | It supports multiple users. |
| Data fetching is slower for the large amount of data. | Data fetching is fast because of the relational approach. |
| The data in a DBMS is subject to low security levels with regards to data manipulation. | There exists multiple levels of data security in a RDBMS. |
| Low software and hardware necessities. | Higher software and hardware necessities. |
| Examples: XML, Window Registry, etc. | Examples: MySQL, PostgreSQL, SQL Server, Oracle, Microsoft Access etc. |

**4. What is the EF Cod Rule of RDBMS Systems?**

→   
**Rule zero**

This rule states that for a system to qualify as an RDBMS, it must be able to manage database entirely through the relational capabilities.

**Rule 1: Information rule**

### All information(including metadata) is to be represented as stored data in cells of tables. The rows and columns have to be strictly unordered.

### **Rule 2: Guaranted Access**

Each unique piece of data(atomic value) should be accesible by : Table Name + Primary Key(Row) + Attribute(column).

NOTE: Ability to directly access via POINTER is a violation of this rule.

### **Rule 3: Systematic treatment of NULL**

Null has several meanings, it can mean missing data, not applicable or no value. It should be handled consistently. Also, Primary key must not be null, ever. Expression on NULL must give null.

### **Rule 4: Active Online Catalog**

Database dictionary(catalog) is the structure description of the complete Database and it must be stored online. The Catalog must be governed by same rules as rest of the database. The same query language should be used on catalog as used to query database.

### **Rule 5: Powerful and Well-Structured Language**

One well structured language must be there to provide all manners of access to the data stored in the database. Example: SQL, etc. If the database allows access to the data without the use of this language, then that is a violation.

### **Rule 6: View Updation Rule**

All the view that are theoretically updatable should be updatable by the system as well.

### **Rule 7: Relational Level Operation**

There must be Insert, Delete, Update operations at each level of relations. Set operation like Union, Intersection and minus should also be supported.

### **Rule 8: Physical Data Independence**

The physical storage of data should not matter to the system. If say, some file supporting table is renamed or moved from one disk to another, it should not effect the application.

### **Rule 9: Logical Data Independence**

If there is change in the logical structure(table structures) of the database the user view of data should not change. Say, if a table is split into two tables, a new view should give result as the join of the two tables. This rule is most difficult to satisfy.

### **Rule 10: Integrity Independence**

The database should be able to enforce its own integrity rather than using other programs. Key and Check constraints, trigger etc, should be stored in Data Dictionary. This also make RDBMS independent of front-end.

### **Rule 11: Distribution Independence**

A database should work properly regardless of its distribution across a network. Even if a database is geographically distributed, with data stored in pieces, the end user should get an impression that it is stored at the same place. This lays the foundation of distributed database.

### **Rule 12: Nonsubversion Rule**

If low level access is allowed to a system it should not be able to subvert or bypass integrity rules to change the data. This can be achieved by some sort of looking or encryption.

**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?**

→ DDL is the short name of **Data Definition Language,** which deals with database schemas and descriptions of how the data should reside in the database.

● CREATE - to create a database and its objects like (table, index, views, store procedure, function, and triggers)

● ALTER - alters the structure of the existing database

● DROP - delete objects from the database

● TRUNCATE - remove all records from a table, including all spaces allocated for the records are removed

● RENAME - rename an object

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

→ DML is short name of **Data Manipulation Language** which deals with data manipulation and includes most common SQL statements such SELECT, INSERT, UPDATE, DELETE, etc., and it is used to store, modify, retrieve, delete and update data in a database.

● SELECT - retrieve data from a database

● INSERT - insert data into a table

● UPDATE - updates existing data within a table

● DELETE - Delete all records from a database table

● MERGE - UPSERT operation (insert or update)

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

→ A key is a set of attributes that can identify each tuple uniquely in the given relation.

**● Primary Key -** A primary key is a candidate key that the database designer selects while designing the database. Primary Keys are unique and NOT NULL.

**● Foreign Key -** An attribute ‘X’ is called as a foreign key to some other attribute ‘Y’ when its values are dependent on the values of attribute ‘Y’. The relation in which attribute ‘Y’ is present is called as the referenced relation. The relation in which attribute ‘X’ is present is called as the referencing relation.

**● Unique Key -** It is unique for all the records of the table. Once assigned, its value cannot be changed i.e. it is non-updatable. It may have a NULL value.

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

→ 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.

Syntax for Savepoint command:

SAVEPOINT SAVEPOINT\_NAME;

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

→ 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.

Ex.

delimiter $$   
 create trigger tr\_insert   
 after insert   
 on employee for each row   
 begin   
 insert into view(emp\_id,name,salary)

values (new.Employee\_id,new.First\_name,new.Salary);   
end   
$$ delimiter ;