

Que-1). What do you understand By Database?

---

Ans-1). A database is a structured collection of information or data that is stored in a computer. It allows you to easily access, manage, and update this data. Think of it like a digital filing system where data is organized in tables and can be quickly retrieved, modified, and analyzed as needed.

A database is a collection of organised or arranged data that can be easily accessed, updated, modified or controlled. Information within the database can be easily placed into rows and columns, or tables.

---

Que – 2). What is Normalization?

---

Ans – 2) Normalization is cleaning up your data. It is organizing data in a database to avoid repetition and keep it accurate. You put related information together and get rid of repeated stuff to make it easier to find and use.

---

### Que – 3). What is Difference between DBMS and RDBMS?

---

Ans – 3). Difference between DBMS and RDBMS :

DBMS	RDBMS
<ul style="list-style-type: none"><li>• Data stored is in the file format</li><li>• Individual access of data elements</li><li>• In this system no connection between data.</li><li>• No support for distributed database</li><li>• Data is stored in a small quantity</li><li>• DBMS supports a single user</li><li>• The software and hardware requirements are low</li><li>• Example: XML, Microsoft Access</li></ul>	<ul style="list-style-type: none"><li>• Data stored is in table format</li><li>• Multiple data elements are accessible together</li><li>• Data in the form of a table are linked together</li><li>• RDBMS supports distributed database</li><li>• Data is stored in a large amount</li><li>• RDBMS supports multiple users</li><li>• The software and hardware requirements are higher</li><li>• Example: Oracle, SQL Server</li></ul>

## Que – 4). What is MF Cod Rule of RDBMS Systems?

---

Ans – 4) The **MF Cod Rule** refers to a set of 12 rules defined by Dr. Edgar F. Codd, the inventor of the relational database model. These rules are used to determine whether a database management system (DBMS) qualifies as a relational DBMS (RDBMS).

Here's a simple breakdown:

1. **Rule 0:** The system must manage data entirely using its relational capabilities.
2. **Rule 1 (Information Rule):** All data should be stored in tables, and each piece of data should be easily accessible by specifying the table name, column, and key.
3. **Rule 2 (Guaranteed Access Rule):** Every piece of data should be accessible without needing to know how it's stored internally.
4. **Rule 3 (Systematic Treatment of NULL Values):** The database should handle missing data (NULL values) in a consistent way across the system.
5. **Rule 4 (Dynamic Online Catalog):** The system's structure (like tables and columns) should also be stored in tables, allowing users to query the database's own structure.
6. **Rule 5 (Comprehensive Data Sub-Language):** The system should support at least one language that can handle all data-related tasks: querying, updating, defining data structures, and managing permissions.
7. **Rule 6 (View Updating Rule):** If the database allows for different views (custom perspectives on the data), it should also allow updates to those views as if they were real tables.
8. **Rule 7 (High-Level Insert, Update, Delete):** The system should allow for bulk data operations—meaning you can insert, update, or delete data in groups, not just one row at a time.
9. **Rule 8 (Physical Data Independence):** Changing how the data is stored (like moving it to a different disk) should not affect how you interact with it at the logical level.

10. **Rule 9 (Logical Data Independence):** Making changes to the structure of the database (like adding columns) should not require you to change the applications that use it.
11. **Rule 10 (Integrity Independence):** Integrity constraints (rules that keep the data accurate) should be stored in the database itself, not in application code.
12. **Rule 11 (Distribution Independence):** The database should work the same way whether the data is stored all in one place or distributed across multiple locations.
13. **Rule 12 (Non-Subversion Rule):** If the system has any low-level access methods (like a backdoor), those methods should not be able to bypass the rules and constraints of the database.

These rules are theoretical ideals that most modern RDBMS systems aim to meet to ensure data integrity, flexibility, and consistency.

---

### **Que – 5). What do you understand By Data Redundancy?**

---

Ans – 5) Data redundancy occurs when the same piece of information is stored multiple times in different places. This can waste storage space, lead to inconsistencies, and make data management more complex. To avoid redundancy, it's often better to store data in a single, centralized location.

---

### **Que – 6). What is DDL Interpreter?**

---

Ans – 6) **DDL** stands for **Data Definition Language**. A DDL interpreter is like a builder for a database. It takes instructions written in DDL (like creating tables or adding columns) and turns them into actions, building the database structure according to your design.

### Que – 7). What is DML Compiler in SQL?

---

Ans – 7) DML stands for Data Manipulation Language. **A DML compiler is the translator in a database.** It takes your commands to add, change, or remove data (like INSERT, UPDATE, DELETE) and converts them into something the computer understands. Think of it as the go-between that lets you talk to the database in plain language.

---

### Que – 8). What is SQL Key Constraints writing an Example of SQL Key Constraints

---

Ans – 8) **SQL Key Constraints** are rules that ensure data integrity in a table.

- **PRIMARY KEY:** This uniquely identifies each row in a table. For example, a 'customer\_id' column could be the primary key.
  - **UNIQUE:** Ensures that all values in a column are different. For instance, a 'social\_security\_number' column should be unique.
  - **FOREIGN KEY:** Links data between two tables. Like a 'order\_id' in an 'order\_details' table referencing the 'order\_id' in an 'orders' table.
- 

### Que – 9). What is save Point? How to create a save Point write a Query?

---

Ans – 9) **Savepoint** is like a bookmark within a database transaction. It lets you undo changes up to that point without affecting the entire transaction.

To create a savepoint, use:

SQL

SAVEPOINT savepoint\_name;

Replace savepoint\_name with your chosen name.

For example:

SQL

```
SAVEPOINT my_bookmark;
```

This creates a savepoint named 'my\_bookmark'.

---

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

---

Ans – 10) 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 : - creating trigger after inserting the record.

```
DELIMITER $$
```

```
create TRIGGER tri_candidate AFTER INSERT on candidate for EACH ROW
```

```
BEGIN
```

```
    insert into test(id, name, action_performed)
```

```
    VALUES(new.id, new.cname, 'Record inserted');
```

```
end
```