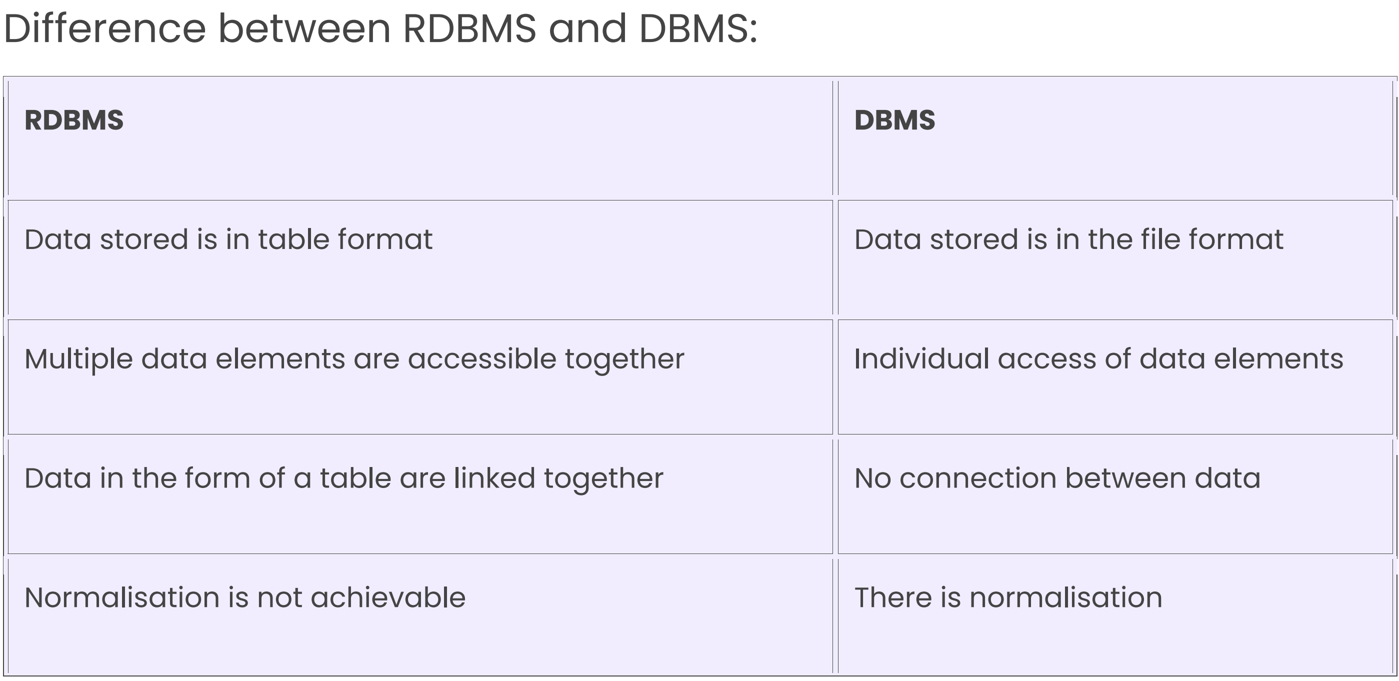
• What do you understand By Database:

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).

• What is Normalization?:

Normalization is a database design technique that reduces data redundancy and eliminates undesirable characteristics like Insertion, Update and Deletion Anomalies.

• What is Difference between DBMS and RDBMS?:



Difference between RDBMS and DBMS:

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.

• What do you understand By Data Redundancy?:

Data redundancy occurs when the same piece of data is stored in two or more separate places and is a common occurrence in many businesses.

As more companies are moving away from siloed data to using a central repository to store information, they are finding that their database is filled with inconsistent duplicates of the same entry.

Although it can be challenging to reconcile — or even benefit from — duplicate data entries, understanding how to reduce and track data redundancy efficiently can help mitigate long-term inconsistency issues for your business.

Data redundancy can be found in a database, which is an organized collection of structured data that's stored by a computer system or the cloud.

A retailer may have a database to track the products they stock. If the same product gets entered twice by mistake, data redundancy takes place.

• What is DDL Interpreter?:

Data Definition Language (DDL) is used to create and modify the structure of objects in a database using predefined commands and a specific syntax. These database objects include tables, sequences,
locations, aliases, schemas and indexes

Command Description:

CREATE - Creates a new table, a view of a table, or other object in database
ALTER - Modifies an existing database object, such as a table.
DROP - Deletes an entire table, a view of a table or other object in the database

• What is DML Compiler in SQL?:

DML stands for Data Manipulation Language. Tables and formulas are helpful when communicating with data stored up to a point in a database through SQL, but a time comes when we actually want to execute some fairly complicated data interactions. We will also need the Data Manipulation Language in that situation.

DML is a way to inform a database precisely what we want it to do by conversing in a manner that it has been built to comprehend from the scratch. When it comes to interacting within existing data, whether adding, moving, or deleting data, it provides a convenient way to do so.

Data manipulation includes introducing data into tables, altering the table's data and deleting the data from the table.

• What is SQL Key Constraints writing an Example of SQL Key Constraints:

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.

These constraints are used in SQL:

- NOT NULL - Ensures that a column cannot have a NULL value
- UNIQUE - Ensures that all values in a column are different
- PRIMARY KEY - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table
- FOREIGN KEY - Prevents actions that would destroy links between tables
- CHECK - Ensures that the values in a column satisfies a specific condition
- DEFAULT - Sets a default value for a column if no value is specified
-
- **CREATE TABLE Student**
- **(**
- **ID int(6) NOT NULL UNIQUE,**
- **NAME varchar(10),**
- **ADDRESS varchar(20),**
- **PRIMARY KEY(ID)**
- **);**

• 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;

This command serves only in the creation of a SAVEPOINT among all the transactional statements. The ROLLBACK command is used to undo a group of transactions.

The syntax for rolling back to a SAVEPOINT is as shown below.

ROLLBACK TO SAVEPOINT_NAME;

*Query:-

```
 START TRANSACTION;

SAVEPOINT ini;

INSERT INTO student VALUES (10, "Komal
patel", 54, "Ahemdabad", "1995-01-06");

 ROLLBACK TO ini ;
```

• What is trigger and how to create a Trigger in SQL?:

Trigger is a statement that a system executes automatically when there is any modification to the database. In a trigger, we first specify when the trigger is to be executed and then the action to be performed when the trigger executes. Triggers are used to specify certain integrity constraints and referential constraints that cannot be specified using the constraint mechanism of SQL.

Create trigger in SQL

```
DELIMITER //
CREATE TRIGGER Check_age
BEFORE INSERT ON employee
FOR EACH ROW
BEGIN
IF NEW.age < 35 THEN
SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = 'ERROR:
AGE MUST BE ATLEAST 25 YEARS!';
END IF;
END; //
```