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

- Database is the collection of data to access,manage,organise electricaly from in the file systems known as the database
- A database is usually controlled by a database management system (DBMS). Together, the data and the DBMS, along with the applications that are associated with them, are referred to as a database system, often shortened to just database.
- The data can then be easily accessed, managed, modified, updated, controlled, and organized. Most databases use structured query language (SQL) for writing and querying data.
- excel,msaccss,Oracle,Mysql,Sql-servers,monogodb,mariadb,sqlite,oformix

2) **What is Normalization?**

- Normalization is the process of minimizing redundancy (duplicity) from a relation or set of relations. Redundancy in relation may cause insertion, deletion and updation anomalies. So, it helps to minimize the redundancy in relations.

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

- The main difference between DBMS and RDBMS is that the DBMS is a software that helps to create and manage databases while RDBMS is a type of DBMS that is based on the relational model.

- DBMS is a software to manage databases that was introduced during the 1960s. It can handle a collection of data and can perform various functionalities such as creating, revising and controlling databases. DBMS is mainly useful for small organizations and individuals. On the other hand, RDBMS is an advanced type of DBMS introduced during the 1970s. It is based on the relational model. The tables in the database in RDBMS are connected to each other. In brief, RDBMS allows to store, organize and manage data more efficiently than a DBMS.

4) **What is MF Cod Rule of RDBMS Systems?**

- **Foundation Rule**： A relational database management system must manage its stored data using only its relational capabilities.

- **Information Rule**： All information in the database should be represented in one and only one way as values in a table.

- **Guaranteed Access Rule**： Each and every datum (atomic value) is guaranteed to be logically accessible by resorting to a combination of table name, primary key value and column name.

- **Systematic Treatment of Null Values**： Null values (distinct from empty character string or a string of blank characters and distinct from zero or any other number) are supported in the fully relational DBMS for representing missing information in a systematic way, independent of data type.

- **Dynamic On-line Catalog Based on the Relational Model**： The database description is represented at the logical level in the same way as ordinary data, so authorized users can apply the same relational language to its interrogation as they apply to regular data.

- **Comprehensive Data Sub-language Rule**： A relational system may support several languages and various modes of terminal use. However, there must be at least one language whose statements are expressible, per some well-defined syntax, as character strings and whose ability to support all of the following is comprehensible.

- **View Updating Rule**： All views that are theoretically update-able are also update-able by the system.

- **High-level Insert, Update and Delete**： The capability of handling a base relation or a derived relation as a single operand applies not only to the retrieval of data but also to the insertion, updation, and deletion of data.

- **Physical Data Independence**： Application programs and terminal activities remain logically unimpaired whenever any changes are made in either storage representation or access methods.

- **Logical Data Independence**：. Application programs and terminal activities remain logically unimpaired when information preserving changes of any kind that theoretically permit un-impairment are made to the base tables.

- **Integrity Independence**： Integrity constraints specific to a particular relational database must be definable in the relational data sub-language and storable in the catalog, not in the application programs.

- **Distribution Independence**： The data manipulation sub-language of a relational DBMS must enable application programs and terminal activities to remain logically unimpaired whether and whenever data are physically centralized or distributed.

- **Non-subversion Rule**： If a relational system has or supports a low-level (single -record -at -a -time) language, that low-level language cannot be used to subvert or bypass the integrity rules or constraints expressed in the higher -level (multiple -records -at -a -time) relational language

5) **What do you understand By Data Redundancy?**

- 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. The concept applies to areas such as databases, computer memory and file storage systems.

- Data redundancy can occur within an organization intentionally or accidentally. If done intentionally, the same data is kept in different locations with the organization making a conscious effort to protect it and ensure its consistency. This data is often used for backups or disaster recovery.

- If carried out by accident, duplicate data may cause data inconsistencies. Even though data redundancy can help minimize the chance of data loss, redundancy issues can affect larger data sets. For example, data that is stored in several places takes up valuable storage space and makes it difficult for the organization to identify which data they should access or update.

6) **What is DDL Interpreter?**

- Data definition language. DDL statements are used **to build and modify the structure of your tables and other objects in the database.** When you execute a DDL statement, it takes effect immediately.

- **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.

- DDL statements are similar to a computer programming language for defining data structures, especially database schemas. Common examples of DDL statements include **CREATE, ALTER, and DROP.**

- Data definition language (DDL) **describes the portion of SQL that creates, alters, and deletes database objects**. These database objects include schemas, tables, views, sequences, catalogs, indexes, variables, masks, permissions, and aliases. Creating a schema.

7) What is DML Compiler in SQL?

- DML stands for Data Manipulation Language. DML compiler **translates the DML statements which are there in a query language into the low-level instructions which the query evaluation engine understands easily.**

8) 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.

- Constraints can be column level or table level. Column level constraints apply to a column, and table level constraints apply to the whole table.

The following constraints are commonly 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 INDEX` - Used to create and retrieve data from the database very quickly

```
CREATE TABLE users(

id int PRIMARY KEY AUTO_INCREMENT,
 name varchar(50),
 email varchar(50) UNIQUE,
 salary int NOT NULL,
 gender varchar(50) DEFAULT 'MALE',
 age int CHECK(age<18)
);
```

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

- 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;**

- This command is used only in the creation of SAVEPOINT among all the transactions.

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

- Trigger is the same as the procedure
- A trigger is a special type of stored procedure that automatically runs when an event occurs in the database server.
-
    - **Type Trigger**

    1) After Insert
    2) After Update
    3) After Delete
    4) Before Insert
    5) Before Update
    6) Before Delete

    - **Create a trigger in sql**

```
create trigger [trigger_name]
[before | after]
{insert | update | delete}
on [table_name]
[for each row]
[trigger_body]
```