Q #1) What is DBMS used for?

Answer: DBMS, commonly known as Database Management System, is an application system whose main purpose revolves around the **data**. This is a system that allows its user to store the data, define it, retrieve it and update the information about the data inside the database.

Q #2) What is meant by a Database?

Answer: In simple terms, Database is a collection of data in some organized way to facilitate its user's to easily access, manage and upload the data.

Q #3) Why is the use of DBMS recommended? Explain by listing some of its major advantages.

Answer: Some of the major advantages of DBMS are as follows:

- **Controlled Redundancy:** DBMS supports a mechanism to control the redundancy of data inside the database by integrating all the data into a single database and as data is stored at only one place, the duplicity of data does not happen.
- **Data Sharing:** Sharing of data among multiple users simultaneously can also be done in DBMS as the same database will be shared among all the users and by different application programs.
- **Backup and Recovery Facility:** DBMS minimizes the pain of creating the backup of data again and again by providing a feature of 'backup and recovery' which automatically creates the data backup and restores the data whenever required.
- **Enforcement of Integrity Constraints:** Integrity Constraints are very important to be enforced on the data so that the refined data after putting some constraints are stored in the database and this is followed by DBMS.
- **Independence of data:** It simply means that you can change the structure of the data without affecting the structure of any of the application programs.

Q #4) What is the purpose of normalization in DBMS?

Answer: Normalization is the process of analyzing the relational schemas which are based on their respective functional dependencies and the primary keys in order to fulfill certain properties.

The properties include:

- To minimize the redundancy of the data.
- To minimize the Insert, Delete and Update Anomalies.

Q #5) What are the different types of languages that are available in the DBMS?

Answer: Basically, there are 3 types of languages in the DBMS as mentioned below:

- **DDL:** DDL is **Data Definition Language** which is used to define the database and schema structure by using some set of SQL Queries like **CREATE**, **ALTER**, **TRUNCATE**, **DROP** and **RENAME.**
- **DCL:** DCL is **Data Control Language** which is used to control the access of the users inside the database by using some set of SQL Queries like **GRANT** and **REVOKE.**
- **DML:** DML is **Data Manipulation Language** which is used to do some manipulations in the database like Insertion, Deletion, etc. by using some set of SQL Queries like **SELECT**, **INSERT**, **DELETE** and **UPDATE.**

Q #6) What is the purpose of SQL?

Answer: SQL stands for **Structured Query Language** whose main purpose is to interact with the relational databases in the form of inserting and updating/modifying the data in the database.

Q #7) Explain the concepts of a Primary key and Foreign Key.

Answer: Primary Key is used to uniquely identify the records in a database table while **Foreign Key** is mainly used to link two or more tables together, as this is a particular field(s) in one of the database tables which are the primary key of some other table.

Example: There are 2 tables – Employee and Department. Both have one common field/column as '**ID'** where ID is the primary key of the **Employee** table while this is the foreign key for the **Department** table.

Q #8) What are the main differences between Primary key and Unique Key?

Answer: Given below are few differences:

- The main difference between the Primary key and Unique key is that the Primary key can never have a null value while the Unique key may consist of null value.
- In each table, there can be only one primary key while there can be more than one unique key in a table.

Q #9) What is the concept of sub-query in terms of SQL?

Answer: Sub-query is basically the query which is included inside some other query and can also be called as an inner query which is found inside the outer query.

Q #10) What is the use of DROP command and what are the differences between DROP, TRUNCATE and DELETE commands?

Answer: DROP command is a DDL command which is used to drop/delete the existing table, database, index or view from the database.

The major difference between DROP, TRUNCATE and DELETE commands are:

DROP and **TRUNCATE** commands are the **DDL** commands which are used to delete tables from the database and once the table gets deleted, all the privileges and indexes that are related to the table also get deleted. These 2 operations cannot be rolled back and so should be used only when necessary.

DELETE command, on the other hand, is a **DML** Command which is also used to delete rows from the table and this can be rolled back.

Note: It is recommended to use the 'WHERE' clause along with the DELETE command else the complete table will get deleted from the database.

Q #11) What is the main difference between UNION and UNION ALL?

Answer: UNION and UNION ALL are used to join the data from 2 or more tables but UNION removes duplicate rows and picks the rows which are distinct after combining the data from the tables whereas UNION ALL does not remove the duplicate rows, it just picks all the data from the tables.

Q #12) Explain the concept of ACID properties in DBMS?

Answer: ACID properties is the combination of Atomicity, Consistency, Isolation, and Durability properties. These properties are very helpful in allowing a safe and secure way of sharing the data among multiple users.

- **Atomicity**: This is based on the concept of "either all or nothing" which basically means that if any update occurs inside the database then that update should either be available to all the others beyond user and application program or it should not be available to anyone beyond the user and application program.
- **Consistency**: This ensures that the consistency is maintained in the database before or after any transaction that takes place inside the database.
- **Isolation**: As the name itself suggests, this property states that each transaction that occurs is in isolation with others i.e. a transaction which has started but not yet completed should be in isolation with others so that the other transaction does not get impacted with this transaction.
- **Durability**: This property states that the data should always be in a durable state i.e. any data which is in the committed state should be available in the same state even if any failure or restart occurs in the system.

Q #13) What is Correlated Subquery in DBMS?

Answer: A Subquery is also known as a nested query i.e. a query written inside some query. When a Subquery is executed for each of the rows of the outer query then it is termed as a Correlated Subquery.

Example of Non-Correlated Subquery is:

```
SELECT * from EMP WHERE 'RIYA' IN (SELECT Name from DEPT WHERE
EMP.EMPID=DEPT.EMPID);
```
Here, the inner query is not executed for each of the rows of the outer query.

Q #14) Explain Entity, Entity Type, and Entity Set in DBMS?

Answer:

Entity is an object, place or thing which has its independent existence in the real world and about which data can be stored in a database. **For Example,** any person, book, etc.

Entity Type is a collection of entities that have the same attributes. **For Example,** the STUDENT table contains rows in which each row is an entity holding the attributes like name, age, and id of the students, hence STUDENT is an Entity Type which holds the entities having the same attributes.

Entity Set is a collection of entities of the same type. **For Example,** A collection of the employees of a firm.

Q #15) What are the different levels of abstraction in the DBMS?

Answer: There are 3 levels of data abstraction in the DBMS.

They include:

- **Physical Level:** This is the lowest level of the data abstraction which states how the data is stored in the database.
- **Logical Level:** This is the next level of the data abstraction which states the type of the data and the relationship among the data that is stored in the database.

- **View Level:** This is the highest level in the data abstraction which shows/states only a part of the database.

Q #16) What integrity rules exist in the DBMS?

Answer: There are 2 major integrity rules that exist in the DBMS.

They are:

- **Entity Integrity:** This states a very important rule that value of a Primary key can never have a NULL value.
- **Referential Integrity:** This rule is related to the Foreign key which states that either the value of a Foreign key is a NULL value or it should be the primary key of any other relation.

Q #17) What is E-R model in the DBMS?

Answer: E-R model is known as an **Entity-Relationship model** in the DBMS which is based on the concept of the Entities and the relationship that exists among these entities.

Q #18) What is a functional dependency in the DBMS?

Answer: This is basically a constraint which is useful in describing the relationship among the different attributes in a relation.

Example: If there is some relation 'R1' which has 2 attributes as Y and Z then the functional dependency among these 2 attributes can be shown as **Y->Z** which states that Z is functionally dependent on Y.

Q #19) What is 1NF in the DBMS?

Answer: 1NF is known as the **First Normal Form**.

This is the easiest form of the normalization process which states that the **d**omain of an attribute should have only atomic values. The objective of this is to remove the duplicate columns that are present in the table.

Q #20) What is 2NF in the DBMS?

Answer: 2NF is the **Second Normal Form**.

Any table is said to have in the 2NF if it satisfies the following 2 conditions:

- A table is in the 1NF.
- Each non-prime attribute of a table is said to be functionally dependent in totality on the primary key.

Q #21) What is 3NF in the DBMS?

Answer: 3NF is the **Third Normal Form.**

Any table is said to have in the 3NF if it satisfies the following 2 conditions:

- A table is in the 2NF.
- Each non-prime attribute of a table is said to be non-transitively dependent on every key of the table.

Q #22) What is BCNF in the DBMS?

Answer: BCNF is the **Boyce Codd Normal Form** which is stricter than the 3NF.

Any table is said to have in the BCNF if it satisfies the following 2 conditions:

- A table is in the 3NF.
- For each of the functional dependency X->Y that exists, X is the super key of a table.

Q #23) What is a CLAUSE in terms of SQL?

Answer: This is used with the SQL queries to fetch specific data as per the requirements on the basis of the conditions that are put in the SQL. This is very helpful in picking the selective records from the complete set of the records.

For Example, There is a query which has WHERE condition or the query with the HAVING clause.

Q #24) How can you get the alternate records from the table in the SQL?

Answer: If you want to fetch the odd numbers then the following query can be used:

```
SELECT EmpId from (SELECT rowno,EmpId from Emp) WHERE mod(rowno,2)=1;
```
If you want to fetch the even numbers, then the following query can be used:

```
SELECT EmpId from (SELECT rowno,EmpId from Emp) WHERE mod(rowno,2)=0;
```
Q #25) How is the pattern matching done in the SQL?

Answer: With the help of the LIKE operator, pattern matching is possible in the SQL.'**%**' is used with the LIKE operator when it matches with the 0 or more characters and **'_'** is used to match the one particular character.

Example:

```
SELECT * from Emp WHERE name like 'b%';
```
```
SELECT * from Emp WHERE name like 'hans_';
```
Q #26) What is a join in the SQL?

Answer: A Join is one of the SQL statements which is used to join the data or the rows from 2 or more tables on the basis of a common field/column among them.

Q #27) What are different types of joins in SQL?

Answer: There are 4 types of SQL Joins:

- **Inner Join:** This type of join is used to fetch the data among the tables which are common in both the tables.
- **Left Join:** This returns all the rows from the table which is on the left side of the join but only the matching rows from the table which is on the right side of the join.
- **Right Join:** This returns all the rows from the table which is on the right side of the join but only the matching rows from the table which is on the left side of the join.
- **Full Join:** This returns the rows from all the tables on which the join condition has put and the rows which do not match hold null values.

Q #28) What is meant by trigger?

Answer: Trigger is one of the very important codes or programs which get executed **automatically in response to the events** that occur in a table or a view. **For Example,** If a new record is inserted in an

employee database then the data gets created automatically in the related tables like salary, department and roles tables.

Q #29) Explain the Stored Procedure.

Answer: Stored Procedure is a group of SQL statements in the form of a function that has some unique name and is stored in relational database management systems(RDBMS) and can be accessed whenever required.

Q #30) What is RDBMS?

Answer: RDBMS is the Relational Database Management System which contains data in the form of the tables and data is accessed on the basis of the common fields among the tables.

Q #31) What are the different type of relationships in the DBMS?

Answer: Relationships in DBMS depicts an association between the tables.

Different types of relationships are:

- **One-to-One:** This basically states that there should be a one-to-one relationship between the tables i.e. there should be one record in both the tables. **Example:** Among a married couple, both wife and husband can have only one spouse.
- **One-to-Many:** This states that there can be many relationships for one i.e. a primary key table hold only one record which can have many, one or none records in the related table. **Example:** A Mother can have many children.
- **Many-to-Many:** This states that both the tables can be related to many other tables. **Example:** One can have many siblings and so do they have.