Q1. What do you understand by intension and extension?

Intension: Intension or most commonly known as Database schema defines the description of the database. This is specified during the database design and mostly remains unchanged.

Extension: Extension is the number of tuples available in the database at any instance of time. This value keeps changing as and when the tuples are created, updated and destroyed. So, the data present in the database at a specific instance of time is known as the extension of the database or most commonly known as the snapshot of the database.

Q2. What do you understand by cursor? Mention the different types of cursor

A cursor is a database object which helps in manipulating data, row by row and represents a result set.

The types of cursor are as follows:

Implicit cursor: This type of cursor is declared automatically as soon as the execution of SQL takes place. Here, the user is not indicated about the declaration of the cursor.

Explicit cursor: This type of cursor is defined by the PL/ SQL, as it handles a query in more than a single row.

Q3. Explain the terms specialization and generalization

Specialization: Specialization is a process of defining a set of subclasses of the entity type. Here, each subclass will contain all the attributes and relationships of the parent entity. Apart from this, the subclasses may contain additional attributes and relationships specific to itself.

Generalization: Generalization is a process of finding relations, common attributes for a particular set of entities; and finally defining a common superclass for them.

Q4. What do you understand by Data Independence?

When you say an application has data independence, it implies that the application is independent of the storage structure and data access strategies of data.

Q5. What are the different integrity rules present in the DBMS?

The different integrity rules present in DBMS are as follows:

- **Entity Integrity:** This rule states that the value of the primary key can never be NULL. So, all the tuples in the column identified as the primary key should have a value.

- **Referential Integrity:** This rule states that either the value of the foreign key is NULL or it should be the primary key of any other relation.

Q6. What does Fill Factor concept mean with respect to indexes?

Fill Factor is used to mention the percentage of space left on every leaf-level page, which is packed with data. Usually, the default value is 100.

Q7. What is Index hunting and how does it help in improving query performance?

The process of boosting a collection of indexes is known as Index hunting. This is done as indexes improve the query performance and the speed at which they are processed.

It helps in improving query performance in the following way:

- The best queries are suggested using the query optimizer.
- Index, query distribution and their performance are used as metrics to check the effect
- Databases are tuned into a small collection of problem queries.

Q8. What are the differences between network and hierarchical database model?

Network Database Model	Hierarchical Database Model
Each parent node can have multiple children nodes and vice versa.	A top-down structure where each parent node can have many child nodes. But, a child node can have only a single parent node.
Supports one-to-one, one-to-many, and many-to-many relationships	Supports one-tone and one-to-many relationships

Q9. Explain what is a deadlock and mention how it can be resolved?

Deadlock is a situation which occurs when two transactions wait on a resource which is locked or other transaction holds. Deadlocks can be prevented by making all the transactions acquire all the locks at the same instance of time. So, once deadlock occurs, the only way to cure is to abort one of the transactions and remove the partially completed work.

Q10. What are the differences between an exclusive lock and a shared lock?

Exclusive Lock	Shared Lock
An exclusive lock is a lock on a data item when a transaction is about to perform the write operation.	A shared lock allows more than one transaction to read the data items.

Q11. What are the differences between DROP, TRUNCATE and DELETE commands?

DROP	TRUNCATE	DELETE
Used to delete a database, table or a view	Used to delete all rows from a table	Used to delete a row in the table
Data cannot be rollbacked	Data cannot be rollbacked	Data can be rollbacked
A DDL command	A DDL command	A DML command.
Slower than TRUNCATE	Faster than DROP and DELETE	Slower than TRUNCATE
Deletes the full structure of the table	Preserves the structure of the table	Deletes the structure of the row from a table

Q12. What is SQL and where is it used?

SQL aka Structured Query Language is the core of the relational database which is used for accessing and managing the databases. This language is used to manipulate and retrieve data from a structured data format in the form of tables and holds relationships between those tables. So, in layman terms, you can use SQL to communicate with the database.

Q13. What do you understand by sub-queries in SQL?

A subquery is a query inside another query where a query is defined to retrieve data or information back from the database. In a subquery, the outer query is called as the main query whereas the inner query is called subquery. Subqueries are always executed first and the result of the subquery is passed on to the main query. It can be nested inside a SELECT, UPDATE or any other query. A subquery can also use any comparison operators such as >,< or =.

Q14. Mention the differences between UNION and UNION ALL

UNION	UNION ALL
Combines the result of two or more SELECT statements consisting of distinct values	Combines the result set of two or more SELECT statements consisting of duplicate values
Syntax: UNION	Syntax: UNION ALL
Has low performance than UNION ALL, as duplicate rows need to be removed.	Has better performance than UNION, as duplicate rows need not have to be removed.

Q15. What do you understand by CLAUSE in SQL?

CLAUSE in SQL is used to limit the result set by mentioning a condition to the query. So, you can use a CLAUSE to filter rows from the entire set of records.

Example: WHERE HAVING clause.

Q16. Mention the differences between HAVING and WHERE clause?

HAVING	WHERE
Used only with SELECT statement	Used in a GROUP BY clause
Used with the GROUP BY function in a query	Applied to each row before they are a part of the GROUP BY function in a query

Q17. How can you perform pattern matching in SQL?

You can perform pattern matching in SQL by using the LIKE operator. With the LIKE operator, you can use the following symbols:

1. %(Percentage sign) – To match zero or more characters.
2. _ (Underscore) –To match exactly one character.

Example:

SELECT * FROM Customers WHERE CustomerName LIKE 's%'

SELECT * FROM Customers WHERE CustomerName like 'xyz_'

Q18. Mention few case manipulation functions in SQL

There are three case manipulation functions in SQL, namely:

- **LOWER: This function returns the string in lowercase. It takes a string as an argument and returns it by converting it into lower case.**

  Syntax: LOWER('string')

- **UPPER: This function returns the string in uppercase. It takes a string as an argument and returns it by converting it into uppercase.**

  Syntax: UPPER('string')

- **INITCAP: This function returns the string with the first letter in uppercase and the rest of the letters in lowercase.**

  Syntax: INITCAP('string')

Q19. What are joins in SQL and what are the different types of joins?

A JOIN clause is used to combine rows from two or more tables, based on a related column between them. It is used to merge two tables or retrieve data from there. There are 4 joins in SQL namely:

- **Inner Join**
- **Right Join**
- **Left Join**
- **Full Join**

Q20. What do you understand by the view and mention the steps to create, update and drop a view?

A view in SQL is a single table, which is derived from other tables. So, a view contains rows and columns similar to a real table and has fields from one or more table.

To create a view, use the following syntax:

```
1    CREATE VIEW ViewName AS
2    SELECT Column1, Column2, ..., ColumnN
3    FROM TableName
4    WHERE Condition;
```

To update a view, use the following syntax:

```
1    CREATE VIEW OR REPLACE ViewName AS
2    SELECT Column1, Column2, ..., ColumnN
3    FROM TableName
4    WHERE Condition;
```

To drop a view, use the following syntax:

```
1.   DROP VIEW ViewName;
```