1. Difference between case expression and case statement
2. **What is Pattern Matching in SQL?**
3. **How to create empty tables with the same structure as another table?**
4. **What is a unique constraint?**
5. **What is a query?**
6. **What is data integrity?**
7. **What is a cross join?**
8. **What is a self join?**
9. **What is join? List its different types.**
10. **What is a Foreign key?**
11. **What is a subquery? And what are its types?**
12. **What is a primary key?**
13. **What are constraints in SQL?**
14. **What are tables and fields?**
15. **Difference between SQL and MySQL**
16. **What is Select statement?**
17. **Difference between DBMS and RDBMS**
18. **What are some common clauses used with SELECT query in SQL?**
19. **Difference between union, minus and intersect**
20. **What is an alias in SQL**
21. **What is a View?**
22. **Difference between truncate, delete and drop?**
23. **Difference between delete and truncate?**
24. **What are aggregate and scalar functions**

Ques3.

**SELECT** \* **INTO** Students\_copy

**FROM** Students **WHERE** 1 = 2;

Ques4.

A UNIQUE constraint ensures that all values in a column are different. This provides uniqueness for the column(s) and helps identify each row uniquely. Unlike primary key, there can be multiple unique constraints defined per table. The code syntax for UNIQUE is quite similar to that of PRIMARY KEY and can be used interchangeably.

**CREATE** **TABLE** Students ( /\* Create table with a single field as unique \*/

ID INT **NOT** **NULL** **UNIQUE**

Name VARCHAR(255)

);

**CREATE** **TABLE** Students ( /\* Create table with multiple fields as unique \*/

ID INT **NOT** **NULL**

LastName VARCHAR(255)

FirstName VARCHAR(255) **NOT** **NULL**

**CONSTRAINT** PK\_Student

**UNIQUE** (ID, FirstName)

);

**ALTER** **TABLE** Students /\* Set a column as unique \*/

**ADD** **UNIQUE** (ID);

**ALTER** **TABLE** Students /\* Set multiple columns as unique \*/

**ADD** **CONSTRAINT** PK\_Student /\* Naming a unique constraint \*/

**UNIQUE** (ID, FirstName)

Ques5.

A query is a request for data or information from a database table or combination of tables. A database query can be either a select query or an action query.

**SELECT** fname, lname /\* select query \*/

**FROM** myDb.students

**WHERE** student\_id = 1;

UPDATE myDB.students /\* action query \*/

**SET** fname = 'Captain', lname = 'America'

**WHERE** student\_id = 1;

Ques7.

Cross join can be defined as a cartesian product of the two tables included in the join. The table after join contains the same number of rows as in the cross-product of the number of rows in the two tables. If a WHERE clause is used in cross join then the query will work like an INNER JOIN.

**SELECT** stu.name, sub.subject

**FROM** students **AS** stu

**CROSS** **JOIN** subjects **AS** sub

Ques8.

A**self JOIN** is a case of regular join where a table is joined to itself based on some relation between its own column(s). Self-join uses the INNER JOIN or LEFT JOIN clause and a table alias is used to assign different names to the table within the query.

**SELECT** A.emp\_id **AS** "Emp\_ID",A.emp\_name **AS** "Employee",

B.emp\_id **AS** "Sup\_ID",B.emp\_name **AS** "Supervisor"

**FROM** employee A, employee B

**WHERE** A.emp\_sup = B.emp\_id;

**Referential integrity** is a property of data stating that all its references are valid. In the context of [relational databases](https://en.wikipedia.org/wiki/Relational_database), it requires that if a value of one attribute (column) of a [relation](https://en.wikipedia.org/wiki/Relation_(database)) (table) references a value of another attribute (either in the same or a different relation), then the referenced value must exist.

Ques13.

Constraints are used to specify the rules concerning data in the table. It can be applied for single or multiple fields in an SQL table during the creation of the table or after creating using the ALTER TABLE command. The constraints are:

* **NOT NULL** - Restricts NULL value from being inserted into a column.
* **CHECK** - Verifies that all values in a field satisfy a condition.
* **DEFAULT** - Automatically assigns a default value if no value has been specified for the field.
* **UNIQUE** - Ensures unique values to be inserted into the field.
* **INDEX** - Indexes a field providing faster retrieval of records.
* **PRIMARY KEY** - Uniquely identifies each record in a table.
* **FOREIGN KEY** - Ensures referential integrity for a record in another table

Ques15.

SQL is a standard language for retrieving and manipulating structured databases. On the contrary, MySQL is a relational database management system, like SQL Server, Oracle or IBM DB2, that is used to manage SQL databases.

Ques19.

The **UNION** operator combines and returns the result-set retrieved by two or more SELECT statements.  
The **MINUS** operator in SQL is used to remove duplicates from the result-set obtained by the second SELECT query from the result-set obtained by the first SELECT query and then return the filtered results from the first.  
The **INTERSECT** clause in SQL combines the result-set fetched by the two SELECT statements where records from one match the other and then returns this intersection of result-sets.

Certain conditions need to be met before executing either of the above statements in SQL -

* Each SELECT statement within the clause must have the same number of columns
* The columns must also have similar data types
* The columns in each SELECT statement should necessarily have the same order

**SELECT** name **FROM** Students /\* Fetch the union of queries \*/

**UNION**

**SELECT** name **FROM** Contacts;

**SELECT** name **FROM** Students /\* Fetch the union of queries with duplicates\*/

**UNION** **ALL**

**SELECT** name **FROM** Contacts;

**SELECT** name **FROM** Students /\* Fetch names from students \*/

MINUS /\* that aren't present in contacts \*/

**SELECT** name **FROM** Contacts;

**SELECT** name **FROM** Students /\* Fetch names from students \*/

**INTERSECT** /\* that are present in contacts as well \*/

**SELECT** name **FROM** Contacts;

Ques21.

A view in SQL is a virtual table based on the result-set of an SQL statement. A view contains rows and columns, just like a real table. The fields in a view are fields from one or more real tables in the database.

Ques22.

**DELETE** statement is used to delete rows from a table.

**DELETE** **FROM** Candidates

**WHERE** CandidateId > 1000;

**TRUNCATE** command is used to delete all the rows from the table and free the space containing the table.

**TRUNCATE** **TABLE** Candidates;

**DROP** command is used to remove an object from the database. If you drop a table, all the rows in the table are deleted and the table structure is removed from the database.

**DROP** **TABLE** Candidates;

Ques23.

The **TRUNCATE** command is used to delete all the rows from the table and free the space containing the table.  
The **DELETE** command deletes only the rows from the table based on the condition given in the where clause or deletes all the rows from the table if no condition is specified. But it does not free the space containing the table