

FOUNDATION WITH SQL (2022) INTERVIEW QUESTIONS

1. What is SQL?

SQL may be a language to work databases; it includes database creation, deletion, fetching rows, modifying rows, etc. SQL is an American National Standards Institute (ANSI) standard language, but there are many different versions of the SQL language. SQL is a Structured query language, which can be a computer language for storing, manipulating, and retrieving data stored during an on-line database.

SQL is that the standard language for electronic database Systems. All the electronic database Management Systems (RDMS) use SQL as their standard database language. SQL is that the quality language for handling Relational Databases. SQL used to insert, search, update and delete database records. SQL can do many other operations like optimizing and maintenance of databases.

2. What is MySQL?

It is the combination of My and SQL, MySQL. It is open-source software. MySQL may be a management system that permits you to manage relational databases. It is open-source software backed by Oracle. It means use MySQL without paying a dime. We can also change its ASCII text file to fit your needs. MySQL is easy as compared to other database software like

Oracle Database, or Microsoft SQL Server. MySQL can run on different platforms like UNIX, Linux, Windows, etc. You can install it on a desktop or even on a server. MySQL is a reliable, scalable, and fast database.

3. Difference Between SQL and MySQL

SQL is a database language whereas MYSQL is software.

SQL is used for storing, manipulating, and retrieving data in a relational database whereas MySQL is an open-source Relational Database Management System that allows us to manage relational databases.

SQL helps us to manage the data in the relational database whereas MySQL helps us to manage relational databases using SQL.

SQL does not change as it is a language whereas MySQL updates as it is software.

4. What is DBMS?

The full form of DBMS is Database Management System. DBMS is a program by using it you can interact with the databases to store, modify and extract the data. In other words, A database management system is a program or system software for creating and managing databases. The database serves as an interface between the database and application programs or users. The DBMS gives applications and users a systematic way to create, read, update and manage the data.

The DBMS manages 3 important things: –

The data

The database engines (database engine is a system for storing and retrieving data)

The database schema (database logical structure).

There are mainly 5 types of database management systems: –

RDBMS: Relationship database management system.

NoSQL DBMS

In-memory database management systems. (IMDBMS)

Columnar database management system (CDBMS)

Cloud-based database management system

5. What is RDBMS?

The full form of RDBMS is a relational database management system. A relational database management system (RDBMS) is nothing but a database management system (DBMS) that is used to manage a relational database. And a relational database is the database that follows the relational model as invented by E.F. Codd. It is the basis for SQL and all modern database systems which store data in the form of related tables. In RDBMS, a single database can be spread across several tables.

6. Difference between Database Engine, Database Server, and Database Software.

Database Software

Database Software is also known as database management systems (DBMS). Database software is a computer software application that serves as an interface between the database and application programs or users. The most popular Database software are MySQL, Oracle, SQL Server by Microsoft, PostgreSQL, and IBM DB2. In this course we will work on a database software or database management system, that is MySQL.

Database Engine

A database engine is also known as a storage Engine. Database Engine is the part of a database management system (DBMS) that actually stores and retrieves data. In other words, Database Engine is an underlying software component used by database management systems (DBMS) to create, read, update and delete (CRUD) data from a database. Many of the modern database management systems support multiple storage engines within the same database.

Database Server

A database server is a program that provides database services to other computer programs or computers using a client-server architecture. The database server may also refer to a computer dedicated to running such a program. Some databases like MySQL completely depend on the client-server model for database access.

7. What is a database?

The structured query language is all about databases. Therefore, we must first understand what a database is? The database is a structured set of data organized in such a way so that a computer can quickly access, manage and update it. In other words, a Database is a collection of one or more tables and therefore the tables contain "data rows" or "data records".

Imagine the database with an example of a refrigerator. The refrigerator is a database. The various compartments in the refrigerator are tables. And the vegetables, foods, and beverages stored inside the refrigerator's compartments are the information.

To interact with any database

The user must have an appropriate privilege to access the database.

The user must select the database before executing any SQL statement.

8. How to create and use a Database?

In order to create a database, first, we should log in to the MySQL database. Once you login into the database server, then you can use the following SQL statement to create a new database in MySQL.

Syntax: CREATE DATABASE DatabaseName;

Example: CREATE DATABASE school;

Here, school is the database name. Hit Enter And you will get Query 1 rows affected message. That is the new database has been created. Now, if you want to see all the existing databases, then you can use the below SQL query in MySQL.

SHOW DATABASES;

Once you execute the above SQL Query, you will see the following out. Further, if you notice then you can see the newly created school database that we just created.

9. How to Drop a Database in MySQL?

If you want to drop an existing database in MySQL, then you need to execute the following SQL statement. The following SQL Statement will delete the school database from the MySQL Server.

DROP DATABASE school; Here, the school is the existing database name. When you execute the above SQL Statement, then you will get Query OK 0 rows affected message, that is our database is removed.

Now again type SHOW DATABASES; SQL statement, to display the list of all the existing databases. And you can see, the school database is gone

10. What is a database table?

A database table consists of systematically structured vertical columns or fields and horizontal rows or records. Each column represents a property of the item while each row represents an item. The cell is the unit where a column and row intersect. Data elements or values are stored in a cell. A database table has a specified number of columns, but it can have

any number of rows. A single database must contain a unique table name, while in multiple databases, the same table name can exist in another database.

11. What is a data type in MySQL?

Understanding the data types in MySQL is the most important factor in database development. By understanding data types, we can create more professional, normalized, relational, and easy to manage database columns and table structures. Every column in a table is defined with its data type, depending upon the data value it's going to store. In other words, a data type defines what kind of values a column can store. There are mainly 3 data types:

Numeric data types

Character data types

Temporal data types.

Numeric Data Types:

Numeric data types store only numeric values. Depending upon the values, the numeric data types can be Integer Numeric, Fixed-Point Numbers, and Floating-point Numbers. The Example of numeric data values is price, rating, salaries, etc.

Character Data Types:

Character data types can be store in alphabets, symbols, and also numbers. The Example of character data values are names, description, alphanumeric string, etc.

Temporal Data Types

Temporal data types store date, time, and date-time together. Data type's name and size can be different in a different database management system.

12. What is the Use of Select Statement in MySQL?

The SELECT Statement is used to return records in the form of a result set (i.e. rows) from one or more database tables or views. The most important point that you need to remember is, the SELECT statement does not store any data itself. It simply displays the data that is stored in database tables. The Select Statement can retrieve and show the data from one or more database tables, from other queries, or from a combination of the above two.

Select Statement Syntax in MySQL:

Syntax1: SELECT All Columns

If you want to select all the columns of a table or view then you can also use "*" as shown below

```
SELECT * FROM Table_Name
```

Note: In the SQL SELECT statement (*) star or asterisk symbol is a special type of keyword that means ALL. Whenever you want to retrieve all columns from a table or view in a SQL SELECT Statement, then you can use *.

Syntax2: SELECT Specific Columns

If you want to select specific columns, then you need to specify such columns before the FROM Clause as shown below.

```
SELECT Column_List FROM Table_Name
```

Note: If you want to select all the columns of a table or view then you can also use "*", but for better performance use the column list instead of using "*".

Syntax3: SELECT All Columns with Conditions

If you want to select All Columns with WHERE (or any other) conditions then you need to use the select statement as shown below

```
SELECT * FROM tables [WHERE conditions];
```

Syntax4: SELECT Specified Columns with Conditions

If you want to select Specified Columns with WHERE (or any other) conditions then you need to use the select statement as shown below. Here, you need to specify the column list that you want to retrieve.

```
SELECT Column_List FROM tables [WHERE conditions];
```

13. What are clauses in MySQL? ^

Clauses are in-built functions available in SQL which is used to deal with data easily stored in the table. It helps us to filter and analyze the data quickly. Clauses are used to get the required data by the user when we have a large number of data stored in the database. That means if you want to provide some additional functionalities such as filtering the records, sorting the records, fetching the records, and grouping the records then you need to use the Clause with the query.

14. Why we need Clauses in MySQL? ^

MySQL Clauses are used for database access and data extraction by applying a filter that describes the conditions. If the condition is satisfied, we will get the particular result rows and the information that does not satisfy the condition is not used. So, SQL Clauses helps us to access proper info and is manageable to deal with different tables in a database.

15. What is a Having Clause in MySQL? ^

The Having Clause in MySQL is also used for filtering the data just like the where clause. Having Clause in MySQL filters the rows from the intermediate result set that is built by using the FROM, WHERE, or GROUP BY clauses in a SELECT statement.

The Having Clause in MySQL is typically used with a GROUP BY clause. That means the Having Clause is used in combination with a GROUP BY clause to restrict the number of groups to be returned by satisfying the condition which is specified using the having clause.

16. Why do we need Having Clause in MySQL?

The WHERE clause could not be used with aggregate data rows, hence the HAVING clause was added to SQL. The purpose of the HAVING clause is to filter the group rows produced by the GROUP BY clause.

The HAVING clause is the same as the WHERE clause. The only difference is WHERE clause FILTERS the intermediate data results, while the HAVING clause operates on group rows. Likewise, WHERE clause, we can use conditions and operators with the HAVING clauses to build complex SQL statements.

Since the HAVING clause acts as a filter on group rows, the only possible column in group rows are columns specified in the GROUP BY clause.

17. Difference Between Where Clause and Having Clause in MySQL

WHERE clause in MySQL cannot be used with aggregate functions whereas HAVING clause can be used with aggregate functions. That means the WHERE clause in MySQL is used for filtering individual rows on a table whereas the HAVING clause in MySQL is used to filtering the groups which are created by the Group by Clause.

The WHERE comes before the GROUP BY clause. That means the WHERE clause filters rows before aggregate calculations are performed. On the other hand, the HAVING clause comes after GROUP by Clause. That means the HAVING clause filters rows after aggregate calculations are performed. So, from a performance standpoint, the HAVING Clause is slower than the WHERE Clause and should be avoided if possible.

WHERE and HAVING clauses can be used together in a single SELECT statement. In that case, the WHERE clause is applied first to filter individual rows. The rows are then grouped and aggregate calculations are performed, and then only the HAVING clause filters the groups in MySQL.

The MySQL WHERE clause can be used with Select, Insert, and Update statements whereas the HAVING clause can only be used with the Select statement.

We can use the Where clause without using the Group by Clause but we can not use the Having Clause without using the

Group by Clause in MySQL.

18. What is Order by Clause in MySQL?

The Order by Clause in MySQL is used for sorting the data either in ascending or descending order based on a specified column or list of columns. That means if you want to sort the output or result of a query either in ascending or descending order then you need to use MySQL Order by Clause.

19. What are the Conditions?

In MySQL, a condition or an expression is made up of keywords, identifiers, and constants and it compares given values with the data rows values in a table. If the condition is matched with the data row it's called a true condition otherwise a false condition.

20. What are Comparison Operators in MySQL?

As the name suggests the Comparison Operators in MySQL are used to compare two values i.e. these operators are used for comparing one expression with another expression. The comparison operators determine whether the two values are equal or a value is greater than the other, or less than the other. The comparison operators can be applied to numbers, strings, and dates. The result of a comparison can be TRUE, FALSE, or NULL (When one or both the expression contains NULL values).

Types of Comparison Operators in MySQL

The different types of comparison operators that are available in MySQL Database are as follows:

Equal (=) Operator

Not Equal (!= or <>) Operator

Greater Than (>) Operator

Less Than (<) Operator

Greater Than or Equal To (>=) Operator

Less Than or Equal To (<=) Operator

21. What are Logical Operators in MySQL? ^

If you want to combine more than one condition, then you need to use the Logical Operators in MySQL. The Logical Operators are used to check for the truthness of some conditions. Logical operators return a Boolean data type with a value of TRUE, FALSE, or UNKNOWN. In MySQL, there are three Logical Operators available. They are as follows:

AND: TRUE if both Boolean expressions are TRUE.

OR: TRUE if one of the Boolean expressions is TRUE.

NOT: Reverses the value of any other Boolean operator.

22. What is LIKE Operator in MySQL? ^

The LIKE Operator in MySQL is basically used with the WHERE clause to search for a specific pattern in a string. The string pattern contains wildcard characters that represent missing characters. The LIKE operator is written as a word "LIKE" followed by a pattern to match with column values.

23. What is EXISTS Operator in MySQL? ^

The EXISTS operator in MySQL is used in combination with a subquery and is considered to be met if the subquery returns at least one row. That means the EXISTS operator is used to check the existence of a result of a subquery. It can be used in a SELECT, INSERT, UPDATE, or DELETE statement.

The EXISTS condition is very similar to IN condition. In the IN condition, we directly define constant values to evaluate with data rows, while in EXISTS condition we use a subquery. If the subquery returns TRUE the main query is evaluated. If the subquery returns FALSE the system won't return the data rows.

24. Difference Between Exists and IN Operator in MySQL

EXISTS is used to check the existence of data in a subquery whereas IN is used to minimize the multiple OR conditions.

The EXISTS operator stops its further execution once it finds the first true occurrence whereas IN operator compares all the values present inside it.

EXISTS operator does not perform a comparison between parent query and child query whereas IN operator performs a comparison between parent query and child query.

25. What are Arithmetic Operators?

Arithmetic Operators are used for performing mathematical calculations such as Addition, Subtraction, Multiplication, Module, and Division represented by the expected +, -, *(star or asterisk), %, and / forward slash symbols respectively on the given operand values. That is, they are used to perform mathematical operations on two expressions of the same or different data types of numeric data.

26. What are SET Operators in MySQL?

The SET Operators in MySQL are basically used to combine the result of more than 1 select statement and return the output as a single result set. In SQL, 4 types of set operators are. They are as follows:

UNION: It is used to combine two or more result sets into a single set, without duplicates.

UNION ALL: It is used to combine two or more result sets into a single set, including duplicates.

INTERSECT: It is used to combine two result sets and returns the data which are common in both the result set.

EXCEPT: It is used to combine two result sets and returns the data from the first result set which is not present in the second result set.

Points to Remember while working with Set Operations:

Every SELECT statement involved in the query must have a similar number of columns.

The columns in the SELECT statement must be in the same order and have similar data types.

In order to sort the result, an ORDER BY clause should be part of the last select statement. The column names or aliases must be found out by the first select statement.

27. What is Data Integrity in MySQL? ^

While learning Constraints we need to understand one database term called Data Integrity. Data integrity means the data contained in the database is accurate, consistent, and reliable. To provide data integrity, RDBMS provides us a set of integrity constraints that ensures that the data entered into the database is going to be accurate, consistent, and reliable. This is the reason why the end-user can trust the data stored in the database.

28. What are Constraints in MySQL? ^

The MySQL Constraints define specific rules to the column(s) data in a database table. While inserting, updating, or deleting the data rows, if the rules of the constraint are not followed, the system will display an error message and the action will be terminated. The SQL Constraints are defined while creating a new table. We can also alter the table and add new SQL Constraints. The MySQL Constraints are mainly used to maintain data integrity.

29. What are the different types of Constraints available in MySQL? ^

MySQL supports the following six types of constraints for maintaining data integrity.

NOT NULL Constraint

UNIQUE KEY Constraint

Default Constraint

CHECK KEY Constraint

PRIMARY KEY Constraint

FOREIGN KEY Constraint.

30. When to use NOT NULL Constraint?

If you don't want to accept NULL values but you want to accept duplicate values to be stored in a column, then you need to apply NOT NULL Constraint to that column in MySQL.

31. What is Default Constraint in MySQL?

As the name suggests the DEFAULT Constraint is used to set a default value for a data column. If the value for the column in the data row is not defined, the default value will be added to the data row column. In simple words, we can say that Default constraints enable MySQL to insert a default value to a column when the user doesn't specify a value.

32. How to Add Default Constraint in MySQL?

We can add the DEFAULT Constraint in MySQL while creating a table using the CREATE TABLE statement, or using the ALTER TABLE statement for the existing table.

33. When to use Default Constraint in MySQL?

In Real-time applications, sometimes we may need to add a default value to a column in MySQL and we don't need that value to come from the user. For example, while creating an order, we may need to store the order date and we want the order date to be stored as the current date and we can easily do this by using the default constraint in MySQL.

Note: In a MySQL table, each column must contain a value including a NULL. While inserting data into a table, if no value is supplied to a column, then the column gets the value set as DEFAULT.

34. What is Primary Key in MySQL?

The Primary Key in MySQL is the combination of UNIQUE and NOT NULL Constraint. That means it will not allow either NULL or Duplicate values into a column or columns on which the primary key constraint is applied. Using the primary key, we can enforce entity integrity i.e. using the primary key we can uniquely identify each record.

A table should contain only 1 Primary Key which can be either on a single or multiple columns i.e. the composite primary key. The Primary Key constraint in MySQL can be applied to numeric and string data types such as INT, VARCHAR, etc.

35. When do we need to choose Primary Key?

When we need the following features on a column, then we need to make that column Primary Key in MySQL

NULLs should not be allowed.

It should be unique

It should not be modified.

Note: The PRIMARY KEY Constraint is useful to restrict storing of duplicate data row values in a given column. The PRIMARY KEY is similar to UNIQUE Constraint but unlike UNIQUE Constraint there can be only one PRIMARY KEY for one table. The PRIMARY KEY automatically sets UNIQUE Constraint for that table column. The PRIMARY KEY column can not contain NULL values. The

Primary key can be defined while creating a new database table or can be added by using ALTER TABLE statement.

36. What is a Foreign Key Constraint in MySQL?

Creating the relationship between the database tables is one of the most important concepts in a database. The relationship between multiple tables provides a mechanism for linking the data stores in multiple tables and retrieving them in an efficient manner.

In order to create a link between two tables, we must specify a Foreign Key in one table that references a column in another table. That means the Foreign Key constraint in MySQL is used for binding two tables with each other and then verify the existence of one table data in other tables.

Note: A foreign key in one TABLE points to either a primary key or a unique key in another table in MySQL. The foreign key constraints are basically used to enforce referential integrity.

37. How to Create Foreign Key Constraint in MySQL?

To Create a Foreign Key Constraint in MySQL, we require the following things

We require two tables for linking with each other and those two tables must have a common column for binding the tables.

The common column that is present in both the tables need not have the same name but their data type must be the same.

The common column that is present under the parent table or master table is known as the reference key column and moreover, the reference key column should not contain any duplicate values. So, we need to impose either UNIQUE or PRIMARY key constraint on that column.

The common column which is present in the child or detailed table is known as the Foreign key column and we need to impose a Foreign key constraint on the column which refers to the reference key column of the master table.

38. What is the difference between Primary Key and Foreign Key Constraint in MySQL?

Primary Key in MySQL:

The Primary Key Constraint in MySQL is uniquely identifying a record in a table.

Primary Key constraint neither accepts null values nor duplicate values on the column on which it is applied.

We can create only one Primary Key on a table in MySQL and that primary key constraint can be created either on a single column or multiple columns.

Foreign Key in MySQL:

The Foreign Key in MySQL is a field in a table that is a unique key (either primary or unique key) in another table.

A Foreign Key can accept both null values and duplicate values.

We can create more than one Foreign key on a table in MySQL.

39. Can we create a table with multiple unique, foreign, or primary keys in MySQL?

We can create a table in MySQL with multiple unique and foreign keys. But it is not possible to create a table with multiple primary Keys.

40. Is it possible that a foreign key references a non-primary key in MySQL?

Yes, it is possible. The point that you need to keep in mind is that a foreign key actually references a key that should contain unique values. So, it may be a primary key or a unique key as both keys maintain the uniqueness of the column of a table.

41. What are Referential Integrity Constraints in MySQL?

The Referential Integrity Constraints in MySQL are nothing but the foreign key constraints that tell MySQL Database to perform certain actions whenever a user attempts to delete or update a primary key for which existing foreign keys point.

42. What are the Actions Performed by MySQL?

In order to tell MySQL what actions to perform whenever a user trying to delete or update a primary key value for which existing foreign key values point, MySQL provided with the following Referential Integrity Constraints which we need to set while creating the foreign key constraints.

SET NULL: If a user tries to delete or update statement(s) that will affect rows in the foreign key table, then those values will be set to NULL when the primary key record is deleted or updated in the Primary key table. The important thing that we need to keep in mind is that the foreign key columns affected must allow NULL values.

CASCADE: If a user tries to delete the statement(s) which will affect the rows in the foreign key table, then those rows will also be deleted when the primary key record is deleted. Similarly, if an update statement affects rows in the foreign key table, then those rows will also be updated with the value from the primary key record after it has been updated.

NO ACTION: This is the default action that MySQL performs. This specifies that if an update or deletes statement affects rows in foreign key tables, then the action will be denied and rolled back. An error message will be raised.

43. Difference Between LENGTH() and CHAR_LENGTH() in MySQL

It is important to understand the difference between the LENGTH() and CHAR_LENGTH() functions. The result of both these functions may appear to be the same even if it's not. The MySQL CHAR_LENGTH() function returns the number of characters in its argument whereas the MySQL LENGTH() function returns the length of a string in bytes (not the number of characters).

The LENGTH function returns the length of the string in bytes. So, it will not suitable for UNICODE character strings since they are multibyte. In that case, you need to use the CHAR_LENGTH function in MySQL to just count the number of characters irrespective of single-byte or multibyte characters.

44. Difference between CAST AND CONVERT Function ^

The main difference between MySQL CAST and CONVERT functions are given below.

CAST Function:

We use the MySQL CAST function to convert one data type into the other data type.

It cannot be used to convert a character set.

It is the part of ANSI-SQL standard

CAST function uses the "AS" keyword for separating the parameter.

CONVERT Function:

We use the MySQL CONVERT function to convert one data type into the other data type.

It can be used to convert a character set.

It is not part of the ANSI-SQL standard

CONVERT function uses the "comma(,)" or "USING" keyword for separating the parameter.

45. What are JOINS in MySQL? ^

The Joins in MySQL are basically used to fetch the rows from two or more related tables. The tables involved in the join are basically related to each other using the primary key and foreign key relationship but the important point is that it is not mandatory. What it means, the tables involved in the MySQL Joins must have a common field. Common field means both the column must be compatible in terms of data type and based on that common field the MySQL JOINS retrieves the data.

46. Why should we use JOINS?

With JOINS we can achieve better MySQL and application performances as it can use indexing. Instead of using multiple queries JOIN simply uses a single query with any search parameters which gives us better performance as compared to subqueries.

47. Types of JOINS in MySQL

There are mainly 3 types of joins in MySQL:

INNER JOIN: The Inner join returns only the matching records from both the tables involved in the Join. Non-matching records are eliminated.

OUTER JOIN: The Outer Join retrieves the matching records as well as non-matching records from both the tables involved in the join in MySQL.

CROSS JOIN: If two or more tables are combined with each other without any condition then we call it cross join in MySQL. In cross join, each record of a table is joins with each record of another table.

48. What is Inner Join in MySQL?

The Inner Join is used to return only the matching rows from both the tables involved in the join. That means it eliminates the non-matching rows. The INNER JOIN is the most common type of join used in real-time applications.

49. When do we need to use Inner JOIN in MySQL?

If you want to retrieve all the matching rows from both the tables involved in the join by removing all the non-matching rows in the result set then you need to use the Inner Join.

50. What is Left Outer Join in MySQL?

The Left Outer Join in MySQL is used to retrieve all the matching records from both the tables as well as non-matching records from the left side table. In that case, the non-matching data will take a null value. The LEFT OUTER JOIN or LEFT JOIN keyword is used to perform the left join in MySQL.

The question that should come to your mind is which is the left table and which is the right table? The answer is, the table which is mentioned to the left of the LEFT OUTER JOIN keyword is the left table, and the table which is mentioned to the right of the LEFT OUTER JOIN keyword is the right table.

51. When do we need to use Left JOIN in MySQL?

If you want to retrieve all the matching rows from both the tables involved in the join as well as all the non-matching rows from the left side table in the result set then you need to use Left Join in MySQL. In that case, the non-matching rows will take a null value.

52. What is Right Outer Join in MySQL?

The Right Outer Join in MySQL is used to retrieve all the matching records from both the tables as well as all the non-matching records from the right-hand side table. In that case, the non-matching data will take a null value. The RIGHT OUTER JOIN or RIGHT JOIN keyword is used to perform the right join operation in MySQL.

The question that should come to your mind is which is the left table and which is the right table? The answer is, the table which is mentioned to the left of the RIGHT OUTER JOIN keyword is the left table, and the table which is mentioned to the right of the RIGHT OUTER JOIN keyword is the right table.

53. When do we need to use right JOIN in MySQL?

If you want to retrieve all the matching rows from both the tables involved in the join as well as all the non-matching rows from the right-hand side table in the result set then you need to use Right Outer Join in MySQL. In that case, the non-matching rows will take a null value.

54. What is Self-Join in MySQL?

The Self Join in MySQL is nothing but joining a table by itself. We need to use Self Join in MySQL when we have some relations between the columns of the same table. If this is not clear at the moment then don't worry, we will discuss this with an example.

When you are implementing the self-join mechanism in MySQL, then you have to create the alias for the table name. You can create any number of aliases for a single table name in MySQL. Aliases are nothing but the alternative name given to a table. The Self Join is not a different kind of join. The Self Join is like any other join except that the two instances of the same table will be joined. So, it can be classified as any type of join, such as

Inner Join

Outer (Left, Right, Full) join

Cross Join