

NAME → Snehal Raghunath Thomake

Batch → 7670. Center→ Thane

Date=28/09/2022 Wed.

DBMS (Database Management System)

The DBMS **manages incoming data, organizes it, and provides ways for the data to be modified or extracted by users or other programs**. Some DBMS examples include MySQL, PostgreSQL, Microsoft Access, SQL Server, FileMaker, Oracle, RDBMS, dBASE, Clipper, and FoxPro.

SQL (Structured Query Language)

Structured Query Language(SQL) as we all know is the database language by the use of which we can perform certain operations on the existing database and also we can use this language to create a database. SQL uses certain commands like Create, Drop, Insert, etc. to carry out the required tasks.

These SQL commands are mainly categorized into five categories as:

1. DDL – Data Definition Language
2. DQL – Data Query Language
3. DML – Data Manipulation Language
4. DCL – Data Control Language
5. TCL – Transaction Control Language

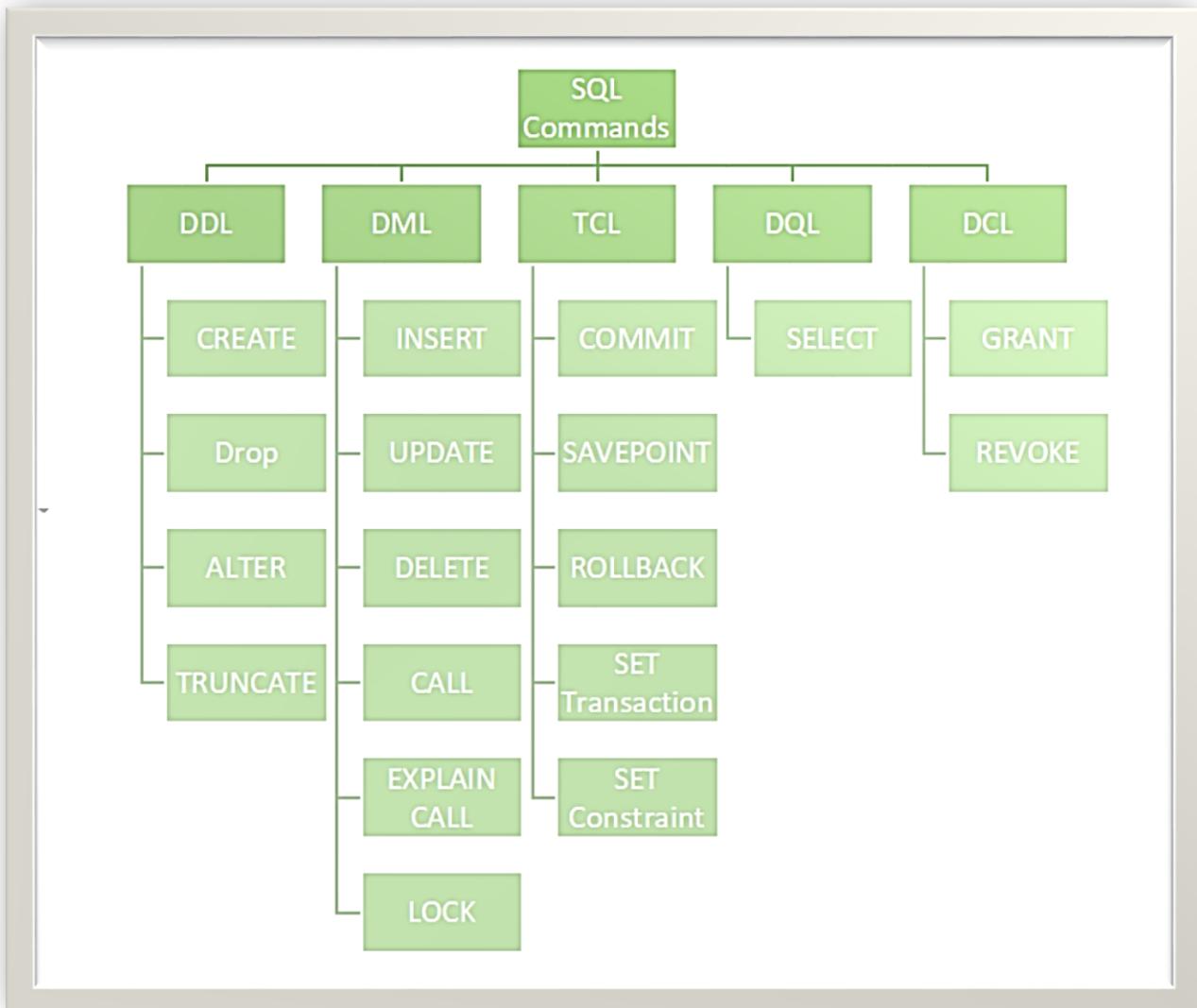
Date=29/09/2022 Thu.

SQL ,MYSQL, ORACLE and Working on web page.

Date=30/09/2022 Friday

These SQL commands are mainly categorized into five categories as:

1. DDL – Data Definition Language
2. DQL – Data Query Language
3. DML – Data Manipulation Language
4. DCL – Data Control Language
5. TCL – Transaction Control Language



1) Data Definition Language (DDL)

- DDL changes the structure of the table like creating a table, deleting a table, altering a table, etc.
- All the command of DDL are auto-committed that means it permanently save all the changes in the database.

Here are some commands that come under DDL:

- **CREATE** → It is used to create a new table in the database.

Syntax

Create table "table name"

```
("column1" "data type",
"column2" "data type",
"column3" "data type",
...
"ColumnN" "data type");
```

Example:

```
CREATE TABLE EMPLOYEE(Name VARCHAR2(20), Email VARCHAR2(100),
DOB DATE , ID INT);
```

- **DROP** → It is used to delete both the structure and record stored in the table.

Syntax

```
DROP TABLE table_name;
```

Example

```
DROP TABLE EMPLOYEE;
```

- **ALTER** → It is used to alter the structure of the database. This change could be either to modify the characteristics of an existing attribute or probably to add a new attribute.

Syntax

```
ALTER TABLE table_name ADD column_name COLUMN-definition;
```

EXAMPLE

```
ALTER TABLE STU_DETAILS ADD(ADDRESS VARCHAR2(20));
```

```
ALTER TABLE STU_DETAILS MODIFY (NAME VARCHAR2(20));
```

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

Syntax

```
TRUNCATE TABLE table_name;
```

Example:

```
TRUNCATE TABLE EMPLOYEE;
```

- **COMMENT** → SQL Comments are used to explain the sections of the SQL statements, and used to prevent the statements of SQL. In many programming languages, comments matter a lot.

Syntax

```
SELECT * FROM table name
```

Example

```
SELECT * FROM Employee
```

- **RENAME** → In some situations, database administrators and users want to change the name of the table in the SQL database because they want to give a more relevant name to the table.

Syntax

```
RENAME old_table_name To new_table_name ;
```

Example

```
RENAME Cars To Car_2021_Details ;
```

2. Data Manipulation Language

- DML commands are used to modify the database. It is responsible for all form of changes in the database.
- The command of DML is not auto-committed that means it can't permanently save all the changes in the database. They can be rollback.

a. INSERT: The INSERT statement is a SQL query. It is used to insert data into the row of a table.

Syntax:

```
INSERT INTO TABLE_NAME  
(col1, col2, col3,... col N)  
VALUES (value1, value2, value3, .... valueN);
```

Or

```
INSERT INTO TABLE_NAME  
VALUES (value1, value2, value3, .... valueN);
```

For example:

```
INSERT INTO javatpoint (Author, Subject) VALUES ("Sonoo", "DBMS");
```

b. UPDATE: This command is used to update or modify the value of a column in the table.

Syntax:

```
UPDATE table_name SET [column_name1= value1,...column_nameN = valueN] [WHERE CO  
NDITION]
```

For example:

```
UPDATE students  
SET User_Name = 'Sonoo'  
WHERE Student_Id = '3'
```

c. DELETE: It is used to remove one or more row from a table.

Syntax:

```
DELETE FROM table_name [WHERE condition];
```

For example:

```
DELETE FROM javatpoint  
WHERE Author="Sonoo";
```

3. Data Control Language

DCL commands are used to grant and take back authority from any database user.

a. Grant: It is used to give user access privileges to a database.

Example

```
GRANT SELECT, UPDATE ON MY_TABLE TO SOME_USER, ANOTHER_USER;
```

b. Revoke: It is used to take back permissions from the user.

Example

```
REVOKE SELECT, UPDATE ON MY_TABLE FROM USER1, USER2;
```

4. Transaction Control Language

TCL commands can only use with DML commands like INSERT, DELETE and UPDATE only.

These operations are automatically committed in the database that's why they cannot be used while creating tables or dropping them.

a. Commit: Commit command is used to save all the transactions to the database.

Syntax:

```
COMMIT;
```

Example:

```
DELETE FROM CUSTOMERS  
WHERE AGE = 25;  
COMMIT;
```

b. Rollback: Rollback command is used to undo transactions that have not already been saved to the database.

Syntax:

```
ROLLBACK;
```

Example:

```
DELETE FROM CUSTOMERS  
WHERE AGE = 25;  
ROLLBACK;
```

c. SAVEPOINT: It is used to roll the transaction back to a certain point without rolling back the entire transaction.

Syntax:

```
SAVEPOINT SAVEPOINT_NAME;
```

5. Data Query Language / Data Retrieval Language

DQL is used to fetch the data from the database.

a. SELECT: This is the same as the projection operation of relational algebra. It is used to select the attribute based on the condition described by WHERE clause.

Syntax:

```
SELECT expressions  
FROM TABLES  
WHERE conditions;
```

For example:

```
SELECT emp_name  
FROM employee  
WHERE age > 20;
```

Data Types:-

CHARACTER [(length)] or CHAR [(length)]

The CHARACTER data type accepts character strings, including Unicode, of a fixed length.

VARCHAR (length)

The VARCHAR data type accepts character strings, including Unicode, of a variable length up to the maximum length specified in the data type declaration.

INTEGER or INT

The INTEGER data type accepts numeric values with an implied scale of zero

Date= 1/10/2022 Sat.

CREAT TABLE

The screenshot shows a Windows desktop environment with a MySQL Command Line Client window open. The window title is "MySQL 8.0 Command Line Client". The command history shows:

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| rock |
| sakila |
| student |
| sys |
| world |
+-----+
8 rows in set (0.01 sec)

mysql> use world;
Database changed
mysql> create table stinfo(rollno int, fname varchar(30), lname varchar(25), sname varchar(20), mark int);
Query OK, 0 rows affected (0.10 sec)

mysql> insert into stinfo(rollno, fname, lname, sname,mark) value (1,'Snehal', 'Thomake', 'Physics', 85);
Query OK, 1 row affected (0.02 sec)

mysql> select * from stinfo
-> ;
+-----+
| rollno | fname | lname | sname | mark |
+-----+
| 1 | Snehal | Thomake | Physics | 85 |
+-----+
1 row in set (0.00 sec)
```

The taskbar at the bottom shows various application icons, and the system tray indicates the date and time as 10/10/2022 11:26 AM.

ALTER

The ALTER TABLE statement is used to add, delete, or modify columns in an existing table.

The ALTER TABLE statement is also used to add and drop various constraints on an existing table.

The screenshot shows a Windows desktop environment with a MySQL Command Line Client window open. The command history shows:

```
mysql> show databases;
+-----+
| Database |
+-----+
| rock |
| sakila |
| student |
| sys |
| world |
+-----+
8 rows in set (0.01 sec)

mysql> use world;
Database changed
mysql> create table stinfo(rollno int, fname varchar(30), lname varchar(25), sname varchar(20), mark int);
Query OK, 0 rows affected (0.10 sec)

mysql> insert into stinfo(rollno, fname, lname, sname,mark) value (1,'Snehal', 'Thomake', 'Physics', 85);
Query OK, 1 row affected (0.02 sec)

mysql> select * from stinfo
-> ;
+-----+
| rollno | fname | lname | sname | mark |
+-----+
| 1 | Snehal | Thomake | Physics | 85 |
+-----+
1 row in set (0.00 sec)

mysql> alter table stinfo add phoneno int;
Query OK, 0 rows affected (0.07 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> select * from stinfo;
+-----+
| rollno | fname | lname | sname | mark | phoneno |
+-----+
| 1 | Snehal | Thomake | Physics | 85 | NULL |
+-----+
1 row in set (0.00 sec)

mysql>
```

The taskbar at the bottom shows various application icons, and the system tray indicates the date and time as 10/10/2022 11:26 AM.

ROW CREATE

```
MySQL Select MySQL 8.0 Command Line Client
mysql> create table stinfo(rollno int, fname varchar(30), lname varchar(25), sname varchar(20), mark int);
Query OK, 0 rows affected (0.10 sec)

mysql> insert into stinfo(rollno, fname, lname, sname,mark) value (1,'Snehal', 'Thomake', 'Physics', 85);
Query OK, 1 row affected (0.02 sec)

mysql> select * from stinfo;
+-----+-----+-----+-----+
| rollno | fname | lname | sname |
+-----+-----+-----+-----+
| 1 | Snehal | Thomake | Physics |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> alter table stinfo add phoneno int;
Query OK, 0 rows affected (0.07 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> select * from stinfo;
+-----+-----+-----+-----+-----+
| rollno | fname | lname | sname | mark | phoneno |
+-----+-----+-----+-----+-----+
| 1 | Snehal | Thomake | Physics | 85 | NULL |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> insert into stinfo(rollno, fname, lname, sname,mark) value (2,'Supriya', 'Kamble', 'Maths', 70);
Query OK, 1 row affected (0.01 sec)

mysql> select * from stinfo;
+-----+-----+-----+-----+-----+
| rollno | fname | lname | sname | mark | phoneno |
+-----+-----+-----+-----+-----+
| 1 | Snehal | Thomake | Physics | 85 | NULL |
| 2 | Supriya | Kamble | Maths | 70 | NULL |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

UPDATE

The UPDATE statement is used to modify the existing records in a table.

```
MySQL Select MySQL 8.0 Command Line Client
Query OK, 1 row affected (0.01 sec)

mysql> select * from stinfo;
+-----+-----+-----+-----+-----+
| rollno | fname | lname | sname | mark | phoneno |
+-----+-----+-----+-----+-----+
| 1 | Snehal | Thomake | Physics | 85 | NULL |
| 2 | Supriya | Kamble | Maths | 70 | NULL |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> update stinfo set phoneno= 7887838057 where rollno= 1;
ERROR 1264 (22003): Out of range value for column 'phoneno' at row 1
mysql> update stinfo set phoneno= '7887838057' where rollno= 1;
ERROR 1264 (22003): Out of range value for column 'phoneno' at row 1
mysql> update stinfo set phoneno= '7887838057';
ERROR 1264 (22003): Out of range value for column 'phoneno' at row 1
mysql> select * from stinfo;
+-----+-----+-----+-----+-----+
| rollno | fname | lname | sname | mark | phoneno |
+-----+-----+-----+-----+-----+
| 1 | Snehal | Thomake | Physics | 85 | NULL |
| 2 | Supriya | Kamble | Maths | 70 | NULL |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> UPDATE stinfo SET sname = 'Chemistry', mark = '90' WHERE rollno = 1;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from stinfo;
+-----+-----+-----+-----+-----+
| rollno | fname | lname | sname | mark | phoneno |
+-----+-----+-----+-----+-----+
| 1 | Snehal | Thomake | Chemistry | 90 | NULL |
| 2 | Supriya | Kamble | Maths | 70 | NULL |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> UPDATE stinfo SET phoneno = '7887838057', mark = '95' WHERE rollno = 1;
ERROR 1264 (22003): Out of range value for column 'phoneno' at row 1
mysql>
```

SELECT

The SELECT statement is used to select data from a database.

The data returned is stored in a result table, called the result-set.

```
MySQL 8.0 Command Line Client
mysql> update stinfo set phoneno= 7887838057 where rollno= 1;
ERROR 1264 (22003): Out of range value for column 'phoneno' at row 1
mysql> update stinfo set phoneno= '7887838057' where rollno= 1;
ERROR 1264 (22003): Out of range value for column 'phoneno' at row 1
mysql> update stinfo set phoneno= '7887838057';
ERROR 1264 (22003): Out of range value for column 'phoneno' at row 1
mysql> select * from stinfo;
+-----+-----+-----+-----+-----+
| rollno | fname | lname | sname | mark | phoneno |
+-----+-----+-----+-----+-----+
| 1 | Snehal | Thomake | Physics | 85 | NULL |
| 2 | Supriya | Kamble | Maths | 70 | NULL |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> UPDATE stinfo SET sname = 'Chemistry', mark = '90' WHERE rollno = 1;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from stinfo;
+-----+-----+-----+-----+-----+
| rollno | fname | lname | sname | mark | phoneno |
+-----+-----+-----+-----+-----+
| 1 | Snehal | Thomake | Chemistry | 90 | NULL |
| 2 | Supriya | Kamble | Maths | 70 | NULL |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> UPDATE stinfo SET phoneno = '7887838057', mark = '95' WHERE rollno = 1;
ERROR 1264 (22003): Out of range value for column 'phoneno' at row 1
mysql> SELECT rollno, lname, lname FROM stinfo;
+-----+-----+-----+
| rollno | lname | lname |
+-----+-----+-----+
| 1 | Thomake | Thomake |
| 2 | Kamble | Kamble |
+-----+-----+-----+
2 rows in set (0.02 sec)

mysql>
```

Date=03/10/2022 Monday

DROP

The DROP TABLE statement is used to drop an existing table in a database.

MySQL 8.0 Command Line Client

```
mysql> select * from school;
+-----+-----+-----+-----+-----+
| SchoolID | School_Name | students_no | Teachers_no | classroom_no | EmailID |
+-----+-----+-----+-----+-----+
| 1 | girls Govp School | 1000 | 80 | 12 | abcd |
| 2 | Boys School | 1500 | 90 | 15 | efgg |
| 3 | English medium School | 20050 | 85 | 19 | bdcj |
| 4 | Marathi medium School | 900 | 70 | 10 | khjb |
| 5 | Navjeevan School | 1750 | 95 | 20 | sneh |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> select * from sub_info;
Empty set (0.01 sec)

mysql> drop "sub_info";
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ''sub_info'' at line 1
mysql> drop sub_info;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'sub_info' at line 1
mysql> DROP TABLE sub_info;
Query OK, 0 rows affected (0.15 sec)

mysql> select * from sub_info;
ERROR 1146 (42S02): Table 'world.sub_info' doesn't exist
mysql>
```

DELETE

The DELETE statement is used to delete existing records in a table.

MySQL 8.0 Command Line Client

```
mysql> select * from school;
+-----+-----+-----+-----+-----+
| SchoolID | School_Name | students_no | Teachers_no | classroom_no | EmailID |
+-----+-----+-----+-----+-----+
| 1 | girls Govp School | 1000 | 80 | 12 | abcd |
| 2 | Boys School | 1500 | 90 | 15 | efgg |
| 3 | English medium School | 20050 | 85 | 19 | bdcj |
| 4 | Marathi medium School | 900 | 70 | 10 | khjb |
| 5 | Navjeevan School | 1750 | 95 | 20 | sneh |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> select * from sub_info;
Empty set (0.01 sec)

mysql> drop "sub_info";
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ''sub_info'' at line 1
mysql> drop sub_info;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'sub_info' at line 1
mysql> DROP TABLE sub_info;
Query OK, 0 rows affected (0.15 sec)

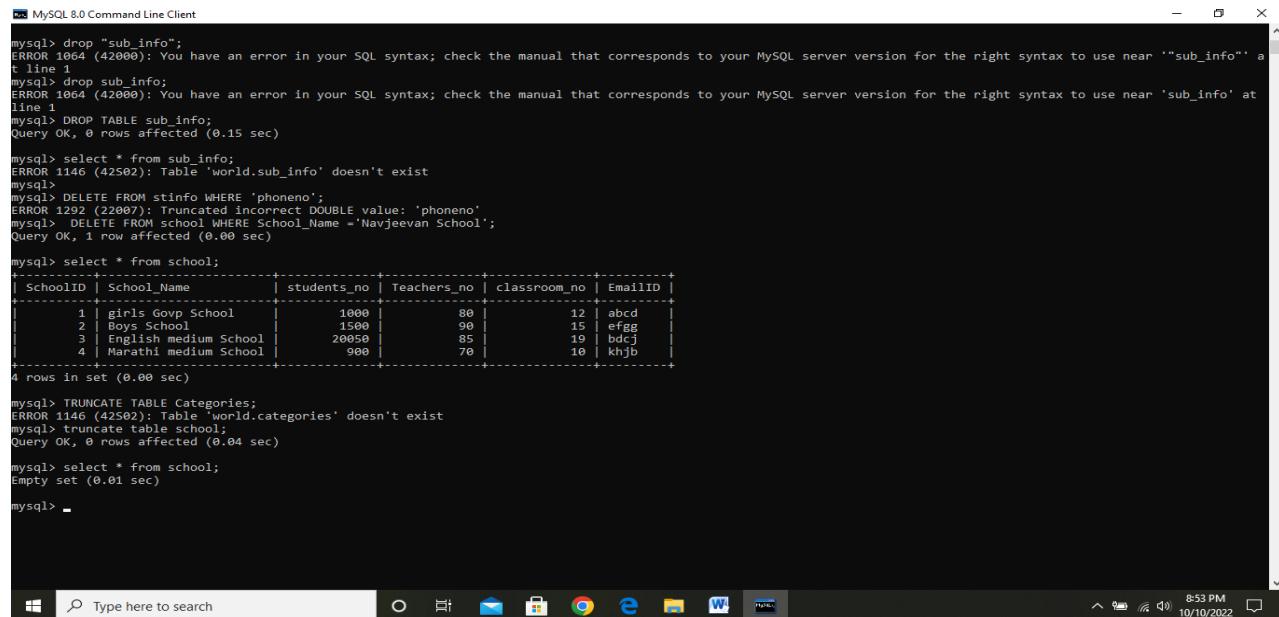
mysql> select * from sub_info;
ERROR 1146 (42S02): Table 'world.sub_info' doesn't exist
mysql>
mysql> DELETE FROM stinfo WHERE 'phoneno';
ERROR 1292 (22007): Truncated incorrect DOUBLE value: 'phoneno'
mysql> DELETE FROM school WHERE School_Name = 'Navjeevan School';
Query OK, 1 row affected (0.00 sec)

mysql> select * from school;
+-----+-----+-----+-----+-----+
| SchoolID | School_Name | students_no | Teachers_no | classroom_no | EmailID |
+-----+-----+-----+-----+-----+
| 1 | girls Govp School | 1000 | 80 | 12 | abcd |
| 2 | Boys School | 1500 | 90 | 15 | efgg |
| 3 | English medium School | 20050 | 85 | 19 | bdcj |
| 4 | Marathi medium School | 900 | 70 | 10 | khjb |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

TRUNCATE TABLE

The TRUNCATE TABLE command deletes the data inside a table, but not the table itself.



```
MySQL 8.0 Command Line Client

mysql> drop "sub_info";
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '"sub_info"' at
t line 1
mysql> drop sub_info;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'sub_info' at
line 1
mysql> DROP TABLE sub_info;
Query OK, 0 rows affected (0.15 sec)

mysql> select * from sub_info;
ERROR 1146 (42S02): Table 'world.sub_info' doesn't exist
mysql>
mysql> DELETE FROM stinfo WHERE 'phoneno';
ERROR 1292 (22007): Truncated incorrect DOUBLE value: 'phoneno'
mysql> DELETE FROM school WHERE School_Name = 'Navjeevan School';
Query OK, 1 row affected (0.00 sec)

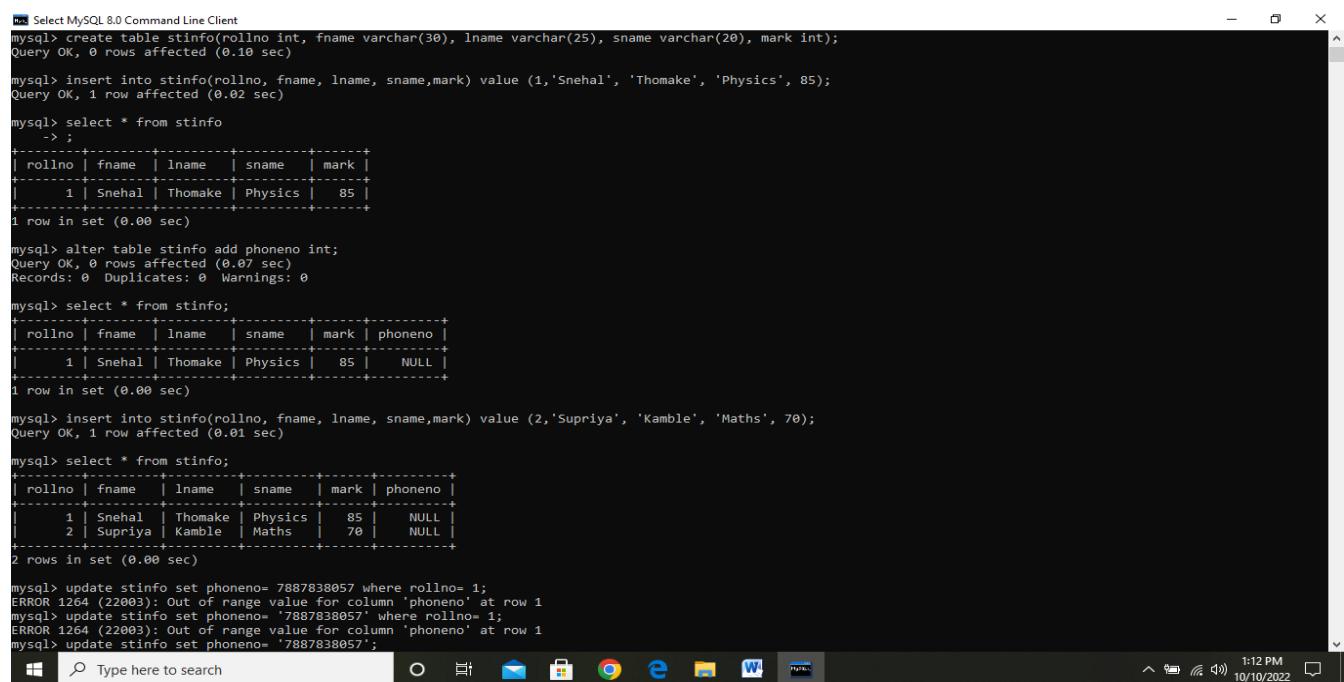
mysql> select * from school;
+-----+-----+-----+-----+
| SchoolID | School_Name | students_no | Teachers_no | classroom_no | EmailID |
+-----+-----+-----+-----+
| 1 | girls Govt School | 1000 | 80 | 12 | abcd |
| 2 | Boys School | 1500 | 90 | 15 | efgg |
| 3 | English medium School | 20050 | 85 | 19 | bdcj |
| 4 | Marathi medium School | 900 | 70 | 10 | khjb |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> TRUNCATE TABLE Categories;
ERROR 1146 (42S02): Table 'world.categories' doesn't exist
mysql> truncate table school;
Query OK, 0 rows affected (0.04 sec)

mysql> select * from school;
Empty set (0.01 sec)

mysql>
```

INSERT = The INSERT INTO statement is used to insert new records in a table.



```
Select MySQL 8.0 Command Line Client

mysql> create table stinfo(rollno int, fname varchar(30), lname varchar(25), sname varchar(20), mark int);
Query OK, 0 rows affected (0.10 sec)

mysql> insert into stinfo(rollno, fname, lname, sname,mark) value (1,'Snehal', 'Thomake', 'Physics', 85);
Query OK, 1 row affected (0.02 sec)

mysql> select * from stinfo
-> ;
+-----+-----+-----+-----+
| rollno | fname | lname | sname | mark |
+-----+-----+-----+-----+
| 1 | Snehal | Thomake | Physics | 85 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> alter table stinfo add phoneno int;
Query OK, 0 rows affected (0.07 sec)
Records: 0  Duplicates: 0  Warnings: 0

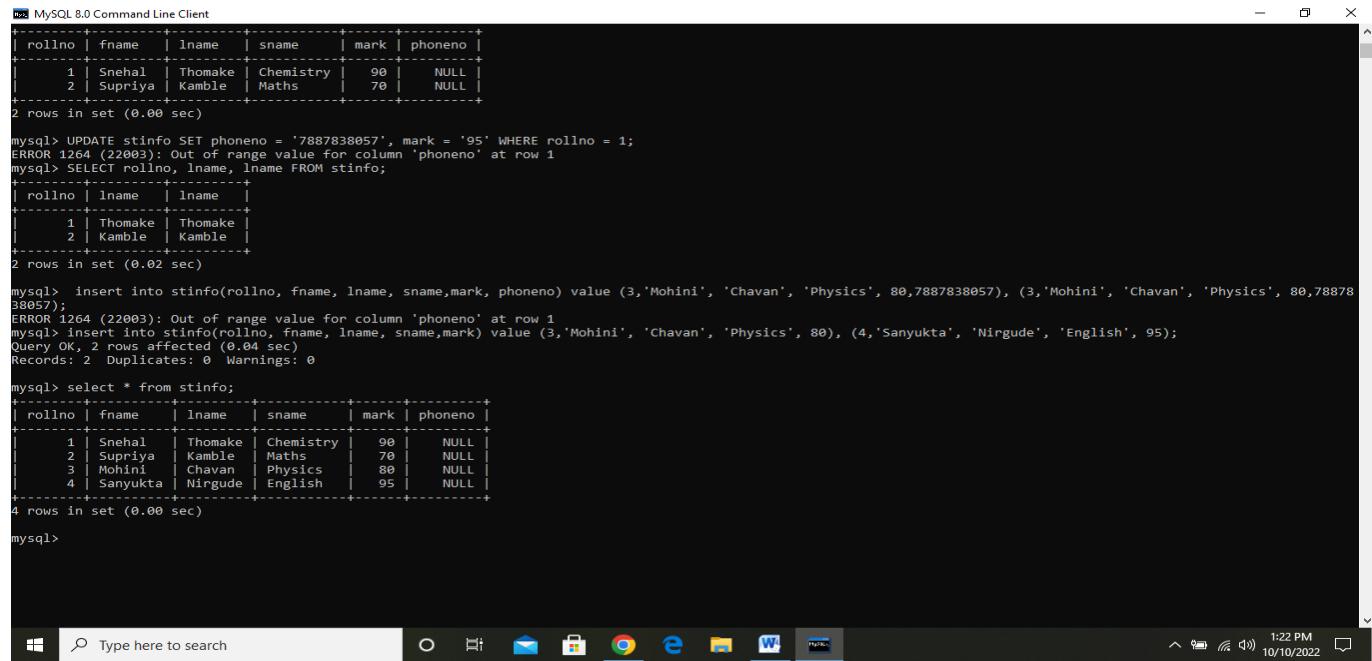
mysql> select * from stinfo;
+-----+-----+-----+-----+-----+
| rollno | fname | lname | sname | mark | phoneno |
+-----+-----+-----+-----+-----+
| 1 | Snehal | Thomake | Physics | 85 | NULL |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> insert into stinfo(rollno, fname, lname, sname,mark) value (2,'Supriya', 'Kamble', 'Maths', 70);
Query OK, 1 row affected (0.01 sec)

mysql> select * from stinfo;
+-----+-----+-----+-----+-----+
| rollno | fname | lname | sname | mark | phoneno |
+-----+-----+-----+-----+
| 1 | Snehal | Thomake | Physics | 85 | NULL |
| 2 | Supriya | Kamble | Maths | 70 | NULL |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> update stinfo set phoneno= 7887838057 where rollno= 1;
ERROR 1264 (22003): Out of range value for column 'phoneno' at row 1
mysql> update stinfo set phoneno= '7887838057' where rollno= 1;
ERROR 1264 (22003): Out of range value for column 'phoneno' at row 1
mysql> update stinfo set phoneno= '7887838057';
```

Insert multiple rows= Sometimes you may need to insert multiple rows of data in MySQL. MySQL allows you to enter multiple rows of information with a single query. In this article, we will look at how to insert multiple rows in MySQL



The screenshot shows a Windows desktop with the MySQL 8.0 Command Line Client window open. The command line shows the following sequence of operations:

```
mysql> UPDATE stinfo SET phoneno = '7887838057', mark = '95' WHERE rollno = 1;
ERROR 1264 (22003): Out of range value for column 'phoneno' at row 1
mysql> SELECT rollno, lname, lname FROM stinfo;
+-----+-----+
| rollno | lname | lname |
+-----+-----+
| 1      | Thomake | Thomake |
| 2      | Kamble  | Kamble  |
+-----+-----+
2 rows in set (0.02 sec)

mysql> insert into stinfo(rollno, fname, lname, sname,mark, phoneno) value (3,'Mohini', 'Chavan', 'Physics', 80,7887838057), (3,'Mohini', 'Chavan', 'Physics', 80,7887838057);
ERROR 1264 (22003): Out of range value for column 'phoneno' at row 1
mysql> insert into stinfo(rollno, fname, lname, sname,mark) value (3,'Mohini', 'Chavan', 'Physics', 80), (4,'Sanyukta', 'Nirgude', 'English', 95);
Query OK, 2 rows affected (0.04 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql> select * from stinfo;
+-----+-----+-----+-----+-----+-----+
| rollno | fname | lname | sname | mark | phoneno |
+-----+-----+-----+-----+-----+-----+
| 1      | Snehali | Thomake | Chemistry | 90 | NULL |
| 2      | Supriya | Kamble  | Maths    | 70 | NULL |
| 3      | Mohini  | Chavan  | Physics   | 80 | NULL |
| 4      | Sanyukta | Nirgude | English   | 95 | NULL |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

RENAME

In some situations, database administrators and users want to change the name of the table in the SQL database because they want to give a more relevant name to the table.

Syntax

```
rename old_table_name to new_table_name;
```

COPY

MySQL copy or clone database is a feature that allows us to create a **duplicate copy of an existing database**, including the table structure, indexes, constraints, default values, etc.

Syntax

```
Select * into new_table_name;
```

Temporary tables

Temporary tables can be created at run-time and can do all kinds of operations that a normal table can do. These temporary tables are created inside tempdb database.

There are two types of temp tables based on the behavior and scope.

1. Local Temp Variable= Local temp tables are only available at current connection time. It is automatically deleted when user disconnects from instances.

Syntax

CREATE TABLE #local temp table (

User id **int**,
Username **varchar** (50),
User address **varchar** (150)
)

2. Global Temp Variable = Global temp tables name starts with double hash (##). Once this table is created, it is like a permanent table. It is always ready for all users and not deleted until the total connection is withdrawn.

Syntax

CREATE TABLE ##new global temp table (

User id **int**,
User **name varchar** (50),
User address **varchar** (150)
)

Date =4/10/2022 and 5/10/2022 is DASARA FUNCTION FOR OFF DAYS

Date= 6/10/2022 Friday

DISTINCT

SQL **DISTINCT** clause is used to remove the duplicates columns from the result set.

- SELECT DISTINCT returns only distinct (**different**) values.
- DISTINCT eliminates duplicate records from the table.
- DISTINCT can be used with aggregates: **COUNT, AVG, MAX**, etc.
- DISTINCT operates on a single column.
- Multiple columns are not supported for DISTINCT

Syntax:

SELECT DISTINCT expressions

FROM tables

[WHERE conditions];

Example and perform →

The screenshot shows a terminal window for MySQL 8.0 Command Line Client. The user has performed several operations:

- Selected all columns from the 'school' table, resulting in 4 rows.
- Inserted four new rows into the 'school' table with SchoolID, School Name, students_no, Teachers_no, classroom_no, and EmailID values corresponding to different schools.
- Selected all columns from the 'school' table again, showing the updated data with 4 rows.
- Used the DISTINCT keyword to select unique values for 'students_no' from the 'school' table, resulting in 3 distinct values (1000, 1200, 2000).

```
MySQL 8.0 Command Line Client
+-----+
| 2 | Supriya | Kamble | Maths | 70 | NULL |
| 3 | Mohini | Chavan | Physics | 80 | NULL |
| 4 | Sanyukta | Nirgude | English | 95 | NULL |
+-----+
4 rows in set (0.01 sec)

mysql> select * from school;
Empty set (0.01 sec)

mysql> insert into school(SchoolID, School_Name, students_no, Teachers_no, classroom_no, EmailID) value (1, 'Boys Town Public School', 1000, 80, 12, 'btps'), (2, 'Guru Govind Singh Public School', 1200, 35, 15, 'ggps'), (3, 'Delhi Public School', 2000, 30, 10, 'dps1'), (4, 'Ashoka Universal School', 1000, 20, 40, 'aus1');

Query OK, 4 rows affected (0.07 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> select * from school;
+-----+
| SchoolID | School_Name | students_no | Teachers_no | classroom_no | EmailID |
+-----+
| 1 | Boys Town Public School | 1000 | 80 | 12 | btps |
| 2 | Guru Govind Singh Public School | 1200 | 35 | 15 | ggps |
| 3 | Delhi Public School | 2000 | 30 | 10 | dps1 |
| 4 | Ashoka Universal School | 1000 | 20 | 40 | aus1 |
+-----+
4 rows in set (0.00 sec)

mysql> SELECT DISTINCT ~CROM school ORDER BY students_no;
mysql>  SELECT DISTINCT students_no from school ORDER BY students_no;
+-----+
| students_no |
+-----+
| 1000 |
| 1200 |
| 2000 |
+-----+
3 rows in set (0.01 sec)

mysql>
```

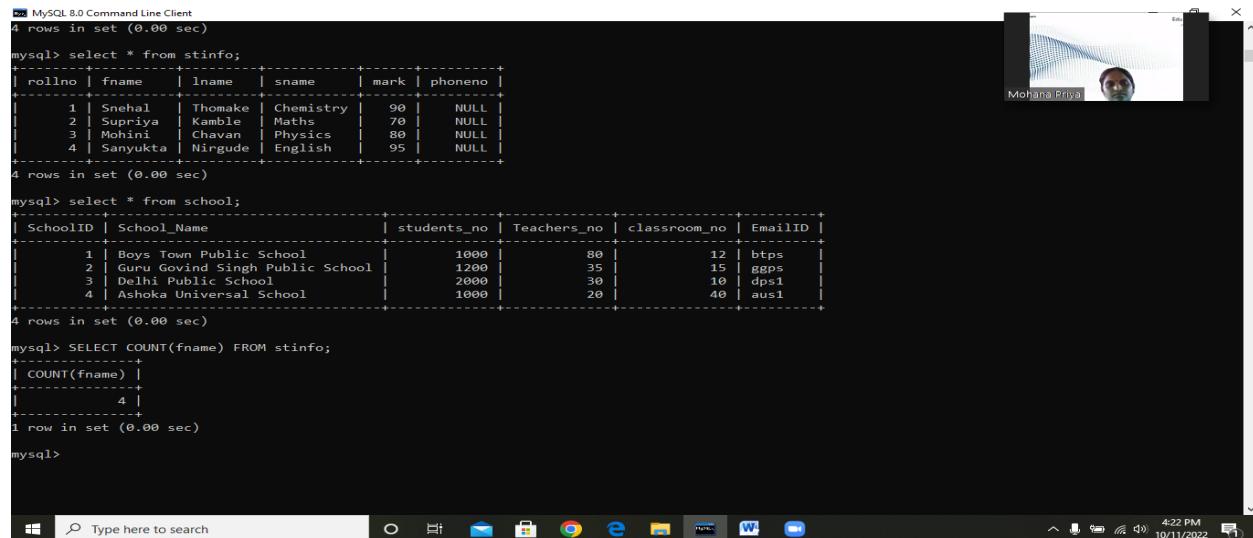
COUNT

The COUNT is a function in Structured Query Language that shows the number of records from the table in the result. In SQL, it is always used in the SELECT query.

Syntax

SELECT COUNT(Name_of_Column) FROM Name_of_Table;

Example and perform →



MySQL 8.0 Command Line Client
4 rows in set (0.00 sec)
mysql> select * from stinfo;
+-----+-----+-----+-----+-----+
| rollno | fname | lname | sname | mark | phoneno |
+-----+-----+-----+-----+-----+
| 1 | Snehal | Thomake | Chemistry | 99 | NULL |
| 2 | Supriya | Kamble | Maths | 70 | NULL |
| 3 | Mohini | Chavan | Physics | 80 | NULL |
| 4 | Sanyukta | Ningude | English | 95 | NULL |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from school;
+-----+-----+-----+-----+-----+
| SchoolID | School_Name | students_no | Teachers_no | classroom_no | EmailID |
+-----+-----+-----+-----+-----+
| 1 | Boys Town Public School | 1000 | 80 | 12 | btps |
| 2 | Guru Govind Singh Public School | 1200 | 35 | 15 | ggps |
| 3 | Delhi Public School | 2000 | 30 | 10 | dps1 |
| 4 | Ashoka Universal School | 1000 | 20 | 40 | aus1 |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> SELECT COUNT(fname) FROM stinfo;
+-----+
| COUNT(fname) |
+-----+
| 4 |
+-----+
1 row in set (0.00 sec)

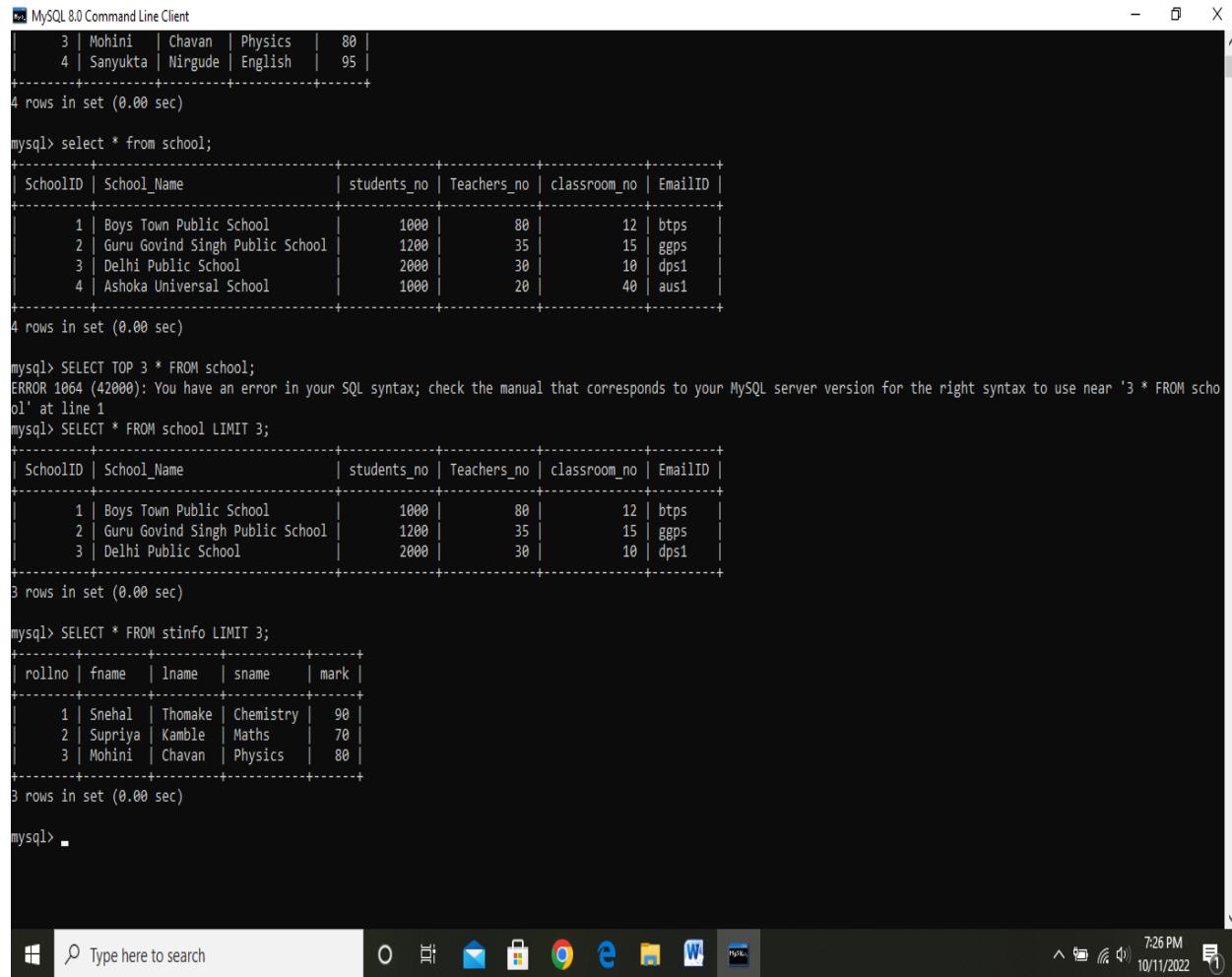
mysql>

TOP

The **SELECT TOP** statement in SQL shows the limited number of records or rows from the database table. The TOP clause in the statement specifies how many rows are returned.

Syntax

```
SELECT column_name(s)
FROM table_name
WHERE condition
LIMIT number;
```



The screenshot shows a terminal window for MySQL 8.0 Command Line Client. It displays several SQL queries and their results:

- First query: `SELECT * from marks;` Returns 4 rows:

| | | | | |
|---|----------|---------|---------|----|
| 3 | Mohini | Chavan | Physics | 80 |
| 4 | Sanyukta | Nirgude | English | 95 |

4 rows in set (0.00 sec)
- Second query: `select * from school;` Returns 4 rows:

| SchoolID | School_Name | students_no | Teachers_no | classroom_no | EmailID |
|----------|---------------------------------|-------------|-------------|--------------|---------|
| 1 | Boys Town Public School | 1000 | 80 | 12 | btps |
| 2 | Guru Govind Singh Public School | 1200 | 35 | 15 | ggps |
| 3 | Delhi Public School | 2000 | 30 | 10 | dps1 |
| 4 | Ashoka Universal School | 1000 | 20 | 40 | aus1 |

4 rows in set (0.00 sec)
- Third query: `SELECT TOP 3 * FROM school;` Returns 3 rows (same result as the second query):

| SchoolID | School_Name | students_no | Teachers_no | classroom_no | EmailID |
|----------|---------------------------------|-------------|-------------|--------------|---------|
| 1 | Boys Town Public School | 1000 | 80 | 12 | btps |
| 2 | Guru Govind Singh Public School | 1200 | 35 | 15 | ggps |
| 3 | Delhi Public School | 2000 | 30 | 10 | dps1 |

3 rows in set (0.00 sec)
- Fourth query: `SELECT * FROM stinfo LIMIT 3;` Returns 3 rows:

| rollno | fname | lname | sname | mark |
|--------|---------|---------|-----------|------|
| 1 | Snehal | Thomake | Chemistry | 90 |
| 2 | Supriya | Kamble | Maths | 70 |
| 3 | Mohini | Chavan | Physics | 80 |

3 rows in set (0.00 sec)
- Fifth query: `mysql> .`

First

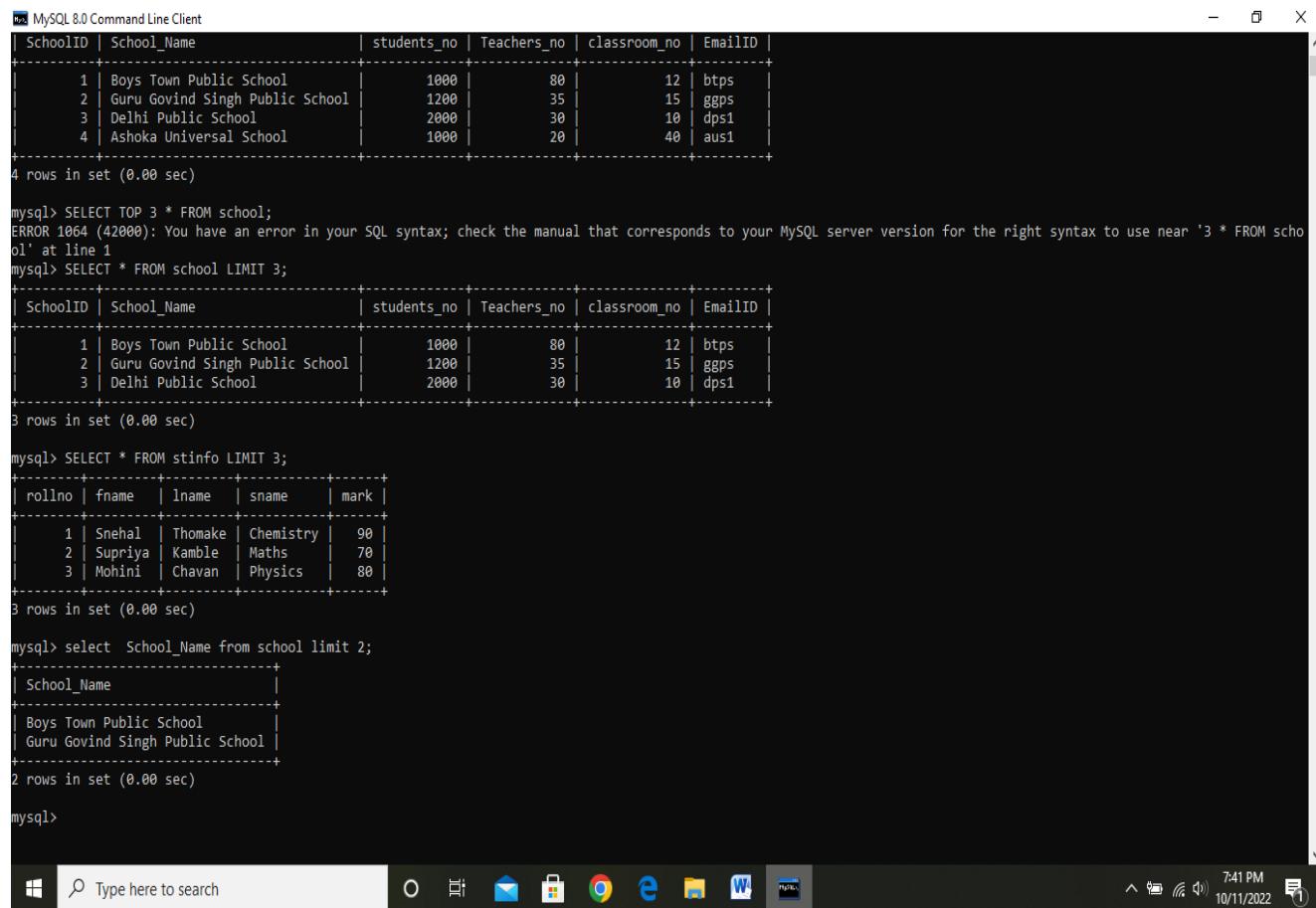
The MySQL first function is used to return the first value of the selected column. Here, we use limit clause to select first record or more.

Syntax:

```
SELECT column_name
```

```
FROM table_name
```

```
LIMIT 1;
```



The screenshot shows a Windows desktop environment with the MySQL 8.0 Command Line Client window open. The client displays several SQL queries and their results. The queries include:

- `SELECT * FROM school;` (Shows 4 rows of data from the school table)
- `SELECT TOP 3 * FROM school;` (Shows an error message: "ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '3 * FROM school' at line 1")
- `SELECT * FROM school LIMIT 3;` (Shows 3 rows of data from the school table)
- `SELECT * FROM stinfo LIMIT 3;` (Shows 3 rows of data from the stinfo table)
- `select School_Name from school limit 2;` (Shows 2 rows of data from the school table)

The results are presented as tabular data with columns labeled SchoolID, School_Name, students_no, Teachers_no, classroom_no, EmailID, fname, lname, sname, and mark.

ROW NUMBER

The ROW_NUMBER() function in MySQL is used to returns the **sequential number** for each row within its partition. It is a kind of window function. The row number starts from 1 to the number of rows present in the partition.

Syntax

ROW_NUMBER() OVER (<partition_definition> <order_definition>)

The screenshot shows a Windows desktop with the MySQL 8.0 Command Line Client window open. The client displays three SQL queries:

```
mysql> select School_Name from school limit 2;
+-----+
| School_Name |
+-----+
| Boys Town Public School |
| Guru Govind Singh Public School |
2 rows in set (0.00 sec)

mysql> SELECT *, ROW_NUMBER() OVER(PARTITION BY sname) AS row_num FROM stinfo;
+-----+-----+-----+-----+-----+
| rollno | fname | lname | sname | mark | row_num |
+-----+-----+-----+-----+-----+
| 1 | Snehal | Thomake | Chemistry | 90 | 1 |
| 4 | Sanyukta | Nirgude | English | 95 | 1 |
| 2 | Supriya | Kamble | Maths | 70 | 1 |
| 3 | Mohini | Chavan | Physics | 80 | 1 |
+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)

mysql> SELECT *,ROW_NUMBER() OVER(PARTITION BY Teachers_no) AS row_num FROM school;
+-----+-----+-----+-----+-----+-----+
| SchoolID | School_Name | students_no | Teachers_no | classroom_no | EmailID | row_num |
+-----+-----+-----+-----+-----+-----+
| 4 | Ashoka Universal School | 1000 | 20 | 40 | aus1 | 1 |
| 3 | Delhi Public School | 2000 | 30 | 10 | dps1 | 1 |
| 2 | Guru Govind Singh Public School | 1200 | 35 | 15 | ggps | 1 |
| 1 | Boys Town Public School | 1000 | 80 | 12 | btps | 1 |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

The client interface includes a search bar at the bottom left and a taskbar with various icons at the bottom right.

MySQL date/time

MySQL date/time functions are used to manipulate temporal values.

```
MySQL 8.0 Command Line Client
+-----+-----+-----+-----+
| 1 | Snehal | Thomake | Chemistry | 90 | 1 |
| 4 | Sanyukta | Nirgude | English | 95 | 1 |
| 2 | Supriya | Kamble | Maths | 70 | 1 |
| 3 | Mohini | Chavan | Physics | 80 | 1 |
+-----+-----+-----+-----+
4 rows in set (0.01 sec)

mysql> SELECT *,ROW_NUMBER() OVER(PARTITION BY Teachers_no) AS row_num FROM school;
+-----+-----+-----+-----+-----+-----+
| SchoolID | School_Name | students_no | Teachers_no | classroom_no | EmailID | row_num |
+-----+-----+-----+-----+-----+-----+
| 4 | Ashoka Universal School | 1000 | 28 | 40 | aus1 | 1 |
| 3 | Delhi Public School | 2000 | 30 | 10 | dps1 | 1 |
| 2 | Guru Govind Singh Public School | 1200 | 35 | 15 | ggps | 1 |
| 1 | Boys Town Public School | 1000 | 88 | 12 | btps | 1 |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select adddate('1999-09-28',30);
+-----+
| adddate('1999-09-28',30) |
+-----+
| 1999-10-28 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT DATE('2022-10-12 01:02:03');
+-----+
| DATE('2022-10-12 01:02:03') |
+-----+
| 2022-10-12 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT ADDDATE('1997-01-02', 31);
+-----+
| ADDDATE('1997-01-02', 31) |
+-----+
| 1997-02-02 |
+-----+
1 row in set (0.00 sec)

mysql>
```

SQL GROUP BY

The SQL **GROUP BY** clause is used in collaboration with the SELECT statement to arrange identical data into groups. This GROUP BY clause follows the WHERE clause in a SELECT statement and precedes the ORDER BY clause.

Syntax

```
SELECT column1, column2  
FROM table_name  
WHERE [ conditions ]  
GROUP BY column1, column2  
ORDER BY column1, column2
```

SUM

```
MySQL 8.0 Command Line Client  
mysql> select * from stinfo;  
+-----+-----+-----+-----+  
| rollno | fname | lname | sname | mark |  
+-----+-----+-----+-----+  
| 1 | Snehal | Thomake | Chemistry | 90 |  
| 2 | Supriya | Kamble | Maths | 70 |  
| 3 | Mohini | Chavan | Physics | 80 |  
| 4 | Sanyukta | Nirgude | English | 95 |  
+-----+-----+-----+-----+  
4 rows in set (0.00 sec)  
  
mysql> select * from school;  
+-----+-----+-----+-----+-----+  
| SchoolID | School_Name | students_no | Teachers_no | classroom_no | EmailID |  
+-----+-----+-----+-----+-----+  
| 1 | Boys Town Public School | 1000 | 80 | 12 | btps |  
| 2 | Guru Govind Singh Public School | 1200 | 35 | 15 | ggps |  
| 3 | Delhi Public School | 2000 | 30 | 10 | dps1 |  
| 4 | Ashoka Universal School | 1000 | 20 | 40 | aus1 |  
+-----+-----+-----+-----+-----+  
4 rows in set (0.00 sec)  
  
mysql> SELECT School_Name, SUM(classroom_no) FROM school GROUP BY School_Name;  
+-----+-----+  
| School_Name | SUM(classroom_no) |  
+-----+-----+  
| Boys Town Public School | 12 |  
| Guru Govind Singh Public School | 15 |  
| Delhi Public School | 10 |  
| Ashoka Universal School | 40 |  
+-----+-----+  
4 rows in set (0.00 sec)  
  
mysql>
```

COUNT

```
MySQL 8.0 Command Line Client
mysql> SELECT School_Name, SUM(classroom_no) FROM school GROUP BY School_Name;
+-----+-----+
| School_Name | SUM(classroom_no) |
+-----+-----+
| Boys Town Public School | 12 |
| Guru Govind Singh Public School | 15 |
| Delhi Public School | 10 |
| Ashoka Universal School | 40 |
+-----+-----+
4 rows in set (0.00 sec)

mysql> SELECT EmailID, COUNT(*) FROM school GROUP BY EmailID;
+-----+-----+
| EmailID | COUNT(*) |
+-----+-----+
| btps | 1 |
| ggps | 1 |
| dps1 | 1 |
| aus1 | 1 |
+-----+-----+
4 rows in set (0.00 sec)

mysql> SELECT sname, COUNT(*) FROM stinfo GROUP BY sname;
+-----+-----+
| sname | COUNT(*) |
+-----+-----+
| Chemistry | 1 |
| Maths | 1 |
| Physics | 1 |
| English | 1 |
+-----+-----+
4 rows in set (0.00 sec)

mysql> SELECT fname, lname, COUNT(*) FROM stinfo GROUP BY fname, lname;
+-----+-----+-----+
| fname | lname | COUNT(*) |
+-----+-----+-----+
| Snehal | Thomake | 1 |
| Supriya | Kamble | 1 |
| Mohini | Chavan | 1 |
| Sanyukta | Nirgude | 1 |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

MIN function

```
MySQL 8.0 Command Line Client
mysql> select * from school;
+-----+-----+-----+-----+-----+
| SchoolID | School_Name | students_no | Teachers_no | classroom_no | EmailID |
+-----+-----+-----+-----+-----+
| 1 | Boys Town Public School | 1000 | 80 | 12 | btps |
| 2 | Guru Govind Singh Public School | 1200 | 35 | 15 | ggps |
| 3 | Delhi Public School | 2000 | 30 | 10 | dps1 |
| 4 | Ashoka Universal School | 1000 | 20 | 40 | aus1 |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from stinfo;
+-----+-----+-----+-----+
| rollno | fname | lname | sname | mark |
+-----+-----+-----+-----+
| 1 | Snehal | Thomake | Chemistry | 90 |
| 2 | Supriya | Kamble | Maths | 70 |
| 3 | Mohini | Chavan | Physics | 80 |
| 4 | Sanyukta | Nirgude | English | 95 |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select School_Name, MIN(Teachers_no) AS 'Minimum Teachers' FROM school GROUP BY School_Name;
+-----+-----+
| School Name | Minimum Teachers |
+-----+-----+
| Boys Town Public School | 80 |
| Guru Govind Singh Public School | 35 |
| Delhi Public School | 30 |
| Ashoka Universal School | 20 |
+-----+-----+
4 rows in set (0.00 sec)

mysql> select msrk, MIN(mark) AS 'Minimum marks' FROM stinfo GROUP BY fname;
ERROR 1054 (42S22): Unknown column 'msrk' in 'field list'
mysql> select mark, MIN(mark) AS 'Minimum marks' FROM stinfo GROUP BY fname;
+-----+-----+
| mark | Minimum marks |
+-----+-----+
| 90 | 90 |
+-----+-----+
4 rows in set (0.00 sec)
```

MAX function

```
MySQL 8.0 Command Line Client
mysql> select mark, MIN(mark) AS 'Minimum marks' FROM stinfo GROUP BY fname;
+-----+-----+
| mark | Minimum marks |
+-----+-----+
| 90   |      90 |
| 70   |      70 |
| 80   |      80 |
| 95   |      95 |
+-----+-----+
4 rows in set (0.00 sec)

mysql> select fname, MIN(mark) AS 'Minimum marks' FROM stinfo GROUP BY fname;
+-----+-----+
| fname | Minimum marks |
+-----+-----+
| Snehal |      90 |
| Supriya |     70 |
| Mohini |     80 |
| Sanyukta |    95 |
+-----+-----+
4 rows in set (0.00 sec)

mysql> select fname, MAX(mark) AS 'Minimum marks' FROM stinfo GROUP BY fname;
+-----+-----+
| fname | Minimum marks |
+-----+-----+
| Snehal |      90 |
| Supriya |     70 |
| Mohini |     80 |
| Sanyukta |    95 |
+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

AVG function

```
MySQL 8.0 Command Line Client
mysql> select fname, MIN(mark) AS 'Minimum marks' FROM stinfo GROUP BY fname;
+-----+-----+
| fname | Minimum marks |
+-----+-----+
| Snehal |      90 |
| Supriya |     70 |
| Mohini |     80 |
| Sanyukta |    95 |
+-----+-----+
4 rows in set (0.00 sec)

mysql> select fname, MAX(mark) AS 'Minimum marks' FROM stinfo GROUP BY fname;
+-----+-----+
| fname | Minimum marks |
+-----+-----+
| Snehal |      90 |
| Supriya |     70 |
| Mohini |     80 |
| Sanyukta |    95 |
+-----+-----+
4 rows in set (0.00 sec)

mysql> SELECT fname, AVG(mark) AS 'Minimum marks' FROM stinfo GROUP BY fname;
+-----+-----+
| fname | Minimum marks |
+-----+-----+
| Snehal |    80.0000 |
| Supriya |    70.0000 |
| Mohini |    80.0000 |
| Sanyukta |    95.0000 |
+-----+-----+
4 rows in set (0.00 sec)

mysql> SELECT School_Name, AVG(Teachers_no) AS 'AVG Teachers' FROM school GROUP BY School_Name;
+-----+-----+
| School Name | AVG Teachers |
+-----+-----+
| Boys Town Public School |    80.0000 |
| Guru Govind Singh Public School |    35.0000 |
| Delhi Public School |    30.0000 |
| Ashoka Universal School |    20.0000 |
+-----+-----+
4 rows in set (0.00 sec)
```

SQL LOGICAL OPERATORS

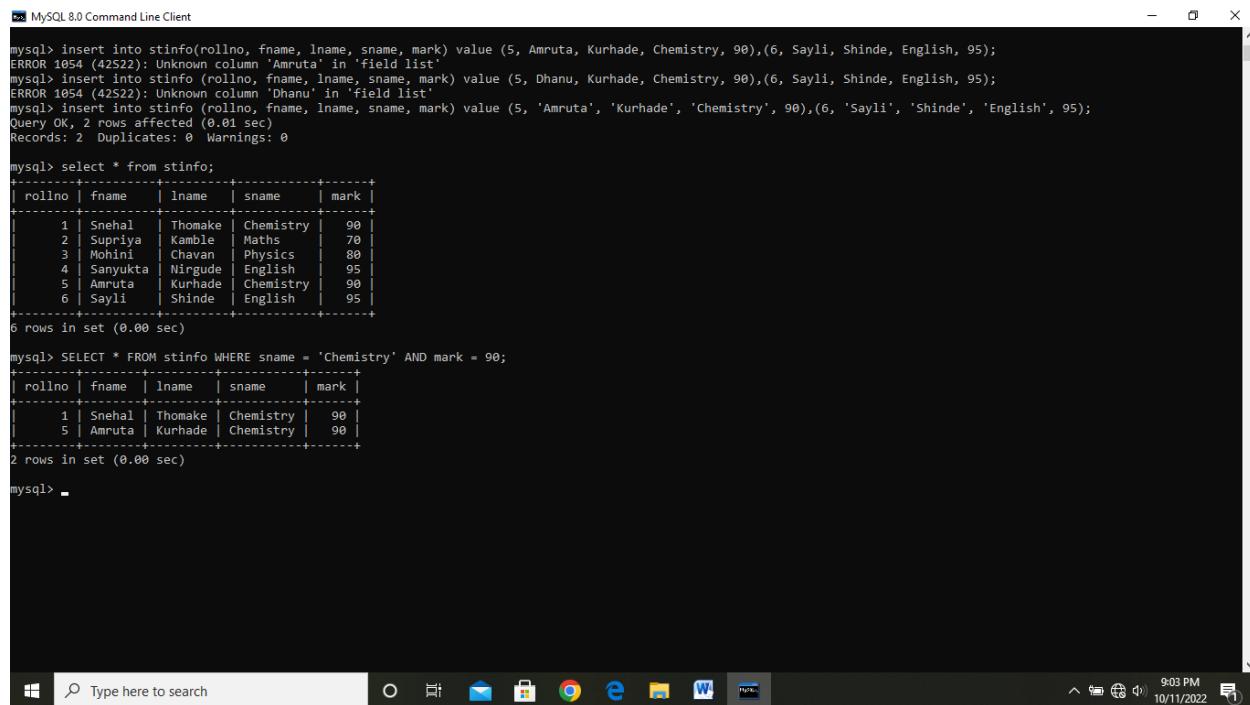
The Logical Operator is nothing but which returns the result in one form, i.e., either it will display the query is true, or the query is false. The results displayed to combine or merge more than one true or false data.

The Logical Operators in SQL are as follows:

1. SQL AND OPERATOR
2. SQL OR OPERATOR
3. SQL NOT OPERATOR
4. SQL BETWEEN OPERATOR
5. SQL IN OPERATOR
6. SQL LIKE OPERATOR

1. SQL AND Operator

The SQL AND operator is used with the where clause in the SQL Query. AND operator in SQL returns only those records which satisfy both the conditions in the SQL query.



MySQL 8.0 Command Line Client

```
mysql> insert into stinfo(rollno, fname, lname, sname, mark) value (5, Amruta, Kurhade, Chemistry, 90),(6, Sayli, Shinde, English, 95);
ERROR 1054 (42S22): Unknown column 'Amruta' in 'field list'
mysql> insert into stinfo (rollno, fname, lname, sname, mark) value (5, Dhanu, Kurhade, Chemistry, 90),(6, Sayli, Shinde, English, 95);
ERROR 1054 (42S22): Unknown column 'Dhanu' in 'field list'
mysql> insert into stinfo (rollno, fname, lname, sname, mark) value (5, 'Amruta', 'Kurhade', 'Chemistry', 90),(6, 'Sayli', 'Shinde', 'English', 95);
Query OK, 2 rows affected (0.01 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql> select * from stinfo;
+-----+-----+-----+-----+
| rollno | fname | lname | sname | mark |
+-----+-----+-----+-----+
| 1 | Snehal | Thomake | Chemistry | 90 |
| 2 | Supriya | Kamble | Maths | 70 |
| 3 | Mohini | Chavan | Physics | 80 |
| 4 | Sanyukta | Nirgude | English | 95 |
| 5 | Amruta | Kurhade | Chemistry | 90 |
| 6 | Sayli | Shinde | English | 95 |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> SELECT * FROM stinfo WHERE sname = 'Chemistry' AND mark = 90;
+-----+-----+-----+-----+
| rollno | fname | lname | sname | mark |
+-----+-----+-----+-----+
| 1 | Snehal | Thomake | Chemistry | 90 |
| 5 | Amruta | Kurhade | Chemistry | 90 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

SQL BETWEEN Operator

This operator displays the records which fall between the given ranges in the SQL query. The results of the BETWEEN operator include begin and end values of the given range.

```
MySQL 8.0 Command Line Client
mysql> insert into stinfo (rollno, fname, lname, sname, mark) value (5, Dhanu, Kurhade, Chemistry, 90),(6, Sayli, Shinde, English, 95);
ERROR 1054 (42S22): Unknown column 'Dhanu' in 'field list'
mysql> insert into stinfo (rollno, fname, lname, sname, mark) value (5, 'Amruta', 'Kurhade', 'Chemistry', 90),(6, 'Sayli', 'Shinde', 'English', 95);
Query OK, 2 rows affected (0.01 sec)
Records: 2 Duplicates: 0 Warnings: 0
mysql> select * from stinfo;
+-----+-----+-----+-----+-----+
| rollno | fname | lname | sname | mark |
+-----+-----+-----+-----+-----+
| 1 | Snehal | Thomake | Chemistry | 90 |
| 2 | Supriya | Kamble | Maths | 70 |
| 3 | Mohini | Chavan | Physics | 80 |
| 4 | Sanyukta | Nirgude | English | 95 |
| 5 | Amruta | Kurhade | Chemistry | 90 |
| 6 | Sayli | Shinde | English | 95 |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> SELECT * FROM stinfo WHERE sname = 'Chemistry' AND mark = 90;
+-----+-----+-----+-----+-----+
| rollno | fname | lname | sname | mark |
+-----+-----+-----+-----+-----+
| 1 | Snehal | Thomake | Chemistry | 90 |
| 5 | Amruta | Kurhade | Chemistry | 90 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> SELECT * FROM stinfo WHERE mark BETWEEN 90 AND 100;
+-----+-----+-----+-----+-----+
| rollno | fname | lname | sname | mark |
+-----+-----+-----+-----+-----+
| 1 | Snehal | Thomake | Chemistry | 90 |
| 4 | Sanyukta | Nirgude | English | 95 |
| 5 | Amruta | Kurhade | Chemistry | 90 |
| 6 | Sayli | Shinde | English | 95 |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

SQL OR Operator

The SQL OR operator is used with the where clause in an SQL Query. AND operator in SQL returns only those records that satisfy any of the conditions in the SQL query.

```
MySQL 8.0 Command Line Client
+-----+-----+-----+-----+-----+
| rollno | fname | lname | sname | mark |
+-----+-----+-----+-----+-----+
| 1 | Snehal | Thomake | Chemistry | 90 |
| 5 | Amruta | Kurhade | Chemistry | 90 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> SELECT * FROM stinfo WHERE mark BETWEEN 90 AND 100;
+-----+-----+-----+-----+-----+
| rollno | fname | lname | sname | mark |
+-----+-----+-----+-----+-----+
| 1 | Snehal | Thomake | Chemistry | 90 |
| 4 | Sanyukta | Nirgude | English | 95 |
| 5 | Amruta | Kurhade | Chemistry | 90 |
| 6 | Sayli | Shinde | English | 95 |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> SQL OR Operator
-> The SQL OR operator is used with the where clause in an SQL Query. AND operator in SQL returns only those records that satisfy any of the conditions in the SQL query
-> ;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'SQL OR Operat
or
The SQL OR operator is used with the where clause in an SQL Quer' at line 1
mysql> mysql>
mysql> SELECT * FROM stinfo WHERE sname = 'English' AND mark = 95;
+-----+-----+-----+-----+-----+
| rollno | fname | lname | sname | mark |
+-----+-----+-----+-----+-----+
| 4 | Sanyukta | Nirgude | English | 95 |
| 6 | Sayli | Shinde | English | 95 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> SELECT * FROM stinfo WHERE sname = 'English' OR mark = 95;
+-----+-----+-----+-----+-----+
| rollno | fname | lname | sname | mark |
+-----+-----+-----+-----+-----+
| 4 | Sanyukta | Nirgude | English | 95 |
| 6 | Sayli | Shinde | English | 95 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

SQL IN Operator

When we want to check for one or more than one value in a single SQL query, we use IN operator with the WHERE clause in a SELECT query.

```
MySQL 8.0 Command Line Client
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> SELECT * FROM stinfo WHERE sname = 'English' OR mark = 95;
+-----+-----+-----+-----+
| rollno | fname | lname | sname | mark |
+-----+-----+-----+-----+
| 4 | Sanyukta | Nirgude | English | 95 |
| 6 | Sayli | Shinde | English | 95 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> SELECT * FROM stinfo;
+-----+-----+-----+-----+
| rollno | fname | lname | sname | mark |
+-----+-----+-----+-----+
| 1 | Snehal | Thomake | Chemistry | 99 |
| 2 | Supriya | Kamble | Maths | 70 |
| 3 | Mohini | Chavan | Physics | 80 |
| 4 | Sanyukta | Nirgude | English | 95 |
| 5 | Amruta | Kurhade | Chemistry | 90 |
| 6 | Sayli | Shinde | English | 95 |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> SELECT * FROM stinfo WHERE sname IN ('Physics', 'English', 'Maths');
ERROR 1054 (42S22): Unknown column 'Physics' in 'where clause'
mysql> SELECT * FROM stinfo WHERE sname IN ('Physics', 'English', 'Maths');
+-----+-----+-----+-----+
| rollno | fname | lname | sname | mark |
+-----+-----+-----+-----+
| 2 | Supriya | Kamble | Maths | 70 |
| 3 | Mohini | Chavan | Physics | 80 |
| 4 | Sanyukta | Nirgude | English | 95 |
| 6 | Sayli | Shinde | English | 95 |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

SQL NOT Operator

NOT operator in SQL shows those records from the table where the criteria is not met. NOT operator is used with where clause in a SELECT query.

```
MySQL 8.0 Command Line Client
+-----+-----+-----+-----+
| 2 | Supriya | Kamble | Maths | 70 |
| 3 | Mohini | Chavan | Physics | 80 |
| 4 | Sanyukta | Nirgude | English | 95 |
| 6 | Sayli | Shinde | English | 95 |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from stinfo;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'from stinfo' at line 1
mysql> select * from stinfo;
+-----+-----+-----+-----+
| rollno | fname | lname | sname | mark |
+-----+-----+-----+-----+
| 1 | Snehal | Thomake | Chemistry | 99 |
| 2 | Supriya | Kamble | Maths | 70 |
| 3 | Mohini | Chavan | Physics | 80 |
| 4 | Sanyukta | Nirgude | English | 95 |
| 5 | Amruta | Kurhade | Chemistry | 90 |
| 6 | Sayli | Shinde | English | 95 |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> SELECT * FROM stinfo WHERE NOT sname = 'Chemistry';
+-----+-----+-----+-----+
| rollno | fname | lname | sname | mark |
+-----+-----+-----+-----+
| 2 | Supriya | Kamble | Maths | 70 |
| 3 | Mohini | Chavan | Physics | 80 |
| 4 | Sanyukta | Nirgude | English | 95 |
| 6 | Sayli | Shinde | English | 95 |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

Unique Key

A unique key in MySQL is a single field or combination of fields that ensure all values going to store into the column will be unique. It means a column cannot stores **duplicate values**. For example, the email addresses and roll numbers of students in the "student_info" table or contact number of employees in the "Employee" table should be unique.

Syntax

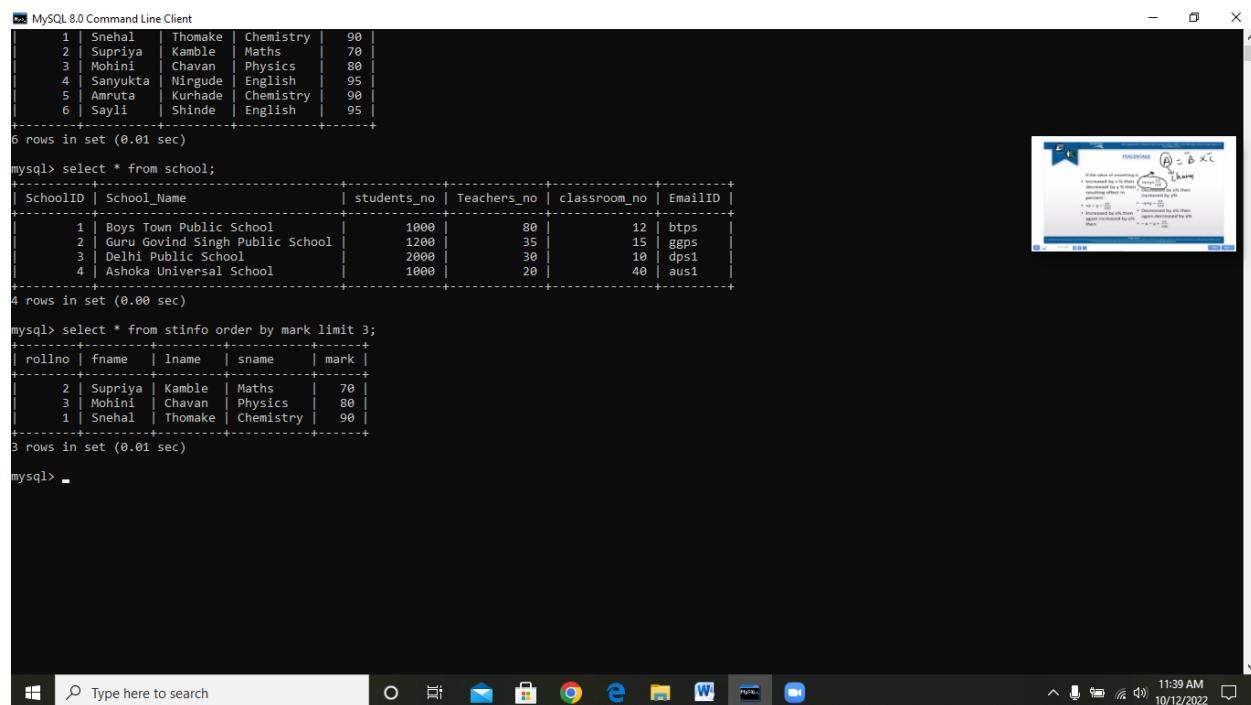
```
CREATE TABLE table_name(  
    col1 col_definition,  
    col2 col_definition,  
    ...  
    [CONSTRAINT constraint_name]  
    UNIQUE(column_name(s))  
);
```

Limit

MySQL Limit query is used to **restrict** the number of rows returns from the result set, rather than **fetching** the whole set in the MySQL database. The Limit clause works with the SELECT statement for returning the specified number of rows only. This query accepts only one or two arguments, and their values should be zero or any positive integer.

Syntax

1. **SELECT** column_list
2. **FROM** table_name
3. **LIMIT** offset, count;



```
MySQL 8.0 Command Line Client
+----+----+----+----+----+
| 1 | Snehal | Thomake | Chemistry | 90 |
| 2 | Supriya | Kamble | Maths      | 70 |
| 3 | Mohini  | Chavan   | Physics    | 80 |
| 4 | Sanyukta | Nirgude | English   | 95 |
| 5 | Amruta   | Kurhade | Chemistry | 90 |
| 6 | Sayli   | Shinde   | English   | 95 |
+----+----+----+----+----+
6 rows in set (0.01 sec)

mysql> select * from school;
+-----+-----+-----+-----+-----+
| SchoolID | School_Name | students_no | Teachers_no | classroom_no | EmailID |
+-----+-----+-----+-----+-----+
| 1 | Boys Town Public School | 1000 | 80 | 12 | btps |
| 2 | Guru Govind Singh Public School | 1200 | 35 | 15 | ggps |
| 3 | Delhi Public School | 2000 | 30 | 10 | dps1 |
| 4 | Ashoka Universal School | 1000 | 20 | 40 | aus1 |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from stinfo order by mark limit 3;
+----+----+----+----+
| rollno | fname | lname | sname | mark |
+----+----+----+----+
| 2 | Supriya | Kamble | Maths | 70 |
| 3 | Mohini  | Chavan | Physics | 80 |
| 1 | Snehal | Thomake | Chemistry | 98 |
+----+----+----+----+
3 rows in set (0.01 sec)

mysql>
```

Date=07/10/2022 **Friday**

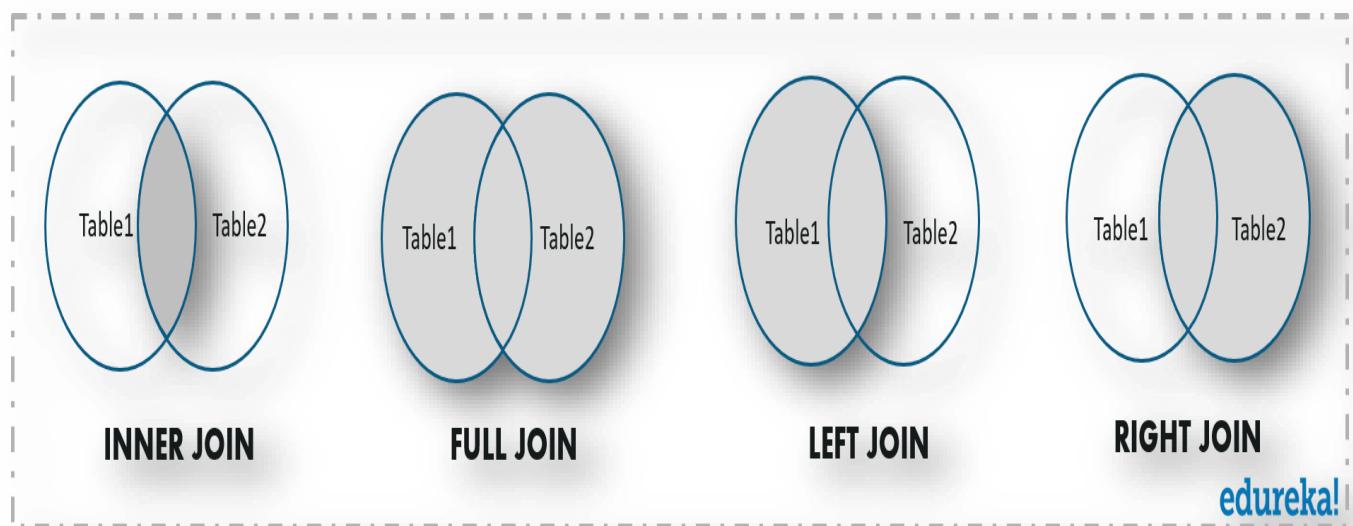
JOIN

JOINS in SQL are commands which are used to combine rows from two or more tables, based on a related column between those tables. There are predominantly used when a user is trying to extract data from tables which have one-to-many or many-to-many relationships between them.

Types of Join

- **INNER JOIN**
- **FULL JOIN**
- **LEFT JOIN**
- **RIGHT JOIN**

You can refer to the below image.



INNER JOIN

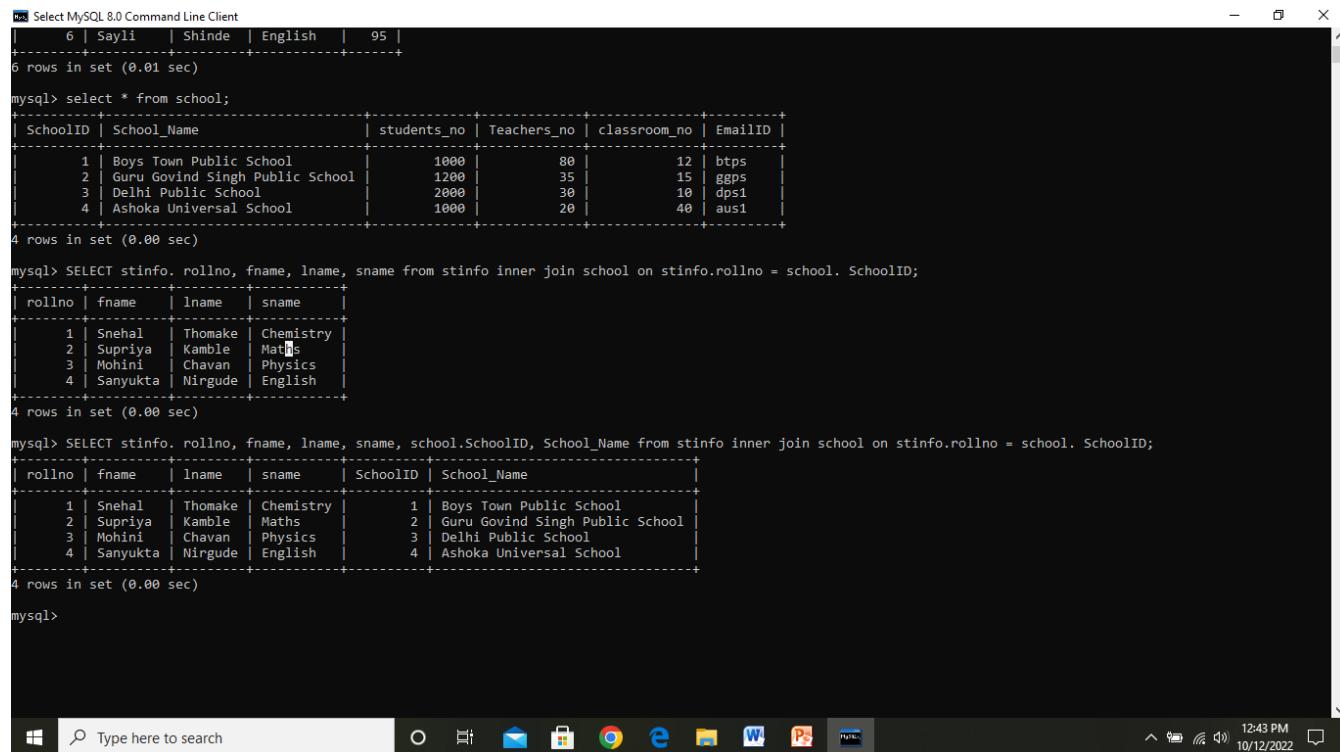
This type of join returns those records which have matching values in both tables. So, if you perform an INNER join operation between the Employee table and the Projects table, all the tuples which have matching values in both the tables will be given as output.

Syntax:

```
SELECT Table1.Column1,Table1.Column2,Table2.Column1,... FROM  
Table1
```

```
INNER JOIN Table2
```

```
ON Table1.MatchingColumnName = Table2.MatchingColumnName;
```



The screenshot shows a terminal window for MySQL 8.0 Command Line Client. It displays three SQL queries and their execution results:

```
mysql> Select MySQL 8.0 Command Line Client
+-----+-----+-----+
| 6 | Sayli | Shinde | English | 95 |
+-----+-----+-----+
6 rows in set (0.01 sec)

mysql> select * from school;
+-----+-----+-----+-----+-----+-----+
| SchoolID | School_Name | students_no | Teachers_no | classroom_no | EmailID |
+-----+-----+-----+-----+-----+-----+
| 1 | Boys Town Public School | 1000 | 80 | 12 | btps |
| 2 | Guru Govind Singh Public School | 1200 | 35 | 15 | ggps |
| 3 | Delhi Public School | 2000 | 30 | 10 | dps1 |
| 4 | Ashoka Universal School | 1000 | 20 | 40 | aus1 |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> SELECT stinfo.rollno, fname, lname, sname from stinfo inner join school on stinfo.rollno = school. SchoolID;
+-----+-----+-----+-----+
| rollno | fname | lname | sname |
+-----+-----+-----+-----+
| 1 | Snehal | Thomake | Chemistry |
| 2 | Supriya | Kamble | Maths |
| 3 | Mohini | Chavan | Physics |
| 4 | Sanyukta | Nirgude | English |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> SELECT stinfo.rollno, fname, lname, sname, school.SchoolID, School_Name from stinfo inner join school on stinfo.rollno = school. SchoolID;
+-----+-----+-----+-----+-----+-----+
| rollno | fname | lname | sname | SchoolID | School_Name |
+-----+-----+-----+-----+-----+-----+
| 1 | Snehal | Thomake | Chemistry | 1 | Boys Town Public School |
| 2 | Supriya | Kamble | Maths | 2 | Guru Govind Singh Public School |
| 3 | Mohini | Chavan | Physics | 3 | Delhi Public School |
| 4 | Sanyukta | Nirgude | English | 4 | Ashoka Universal School |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

FULL JOIN

FULL JOIN creates the result-set by combining results of both LEFT JOIN and RIGHT JOIN. The result-set will contain all the rows from both tables. For the rows for which there is no matching, the result-set will contain NULL values.

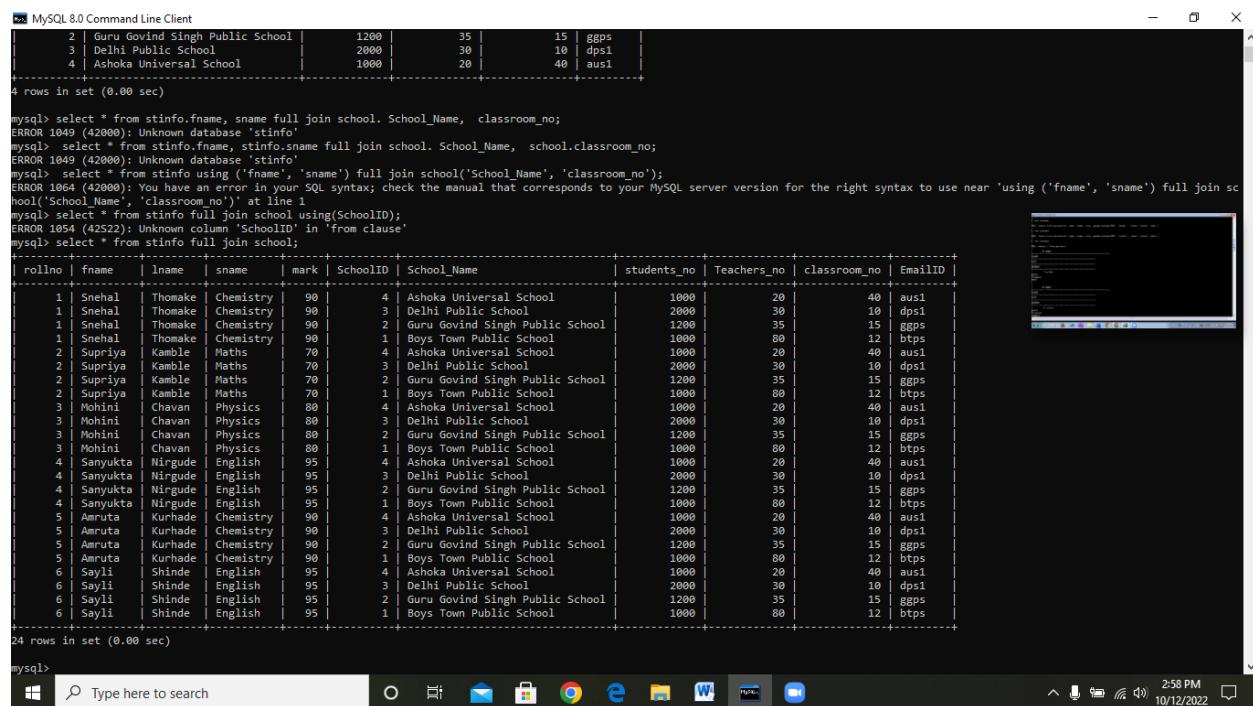
Syntax

SELECT table1.column1,table1.column2,table2.column1,....

FROM table1

FULL JOIN table2

ON table1.matching_column = table2.matching_column;



The screenshot shows the MySQL 8.0 Command Line Client interface. The command entered was:

```
mysql> select * from stinfo.fname, sname full join school. School_Name, classroom_no;
```

MySQL returned several error messages:

- ERROR 1049 (42000): Unknown database 'stinfo'
- ERROR 1049 (42000): Unknown database 'stinfo'
- ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'using ('fname', 'sname') full join sc' at line 1
- ERROR 1054 (42S22): Unknown column 'SchoolID' in 'from clause'
- ERROR 1054 (42S22): Unknown column 'SchoolID' in 'from clause'

After correcting the syntax, the query was executed successfully:

```
mysql> select * from stinfo full join school;
```

The resulting output is a combined result set from the stinfo and school tables. It includes columns from both tables and handles missing matches by returning NULL values where applicable. The output is as follows:

| rollno | fname | lname | sname | mark | SchoolID | School_Name | students_no | Teachers_no | classroom_no | EmailID |
|--------|----------|---------|-----------|------|----------|---------------------------------|-------------|-------------|--------------|---------|
| 1 | Snehal | Thomake | Chemistry | 90 | 4 | Ashoka Universal School | 1000 | 20 | 40 | aus1 |
| 1 | Snehal | Thomake | Chemistry | 90 | 3 | Delhi Public School | 2000 | 30 | 10 | dps1 |
| 1 | Snehal | Thomake | Chemistry | 90 | 2 | Guru Govind Singh Public School | 1200 | 35 | 15 | ggps |
| 1 | Snehal | Thomake | Chemistry | 90 | 1 | Boys Town Public School | 1000 | 80 | 12 | btps |
| 2 | Supriya | Kamble | Maths | 70 | 4 | Ashoka Universal School | 1000 | 20 | 40 | aus1 |
| 2 | Supriya | Kamble | Maths | 70 | 3 | Delhi Public School | 2000 | 30 | 10 | dps1 |
| 2 | Supriya | Kamble | Maths | 70 | 2 | Guru Govind Singh Public School | 1200 | 35 | 15 | ggps |
| 2 | Supriya | Kamble | Maths | 70 | 1 | Boys Town Public School | 1000 | 80 | 12 | btps |
| 3 | Mohini | Chavan | Physics | 80 | 4 | Ashoka Universal School | 1000 | 20 | 40 | aus1 |
| 3 | Mohini | Chavan | Physics | 80 | 3 | Delhi Public School | 2000 | 30 | 10 | dps1 |
| 3 | Mohini | Chavan | Physics | 80 | 2 | Guru Govind Singh Public School | 1200 | 35 | 15 | ggps |
| 3 | Mohini | Chavan | Physics | 80 | 1 | Boys Town Public School | 1000 | 80 | 12 | btps |
| 4 | Sanyukta | Mirgude | English | 95 | 4 | Ashoka Universal School | 1000 | 20 | 40 | aus1 |
| 4 | Sanyukta | Mirgude | English | 95 | 3 | Delhi Public School | 2000 | 30 | 10 | dps1 |
| 4 | Sanyukta | Mirgude | English | 95 | 2 | Guru Govind Singh Public School | 1200 | 35 | 15 | ggps |
| 4 | Sanyukta | Mirgude | English | 95 | 1 | Boys Town Public School | 1000 | 80 | 12 | btps |
| 5 | Amruta | Kunhade | Chemistry | 90 | 4 | Ashoka Universal School | 1000 | 20 | 40 | aus1 |
| 5 | Amruta | Kunhade | Chemistry | 90 | 3 | Delhi Public School | 2000 | 30 | 10 | dps1 |
| 5 | Amruta | Kunhade | Chemistry | 90 | 2 | Guru Govind Singh Public School | 1200 | 35 | 15 | ggps |
| 5 | Amruta | Kunhade | Chemistry | 90 | 1 | Boys Town Public School | 1000 | 80 | 12 | btps |
| 6 | Sayli | Shinde | English | 95 | 4 | Ashoka Universal School | 1000 | 20 | 40 | aus1 |
| 6 | Sayli | Shinde | English | 95 | 3 | Delhi Public School | 2000 | 30 | 10 | dps1 |
| 6 | Sayli | Shinde | English | 95 | 2 | Guru Govind Singh Public School | 1200 | 35 | 15 | ggps |
| 6 | Sayli | Shinde | English | 95 | 1 | Boys Town Public School | 1000 | 80 | 12 | btps |

24 rows in set (0.00 sec)

LEFT JOIN

This join returns all the rows of the table on the left side of the join and matches rows for the table on the right side of the join. For the rows for which there is no matching row on the right side, the result-set will contain null. LEFT JOIN is also known as LEFT OUTER JOIN.

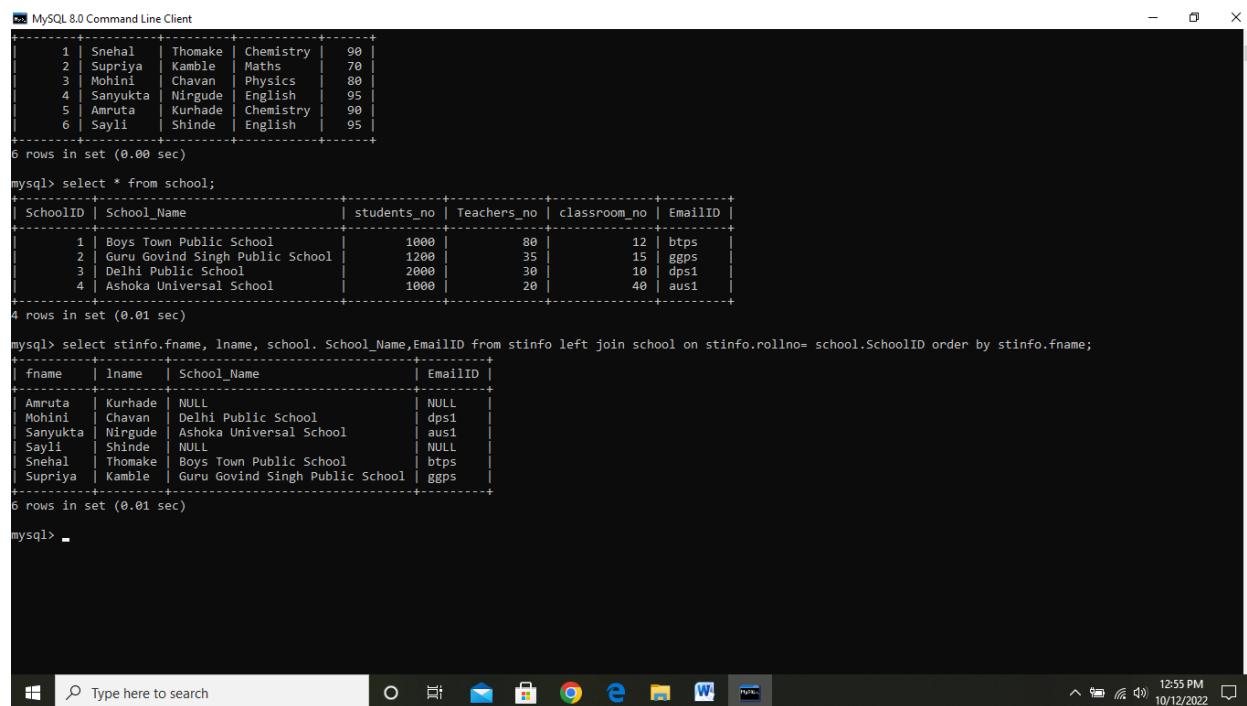
Syntax:

```
SELECT table1.column1,table1.column2,table2.column1,....
```

```
FROM table1
```

```
LEFT JOIN table2
```

```
ON table1.matching_column = table2.matching_column;
```



```
MySQL 8.0 Command Line Client
+-----+
| 1 | Snehal   | Thomake | Chemistry | 90 |
| 2 | Supriya  | Kamble  | Maths     | 70 |
| 3 | Mohini   | Chavan  | Physics   | 80 |
| 4 | Sanyukta | Nirgude | English   | 95 |
| 5 | Amruta   | Kurhade | Chemistry | 99 |
| 6 | Sayli    | Shinde  | English   | 95 |
+-----+
6 rows in set (0.00 sec)

mysql> select * from school;
+-----+
| SchoolID | School_Name           | students_no | Teachers_no | classroom_no | EmailID |
+-----+
| 1 | Boys Town Public School | 1000        | 80          | 12           | btps      |
| 2 | Guru Govind Singh Public School | 1200        | 35          | 15           | ggps      |
| 3 | Delhi Public School     | 2000        | 30          | 10           | dps1      |
| 4 | Ashoka Universal School | 1000        | 20          | 40           | aus1      |
+-----+
4 rows in set (0.01 sec)

mysql> select stinfo.fname, lname, school.School_Name, EmailID from stinfo left join school on stinfo.rollno= school.SchoolID order by stinfo.fname;
+-----+
| fname  | lname   | School_Name           | EmailID |
+-----+
| Amruta | Kurhade | NULL                 | NULL    |
| Mohini | Chavan  | Delhi Public School  | dps1    |
| Sanyukta | Nirgude | Ashoka Universal School | aus1    |
| Sayli  | Shinde  | NULL                 | NULL    |
| Snehal | Thomake | Boys Town Public School | btps    |
| Supriya | Kamble  | Guru Govind Singh Public School | ggps    |
+-----+
6 rows in set (0.01 sec)

mysql>
```

RIGHT JOIN

RIGHT JOIN is similar to LEFT JOIN. This join returns all the rows of the table on the right side of the join and matching rows for the table on the left side of the join. For the rows for which there is no matching row on the left side, the result-set will contain null. RIGHT JOIN is also known as RIGHT OUTER JOIN.

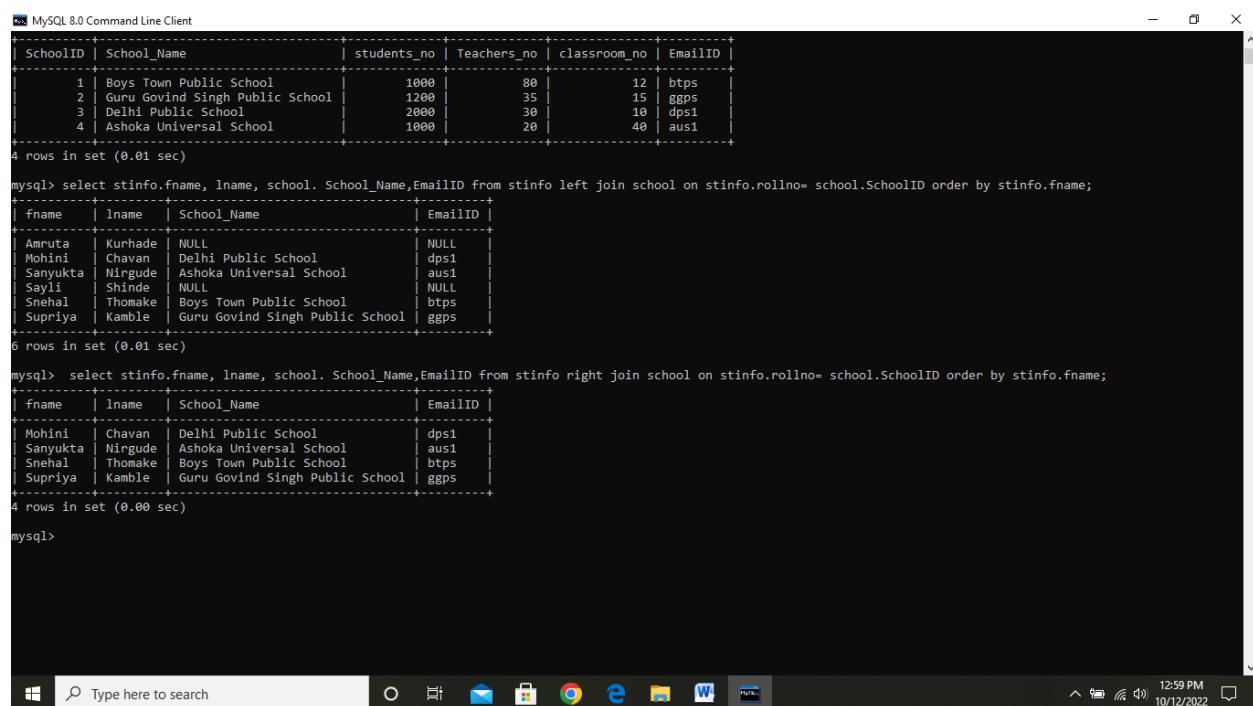
Syntax

```
SELECT table1.column1,table1.column2,table2.column1,....
```

```
FROM table1
```

```
RIGHT JOIN table2
```

```
ON table1.matching_column = table2.matching_column;
```



The screenshot shows the MySQL 8.0 Command Line Client interface. It displays three SQL queries and their results:

```
MySQL 8.0 Command Line Client
+-----+-----+-----+-----+-----+
| SchoolID | School_Name | students_no | Teachers_no | classroom_no | EmailID |
+-----+-----+-----+-----+-----+
| 1 | Boys Town Public School | 1000 | 80 | 12 | btps |
| 2 | Guru Govind Singh Public School | 1200 | 35 | 15 | ggps |
| 3 | Delhi Public School | 2000 | 50 | 10 | dps1 |
| 4 | Ashoka Universal School | 1000 | 28 | 40 | aus1 |
+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)

mysql> select stinfo.fname, lname, school.School_Name,EmailID from stinfo left join school on stinfo.rollno= school.SchoolID order by stinfo.fname;
+-----+-----+-----+-----+
| fname | lname | School_Name | EmailID |
+-----+-----+-----+-----+
| Amruta | Kurhade | NULL | NULL |
| Mohini | Chavan | Delhi Public School | dps1 |
| Sanyukta | Nirgude | Ashoka Universal School | aus1 |
| Sayali | Shinde | NULL | NULL |
| Snehal | Thomake | Boys Town Public School | btps |
| Supriya | Kamble | Guru Govind Singh Public School | ggps |
+-----+-----+-----+-----+
6 rows in set (0.01 sec)

mysql> select stinfo.fname, lname, school.School_Name,EmailID from stinfo right join school on stinfo.rollno= school.SchoolID order by stinfo.fname;
+-----+-----+-----+-----+
| fname | lname | School_Name | EmailID |
+-----+-----+-----+-----+
| Mohini | Chavan | Delhi Public School | dps1 |
| Sanyukta | Nirgude | Ashoka Universal School | aus1 |
| Snehal | Thomake | Boys Town Public School | btps |
| Supriya | Kamble | Guru Govind Singh Public School | ggps |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

Date=10/10/2022 Monday

SQL HAVING Clause

The HAVING clause was added to SQL because the WHERE keyword cannot be used with aggregate functions.

HAVING Syntax

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
HAVING condition
ORDER BY column_name(s);
```



```
MySQL 8.0 Command Line Client
Empty set (0.02 sec)

mysql> use world;
Database changed
mysql> select * from stinfo;
+-----+-----+-----+-----+-----+
| rollno | fname | lname | sname   | mark | phoneno |
+-----+-----+-----+-----+-----+
| 1     | Snehal | Thomake | Chemistry | 90    | NULL    |
| 2     | Supriya | Kamble | Maths    | 70    | NULL    |
| 3     | Mohini | Chavan | Physics  | 80    | NULL    |
| 4     | Sanyukta | Nirgude | English  | 95    | NULL    |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select count(rollno), sname from stinfo group by sname having count(rollno) > 4;
Empty set (0.00 sec)

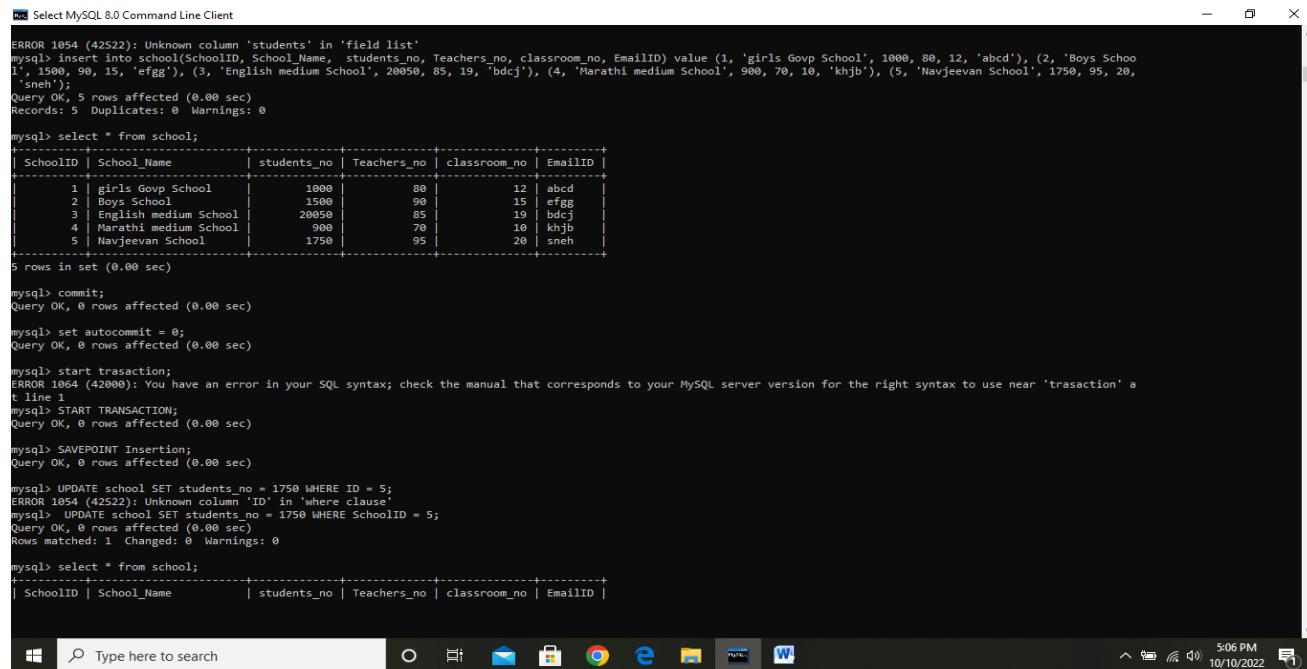
mysql> select count(rollno), sname from stinfo group by sname having count(rollno);
+-----+-----+
| count(rollno) | sname   |
+-----+-----+
| 1           | Chemistry |
| 1           | Maths    |
| 1           | Physics  |
| 1           | English  |
+-----+-----+
4 rows in set (0.00 sec)

mysql> select sum(rollno), sname from stinfo group by sname having sum(rollno);
+-----+-----+
| sum(rollno) | sname   |
+-----+-----+
| 1           | Chemistry |
| 2           | Maths    |
| 3           | Physics  |
| 4           | English  |
+-----+-----+
4 rows in set (0.00 sec)

mysql> select min(rollno), sname from stinfo group by sname having count(rollno)
-> ;
+-----+-----+
| min(rollno) | sname   |
+-----+-----+
```

TCL Commands in SQL

- In SQL, TCL stands for **Transaction control language**.
- A single unit of work in a database is formed after the consecutive execution of commands is known as a transaction.
- There are certain commands present in SQL known as TCL commands that help the user manage the transactions that take place in a database.
- **COMMIT**, **ROLLBACK** and **SAVEPOINT** are the most commonly used TCL commands in SQL.



```
mysql Select MySQL 8.0 Command Line Client

ERROR 1054 (42S22): Unknown column 'students' in 'field list'
mysql> Insert into school(SchoolID, School_Name, students_no, Teachers_no, classroom_no, EmailID) value (1, 'girls Govp School', 1000, 80, 12, 'abcd'), (2, 'Boys Schoo
l', 1500, 90, 15, 'efgg'), (3, 'English medium School', 20050, 85, 19, 'bdcj'), (4, 'Marathi medium School', 900, 70, 10, 'khjb'), (5, 'Navjeevan School', 1750, 95, 20,
'sneh');
Query OK, 5 rows affected (0.00 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> select * from school;
+-----+-----+-----+-----+-----+
| SchoolID | School_Name | students_no | Teachers_no | classroom_no | EmailID |
+-----+-----+-----+-----+-----+
| 1 | girls Govp School | 1000 | 80 | 12 | abcd |
| 2 | Boys School | 1500 | 90 | 15 | efgg |
| 3 | English medium School | 20050 | 85 | 19 | bdcj |
| 4 | Marathi medium School | 900 | 70 | 10 | khjb |
| 5 | Navjeevan School | 1750 | 95 | 20 | sneh |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> commit;
Query OK, 0 rows affected (0.00 sec)

mysql> set autocommit = 0;
Query OK, 0 rows affected (0.00 sec)

mysql> start transaction;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'trasaction' a
t line 1
mysql> START TRANSACTION;
Query OK, 0 rows affected (0.00 sec)

mysql> SAVEPOINT Insertion;
Query OK, 0 rows affected (0.00 sec)

mysql> UPDATE school SET students_no = 1750 WHERE ID = 5;
ERROR 1054 (42S22): Unknown column 'ID' in 'where clause'
mysql> UPDATE school SET students_no = 1750 WHERE SchoolID = 5;
Query OK, 0 rows affected (0.00 sec)
Rows matched: 1  Changed: 0  Warnings: 0

mysql> select * from school;
+-----+-----+-----+-----+-----+
| SchoolID | School_Name | students_no | Teachers_no | classroom_no | EmailID |
+-----+-----+-----+-----+-----+
```

```

Select MySQL 8.0 Command Line Client

mysql> select * from school;
+-----+-----+-----+-----+-----+
| SchoolID | School_Name | students_no | Teachers_no | classroom_no | EmailID |
+-----+-----+-----+-----+-----+
| 1 | girls Govp School | 1000 | 80 | 12 | abcd |
| 2 | Boys School | 1500 | 90 | 15 | efgg |
| 3 | English medium School | 20050 | 85 | 19 | bdcj |
| 4 | Marathi medium School | 900 | 70 | 10 | khjb |
| 5 | Navjeevan School | 1750 | 95 | 20 | sneh |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> UPDATE school SET students_no = 2000 WHERE SchoolID = 5;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from school;
+-----+-----+-----+-----+-----+
| SchoolID | School_Name | students_no | Teachers_no | classroom_no | EmailID |
+-----+-----+-----+-----+-----+
| 1 | girls Govp School | 1000 | 80 | 12 | abcd |
| 2 | Boys School | 1500 | 90 | 15 | efgg |
| 3 | English medium School | 20050 | 85 | 19 | bdcj |
| 4 | Marathi medium School | 900 | 70 | 10 | khjb |
| 5 | Navjeevan School | 2000 | 95 | 20 | sneh |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> SAVEPOINT Updation;
Query OK, 0 rows affected (0.00 sec)

mysql> ROLLBACK TO Insertion;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from school;
+-----+-----+-----+-----+-----+
| SchoolID | School_Name | students_no | Teachers_no | classroom_no | EmailID |
+-----+-----+-----+-----+-----+
| 1 | girls Govp School | 1000 | 80 | 12 | abcd |
| 2 | Boys School | 1500 | 90 | 15 | efgg |
| 3 | English medium School | 20050 | 85 | 19 | bdcj |
| 4 | Marathi medium School | 900 | 70 | 10 | khjb |
| 5 | Navjeevan School | 1750 | 95 | 20 | sneh |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

```

COMMIT :

This command is used to save the data permanently.

Syntax:

commit;

SAVEPOINT :

This command is used to save the data at a particular point temporarily, so that whenever needed can be rollback to that particular point.

Syntax:

Savepoint A;

ROLL BACK :

This command is used to get the data or restore the data to the last savepoint or last committed state.

Syntax:

rollback;

Date=10/10/2022 extra work

1) LIKE Operator

The LIKE operator is used in a WHERE clause to search for a specified pattern in a column.

There are two wildcards often used in conjunction with the LIKE operator:

- The percent sign (%) represents zero, one, or multiple characters
- The underscore sign (_) represents one, single character

The percent sign and the underscore can also be used in combinations!

LIKE Syntax

```
SELECT column1, column2, ...
FROM table_name
WHERE columnN LIKE pattern;
```

2) Aliases

Aliases are used to give a table, or a column in a table, a temporary name.

Aliases are often used to make column names more readable.

An alias only exists for the duration of that query.

An alias is created with the AS keyword.

Alias Column Syntax

```
SELECT column_name AS alias_name
FROM table_name;
```

Alias Table Syntax

```
SELECT column_name(s)
FROM table_name AS alias_name;
```

```

MySQL 8.0 Command Line Client
-> FROM table_name
->
-> ;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'The MySQL LIK
E Operator
The LIKE operator is used in a WHERE clause to search fo' at line 1
mysql> \c
mysql> SELECT * FROM stinfo where ustomerName LIKE '%';
ERROR 1054 (42S22): Unknown column 'ustomerName' in 'where clause'
mysql> SELECT * FROM stinfo where fname LIKE 'a%';
Empty set (0.00 sec)

mysql> select * from stinfo;
+-----+-----+-----+-----+
| rollno | fname | lname | sname | mark | phoneno |
+-----+-----+-----+-----+
| 1 | Snehal | Thomake | Chemistry | 90 | NULL |
| 2 | Supriya | Kamble | Maths | 70 | NULL |
| 3 | Mohini | Chavan | Physics | 80 | NULL |
| 4 | Sanyukta | Nirgude | English | 95 | NULL |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> SELECT * FROM stinfo where rollno like 'a%';
Empty set (0.00 sec)

mysql> SELECT * FROM stinfo where rollno like "%or%";
Empty set (0.00 sec)

mysql> SELECT rollno AS ID, fname AS fNameame from stinfo;
+-----+-----+
| ID | fname |
+-----+-----+
| 1 | Snehal |
| 2 | Supriya |
| 3 | Mohini |
| 4 | Sanyukta |
+-----+-----+
4 rows in set (0.00 sec)

mysql>

```

LOWER CASE() Function

```

MySQL 8.0 Command Line Client
Empty set (0.00 sec)

mysql> SELECT rollno AS ID, fname AS fNameame from stinfo;
+-----+-----+
| ID | fname |
+-----+-----+
| 1 | Snehal |
| 2 | Supriya |
| 3 | Mohini |
| 4 | Sanyukta |
+-----+-----+
4 rows in set (0.00 sec)

mysql> SELECT LCASE("SQL Tutorial is FUN!") AS LowercaseText;
+-----+
| LowercaseText |
+-----+
| sql tutorial is fun! |
+-----+
1 row in set (0.01 sec)

mysql> SELECT LCASE("SQL Tutorial is FUN!") AS LowercaseText;
+-----+
| LowercaseText |
+-----+
| sql tutorial is fun! |
+-----+
1 row in set (0.00 sec)

mysql> SELECT LCASE("SQL Tutorial is FUN!") AS LowercaseText;
+-----+
| LowercaseText |
+-----+
| sql tutorial is fun! |
+-----+
1 row in set (0.00 sec)

mysql>

```

UPPER CASE() Function

```
MySQL 8.0 Command Line Client
mysql> SELECT LCASE("SQL Tutorial is FUN!") AS LowercaseText;
+-----+
| LowercaseText |
+-----+
| sql tutorial is fun! |
+-----+
1 row in set (0.01 sec)

mysql> SELECT LCASE("SQL Tutorial is FUN!") AS LowercaseText;
+-----+
| LowercaseText |
+-----+
| sql tutorial is fun! |
+-----+
1 row in set (0.00 sec)

mysql> SELECT LCASE("SQL Tutorial is FUN!") AS LowercaseText;
+-----+
| LowercaseText |
+-----+
| sql tutorial is fun! |
+-----+
1 row in set (0.00 sec)

mysql>
mysql>
mysql>
mysql>
mysql>
mysql> SELECT UCASE("SQL Tutorial is FUN!") AS UppercaseText;
+-----+
| UppercaseText |
+-----+
| SQL TUTORIAL IS FUN! |
+-----+
1 row in set (0.00 sec)

mysql>
```

DEGREES () Function

```
MySQL 8.0 Command Line Client
+-----+
| sql tutorial is fun! |
+-----+
1 row in set (0.00 sec)

mysql> SELECT LCASE("SQL Tutorial is FUN!") AS LowercaseText;
+-----+
| LowercaseText |
+-----+
| sql tutorial is fun! |
+-----+
1 row in set (0.00 sec)

mysql>
mysql>
mysql>
mysql>
mysql>
mysql> SELECT UCASE("SQL Tutorial is FUN!") AS UppercaseText;
+-----+
| UppercaseText |
+-----+
| SQL TUTORIAL IS FUN! |
+-----+
1 row in set (0.00 sec)

mysql> SELECT DEGREES(1.5);
+-----+
| DEGREES(1.5) |
+-----+
| 85.94366926962348 |
+-----+
1 row in set (0.01 sec)

mysql> select degrees(5);
+-----+
| degrees(5) |
+-----+
| 286.4788975654116 |
+-----+
1 row in set (0.00 sec)

mysql>
```

Date= 11/10/2022 Tue

SQL CONCAT Function

The CONCAT function in SQL is a String function, which is used to merge two or more strings. The Concat service converts the Null values to an Empty string when we display the result. This function is used to concatenate two strings to make a single string. The **operator** is used to link **character strings** and **column string**.

Syntax of CONCAT function

SELECT CONCAT (String 1, String 2, String3.., String N)

FROM [Source]



```
MySQL 8.0 Command Line Client
mysql> use world;
Database changed
mysql> select * from stinfo;
+-----+-----+-----+-----+-----+
| rollno | fname | lname | sname | mark | phoneno |
+-----+-----+-----+-----+-----+
| 1 | Snehal | Thomake | Chemistry | 90 | NULL |
| 2 | Supriya | Kamble | Maths | 70 | NULL |
| 3 | Mohini | Chavan | Physics | 80 | NULL |
| 4 | Sanyukta | Nirgude | English | 95 | NULL |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from school;
+-----+-----+-----+-----+-----+
| SchoolID | School_Name | students_no | Teachers_no | classroom_no | EmailID |
+-----+-----+-----+-----+-----+
| 1 | Boys Town Public School | 1000 | 88 | 12 | btps |
| 2 | Guru Govind Singh Public School | 1200 | 35 | 15 | ggps |
| 3 | Delhi Public School | 2000 | 30 | 10 | dps1 |
| 4 | Ashoka Universal School | 1000 | 28 | 40 | aus1 |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select concat ( SchoolID, School_Name, EmailID) from school;
+-----+
| concat ( SchoolID, School_Name, EmailID) |
+-----+
| 1Boys Town Public Schoolbtps |
| 2Guru Govind Singh Public Schoolggps |
| 3Delhi Public Schooldps1 |
| 4Ashoka Universal Schoolaus1 |
+-----+
4 rows in set (0.01 sec)

mysql>
```

Data Control Language

DCL commands are used to grant and take back authority from any database user.

Grant: It is used to give user access privileges to a database.

Example

```
GRANT SELECT, UPDATE ON MY_TABLE TO SOME_USER, ANOTHER_USER;
```



```
MySQL 8.0 Command Line Client
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select concat ( SchoolID, School_Name, EmailID) from school;
+-----+-----+
| concat ( SchoolID, School_Name, EmailID) |
+-----+-----+
| 1Boys Town Public Schoolbtps |
| 2Guru Govind Singh Public Schoolgps |
| 3Delhi Public Schoolpsi |
| 4Ashoka Universal Schoolaus1 |
+-----+-----+
4 rows in set (0.01 sec)

mysql> create user snehal@localhost identified by 'thomake1999';
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'create user sn
ehal@localhost identified by 'thomake1999'' at line 1
mysql> CREATE USER john@localhost IDENTIFIED BY 'jtp12345';
Query OK, 0 rows affected (0.15 sec)

mysql> SHOW GRANTS FOR john@localhost;
+-----+
| Grants for john@localhost |
+-----+
| GRANT USAGE ON *.* TO `john`@`localhost` |
+-----+
1 row in set (0.00 sec)

mysql> GRANT ALL ON mystudentdb.* TO john@localhost;
Query OK, 0 rows affected (0.01 sec)

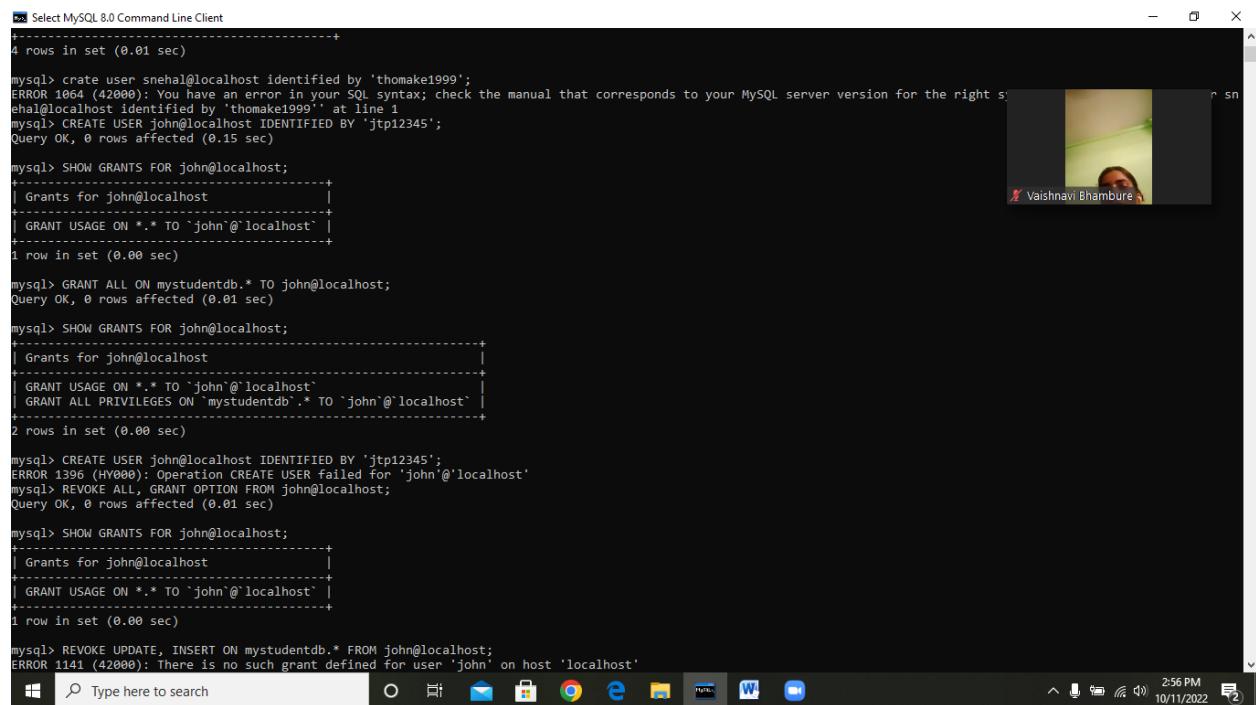
mysql> SHOW GRANTS FOR john@localhost;
+-----+
| Grants for john@localhost |
+-----+
| GRANT USAGE ON *.* TO `john`@`localhost` |
| GRANT ALL PRIVILEGES ON `mystudentdb`.* TO `john`@`localhost` |
+-----+
2 rows in set (0.00 sec)

mysql>
```

Revoke: It is used to take back permissions from the user.

Example

REVOKE SELECT, UPDATE ON MY_TABLE FROM USER1, USER2;



The screenshot shows a Windows desktop environment with a MySQL Command Line Client window open. The window title is "Select MySQL 8.0 Command Line Client". The command history is as follows:

```
mysql> CREATE USER snehal@localhost IDENTIFIED BY 'thomake1999';
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'IDENTIFIED BY 'thomake1999'' at line 1
mysql> CREATE USER john@localhost IDENTIFIED BY 'jtp12345';
Query OK, 0 rows affected (0.15 sec)

mysql> SHOW GRANTS FOR john@localhost;
+-----+
| Grants for john@localhost |
+-----+
| GRANT USAGE ON *.* TO `john`@`localhost` |
+-----+
1 row in set (0.00 sec)

mysql> GRANT ALL ON mystudentdb.* TO john@localhost;
Query OK, 0 rows affected (0.01 sec)

mysql> SHOW GRANTS FOR john@localhost;
+-----+
| Grants for john@localhost |
+-----+
| GRANT USAGE ON *.* TO `john`@`localhost` |
| GRANT ALL PRIVILEGES ON `mystudentdb`.* TO `john`@`localhost` |
+-----+
2 rows in set (0.00 sec)

mysql> CREATE USER john@localhost IDENTIFIED BY 'jtp12345';
ERROR 1396 (HY000): Operation CREATE USER failed for 'john'@'localhost'
mysql> REVOKE ALL, GRANT OPTION FROM john@localhost;
Query OK, 0 rows affected (0.01 sec)

mysql> SHOW GRANTS FOR john@localhost;
+-----+
| Grants for john@localhost |
+-----+
| GRANT USAGE ON *.* TO `john`@`localhost` |
+-----+
1 row in set (0.00 sec)

mysql> REVOKE UPDATE, INSERT ON mystudentdb.* FROM john@localhost;
ERROR 1141 (42000): There is no such grant defined for user 'john' on host 'localhost'
```

The taskbar at the bottom shows icons for File Explorer, Task View, Mail, Edge, File Explorer, Task View, Mail, Word, and File Explorer. The system tray shows the date and time as 10/11/2022 2:56 PM.

SQL ORDER BY Clause

- Whenever we want to sort the records based on the columns stored in the tables of the SQL database, then we consider using the ORDER BY clause in SQL.

Syntax to sort the records in ascending order:

SELECT ColumnName1,...,ColumnNameN FROM TableName ORDER BY ColumnName ASC;

MySQL 8.0 Command Line Client

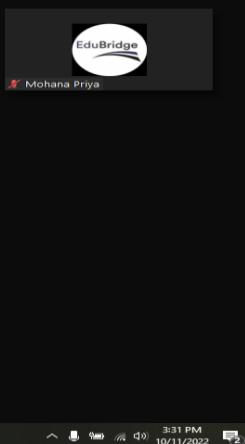
```

mysql> SELECT * FROM school ORDER BY classroom_no DESC;
+----+-----+-----+-----+-----+
| SchoolID | School_Name | students_no | Teachers_no | classroom_no | EmailID |
+----+-----+-----+-----+-----+
| 3 | Delhi Public School | 2000 | 30 | 10 | dps1 |
| 1 | Boys Town Public School | 1000 | 80 | 12 | btps |
| 2 | Guru Govind Singh Public School | 1200 | 35 | 15 | ggps |
| 4 | Ashoka Universal School | 1000 | 20 | 40 | aus1 |
+----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> SELECT * FROM school ORDER BY classroom_no ASC;
+----+-----+-----+-----+-----+
| SchoolID | School_Name | students_no | Teachers_no | classroom_no | EmailID |
+----+-----+-----+-----+-----+
| 4 | Ashoka Universal School | 1000 | 20 | 40 | aus1 |
| 3 | Delhi Public School | 2000 | 30 | 10 | dps1 |
| 2 | Guru Govind Singh Public School | 1200 | 35 | 15 | ggps |
| 1 | Boys Town Public School | 1000 | 80 | 12 | btps |
| 5 | Delhi Public School | 2000 | 30 | 10 | dps1 |
+----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>

```



Syntax to sort the records in descending order:

SELECT ColumnName1,...,ColumnNameN FROM TableName ORDER BY ColumnNameDESC;

MySQL 8.0 Command Line Client

```

mysql> SELECT * FROM school ORDER BY students_no DESC;
+----+-----+-----+-----+-----+
| SchoolID | School_Name | students_no | Teachers_no | classroom_no | EmailID |
+----+-----+-----+-----+-----+
| 4 | Ashoka Universal School | 1000 | 20 | 40 | aus1 |
| 3 | Delhi Public School | 2000 | 30 | 10 | dps1 |
| 2 | Guru Govind Singh Public School | 1200 | 35 | 15 | ggps |
| 1 | Boys Town Public School | 1000 | 80 | 12 | btps |
+----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

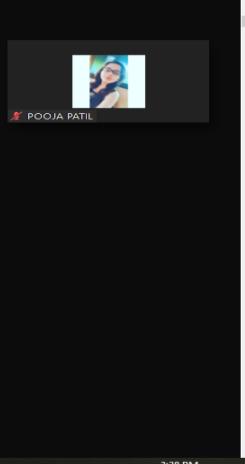
mysql> SELECT * FROM school ORDER BY students_no ASC;
+----+-----+-----+-----+-----+
| SchoolID | School_Name | students_no | Teachers_no | classroom_no | EmailID |
+----+-----+-----+-----+-----+
| 3 | Delhi Public School | 2000 | 30 | 10 | dps1 |
| 2 | Guru Govind Singh Public School | 1200 | 35 | 15 | ggps |
| 1 | Boys Town Public School | 1000 | 80 | 12 | btps |
| 4 | Ashoka Universal School | 1000 | 20 | 40 | aus1 |
+----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> SELECT * FROM school ORDER BY classroom_no, Teachers_no DESC;
+----+-----+-----+-----+-----+
| SchoolID | School_Name | students_no | Teachers_no | classroom_no | EmailID |
+----+-----+-----+-----+-----+
| 1 | Boys Town Public School | 1000 | 80 | 12 | btps |
| 2 | Guru Govind Singh Public School | 1200 | 35 | 15 | ggps |
| 3 | Delhi Public School | 2000 | 30 | 10 | dps1 |
| 4 | Ashoka Universal School | 1000 | 20 | 40 | aus1 |
+----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> SELECT * FROM school ORDER BY classroom_no DESC;
+----+-----+-----+-----+-----+
| SchoolID | School_Name | students_no | Teachers_no | classroom_no | EmailID |
+----+-----+-----+-----+-----+
| 4 | Ashoka Universal School | 1000 | 20 | 40 | aus1 |
| 2 | Guru Govind Singh Public School | 1200 | 35 | 15 | ggps |
| 1 | Boys Town Public School | 1000 | 80 | 12 | btps |
| 3 | Delhi Public School | 2000 | 30 | 10 | dps1 |
+----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

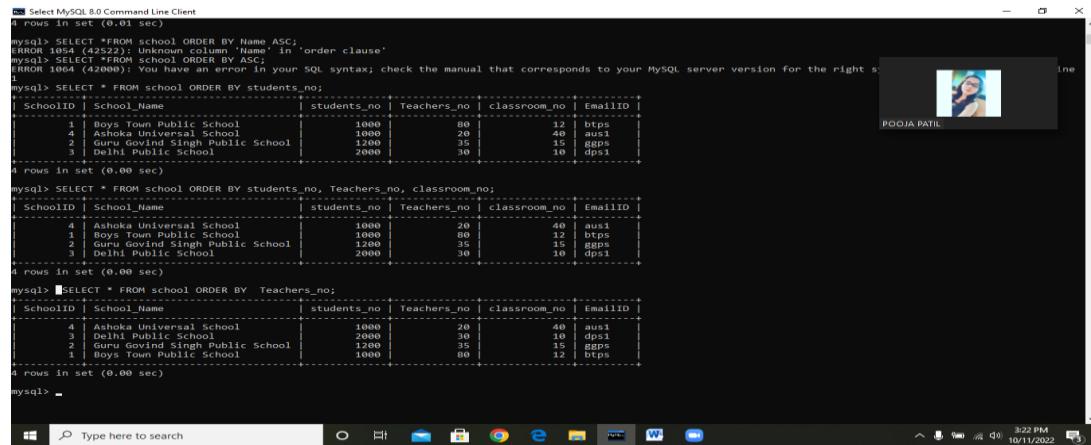
mysql>

```



Syntax to sort the records in ascending order without using ASC keyword:

SELECT ColumnName1,...,ColumnNameN FROM TableName ORDER BY ColumnName



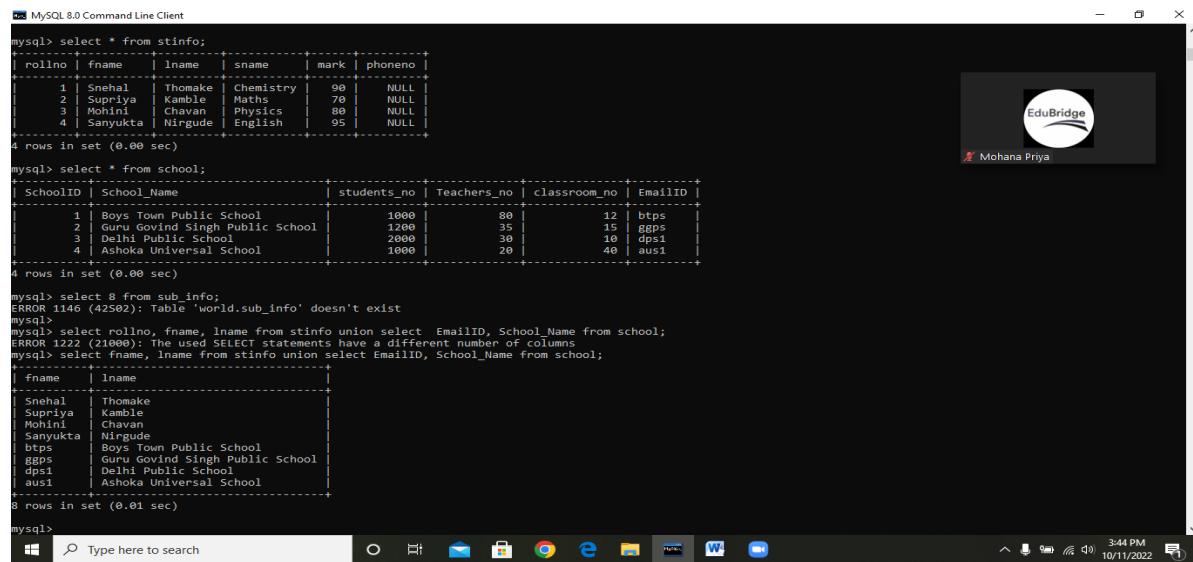
```
mysql> Select * FROM school ORDER BY Name ASC;
ERROR 1048 (42002): Unknown column 'Name' in 'order clause'
mysql> SELECT * FROM school ORDER BY Name AS;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'AS' at line 1
mysql> SELECT * FROM school ORDER BY students_no;
+-----+-----+-----+-----+-----+
| SchoolID | School_Name | students_no | Teachers_no | classroom_no | EmailID |
+-----+-----+-----+-----+-----+
| 1 | Boys Town Public School | 1000 | 80 | 20 | btps |
| 4 | Ashoka Universal School | 1000 | 20 | 40 | ausi |
| 2 | Guru Govind Singh Public School | 1200 | 35 | 15 | ggps |
| 3 | Delhi Public School | 2000 | 30 | 10 | dps1 |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> SELECT * FROM school ORDER BY students_no, Teachers_no, classroom_no;
+-----+-----+-----+-----+-----+
| SchoolID | School_Name | students_no | Teachers_no | classroom_no | EmailID |
+-----+-----+-----+-----+-----+
| 4 | Ashoka Universal School | 1000 | 20 | 40 | ausi |
| 1 | Boys Town Public School | 1000 | 80 | 12 | btps |
| 2 | Guru Govind Singh Public School | 1200 | 35 | 15 | ggps |
| 3 | Delhi Public School | 2000 | 30 | 10 | dps1 |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> SELECT * FROM school ORDER BY Teachers_no;
+-----+-----+-----+-----+-----+
| SchoolID | School_Name | students_no | Teachers_no | classroom_no | EmailID |
+-----+-----+-----+-----+-----+
| 4 | Ashoka Universal School | 1000 | 20 | 40 | ausi |
| 3 | Delhi Public School | 2000 | 30 | 10 | dps1 |
| 2 | Guru Govind Singh Public School | 1200 | 35 | 15 | ggps |
| 1 | Boys Town Public School | 1000 | 80 | 12 | btps |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

MySQL Union

MySQL Union is an operator that allows us to combine two or more results from multiple SELECT queries into a single result set. It comes with a default feature that removes the **duplicate** rows from the result set. MySQL always uses the name of the column in the first SELECT statement will be the column names of the result set(output).



```
mysql> select * from stinfo;
+-----+-----+-----+-----+-----+
| rollno | fname | lname | sname | mark | phoneno |
+-----+-----+-----+-----+-----+
| 1 | Snehal | Thomake | Chemistry | 99 | NULL |
| 2 | Supriya | Kamble | Maths | 70 | NULL |
| 3 | Mohini | Chavan | Physics | 80 | NULL |
| 4 | Sanyukta | Nirgude | English | 95 | NULL |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from school;
+-----+-----+-----+-----+-----+
| SchoolID | School_Name | students_no | Teachers_no | classroom_no | EmailID |
+-----+-----+-----+-----+-----+
| 1 | Boys Town Public School | 1000 | 80 | 20 | btps |
| 2 | Guru Govind Singh Public School | 1200 | 35 | 15 | ggps |
| 3 | Delhi Public School | 2000 | 30 | 10 | dps1 |
| 4 | Ashoka Universal School | 1000 | 20 | 40 | ausi |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select 8 from sub_info;
ERROR 1146 (42502): Table 'world.sub_info' doesn't exist
mysql>
mysql> select rollno, fname, lname from stinfo union select EmailID, School_Name from school;
ERROR 1060 (42000): The used SELECT statements have a different number of columns
mysql> select fname, lname from stinfo union select EmailID, School_Name from school;
+-----+-----+
| fname | lname |
+-----+-----+
| Snehal | Thomake |
| Supriya | Kamble |
| Mohini | Chavan |
| Sanyukta | Nirgude |
| btps | Boys Town Public School |
| ggps | Guru Govind Singh Public School |
| dps1 | Delhi Public School |
| ausi | Ashoka Universal School |
+-----+-----+
8 rows in set (0.01 sec)
```

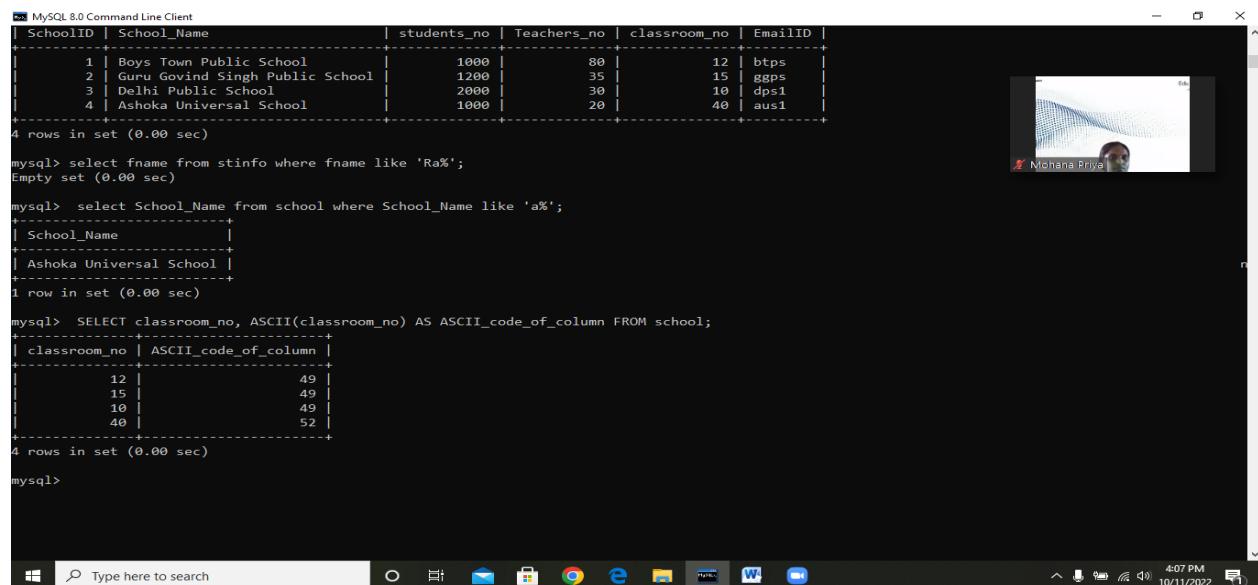
ASCII String Function

This function in SQL returns the ASCII value of the character in the output. It gives the ASCII value of the left-most character of the string.

Syntax of ASCII String Function:

Syntax1: This syntax uses ASCII with the table column:

1. **SELECT ASCII(Column_Name) as ASCII_Name FROM Table_Name;**



The screenshot shows a Windows desktop with the MySQL 8.0 Command Line Client window open. The client displays several SQL queries and their results. The queries demonstrate how to use the ASCII function on table columns and strings.

```
MySQL 8.0 Command Line Client
+-----+-----+-----+-----+-----+
| SchoolID | School_Name | students_no | Teachers_no | classroom_no | EmailID |
+-----+-----+-----+-----+-----+
| 1 | Boys Town Public School | 1000 | 80 | 12 | btps |
| 2 | Guru Govind Singh Public School | 1200 | 35 | 15 | ggps |
| 3 | Delhi Public School | 2000 | 30 | 10 | dps1 |
| 4 | Ashoka Universal School | 1000 | 20 | 40 | aus1 |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select fname from stinfo where fname like 'Ra%';
Empty set (0.00 sec)

mysql> select School_Name from school where School_Name like 'a%';
+-----+
| School_Name |
+-----+
| Ashoka Universal School |
+-----+
1 row in set (0.00 sec)

mysql> SELECT classroom_no, ASCII(classroom_no) AS ASCII_code_of_column FROM school;
+-----+-----+
| classroom_no | ASCII_code_of_column |
+-----+-----+
| 12 | 49 |
| 15 | 49 |
| 10 | 49 |
| 40 | 52 |
+-----+
4 rows in set (0.00 sec)

mysql>
```

Date = 12/10/2022

```
MySQL 8.0 Command Line Client
+-----+
24 rows in set (0.00 sec)

mysql> CREATE INDEX i.i_l_lastname
    > ON Persons (lastName);
ERROR 1146 (42002): Table 'world.persons' doesn't exist
mysql> CREATE INDEX fname on stinfo (fname);
Query OK, 0 rows affected (0.28 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> select * from stinfo;
+-----+-----+-----+-----+
| rollno | fname | lname | sname | mark |
+-----+-----+-----+-----+
|       1 | Snehal | Thomake | Chemistry | 90 |
|       2 | Supriya | Raghunath | Maths | 70 |
|       3 | Mohini | Chavan | Politics | 80 |
|       4 | Sanyukta | Nirgude | English | 95 |
|       5 | Amruta | Kurhade | Chemistry | 90 |
|       6 | Sayali | Shinde | English | 95 |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> SELECT CHAR_LENGTH("Snehal Raghunath Thomake") AS LengthOfString;
+-----+
| LengthOfString |
+-----+
|          24 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT FIELD("q", "s", "q", "l");
+-----+
| FIELD("q", "s", "q", "l") |
+-----+
|          2 |
+-----+
1 row in set (0.00 sec)

mysql> select field("r", "m", "h", "r", "h");
+-----+
| field("r", "m", "h", "r", "h") |
+-----+
|          3 |
+-----+
1 row in set (0.00 sec)
```



Priyanka

```
MySQL 8.0 Command Line Client
+-----+
1 row in set (0.00 sec)

mysql> SELECT FIND_IN_SET("q", "s,q,l");
+-----+
| FIND_IN_SET("q", "s,q,l") |
+-----+
|          2 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT FORMAT(250500.5634, 0);
+-----+
| FORMAT(250500.5634, 0) |
+-----+
| 250,501 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT FORMAT(250500.5634, 2);
+-----+
| FORMAT(250500.5634, 2) |
+-----+
| 250,500.56 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT LOCATE("3", "W3Schools.com") AS MatchPosition;
+-----+
| MatchPosition |
+-----+
|          2 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT LOCATE("3", "i got a news 3 from supriya") AS MatchPosition;
+-----+
| MatchPosition |
+-----+
|         14 |
+-----+
1 row in set (0.00 sec)

mysql> select * from school;
+-----+-----+-----+-----+
| SchoolID | School_Name | students_no | Teachers_no | classroom_no | EmailID |
+-----+-----+-----+-----+
|       1 | Boys Town Public School |      1000 |        80 |          12 | btps |
|       2 | Guru Govind Singh Public School |     1200 |        35 |          15 | ggps |
+-----+-----+-----+-----+
```



Mohana Priya

```

MySQL 8.0 Command Line Client
1 row in set (0.00 sec)

mysql> select * from school;
+-----+-----+-----+-----+-----+
| SchoolID | School_Name | students_no | Teachers_no | classroom_no | EmailID |
+-----+-----+-----+-----+-----+
| 1 | Boys Town Public School | 1000 | 80 | 12 | btps |
| 2 | Guru Govind Singh Public School | 1200 | 35 | 15 | ggps |
| 3 | Delhi Public School | 2000 | 30 | 10 | dps1 |
| 4 | Ashoka Universal School | 1000 | 20 | 40 | aus1 |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select locate ("n", School_Name) from school;
+-----+
| locate ("n", School_Name) |
+-----+
| 9 |
| 10 |
| 0 |
| 9 |
+-----+
4 rows in set (0.00 sec)

mysql> SELECT LPAD("snehal", 10, "ABC");
+-----+
| LPAD("snehal", 10, "ABC") |
+-----+
| ABCsnehal |
+-----+
1 row in set (0.00 sec)

mysql> SELECT RPAD("snehal", 10, "ABC");
+-----+
| RPAD("snehal", 10, "ABC") |
+-----+
| ABCAsnehal |
+-----+
1 row in set (0.00 sec)

mysql> SELECT LTRIM("      SQL Tutorial") AS LeftTrimmedString;
+-----+
| LeftTrimmedString |
+-----+
| SQL Tutorial |
+-----+
1 row in set (0.00 sec)

```

Mohana Priya

```

MySQL 8.0 Command Line Client
mysql> ^C
mysql> ^C
mysql> SELECT LTRIM("      Snehal Thomake") AS Left Trimmed;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'Left Trimmed' at line 1
mysql> SELECT LTRIM("      snehal thomake") AS LeftTrimmedString;
+-----+
| LeftTrimmedString |
+-----+
| snehal thomake |
+-----+
1 row in set (0.00 sec)

mysql> ^C
mysql> SELECT MID("SQL Tutorial", 5, 3) AS ExtractString;
+-----+
| ExtractString |
+-----+
| Tut |
+-----+
1 row in set (0.00 sec)

mysql>
mysql> SELECT MID("snehal thomake", 6, 4) AS ExtractString;
+-----+
| ExtractString |
+-----+
| l th |
+-----+
1 row in set (0.00 sec)

mysql> SELECT COS(2);
+-----+
| COS(2) |
+-----+
| -0.4161468365471424 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT 10 DIV 5;
+-----+
| 10 DIV 5 |
+-----+
| 2 |
+-----+
1 row in set (0.00 sec)

```

Mohana Priya

EON 7670 ST AN
Pooja Patil Edu: pooja patil 7670
12-10-2022
Google Chrome • web.whatsapp.com

4:46 PM
10/12/2022

MySQL 8.0 Command Line Client

```
+-----+  
1 row in set (0.00 sec)  
  
mysql> SELECT FLOOR(25.75);  
+-----+  
| FLOOR(25.75) |  
+-----+  
| 25 |  
+-----+  

```

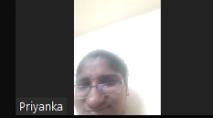


Mohana Priya

MySQL 8.0 Command Line Client

```
+-----+  
1 row in set (0.00 sec)  
  
mysql> SELECT LOG(2);  
+-----+  
| LOG(2) |  
+-----+  
| 0.693147180559453 |  
+-----+  

```



Priyanka

```
MySQL 8.0 Command Line Client
+-----+
1 row in set (0.00 sec)

mysql> SELECT CURRENT_DATE();
+-----+
| CURRENT_DATE() |
+-----+
| 2022-10-12 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT CURRENT_TIME();
+-----+
| CURRENT_TIME() |
+-----+
| 16:21:58 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT CURRENT_TIMESTAMP();
+-----+
| CURRENT_TIMESTAMP() |
+-----+
| 2022-10-12 16:23:43 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT CURTIME();
+-----+
| CURTIME() |
+-----+
| 16:24:31 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT DAY("2022-06-15 09:34:21");
+-----+
| DAY("2022-06-15 09:34:21") |
+-----+
| 15 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT DAYNAME("2022-10-28");
+-----+
| DAYNAME("2022-10-28") |
+-----+
| Friday |
+-----+
1 row in set (0.00 sec)

Windows Taskbar: Type here to search, Start, File Explorer, Mail, Google Chrome, Microsoft Edge, File, Power, Camera, 4:47 PM, 10/12/2022
```

```
Select MySQL 8.0 Command Line Client
mysql> SELECT CURTIME();
+-----+
| CURTIME() |
+-----+
| 16:24:31 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT DAY("2022-06-15 09:34:21");
+-----+
| DAY("2022-06-15 09:34:21") |
+-----+
| 15 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT DAYNAME("2022-10-28");
+-----+
| DAYNAME("2022-10-28") |
+-----+
| Friday |
+-----+
1 row in set (0.01 sec)

mysql> SELECT LOCALTIME();
+-----+
| LOCALTIME() |
+-----+
| 2022-10-12 16:51:23 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT WEEKDAY("2017-06-15");
+-----+
| WEEKDAY("2017-06-15") |
+-----+
| 3 |
+-----+
1 row in set (0.00 sec)

mysql> select * from stinfo;
+-----+
| rollno | fname | lname | sname | mark |
+-----+
| 1 | Snehal | Thomake | Chemistry | 90 |
| 2 | Supriya | Kamble | Maths | 70 |
| 3 | Mohini | Chavan | Physics | 80 |
+-----+
3 rows in set (0.00 sec)

Windows Taskbar: Type here to search, Start, File Explorer, Mail, Google Chrome, Microsoft Edge, File, Power, Camera, 4:48 PM, 10/12/2022
```

```
MySQL [Select MySQL 8.0 Command Line Client] 2022-10-12 16:31:23
+-----+
| LOCALTIME()           |
+-----+
| 2022-10-12 16:31:23 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT WEEKDAY("2017-06-15");
+-----+
| WEEKDAY("2017-06-15") |
+-----+
|            3          |
+-----+
1 row in set (0.00 sec)

mysql> select * from stinfo;
+-----+
| rollno | fname   | lname    | sname    | mark |
+-----+
|      1 | Snehal  | Thomake | Chemistry | 90   |
|      2 | Supriya | Kamble  | Maths     | 78   |
|      3 | Mohini  | Chavan  | Physics   | 88   |
|      4 | Sanyukta | Nirgude | English   | 95   |
|      5 | Amruta  | Kurhade | Chemistry | 90   |
|      6 | Sayali  | Shinde  | English   | 95   |
+-----+
6 rows in set (0.00 sec)

mysql> select concat(fname , lname) from stinfo;
+-----+
| concat(fname , lname) |
+-----+
| SnehalThomake        |
| SupriyaKamble         |
| MohiniChavan          |
| SanyuktaNirgude       |
| AmrutaKurhade         |
| SayaliShinde          |
+-----+
6 rows in set (0.00 sec)

mysql>
```