**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](https://www.geeksforgeeks.org/structured-query-language/) uses certain commands like Create, Drop, Insert, etc. to carry out the required tasks.

These [SQL](https://www.geeksforgeeks.org/sql-concepts-and-queries/)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

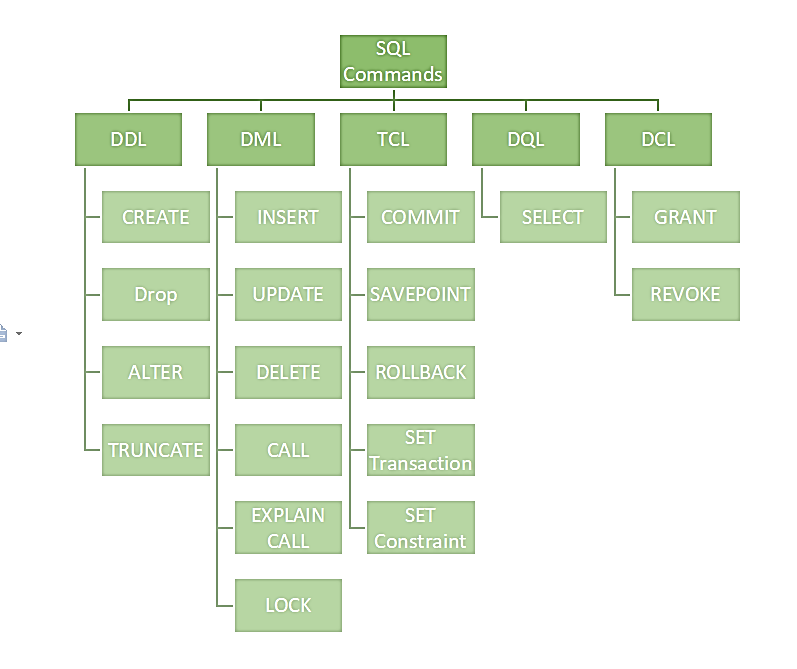
**Date=29/09/2022 Thu.**

SQL ,MYSQL, ORACLE and Working on web page.

**Date=30/09/2022 Friday**

These [SQL](https://www.geeksforgeeks.org/sql-concepts-and-queries/)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



### 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:**   1. INSERT INTO TABLE\_NAME 2. (col1, col2, col3,.... col N) 3. VALUES (value1, value2, value3, .... valueN);   Or   1. INSERT INTO TABLE\_NAME 2. VALUES (value1, value2, value3, .... valueN);   **For example:**   1. 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 CONDITION]  **For example:**   1. UPDATE students 2. SET User\_Name = 'Sonoo' 3. WHERE Student\_Id = '3'   **c. DELETE:** It is used to remove one or more row from a table.  **Syntax:**   1. DELETE FROM table\_name [WHERE condition];   **For example:**   1. DELETE FROM javatpoint 2. 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**   1. GRANT SELECT, UPDATE ON MY\_TABLE TO SOME\_USER, ANOTHER\_USER;   **b. Revoke:** It is used to take back permissions from the user.  **Example**   1. 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:**   1. COMMIT;   **Example:**   1. DELETE FROM CUSTOMERS 2. WHERE AGE = 25; 3. COMMIT;   **b. Rollback:** Rollback command is used to undo transactions that have not already been saved to the database.  **Syntax:**   1. ROLLBACK;   **Example:**   1. DELETE FROM CUSTOMERS 2. WHERE AGE = 25; 3. ROLLBACK;   **c. SAVEPOINT:** It is used to roll the transaction back to a certain point without rolling back the entire transaction.  **Syntax:**   1. 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:**   1. SELECT expressions 2. FROM TABLES 3. WHERE conditions;   **For example:**   1. SELECT emp\_name 2. FROM employee 3. 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)

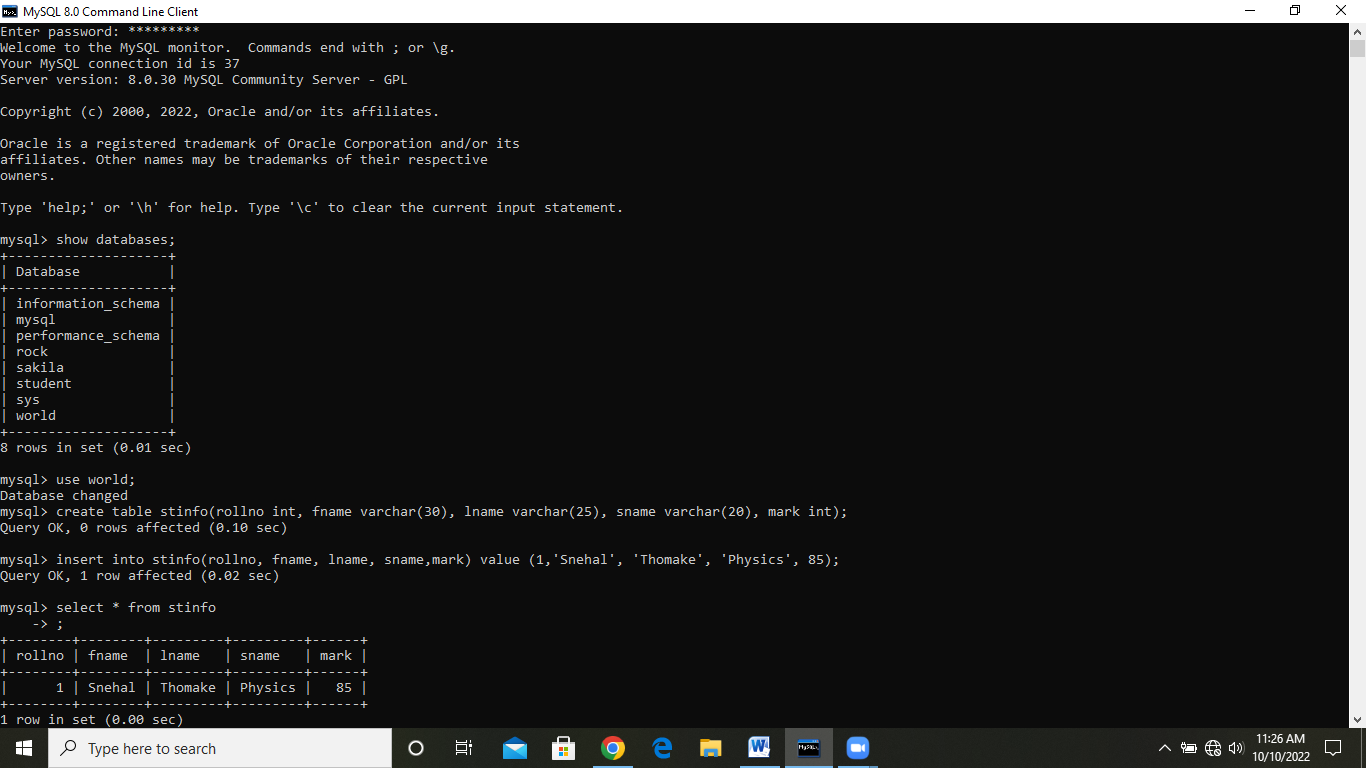
The VARCHAR data type accepts character strings, including Unicode, of a variable length is up to the maximum length specified in the data type declaration.

#### INTEGER or INT

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

Date= 1/10/2022 Sat.

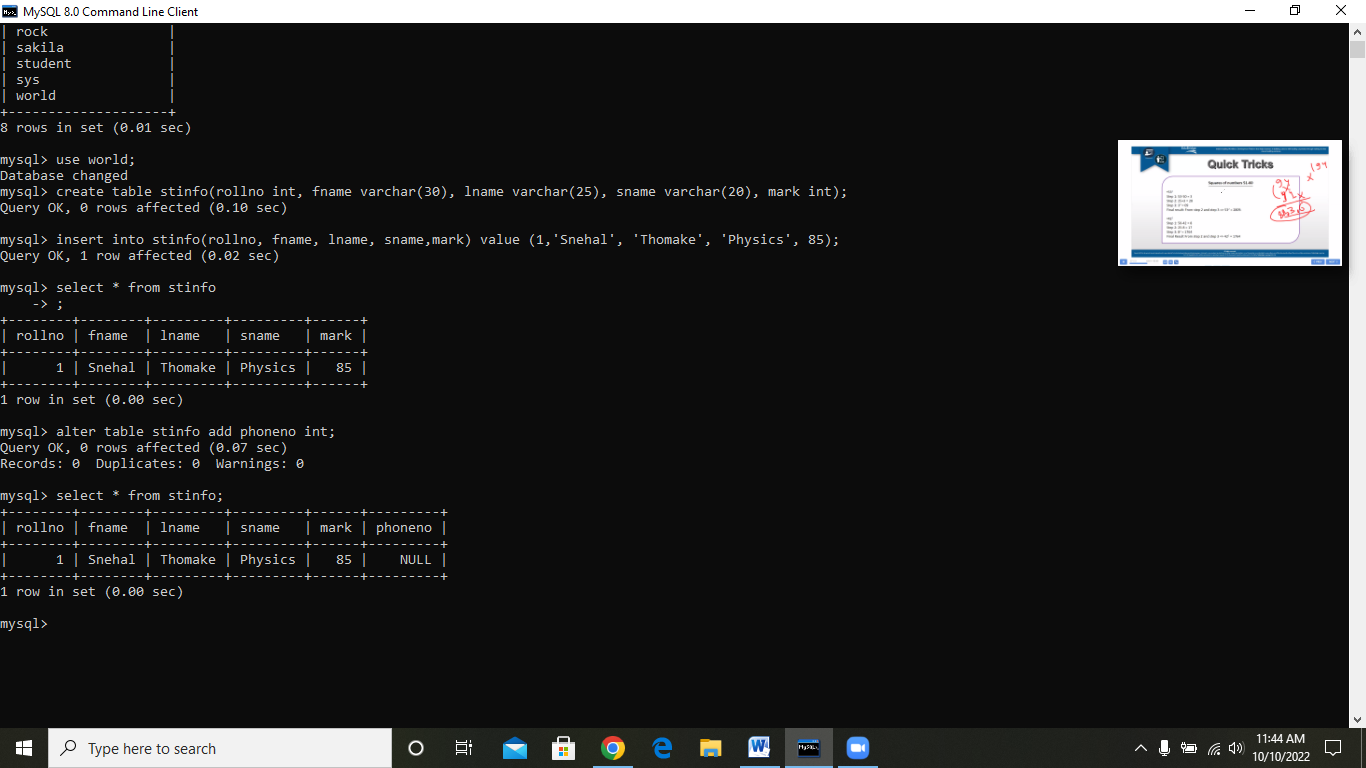
CREAT TABLE



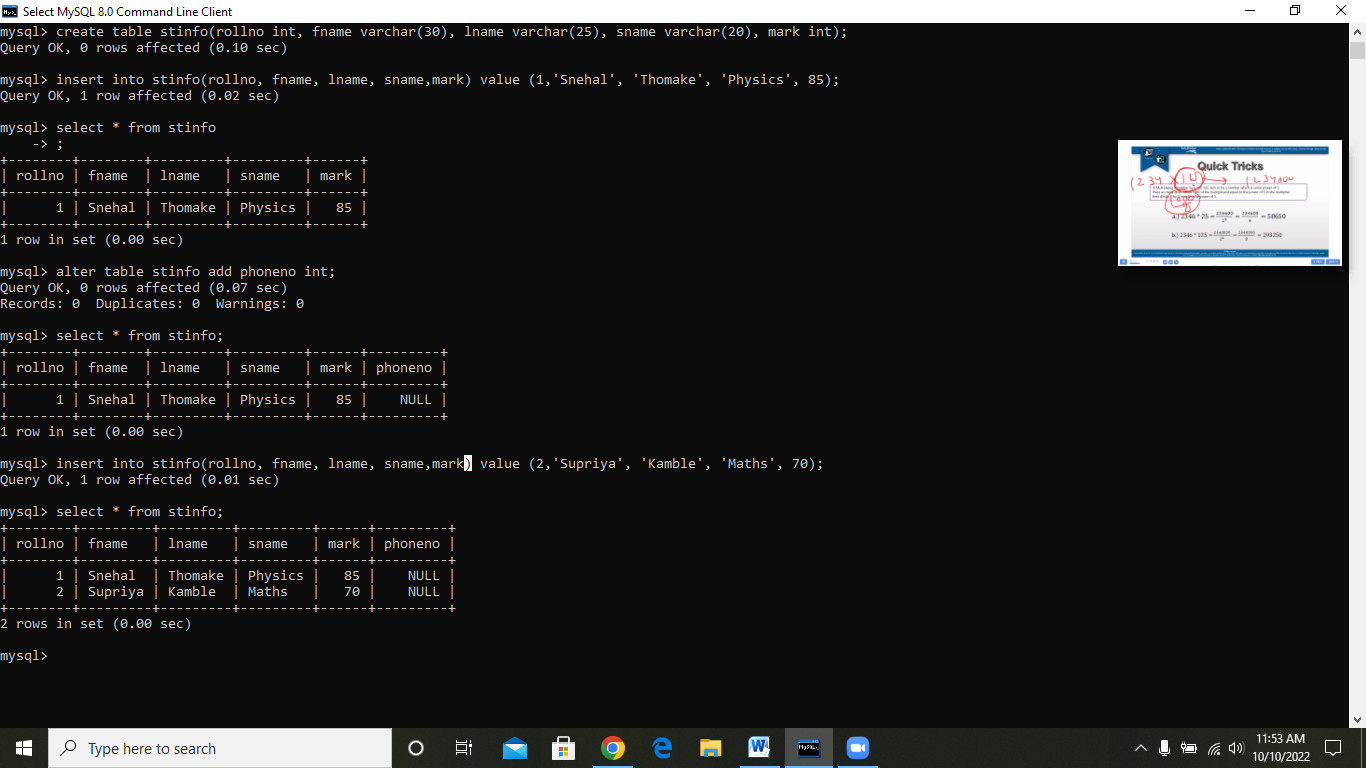
ALTER

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.

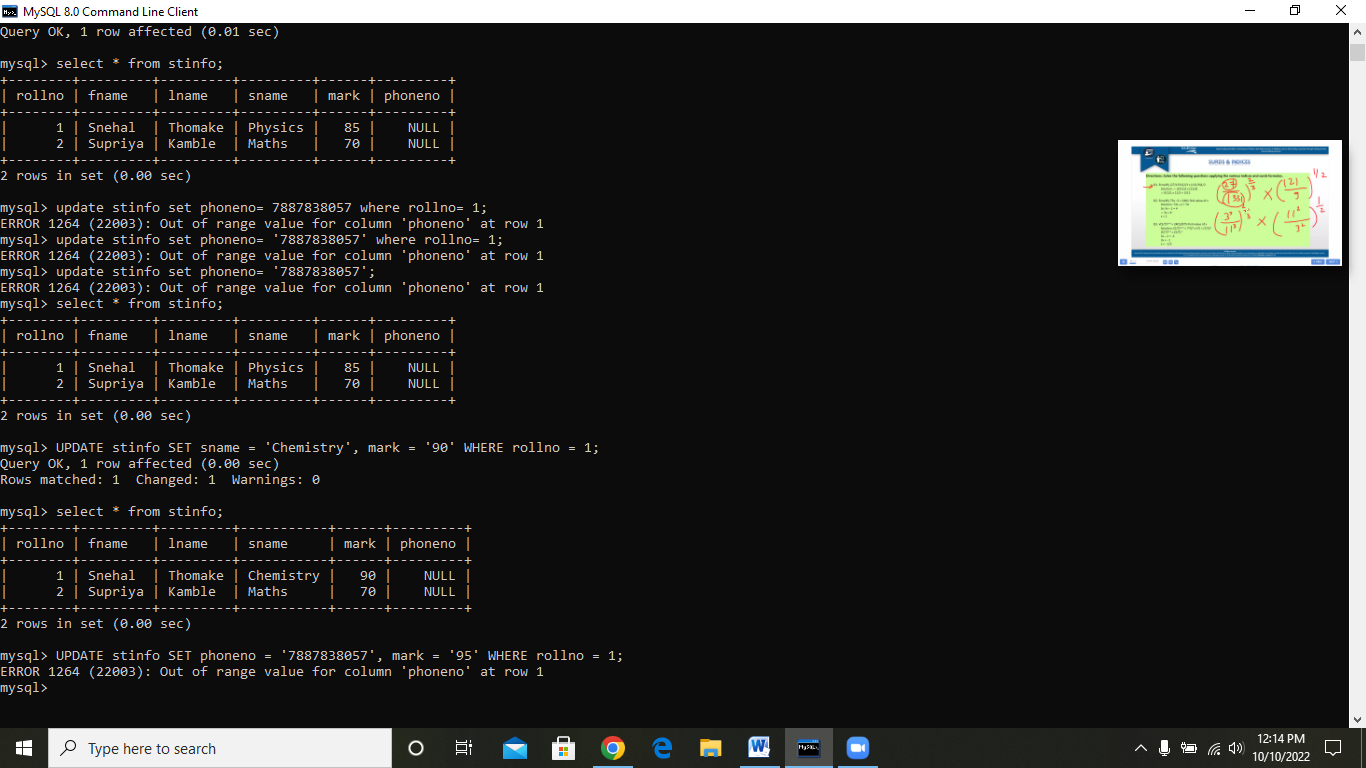


ROW CREATE



UPDATE

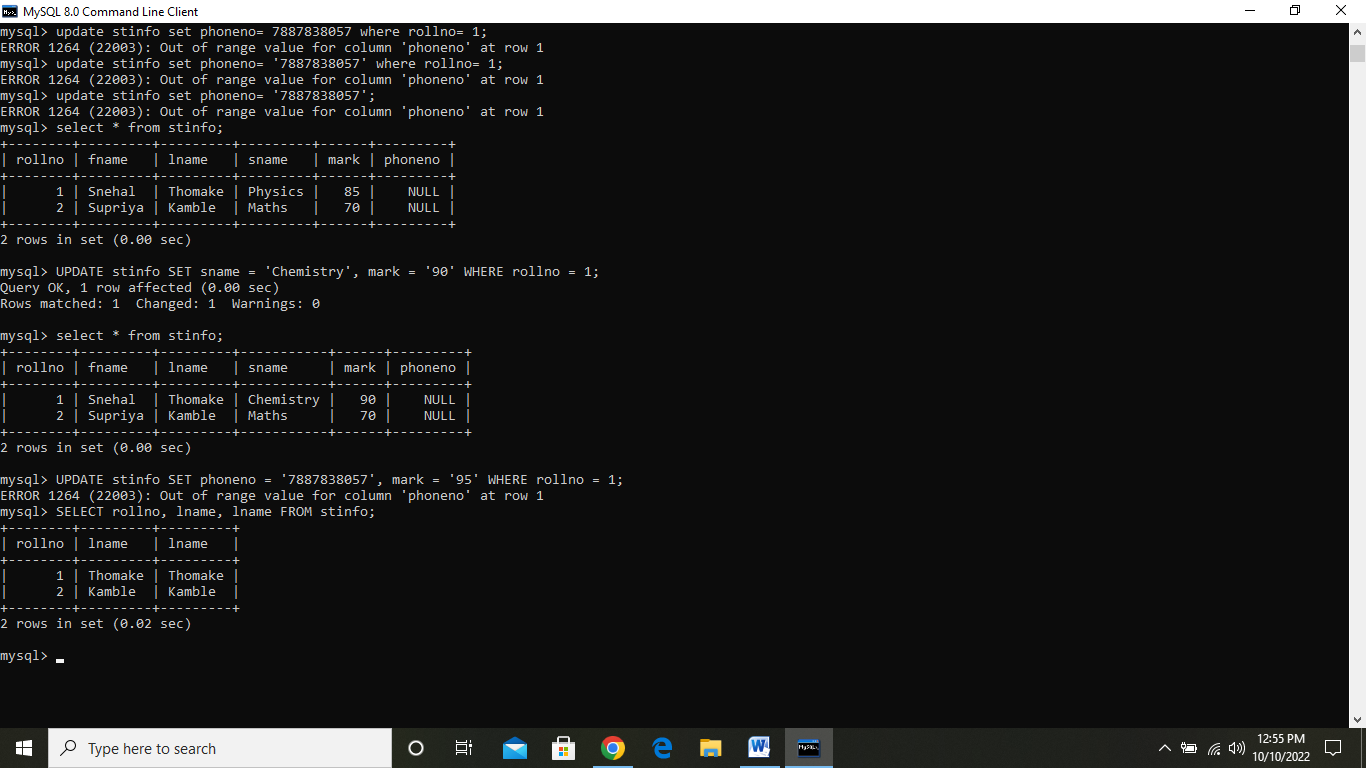
The UPDATE statement is used to modify the existing records in a table.



SELECT

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

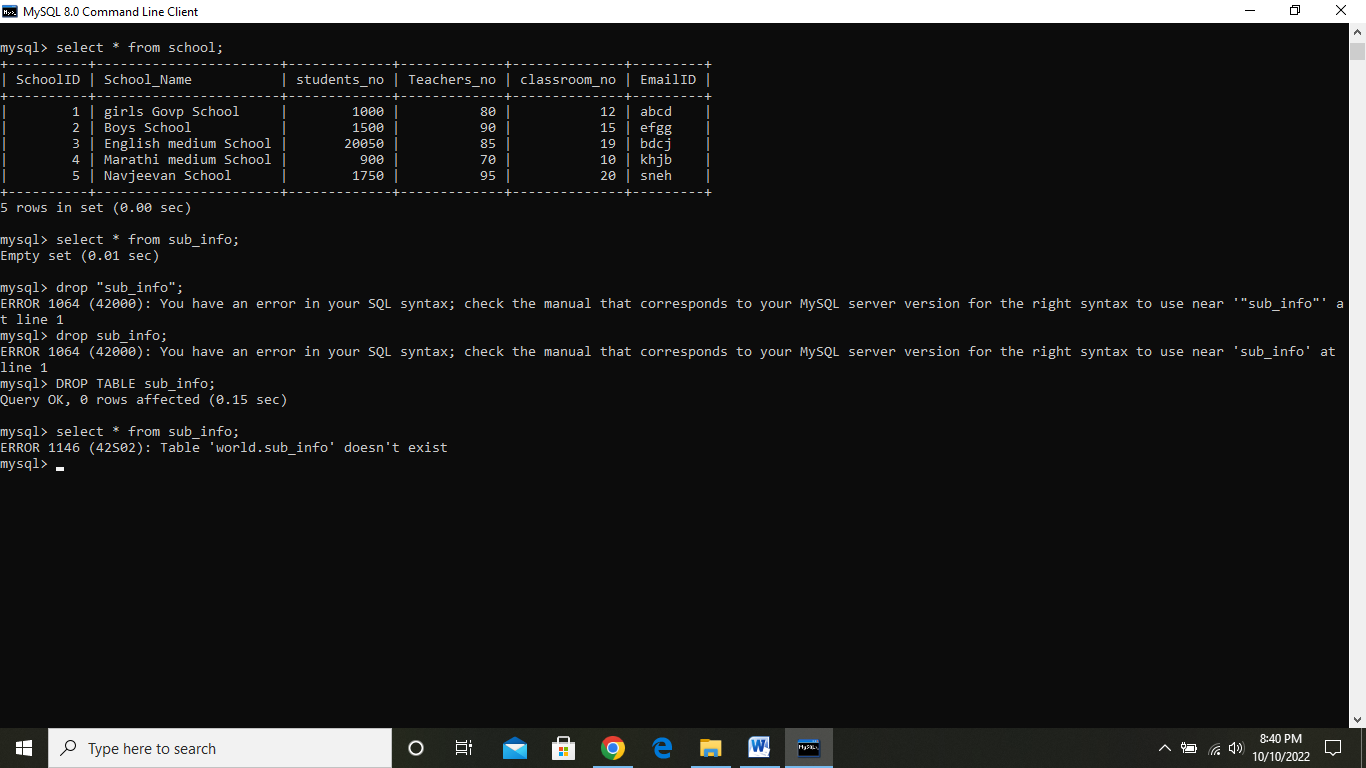
The data returned is stored in a result table, called the result-set.



**Date=03/10/2022 Monday**

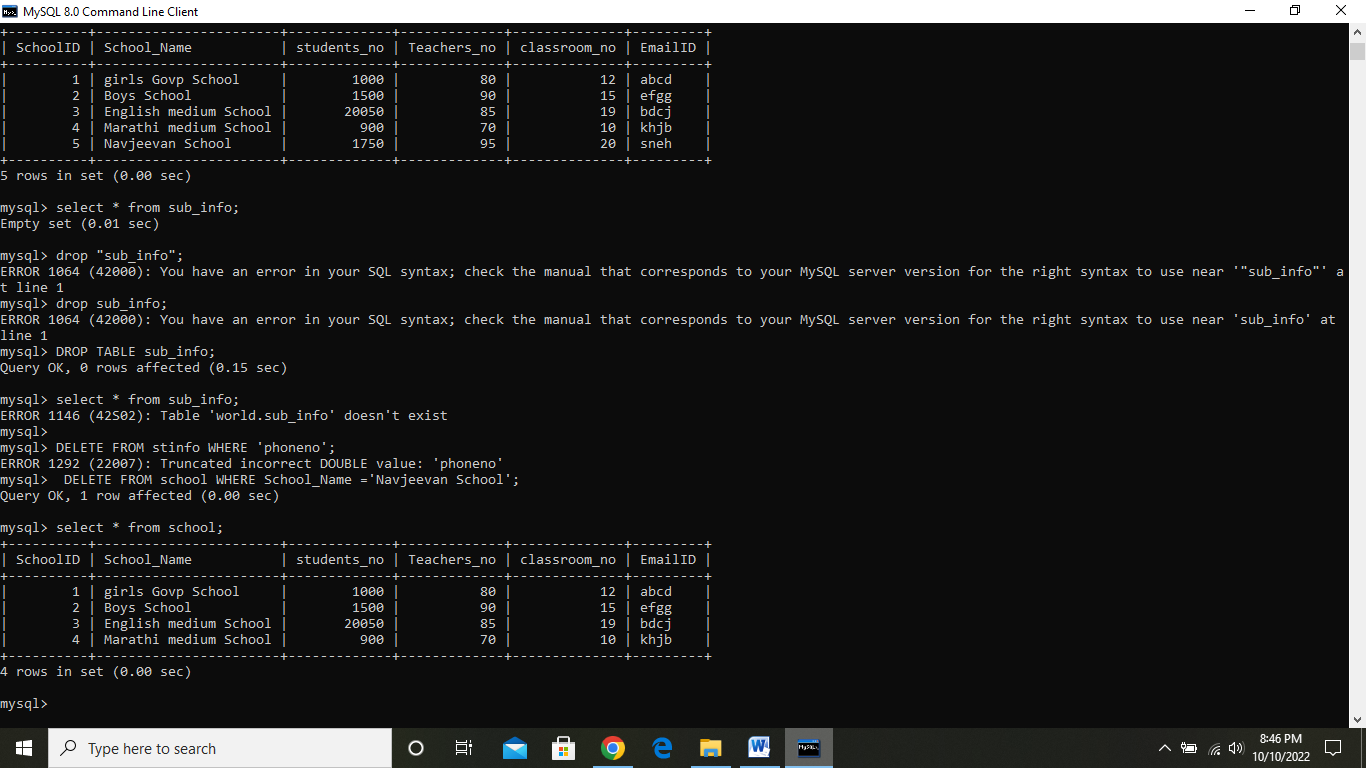
**DROP**

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



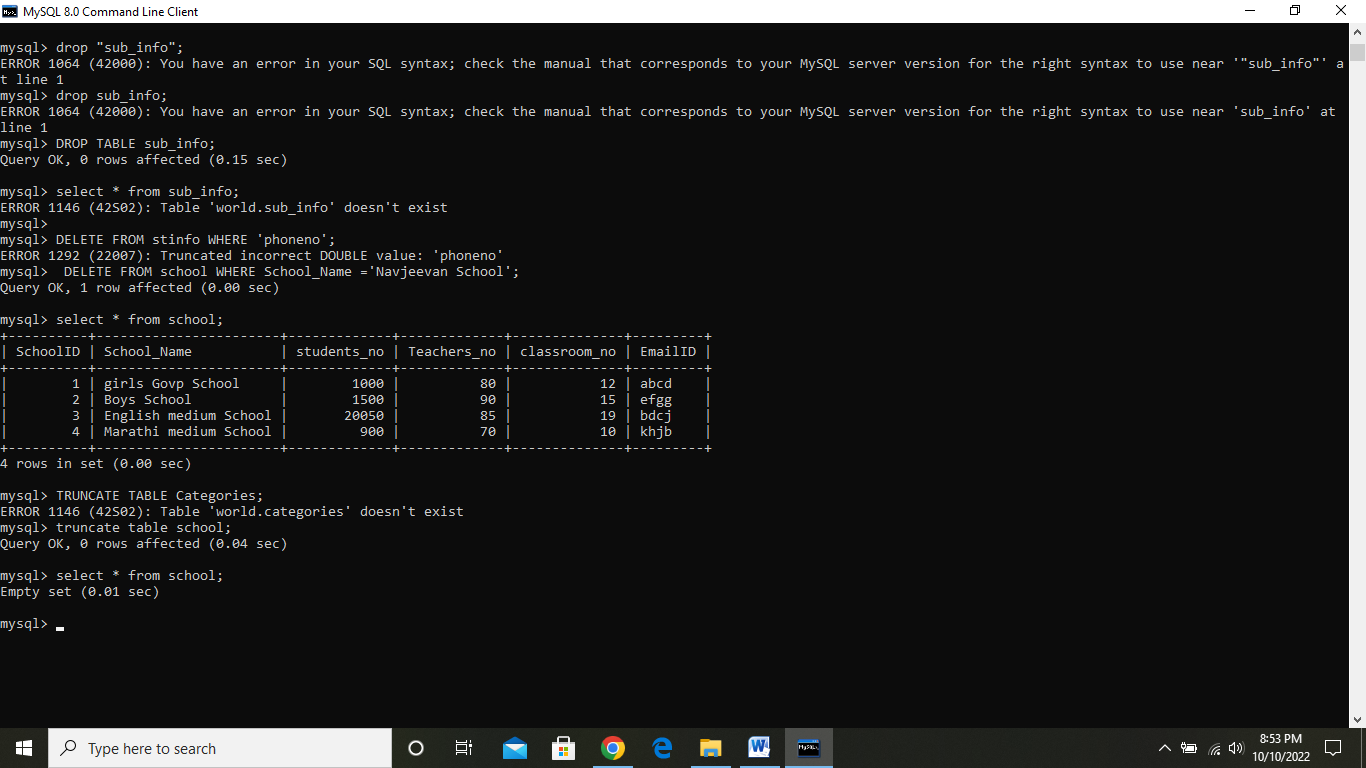
**DELETE**

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

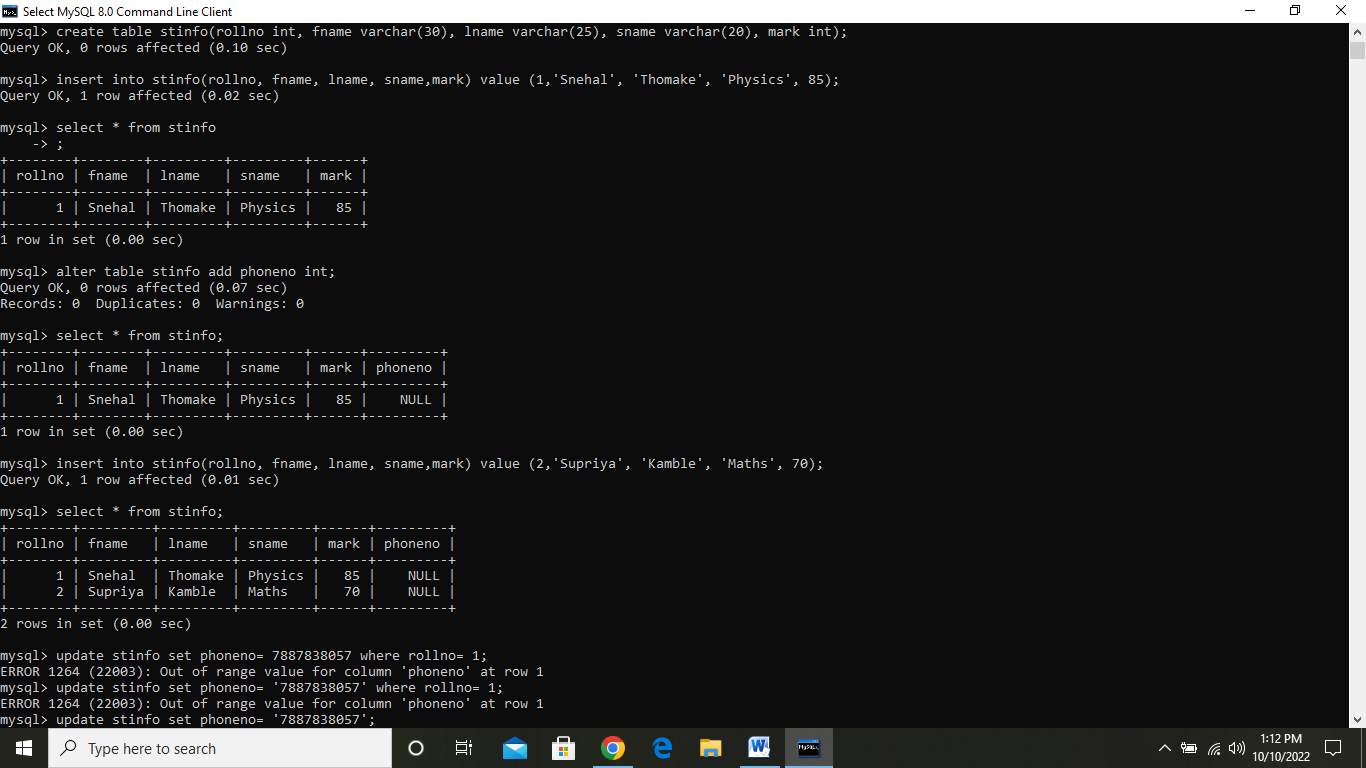


**TRUNCATE TABLE**

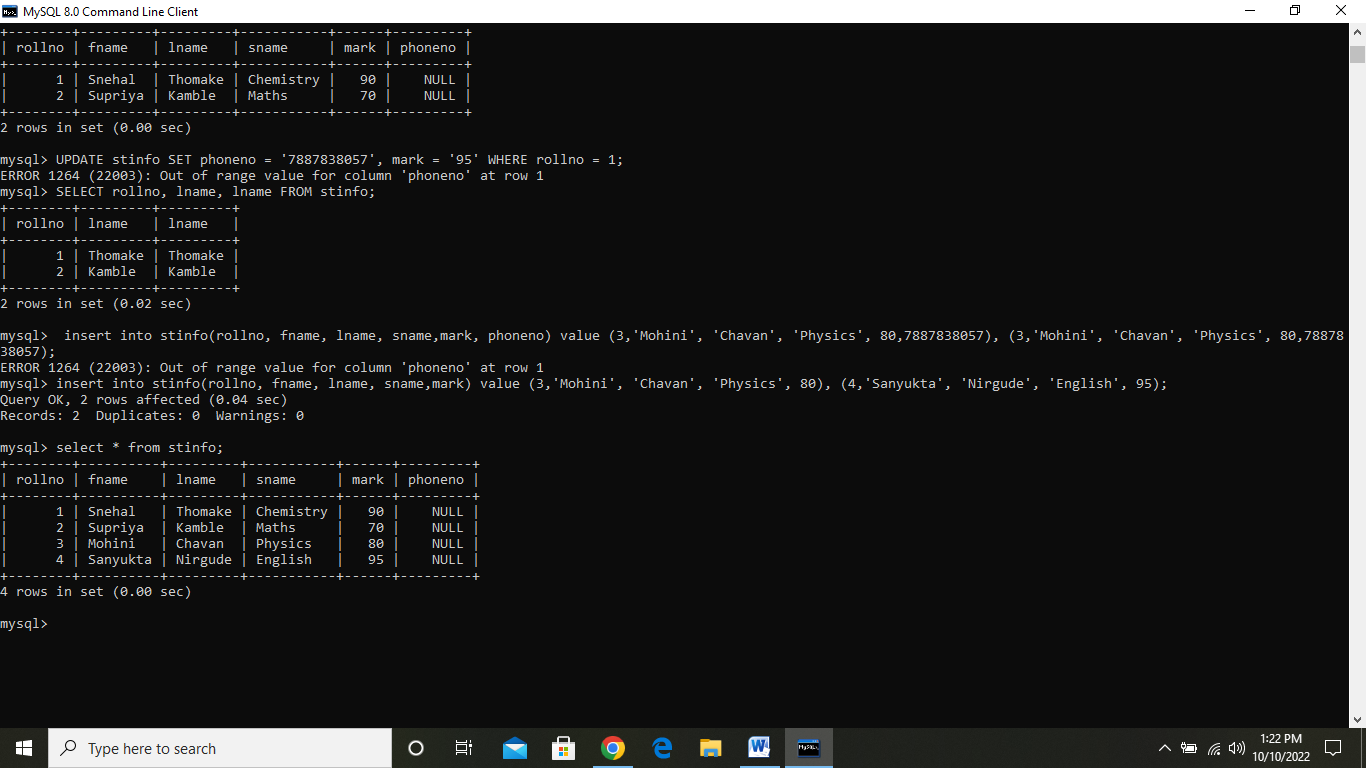
The TRUNCATE TABLE command deletes the data inside a table, but not the table itself.



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



**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

**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)

)

1. 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

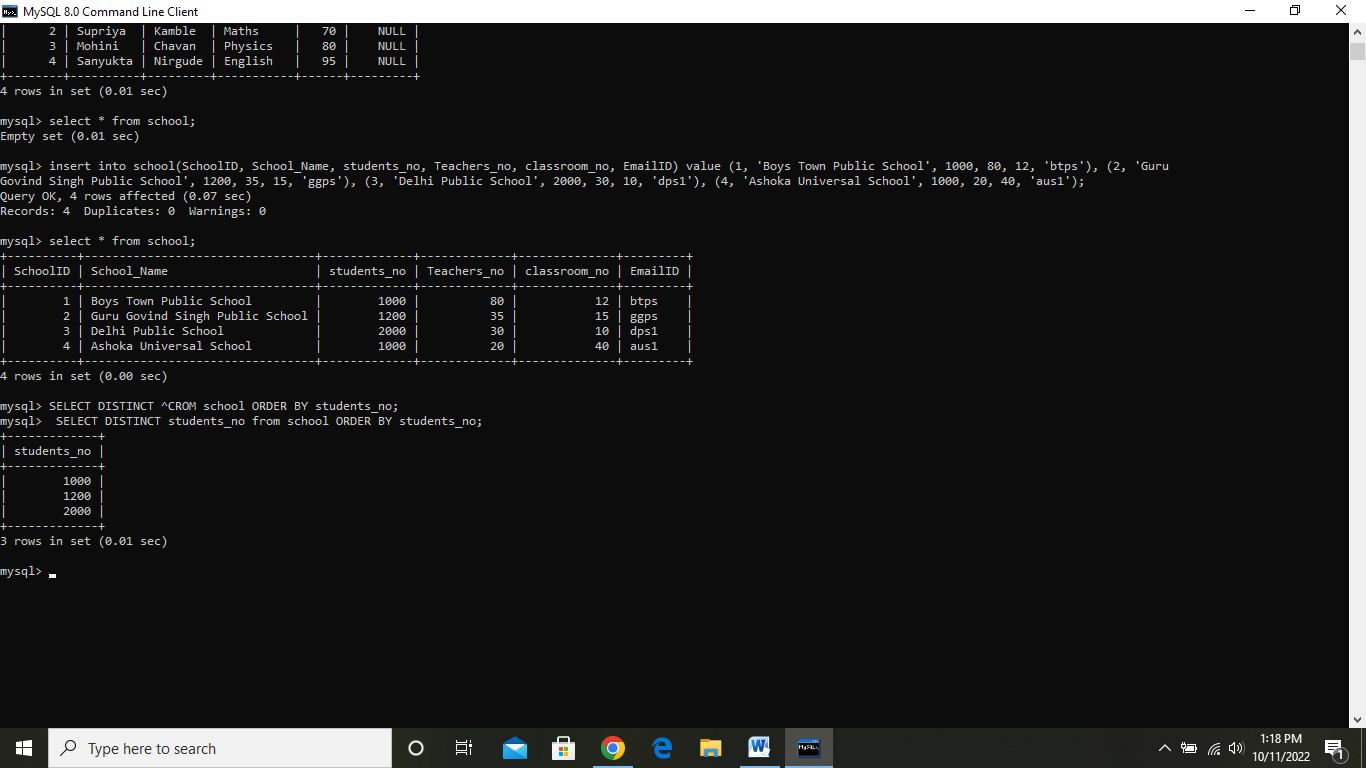
### Syntax:

**SELECT** **DISTINCT** expressions

**FROM** tables

[**WHERE** conditions];

Example and perform 🡪



# 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 🡪

# 

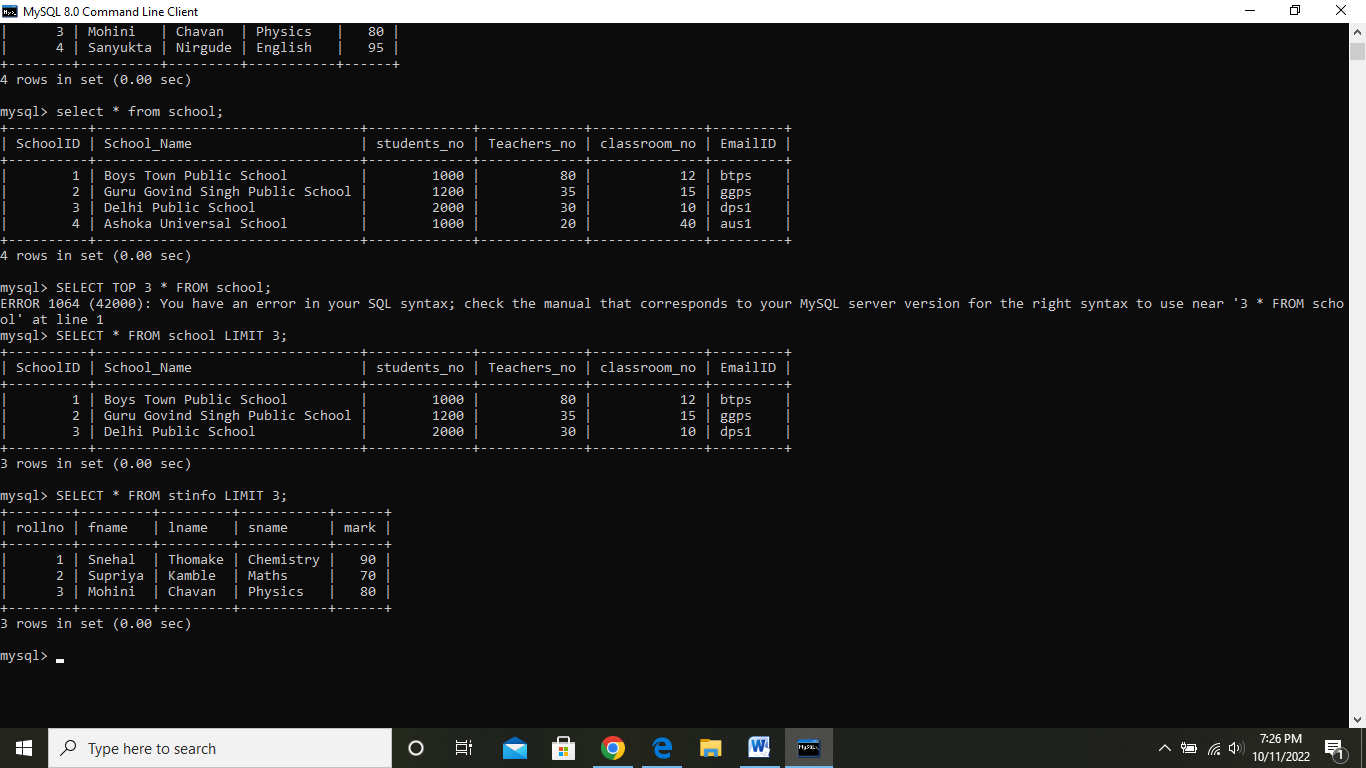
# 

# 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\_nameWHERE condition  
LIMIT number;



**First**

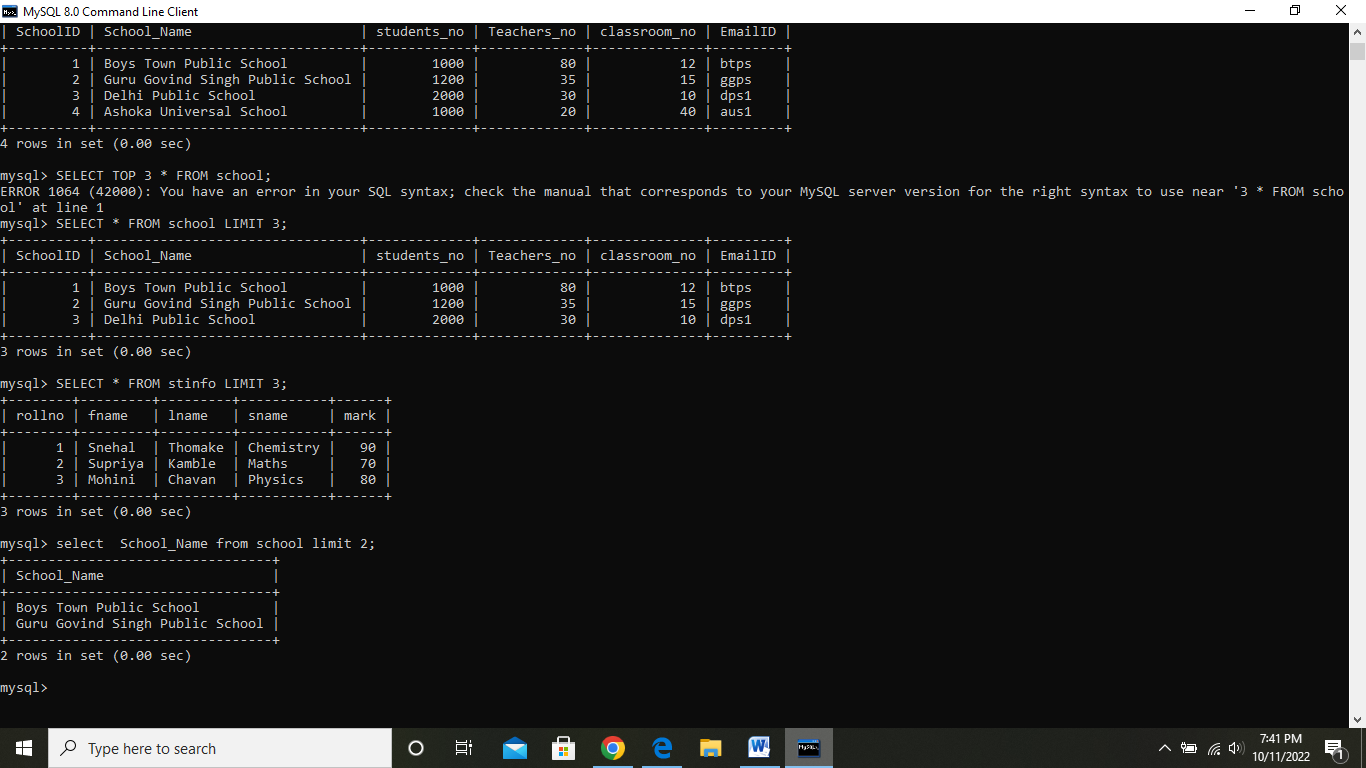
The MySQL first function is used to return the first value of the selected column. Here, we use limit clause to select first record or more.

**Syntax:**

**SELECT** column\_name

**FROM** table\_name

LIMIT 1;

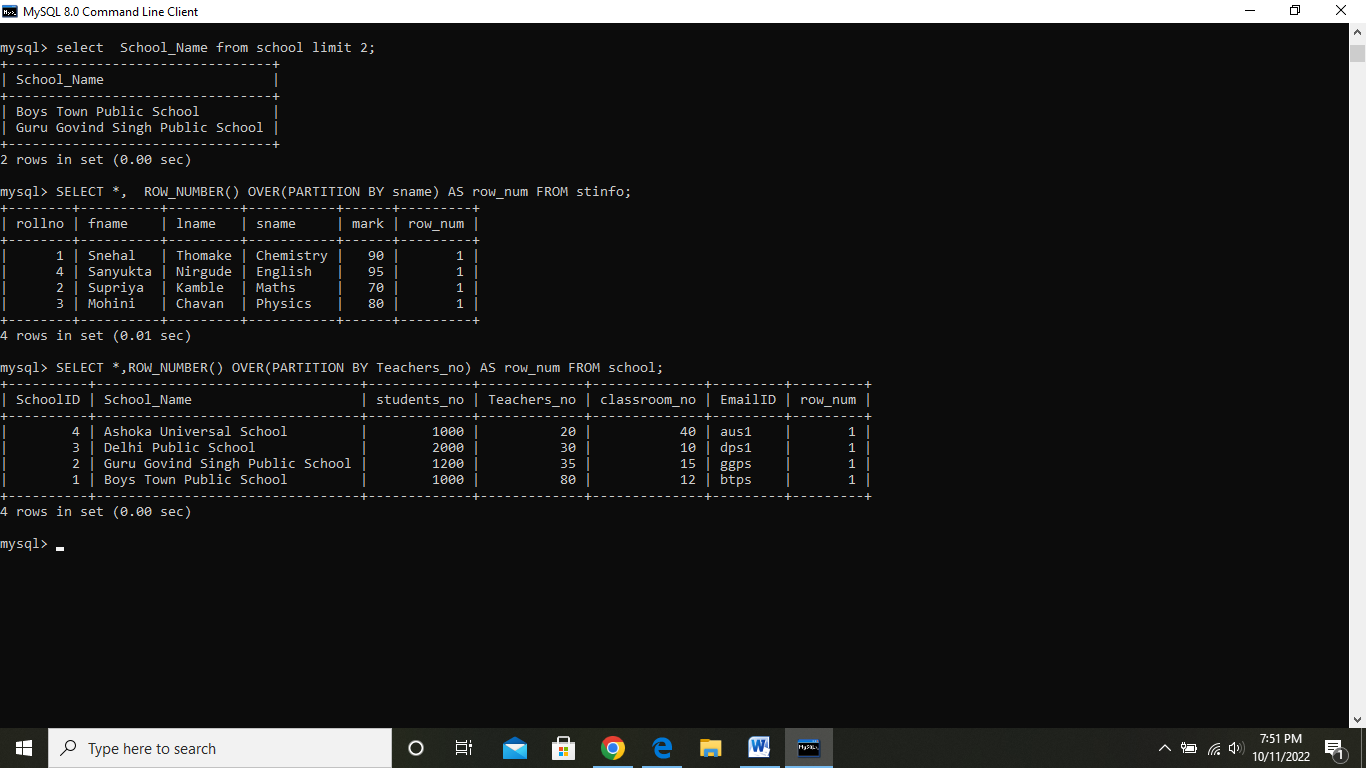


# ROW NUMBER

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.

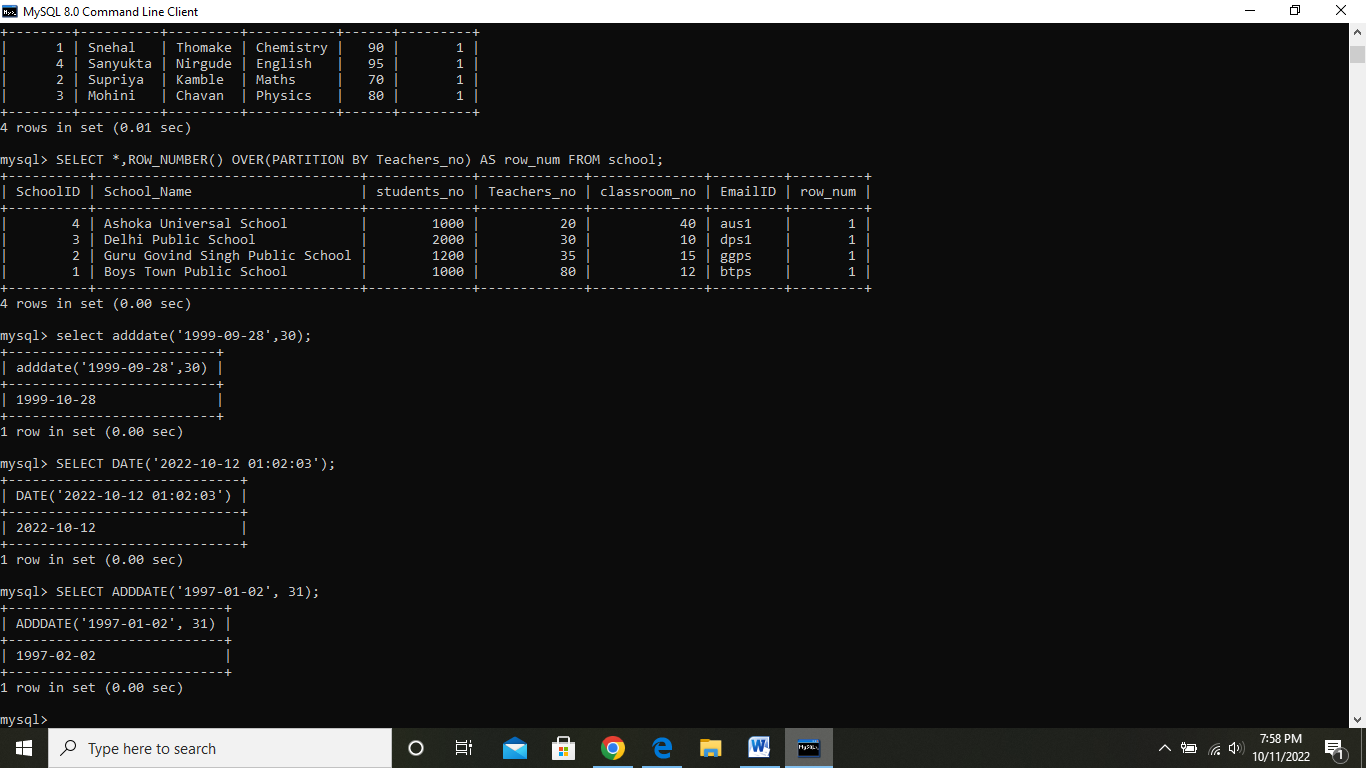
Syntax

ROW\_NUMBER() OVER (<partition\_definition> <order\_definition>)



## MySQL date/time

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



# 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

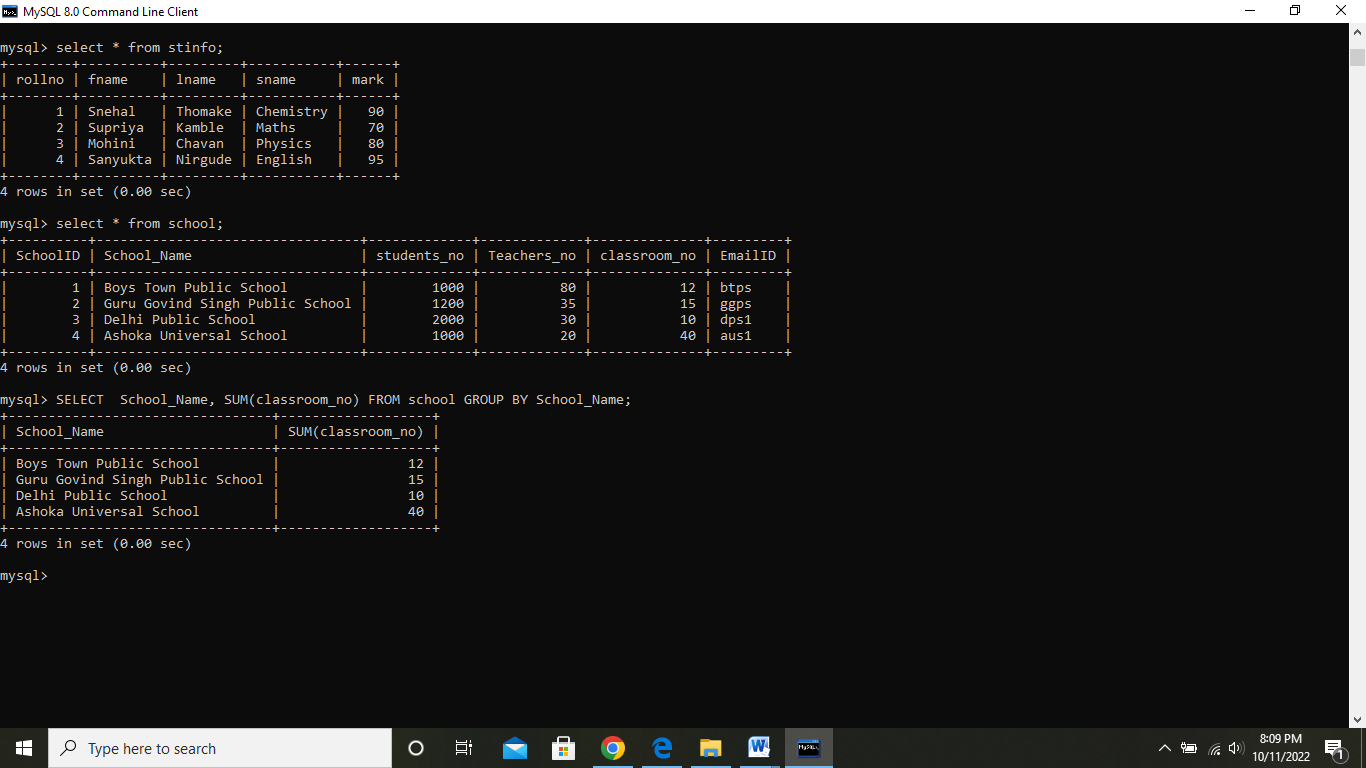
FROM table\_name

WHERE [ conditions ]

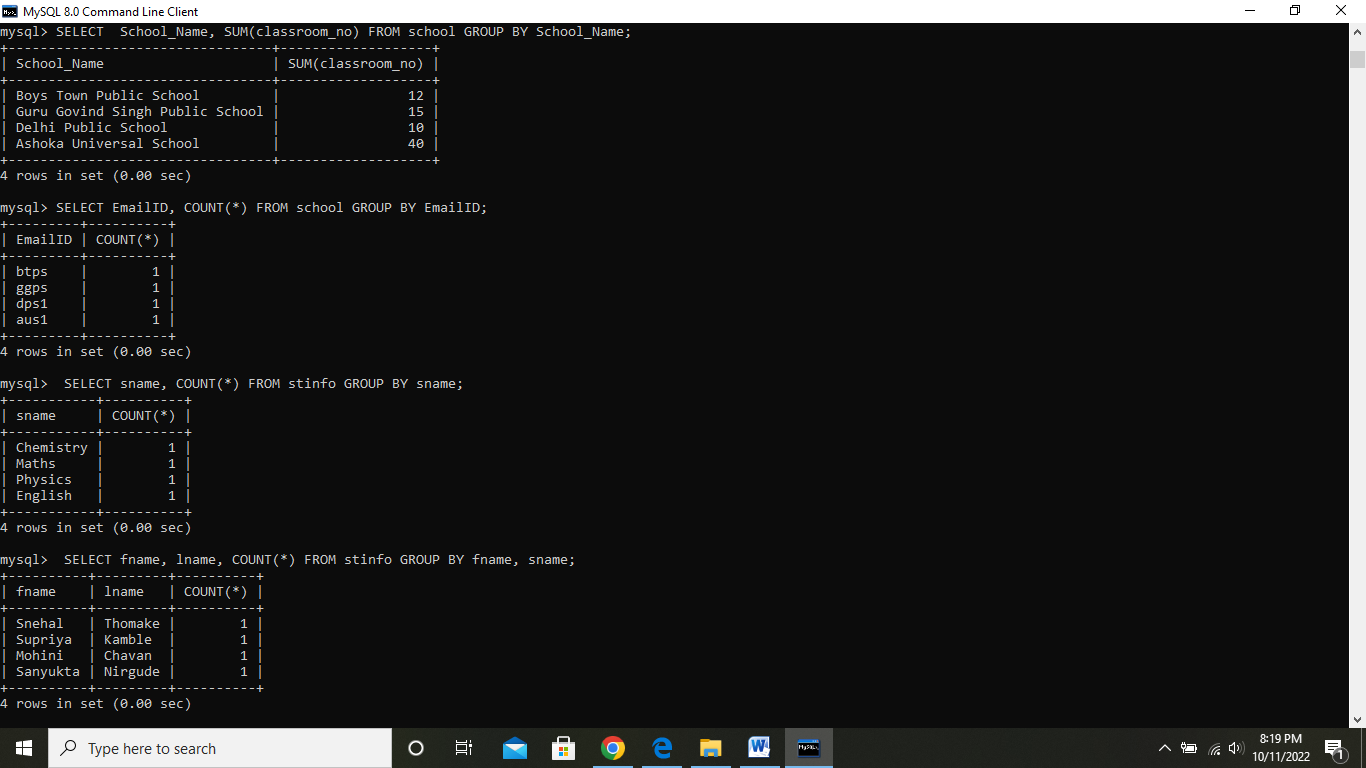
GROUP BY column1, column2

ORDER BY column1, column2

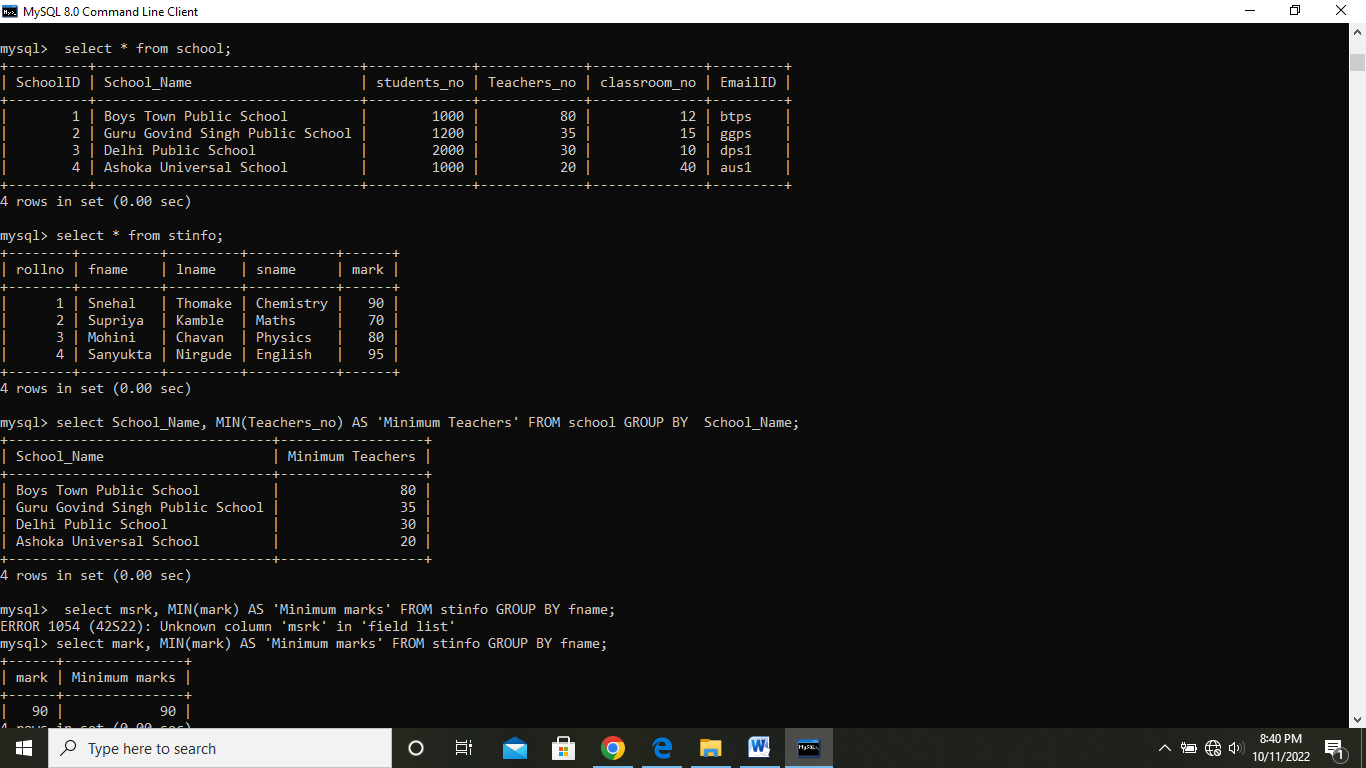
**SUM**



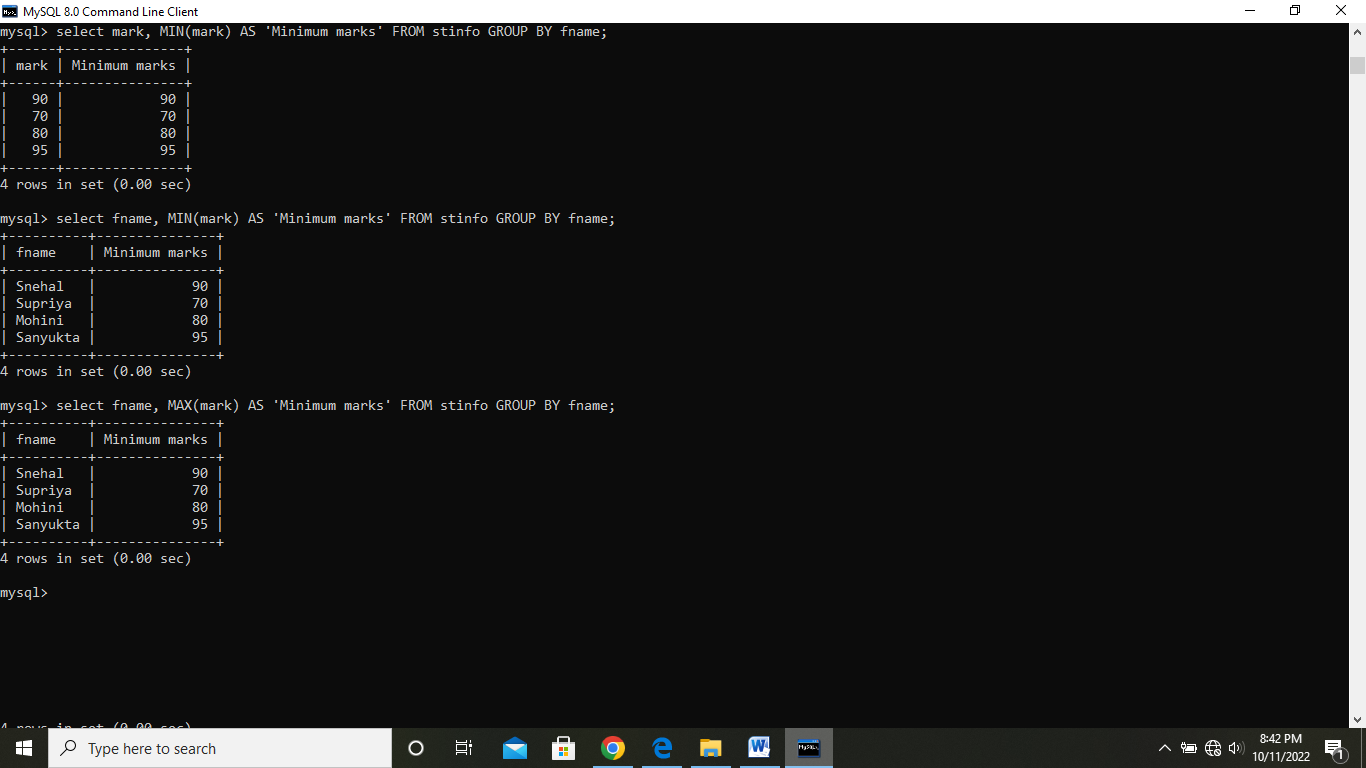
## COUNT



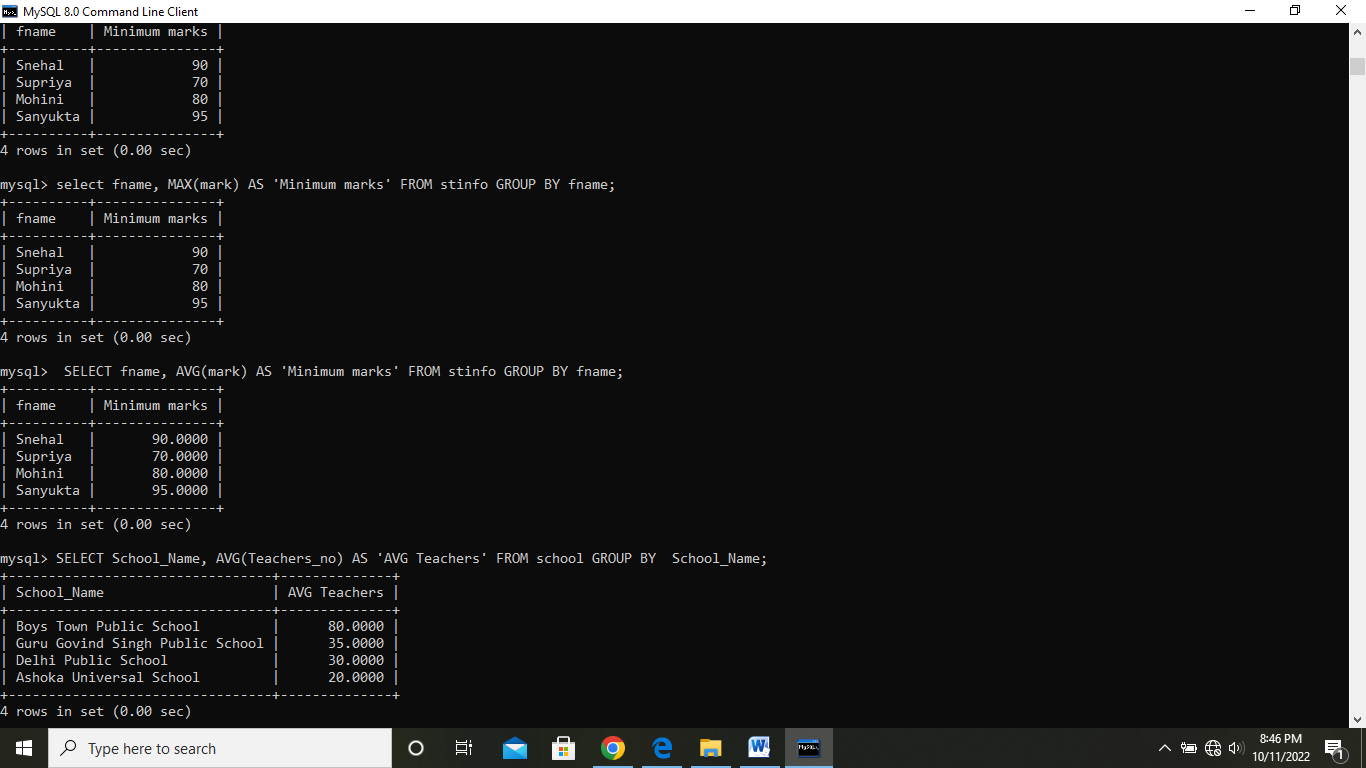
## MIN function



## MAX function



## AVG function



SQL LOGICAL OPERATORS

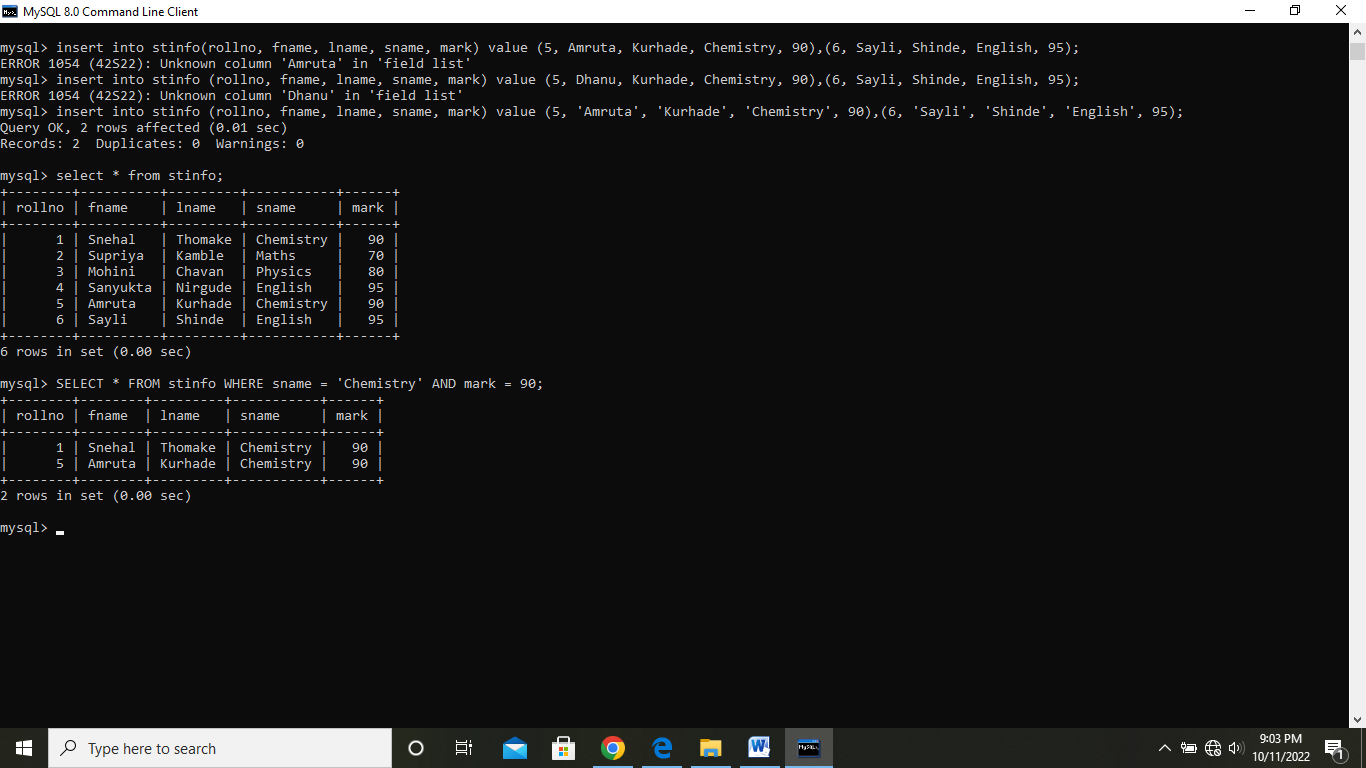
The Logical Operator is nothing but which returns the result in one form, i.e., either it will display the query is true, or the query is false. The results displayed to combine or merge more than one true or false data.

**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

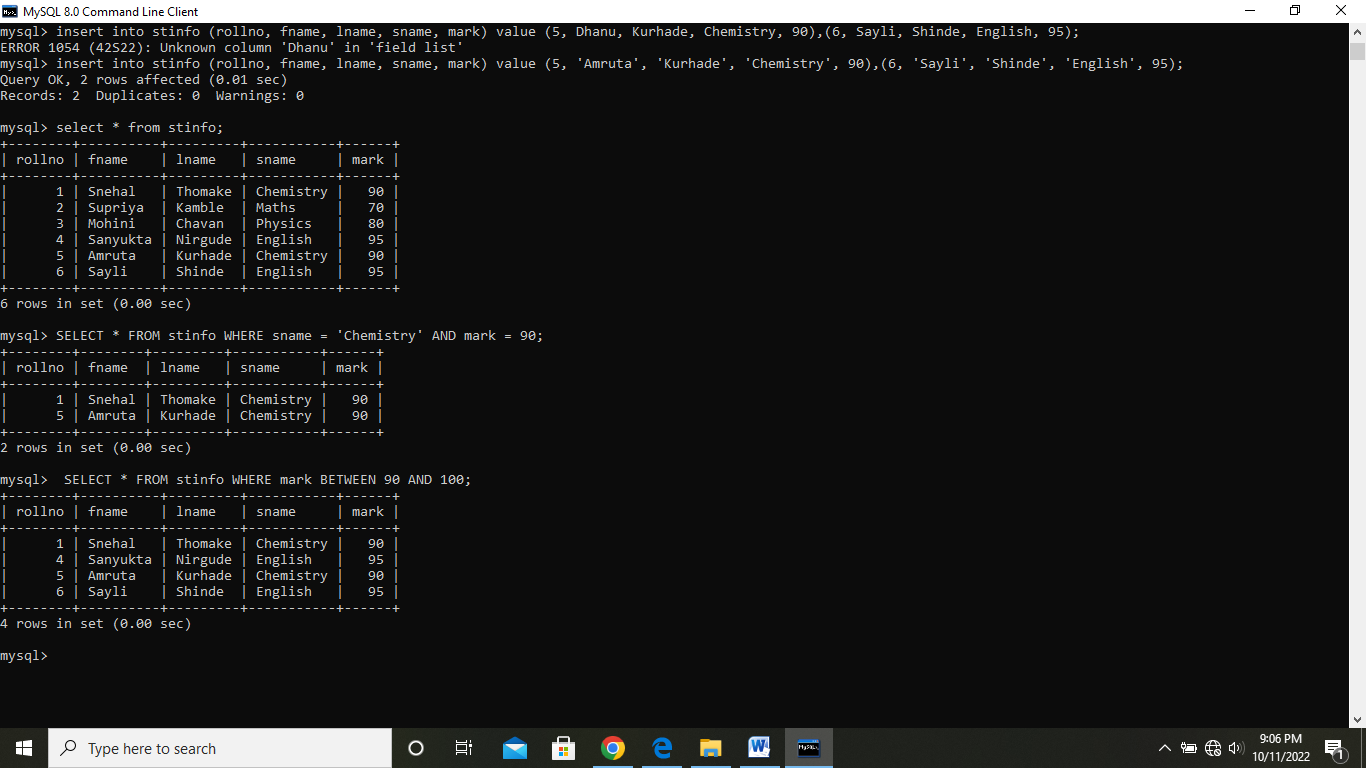
## 1. SQL AND Operator

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.



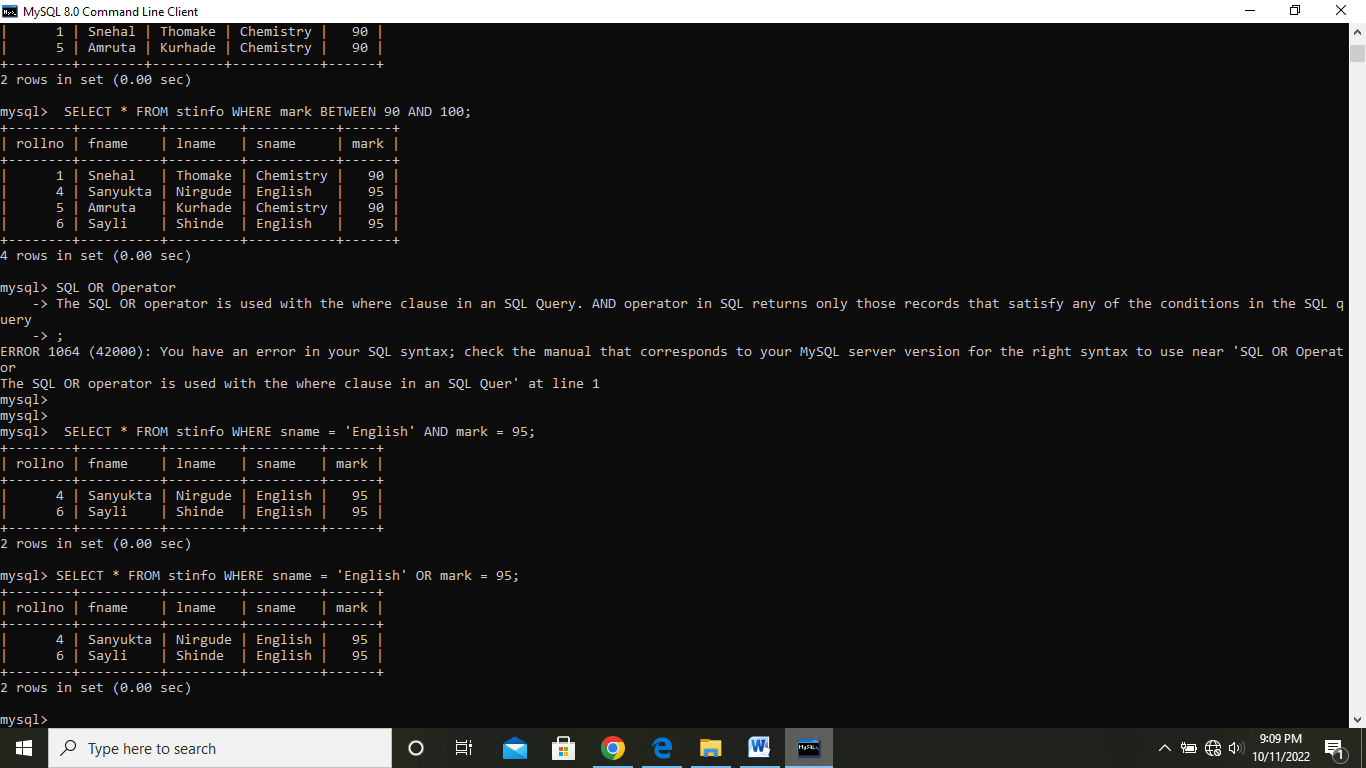
## SQL BETWEEN Operator

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.



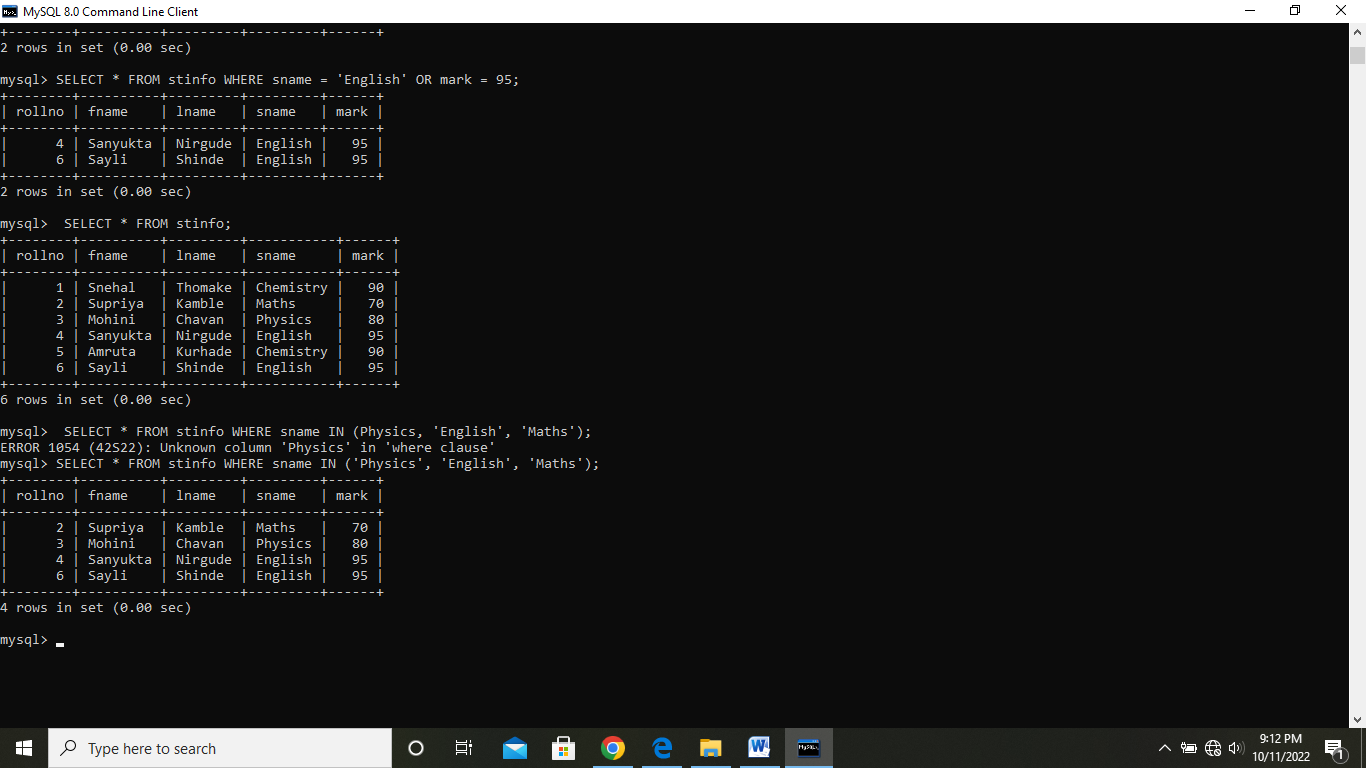
## SQL OR Operator

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.



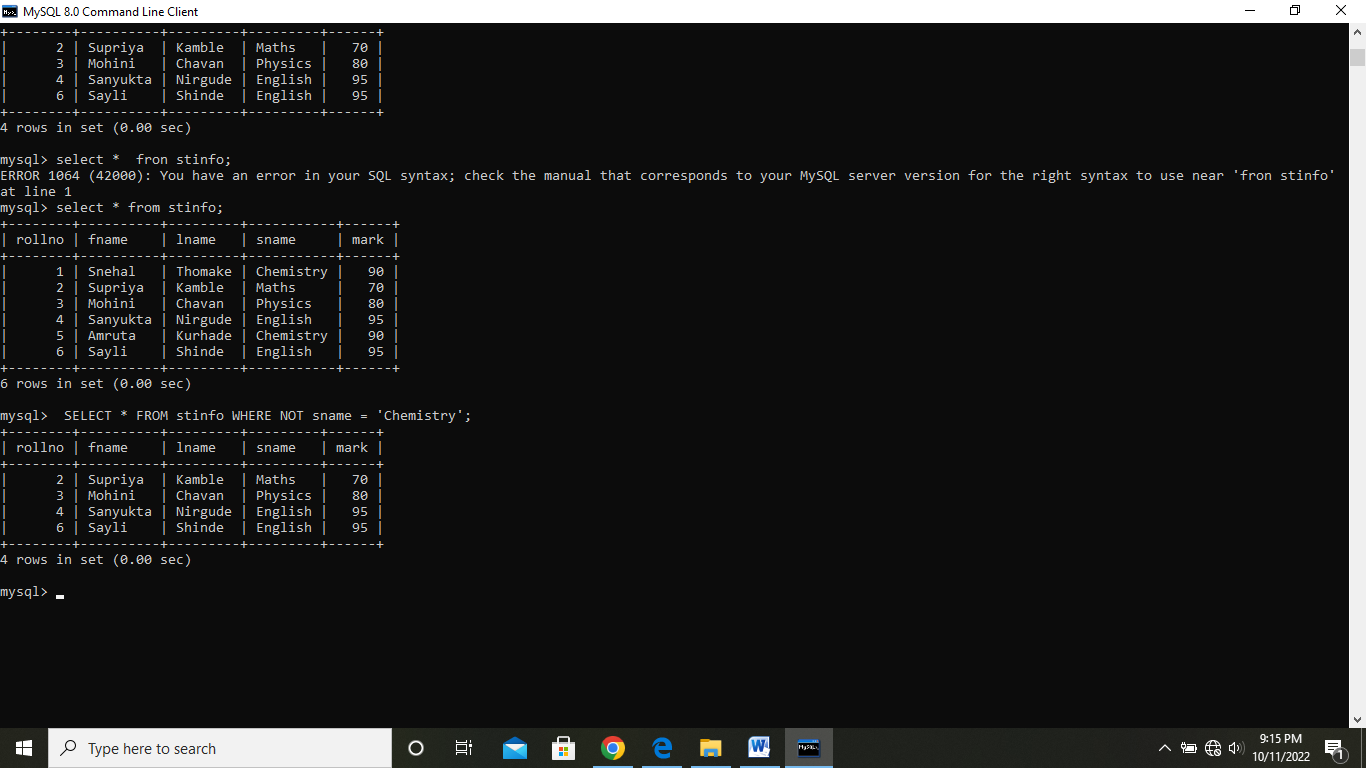
**SQL IN Operator**

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.



## 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.



**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))

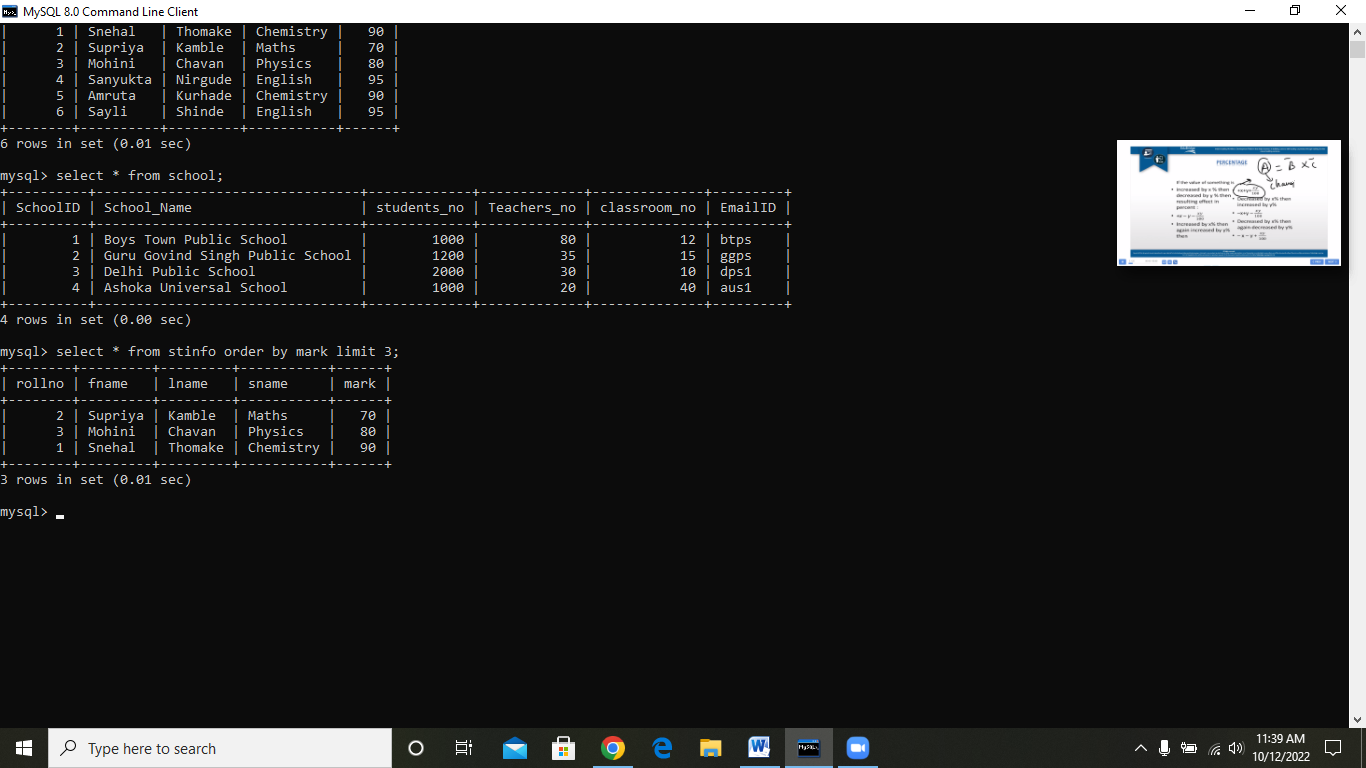
);

# 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;



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

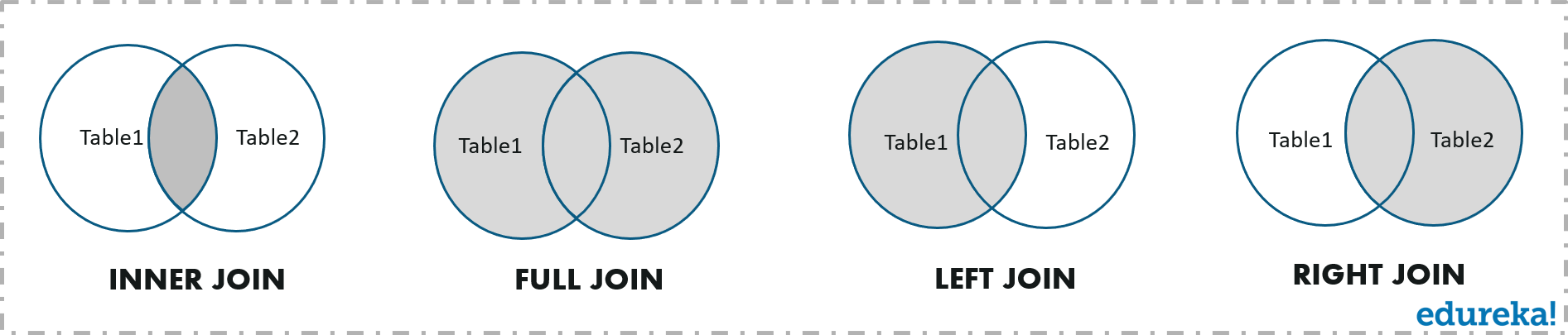
**JOIN**

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**

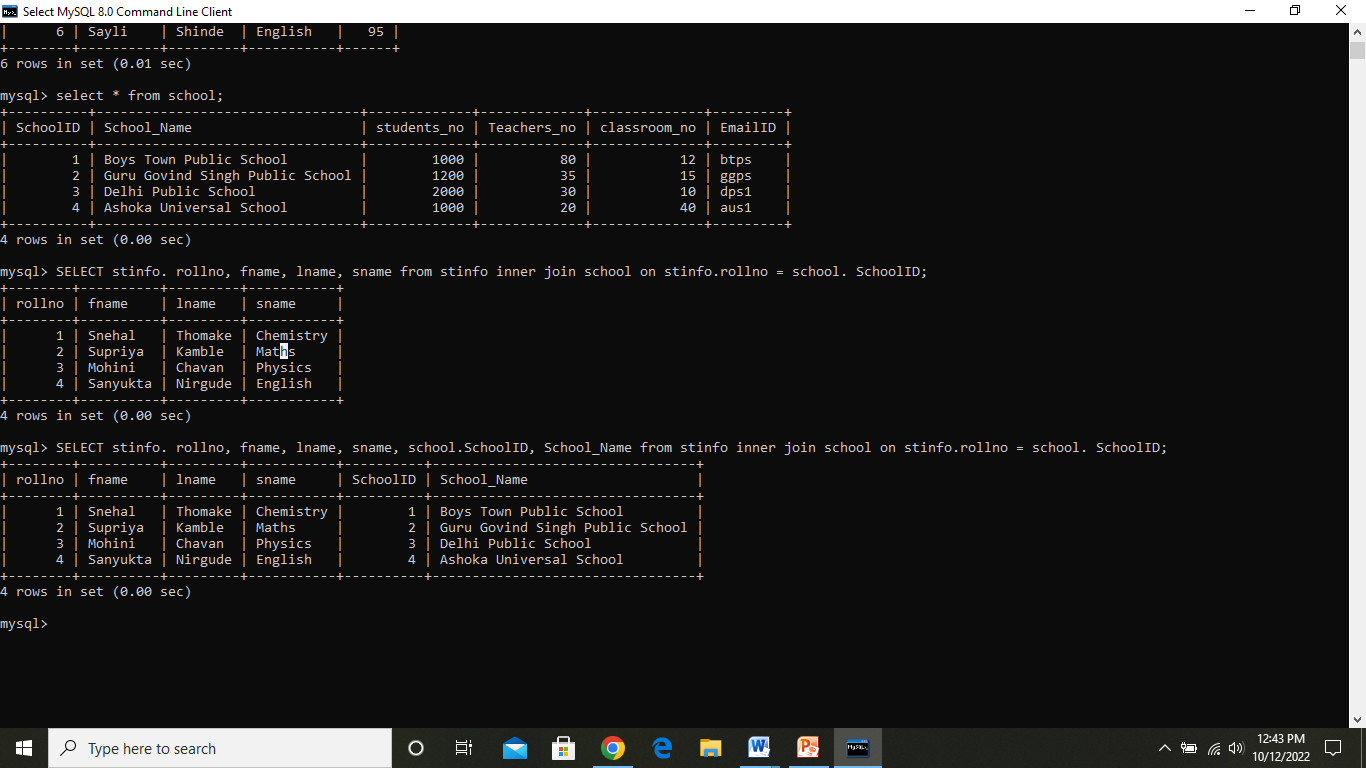
This type of join returns those records which have matching values in both tables. So, if you perform an INNER join operation between the Employee table and the Projects table, all the tuples which have matching values in both the tables will be given as output.

**Syntax:**

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

INNER JOIN Table2

ON Table1.MatchingColumnName = Table2.MatchingColumnName;



**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.

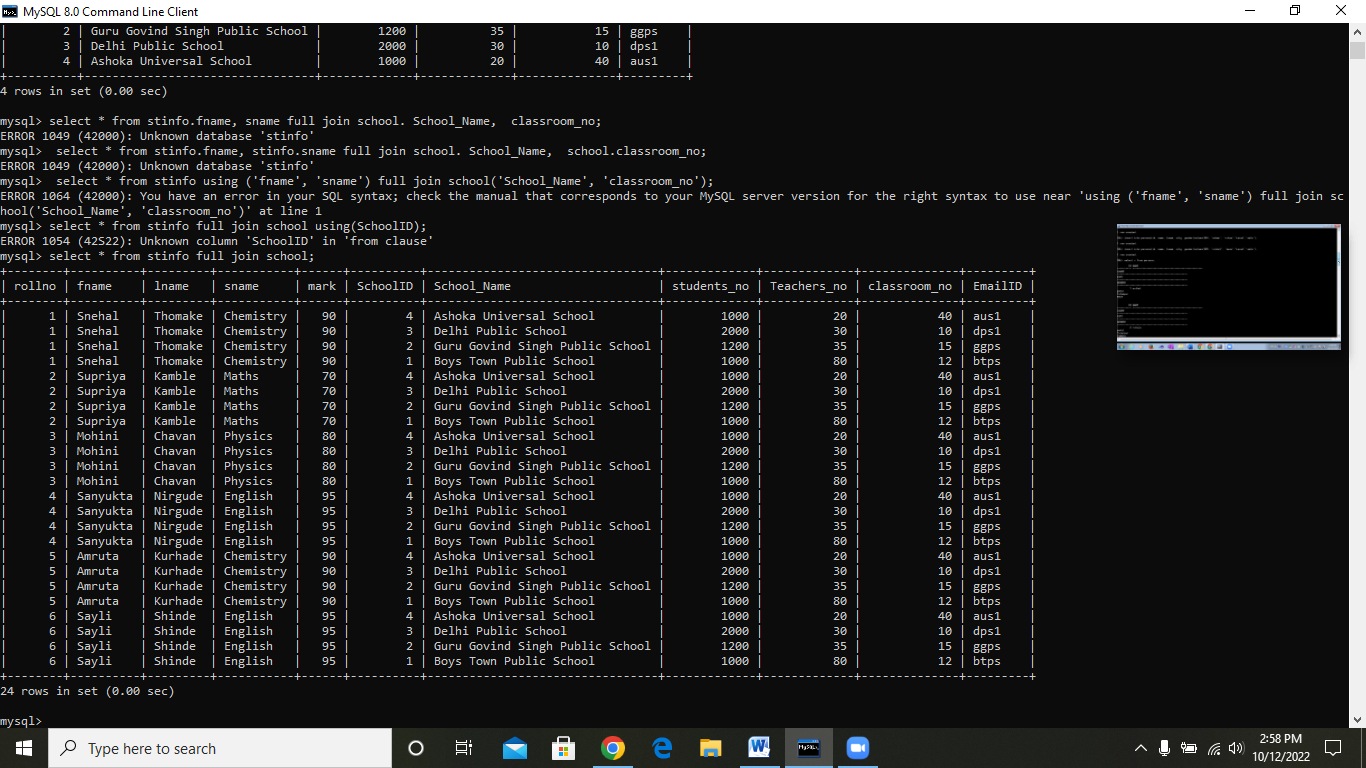
**Syntax**

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

FROM table1

FULL JOIN table2

ON table1.matching\_column = table2.matching\_column;



**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.

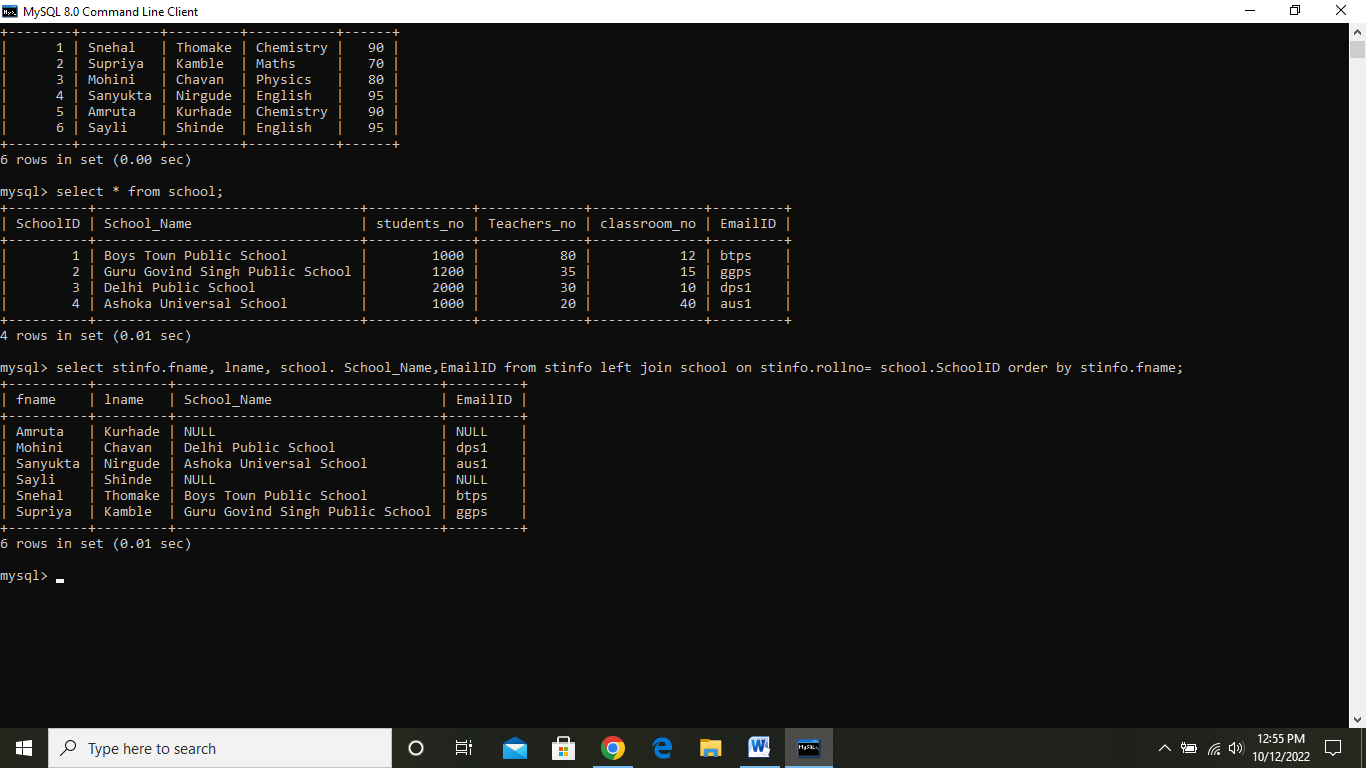
**Syntax:**

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

FROM table1

LEFT JOIN table2

ON table1.matching\_column = table2.matching\_column;



**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.

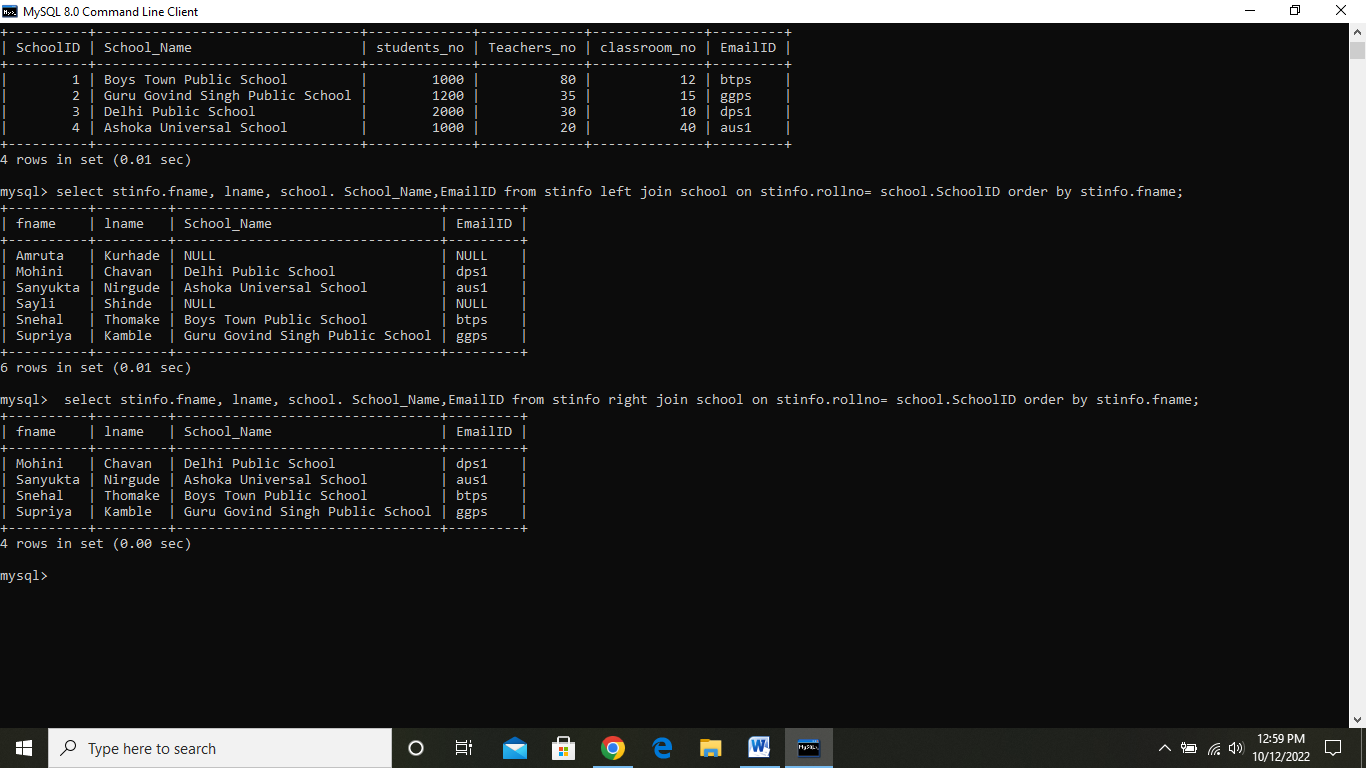
**Syntax**

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

FROM table1

RIGHT JOIN table2

ON table1.matching\_column = table2.matching\_column;



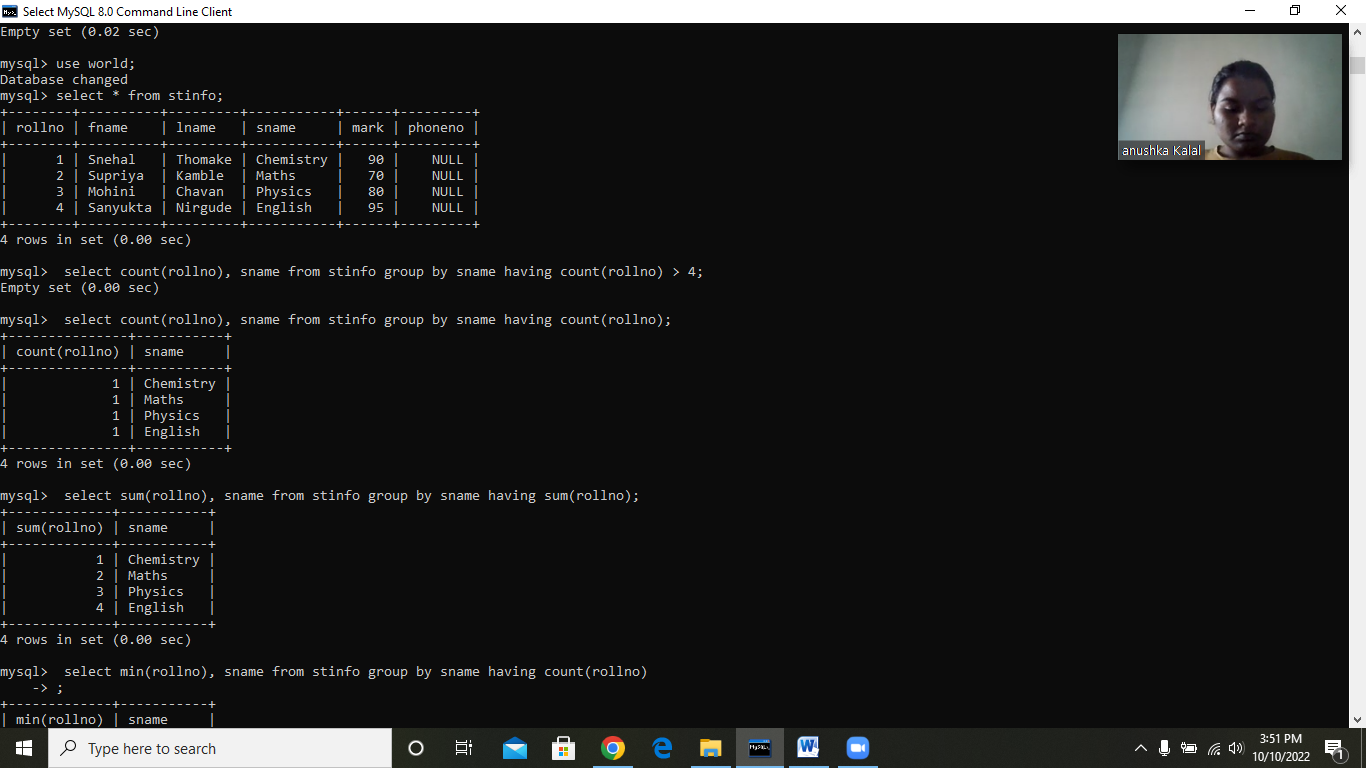
Date=10/10/2022 **Monday**

## SQL HAVING Clause

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

### HAVING Syntax

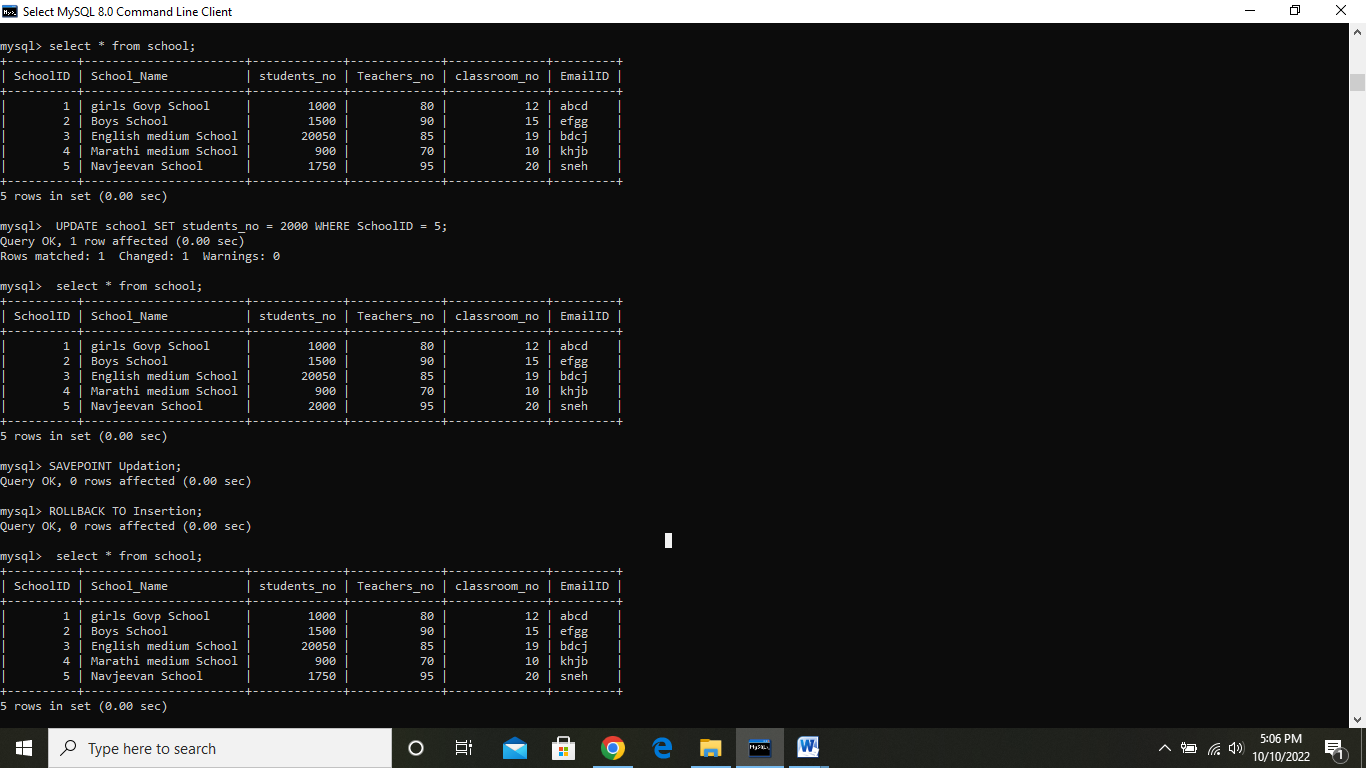
SELECT column\_name(s)  
FROM table\_name  
WHERE condition  
GROUP BY column\_name(s)HAVING conditionORDER BY column\_name(s);



# 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.





**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**

## 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;

## 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.

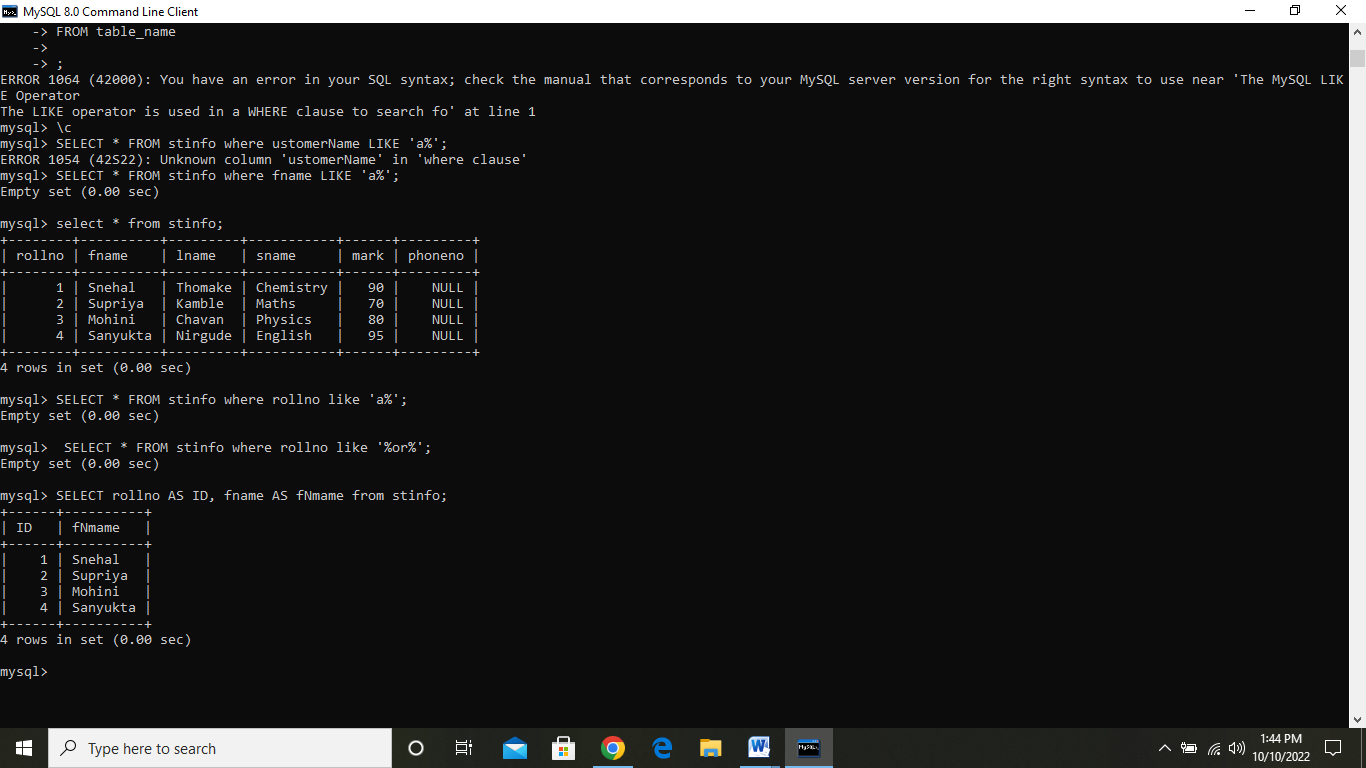
An alias is created with the AS keyword.

### Alias Column Syntax

SELECT column\_name AS alias\_name  
FROM table\_name;

### Alias Table Syntax

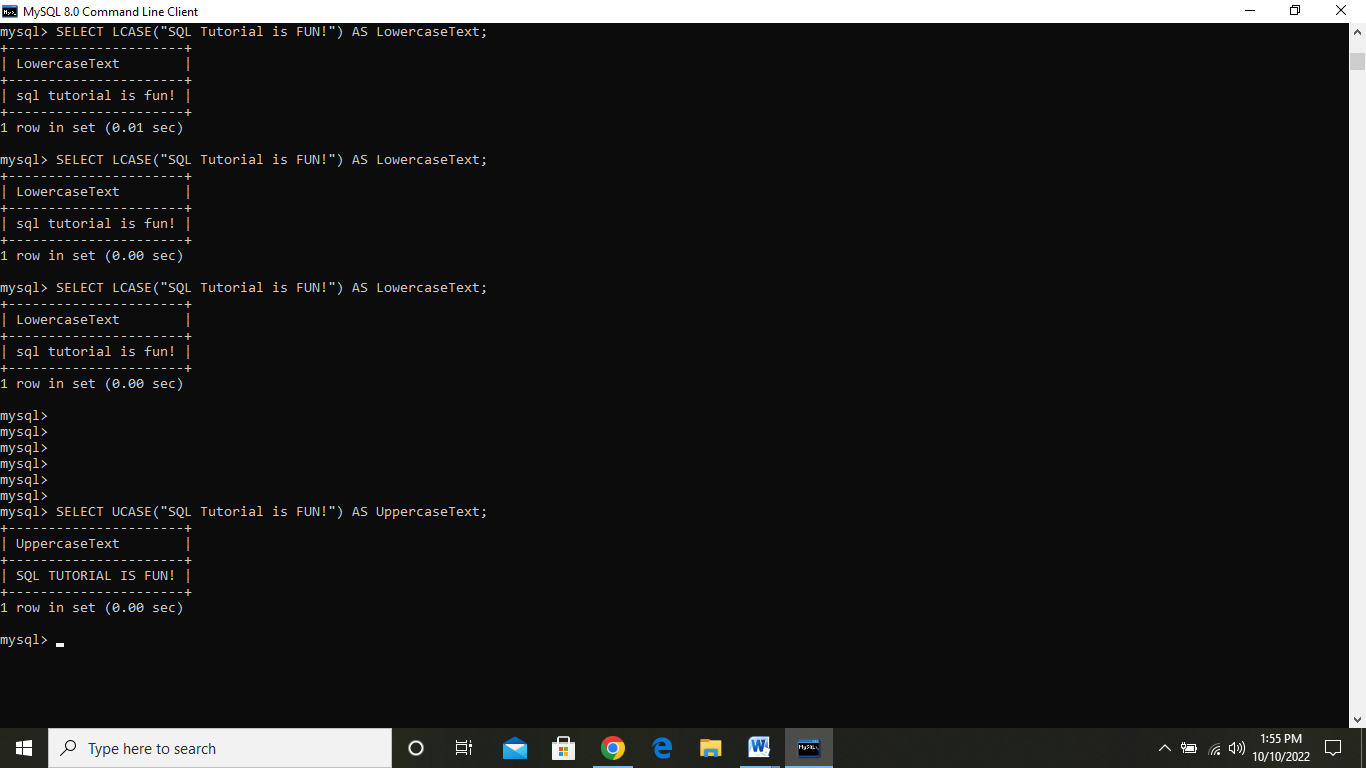
SELECT column\_name(s)  
FROM table\_name AS alias\_name;



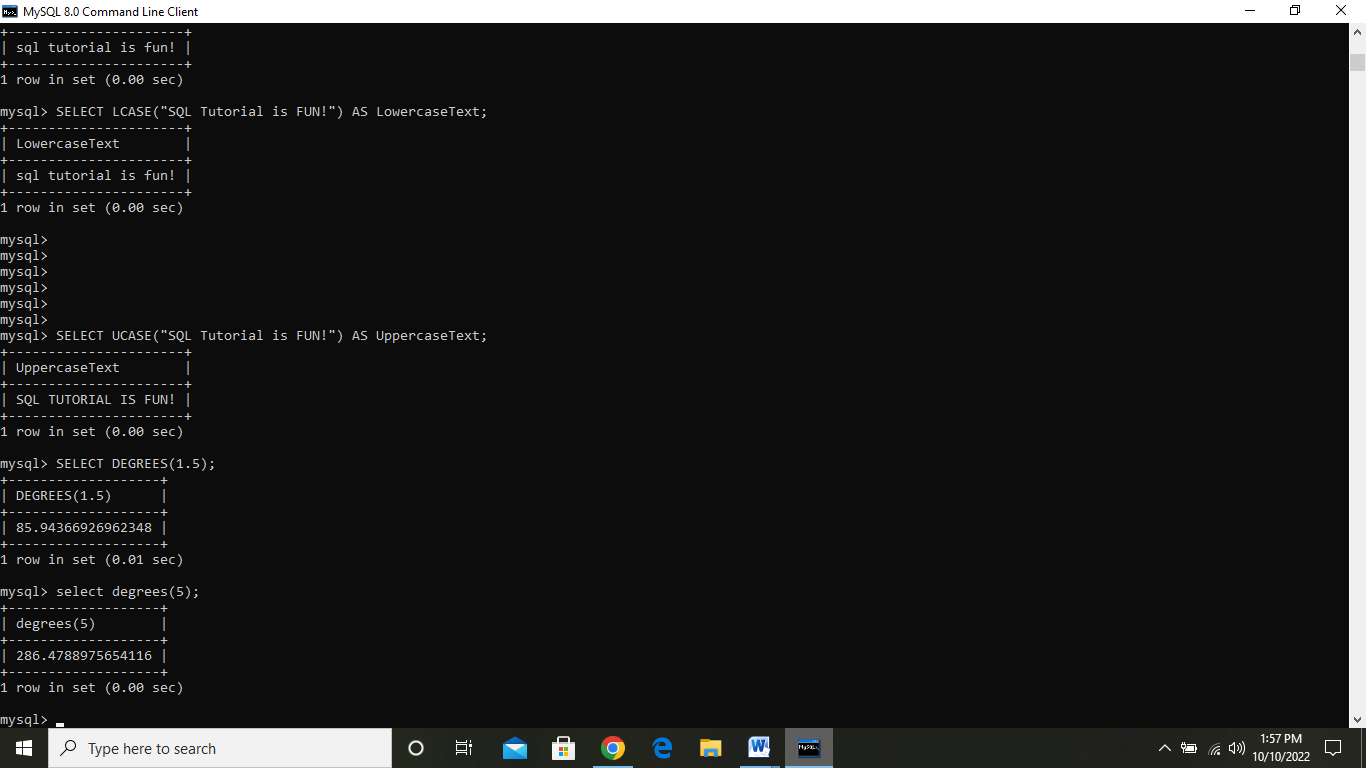
# LOWER CASE() Function

# 

# UPPER CASE() Function



**DEGREES () Function**



**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]



### 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**

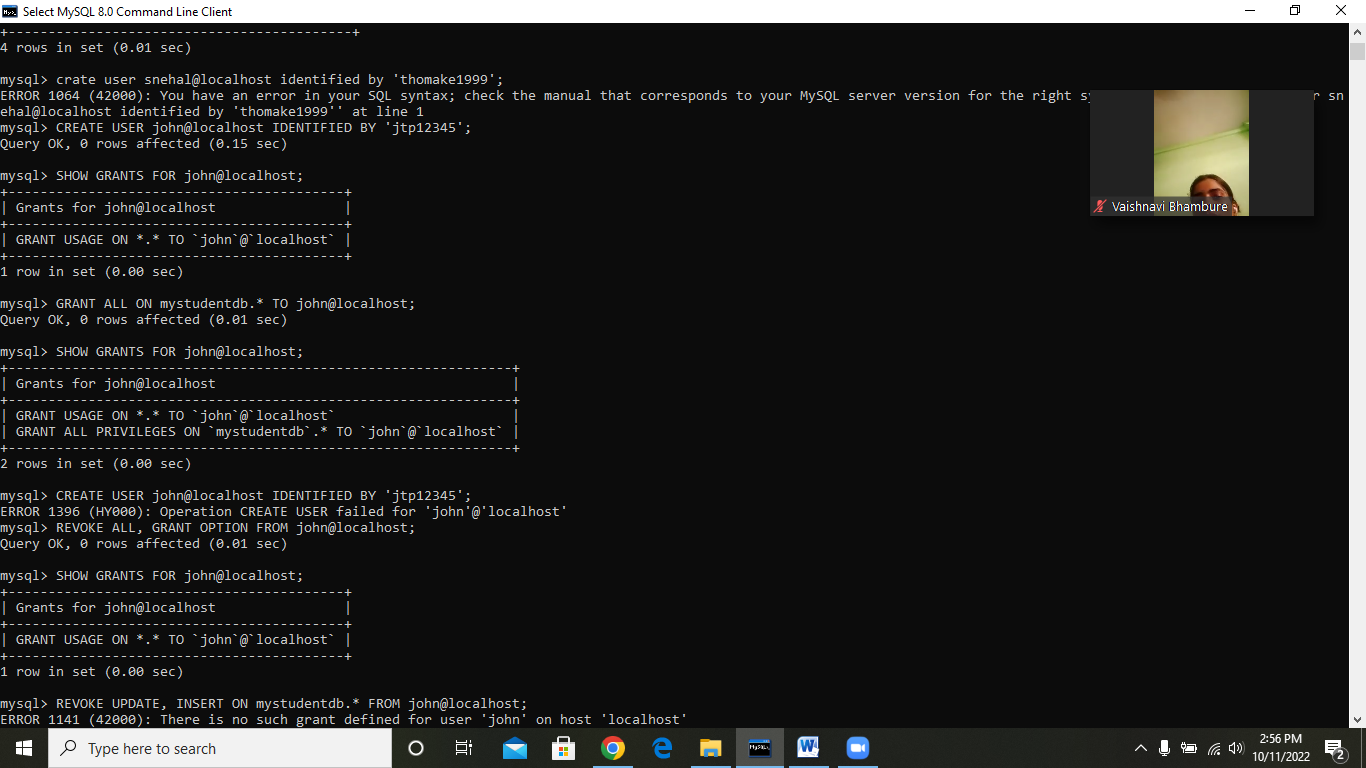
GRANT SELECT, UPDATE ON MY\_TABLE TO SOME\_USER, ANOTHER\_USER;



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

**Example**

REVOKE SELECT, UPDATE ON MY\_TABLE FROM USER1, USER2;

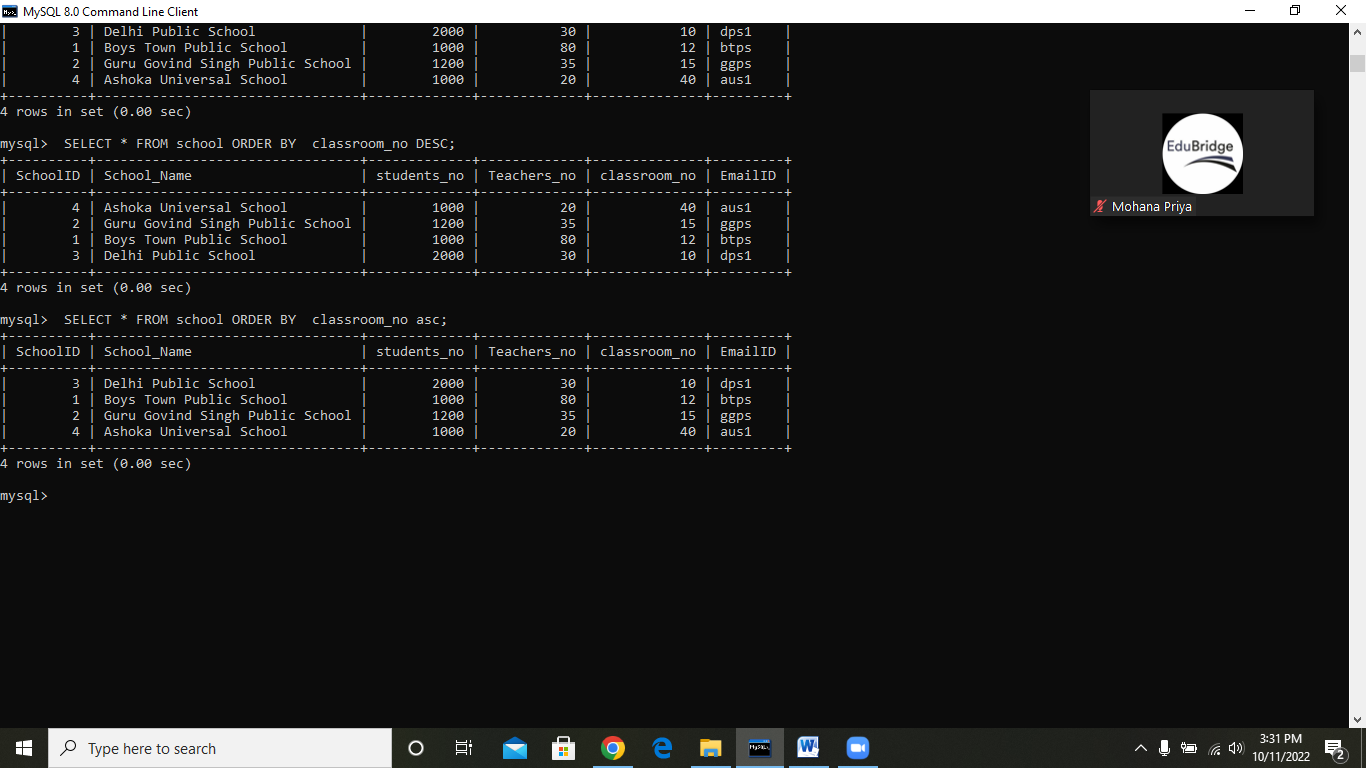


SQL ORDER BY Clause

* 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.

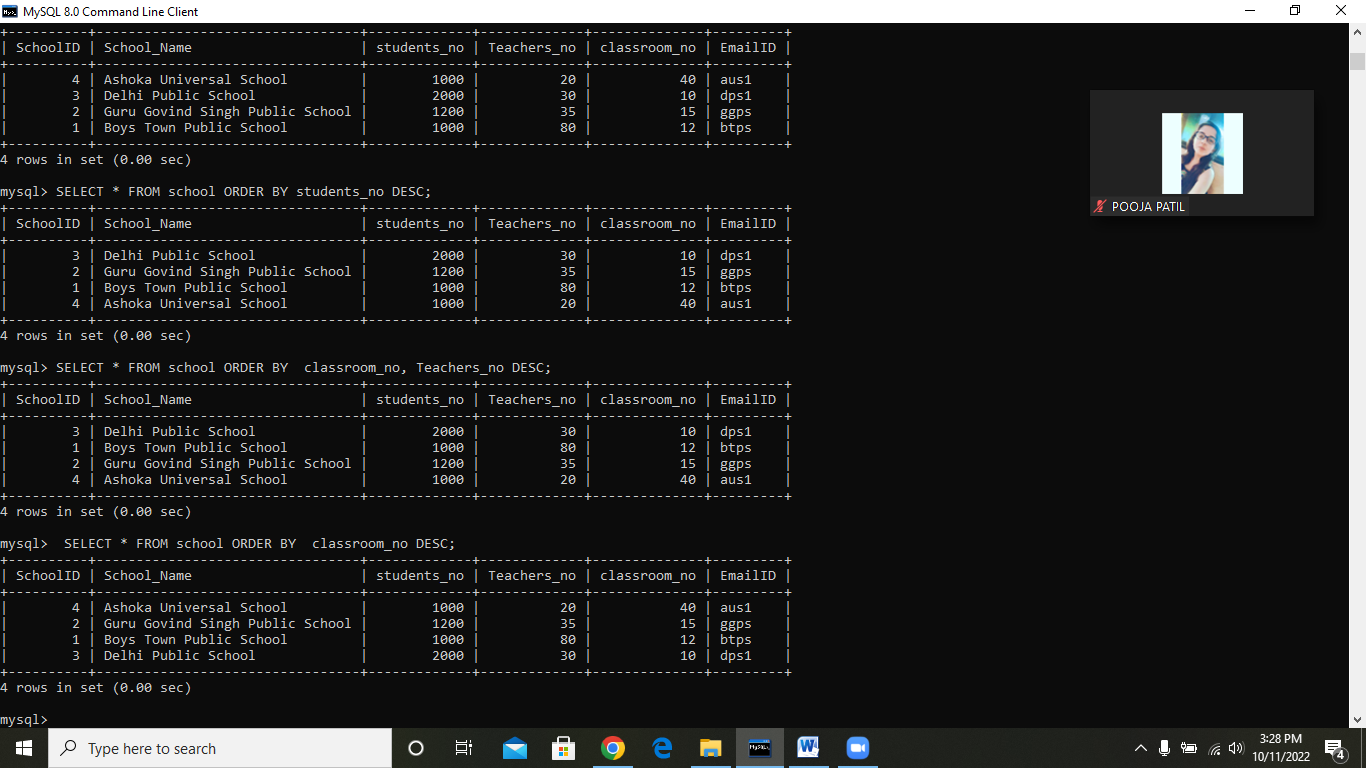
### Syntax to sort the records in ascending order:

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



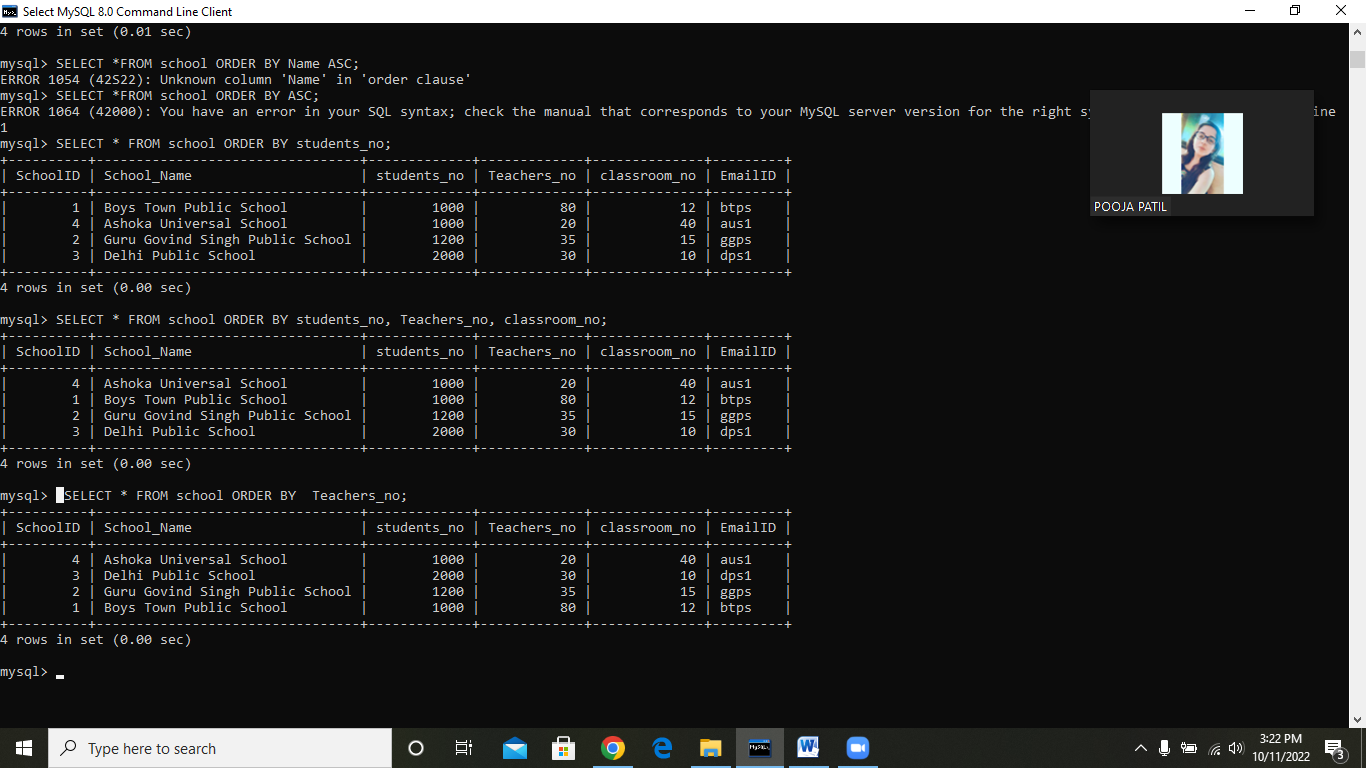
### Syntax to sort the records in descending order:

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



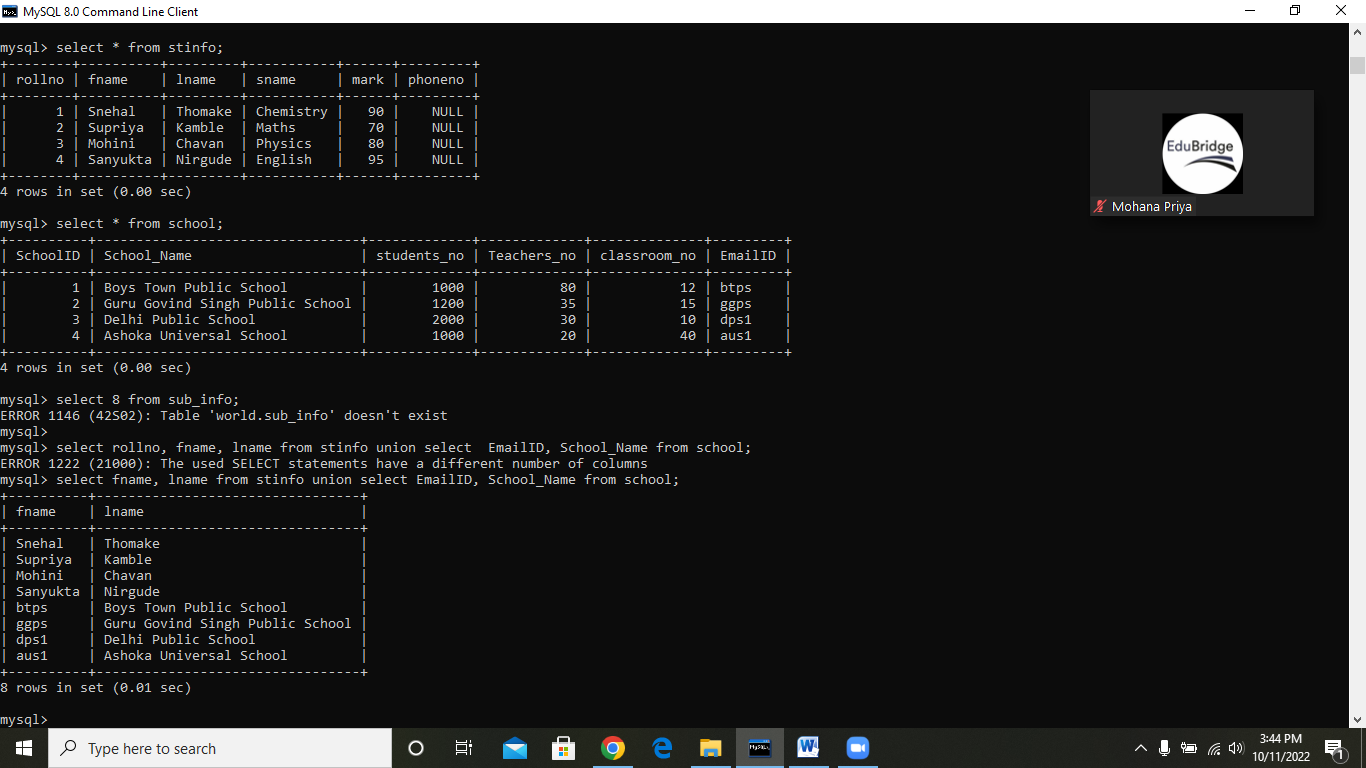
**Syntax to sort the records in ascending order without using ASC keyword:**

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



# 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).



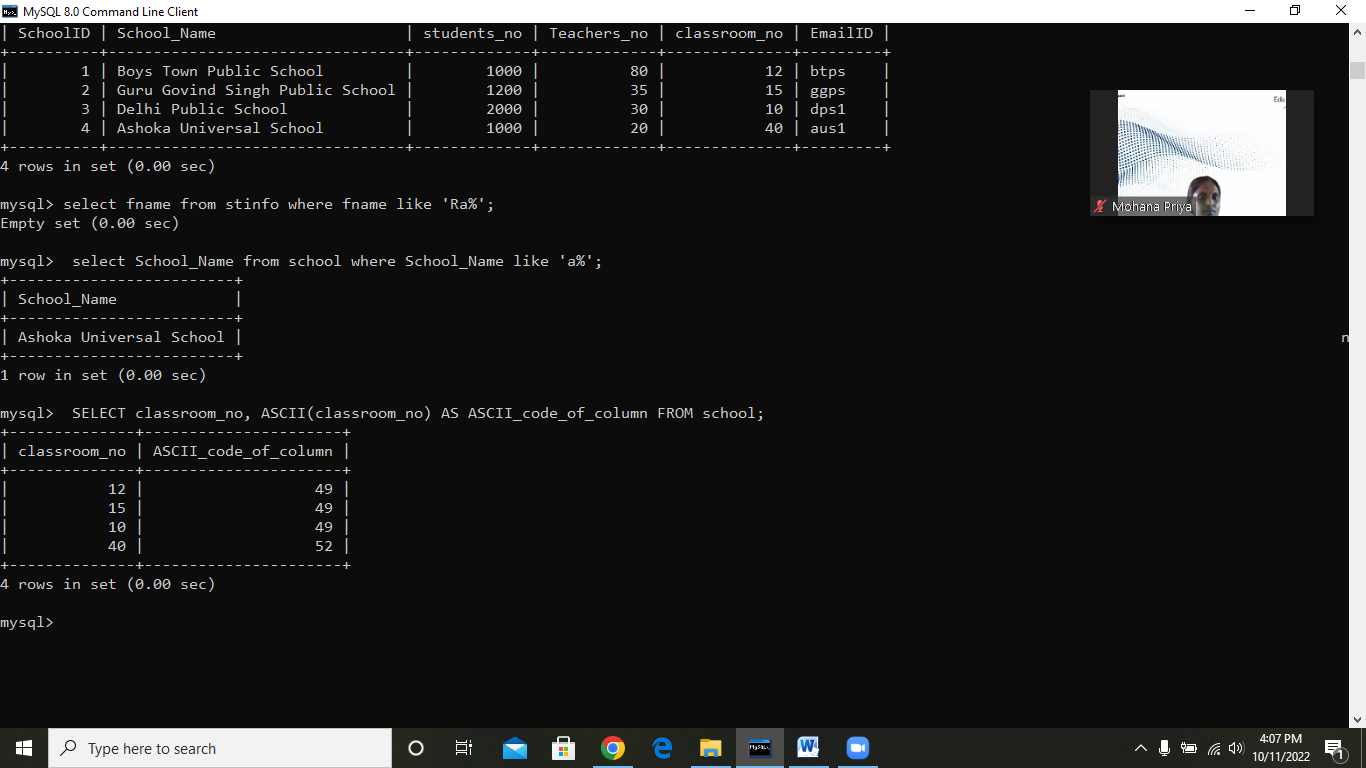
## ASCII String Function

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;



**Date = 12/10/2022**

