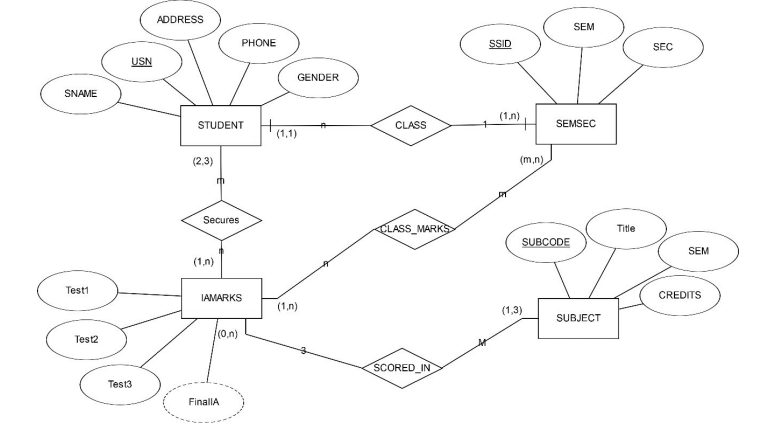
**Database Management System**

**Practical File**

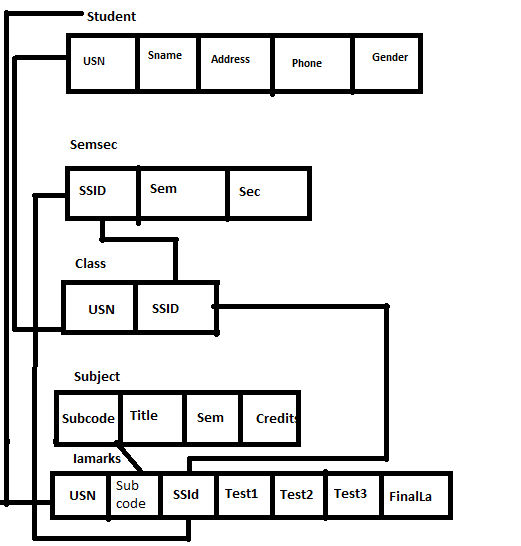
**Experiment 1**

Draw an E-R diagram from given entities and their attributes

****

**Experiment 2**

Convert the E-R diagram into a Relational model with proper constraints.

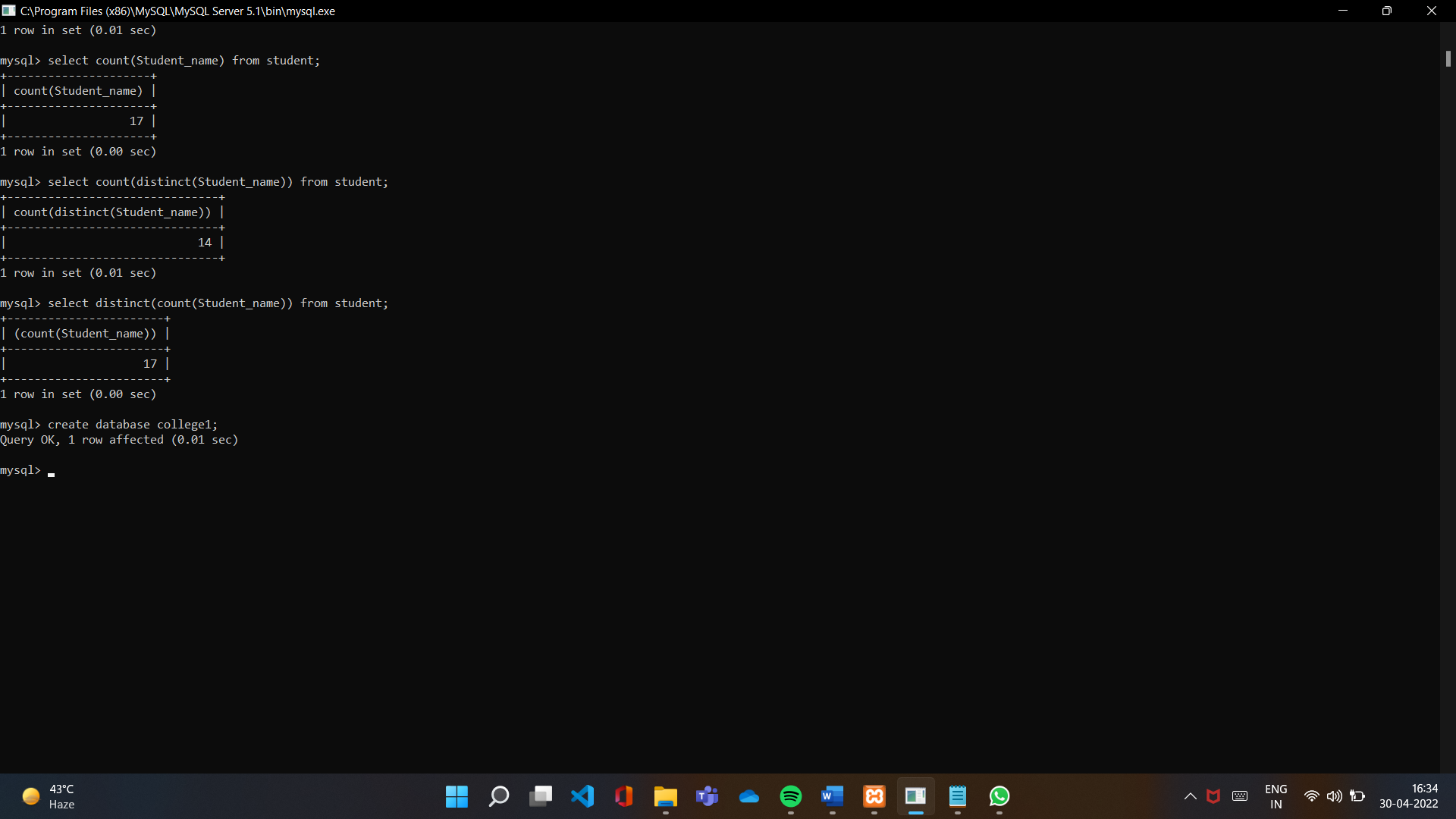


**Q..Create a Database and name as College1.**

**Query:--**

**Create database <databasename>;**

Create database college;



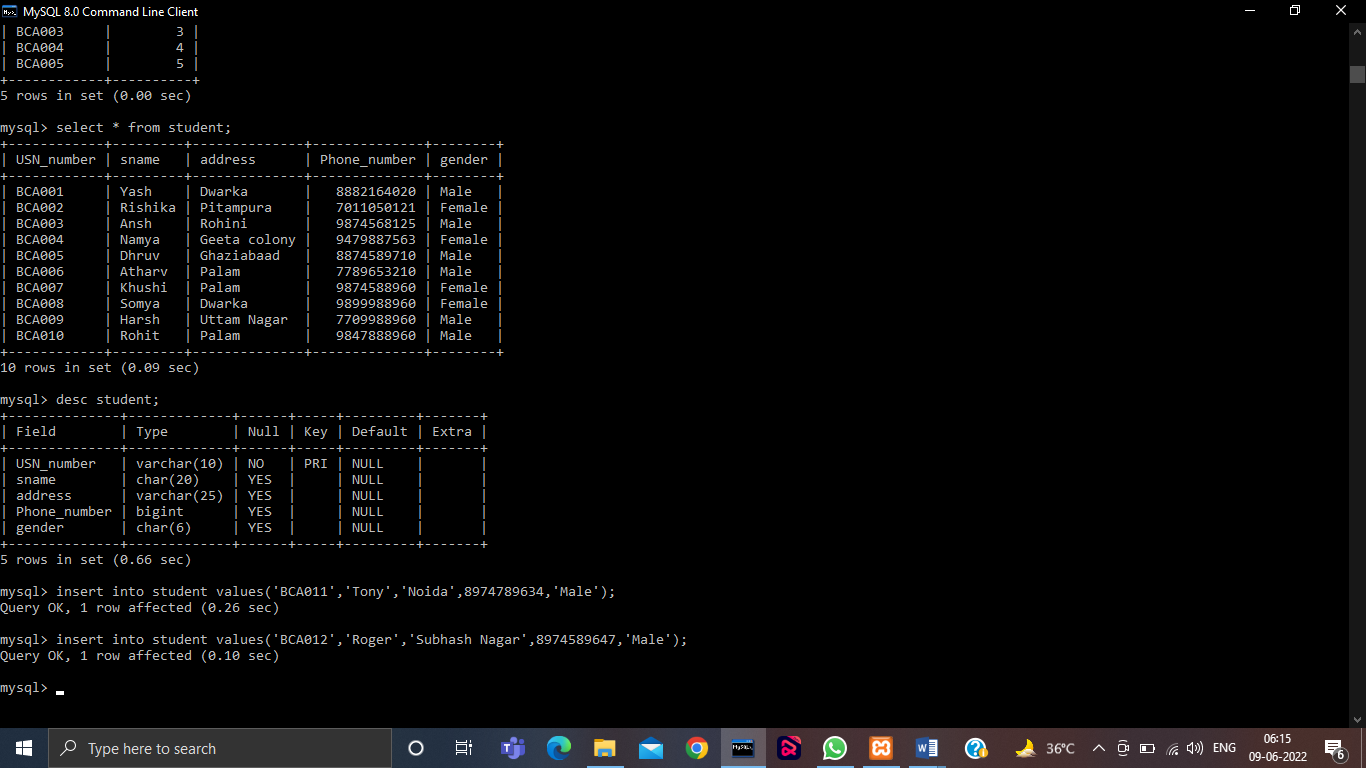
**Q.. Write a query to create a table named as student.**

**Create table <tablename> (<attribute> type,<attribute> type….);**

**Q..Write a query to insert the values in a table.**

**Insert into <tablename> values();**

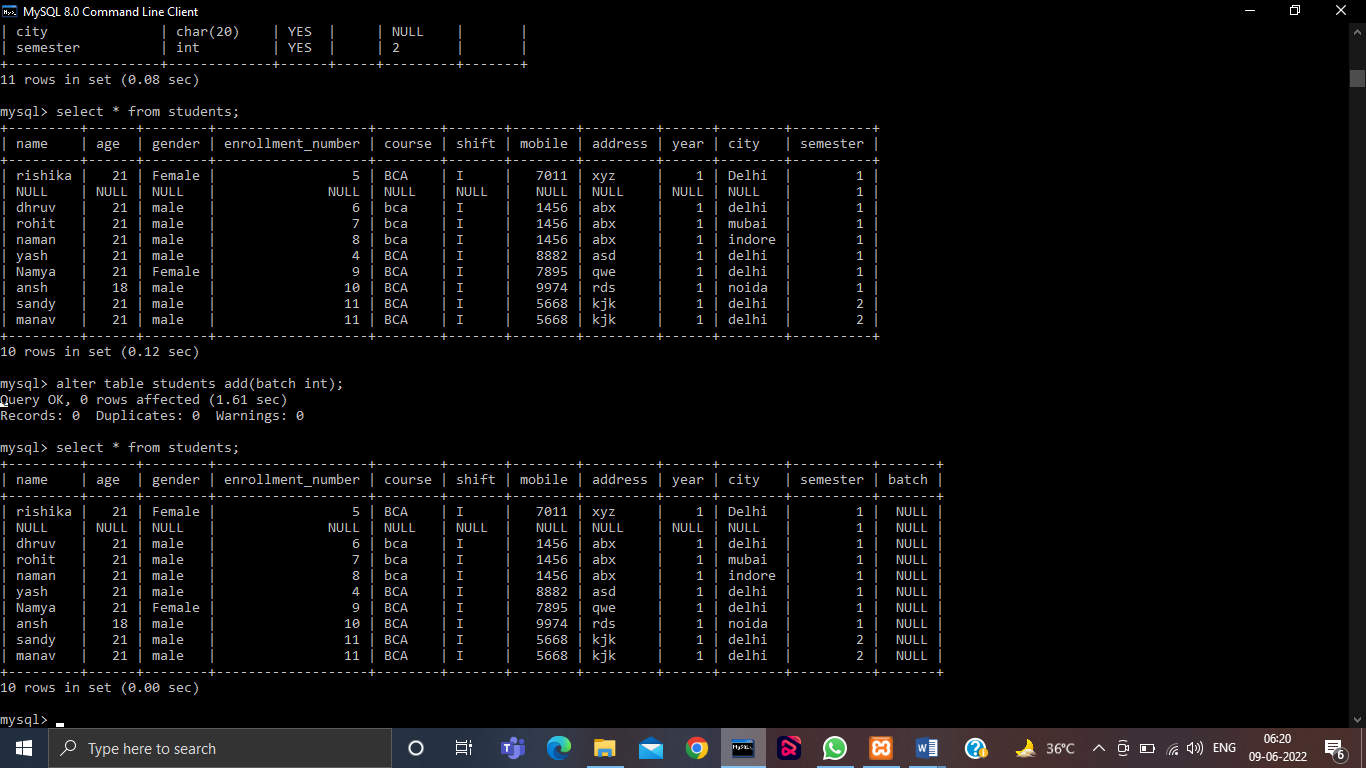
insert into student values('BCA011','Tony','Noida',897478634,'Male');



**Q..Write a query to add a new column in a existing table.**

**Alter table <tablename> add(<columnname> <type>);**

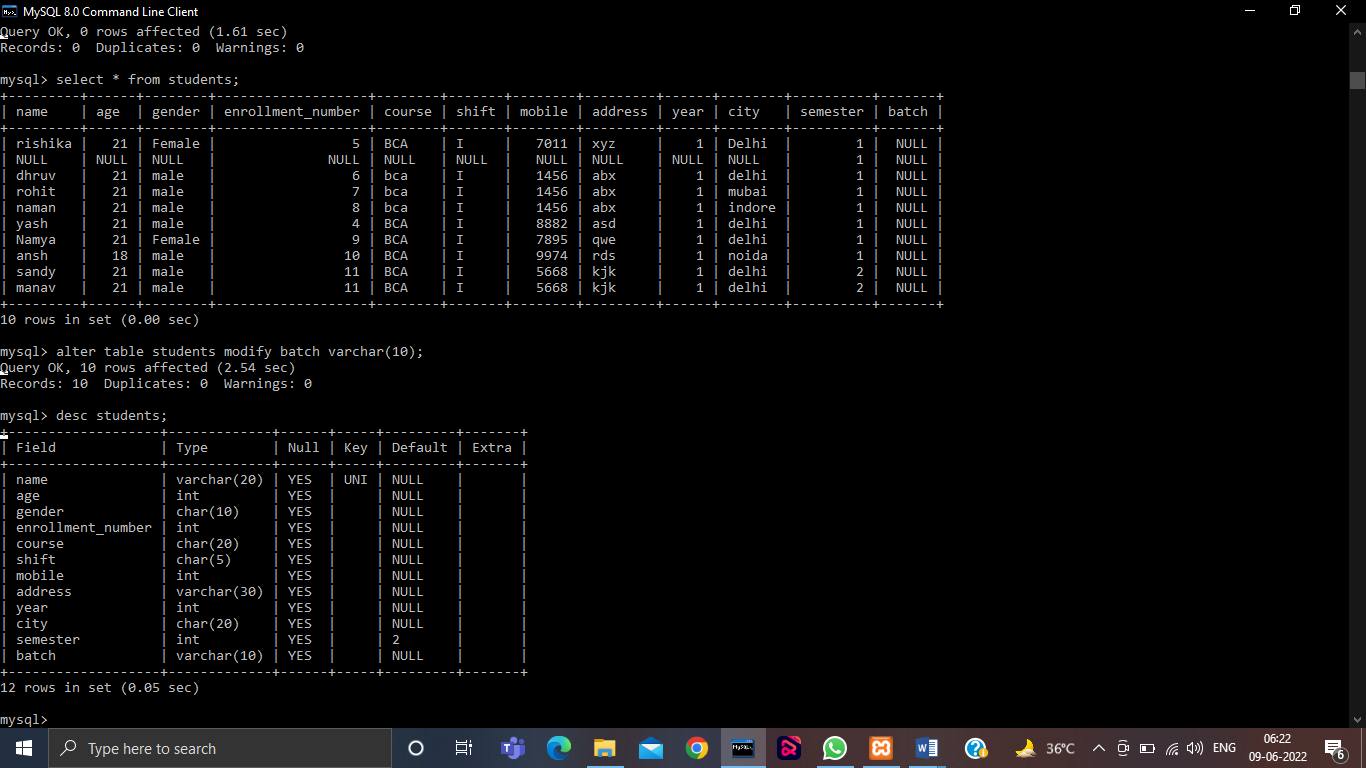
alter table studenst add(batch int);



**Q..Write a query to change the type of column in pre-existing table.**

**alter table <tablename> modify<columnname> <newtype>;**

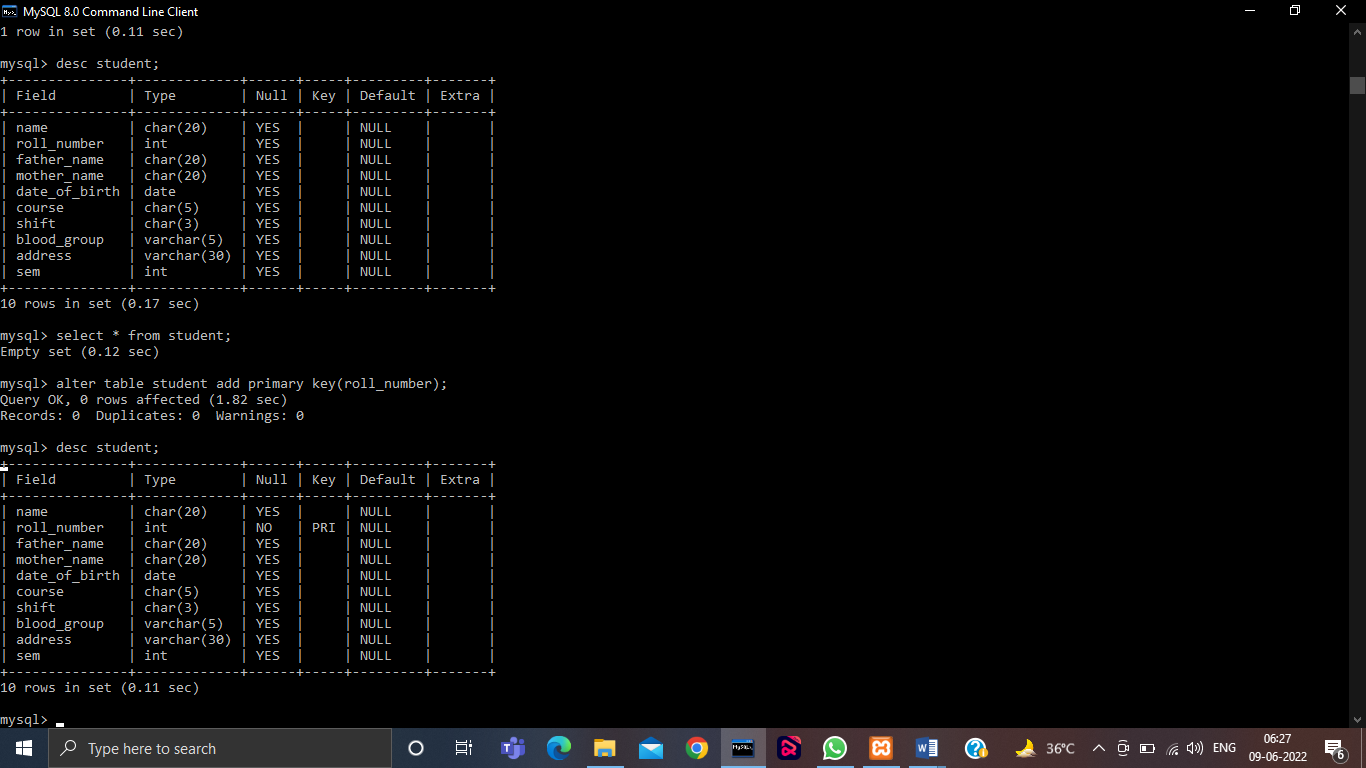
alter table student modify batch varchar(10);



**Q..Write a query to add a primary key in a column.**

**Alter table <tablename> add primary key(<column\_name>);**

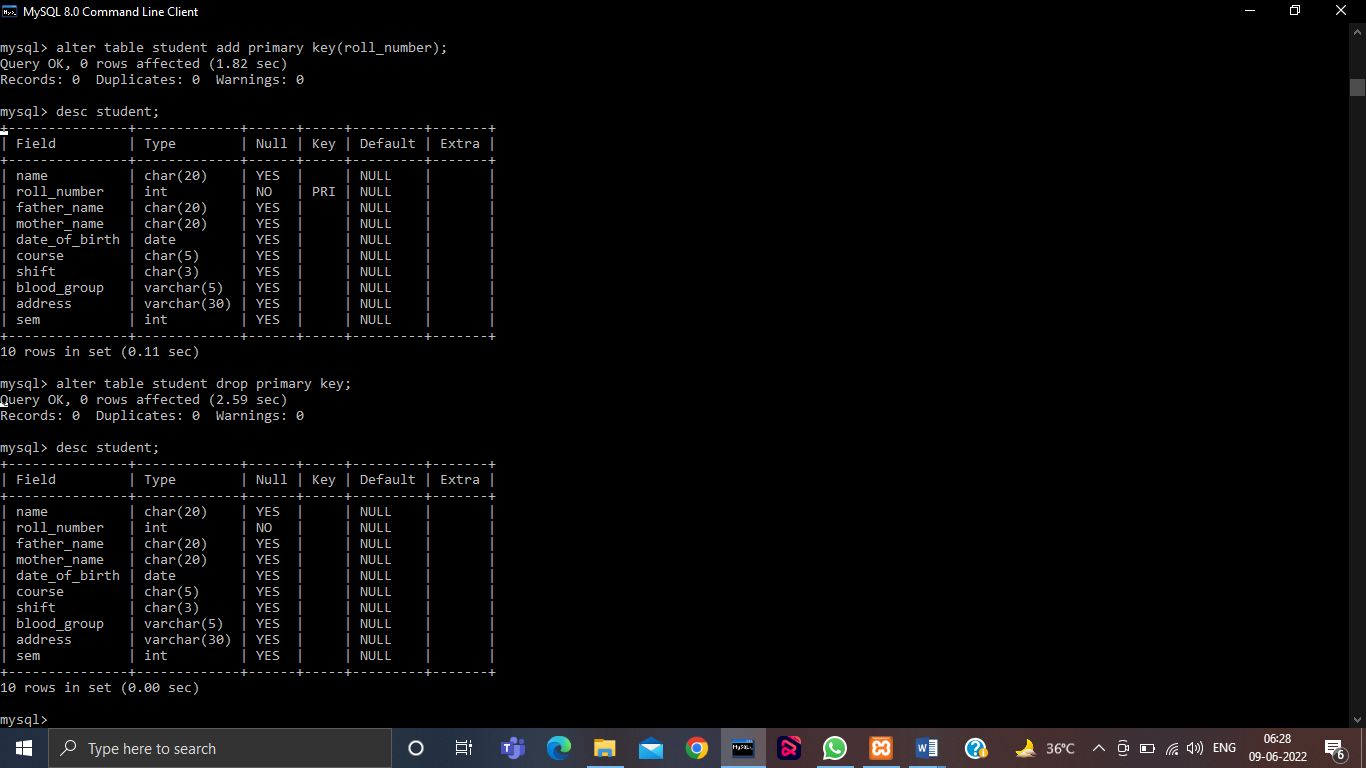
alter table student add primary key(roll\_number);



**Q..Write a query to drop a primary key of a column.**

**Alter table <tablename> drop primary key;**

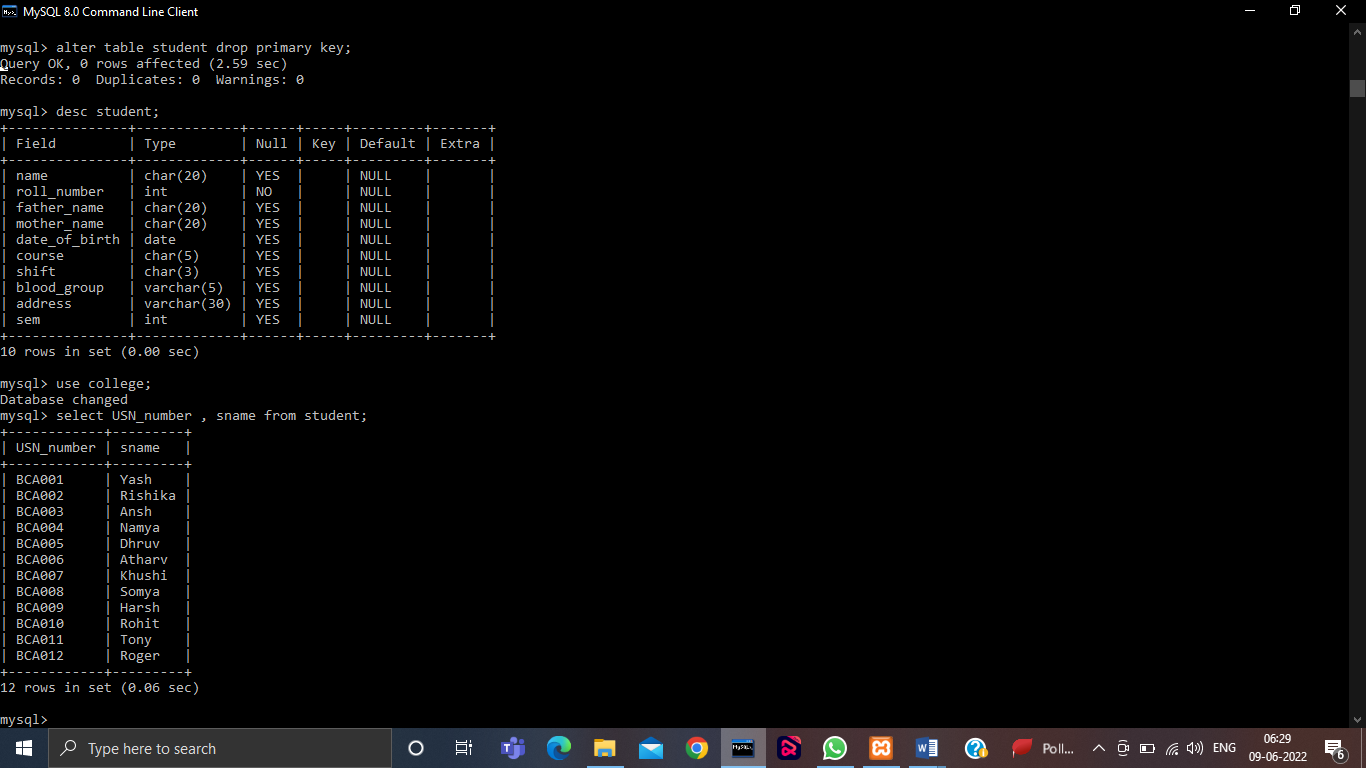
Alter table student drop primary key;



**Q..Write a Query to display the particular columns of the table.**

**Select <columnname> from <tablename>;**

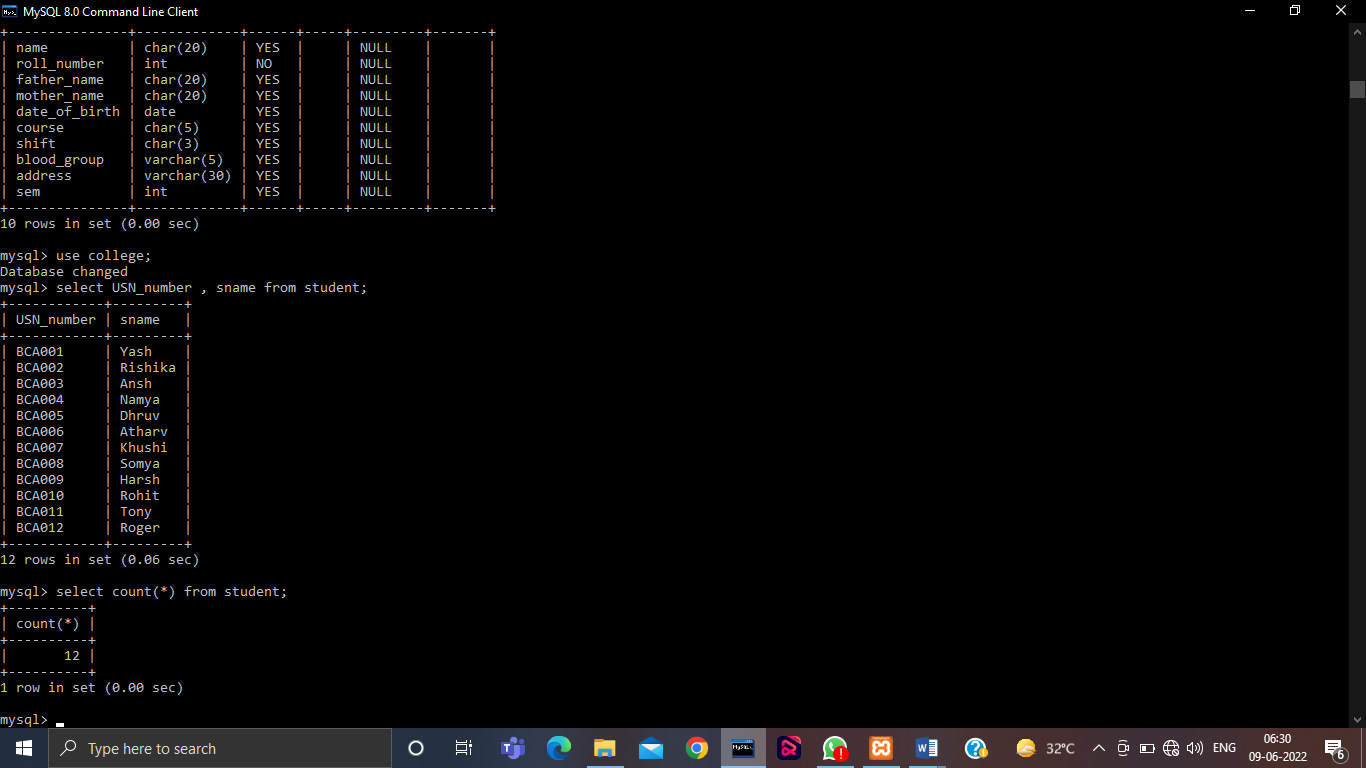
Select USN\_number,sname from student;



**Q.. Write a query to count the number of records in a table.**

**Select count(\*) from <tablename>;**

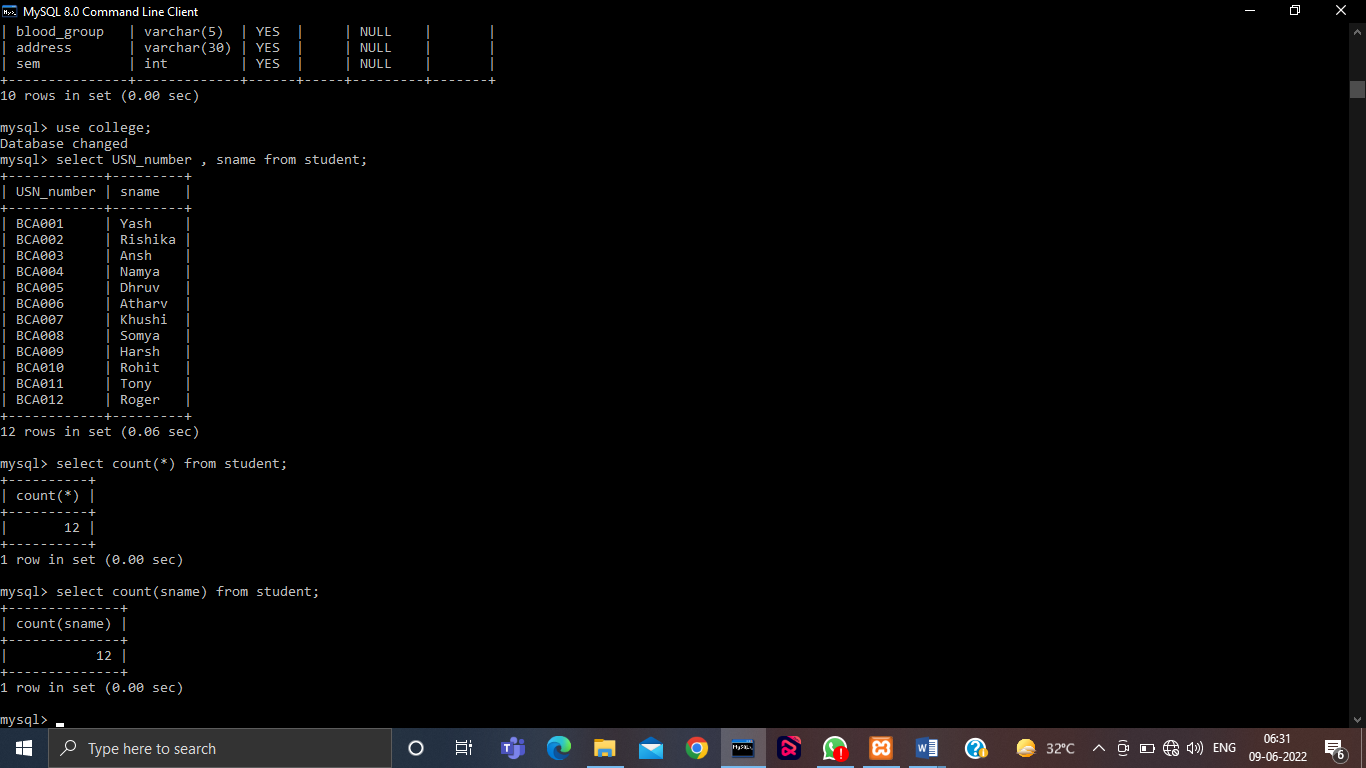
Select count(\*) from student;



**Q.. Write a query to count the records of a particular column of a table.**

**Select count(<column\_name>) from <tablename>;**

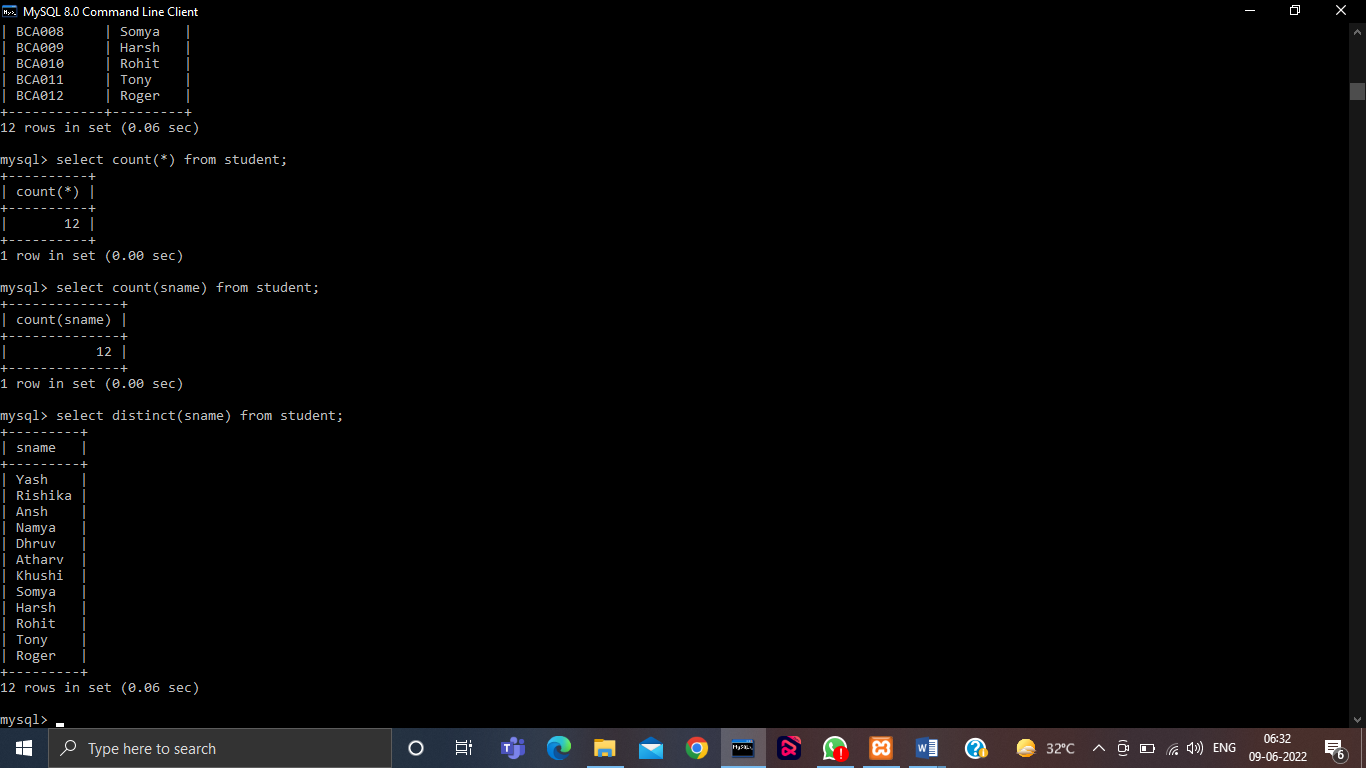
Select count(sname) from student;



**Q.. Write a query to display the distinct records of a particular column of a table.**

Select distinct(<column\_name>) from <table\_name>;

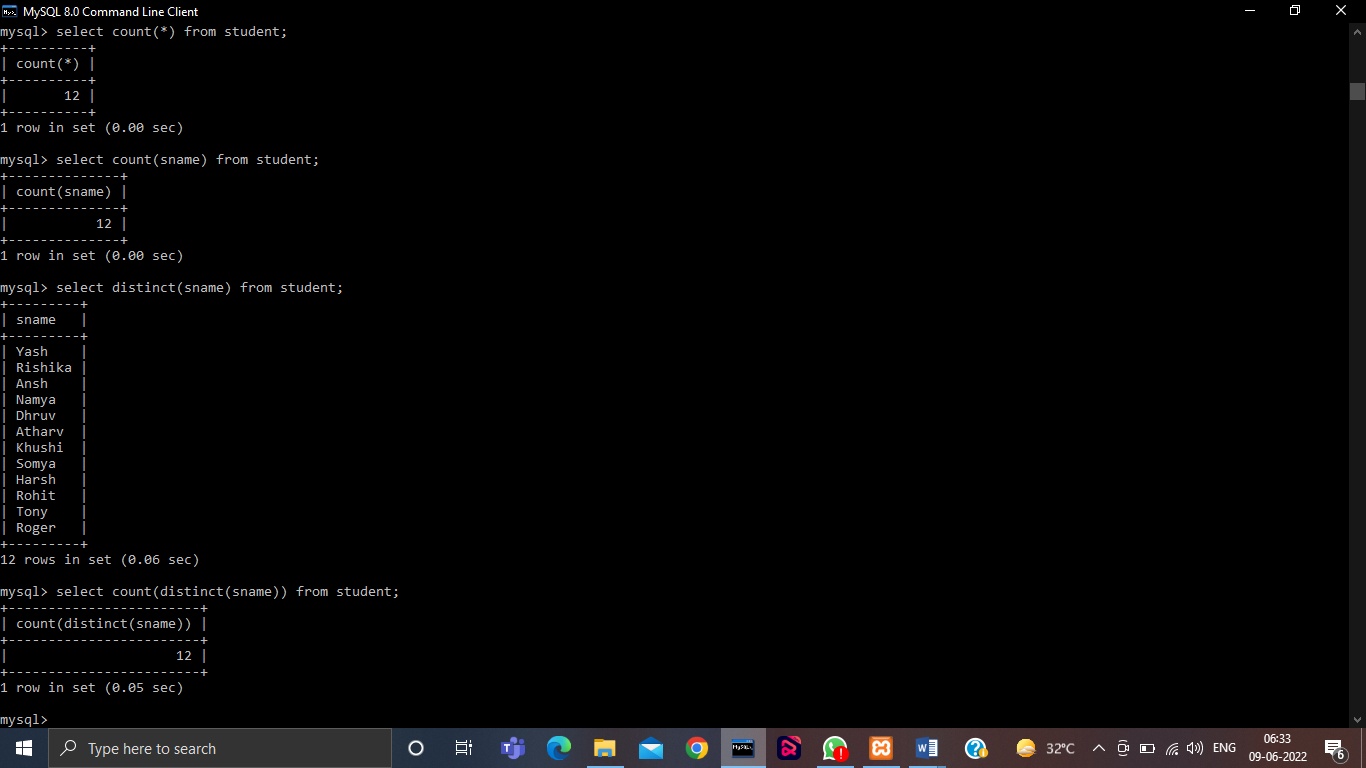
Select distinct(sname) from student;



**Q.. Write a query to count the number of distinct records from a table.**

Select count(distinct(<column\_name>)) from <table\_name>;

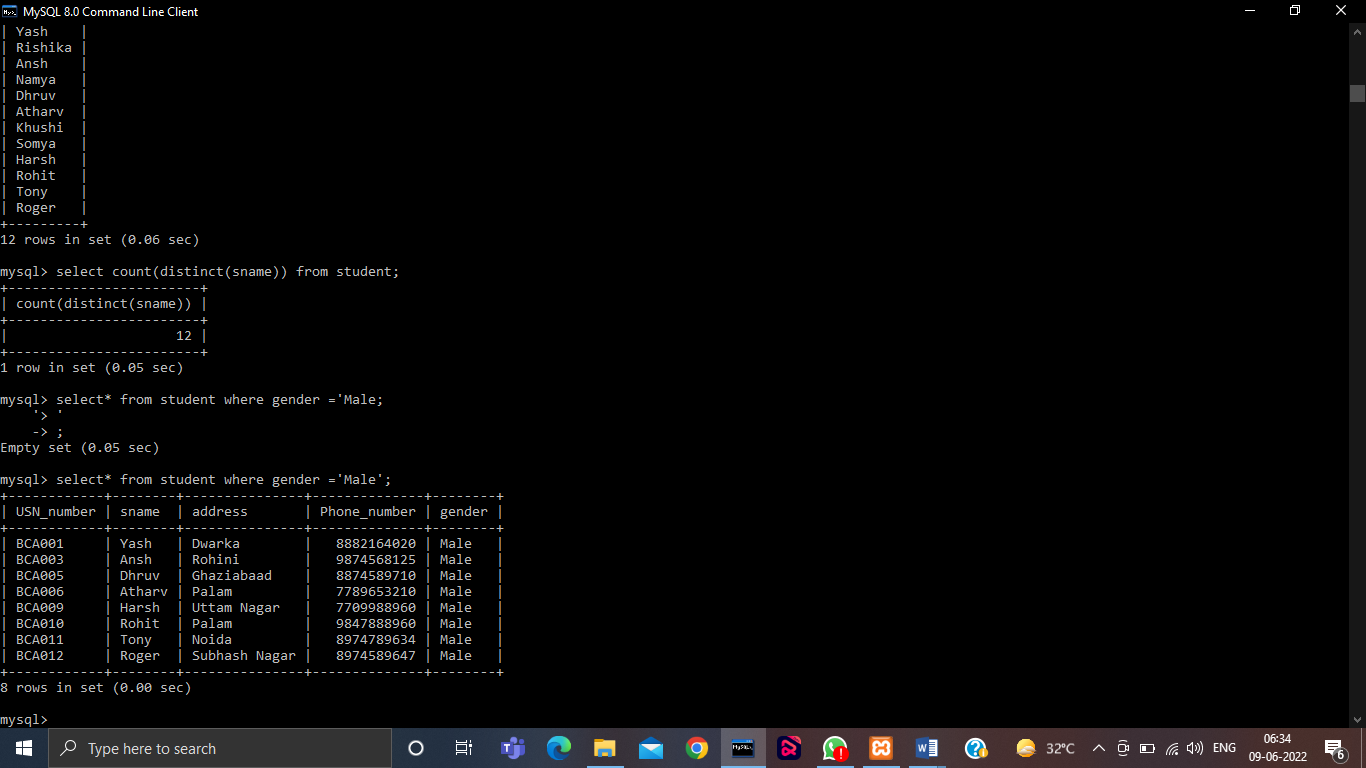
Select count(distinct(sname)) from student;



**WHERE Clause**

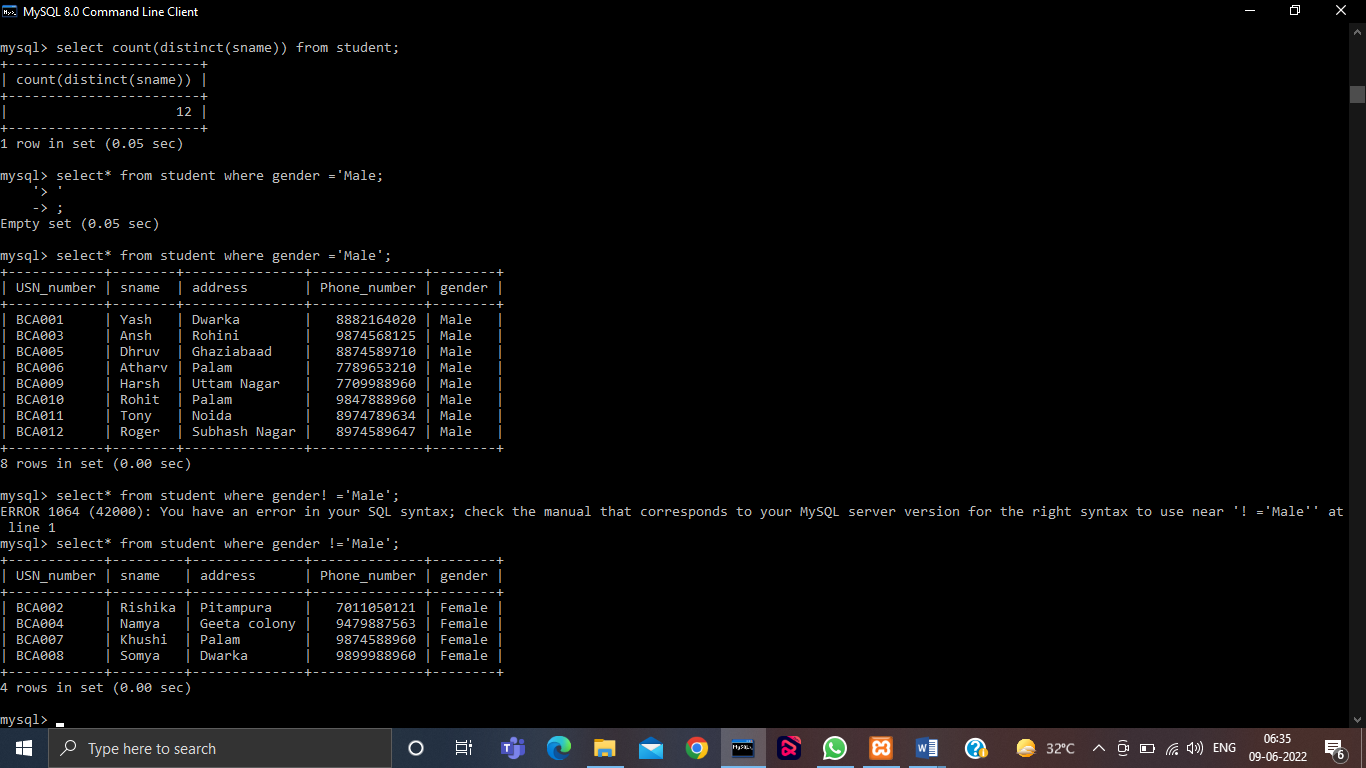
**Q.. Write a query to print all the information of gender male.**

**Select \* from student where gender=’Male’;**



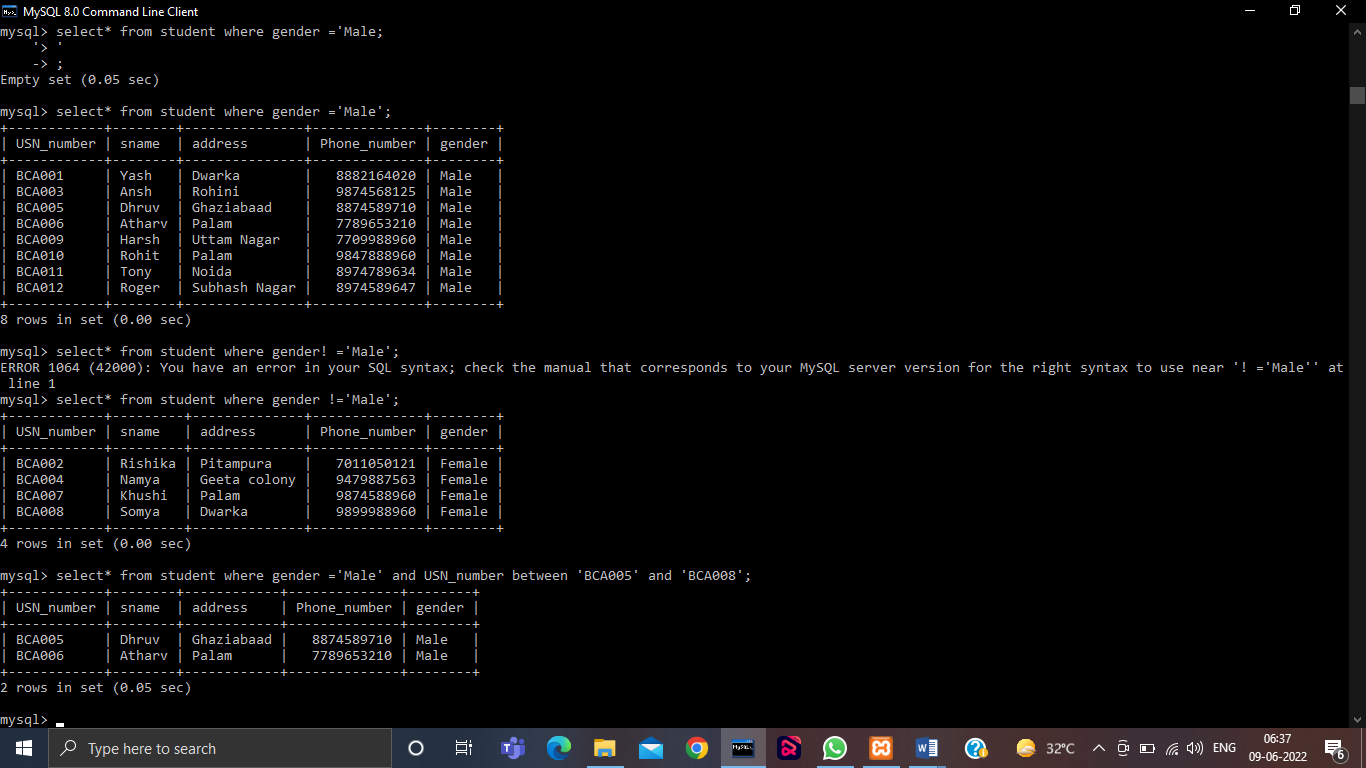
**Q.. Write a query to print all the information of female students using not operator.**

**Select \* from student where gender !=’Male’;**



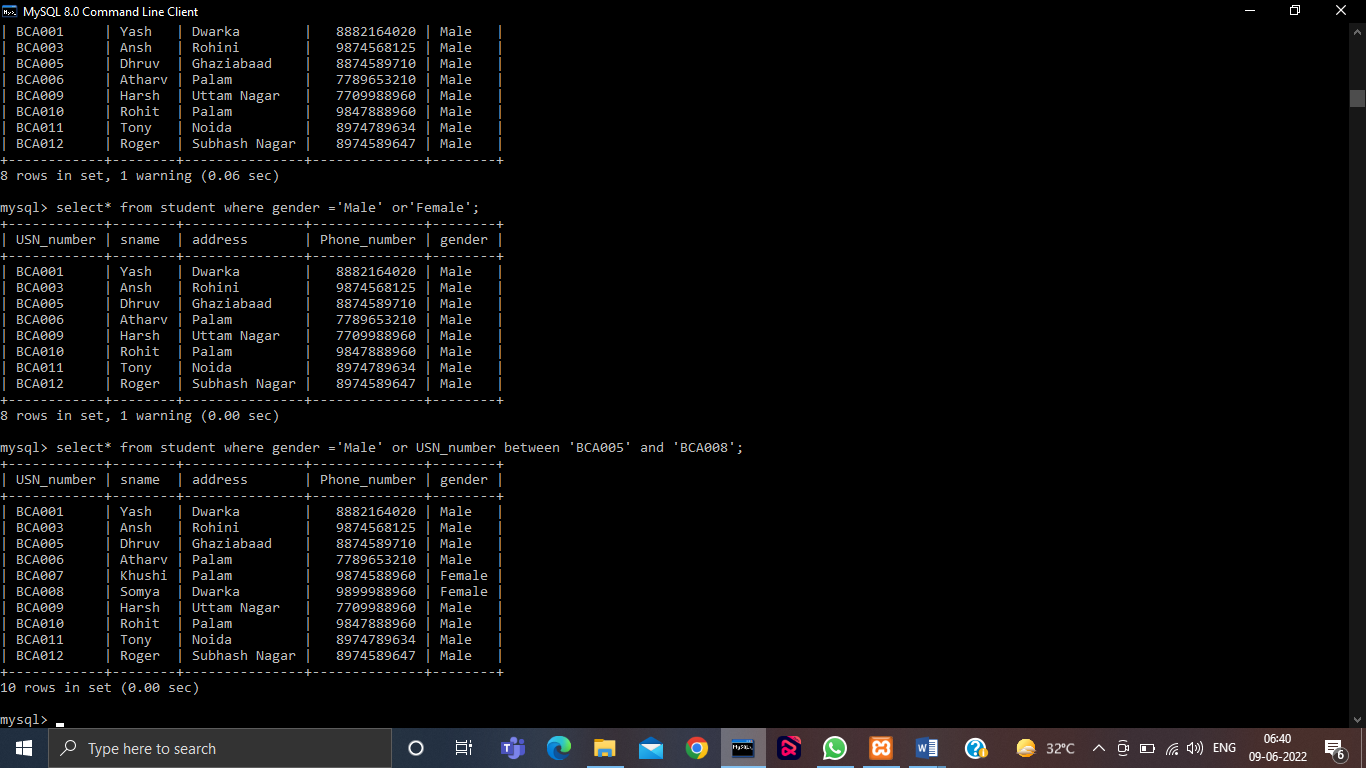
**Q.. Write a query to show the use of AND operator.**

**Select \* from student where gender=’Male’ and USN\_Number between 'BCA005' and 'BCA008';**



**Q..Write a query to show the use of OR operator.**

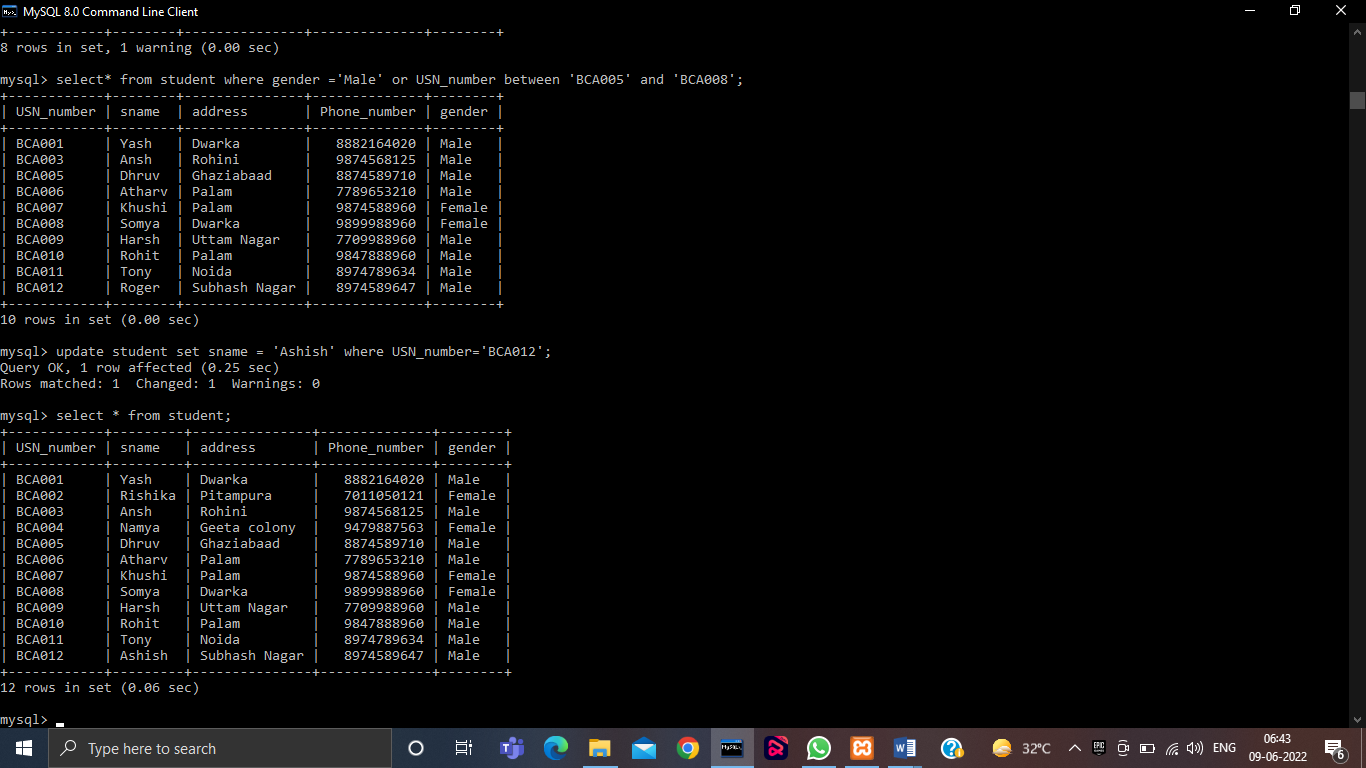
**Select \* from student where gender=’Male’ or USN\_Number between 'BCA005' and 'BCA008';**



**Update Query:**

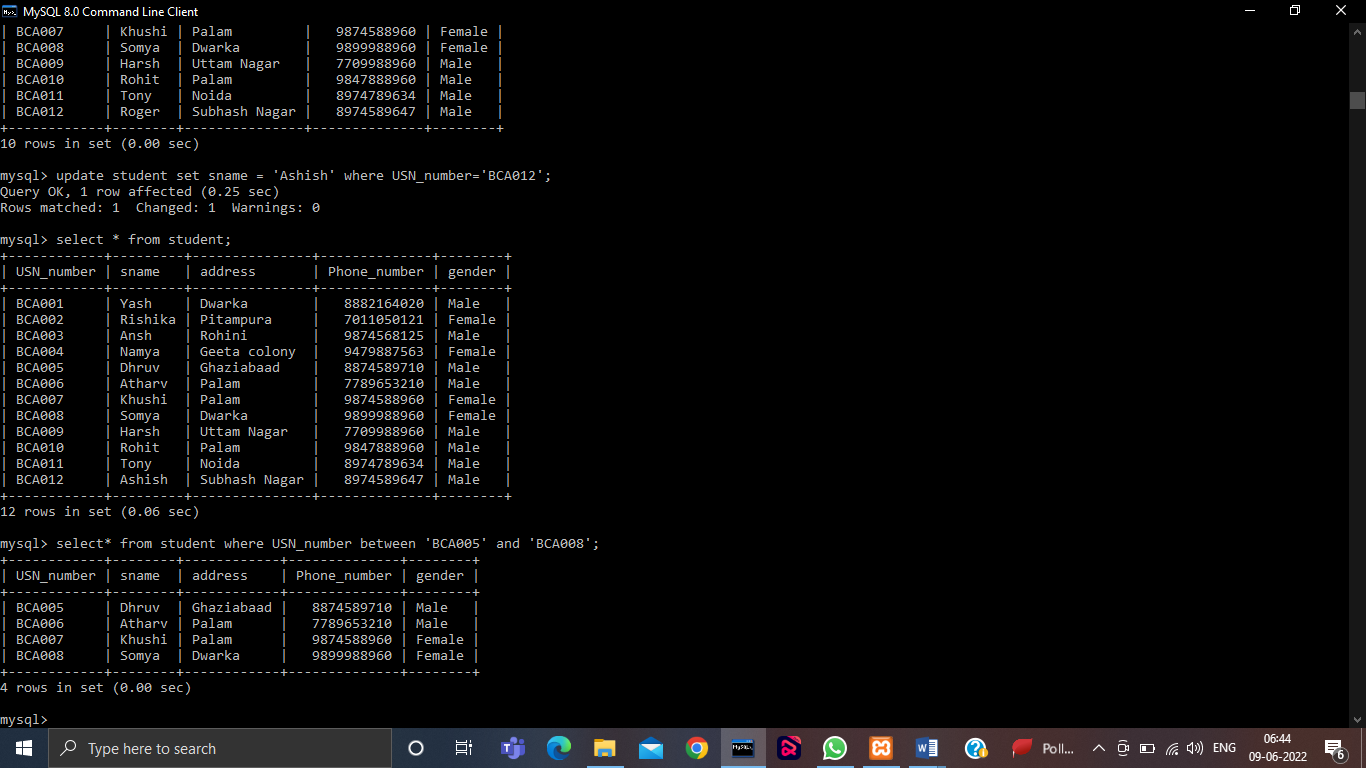
**Update <tablename> set <column\_name> = <value> where condition;**

**update student set sname=’Ashish’ where USN\_number between =’BCA012’;**



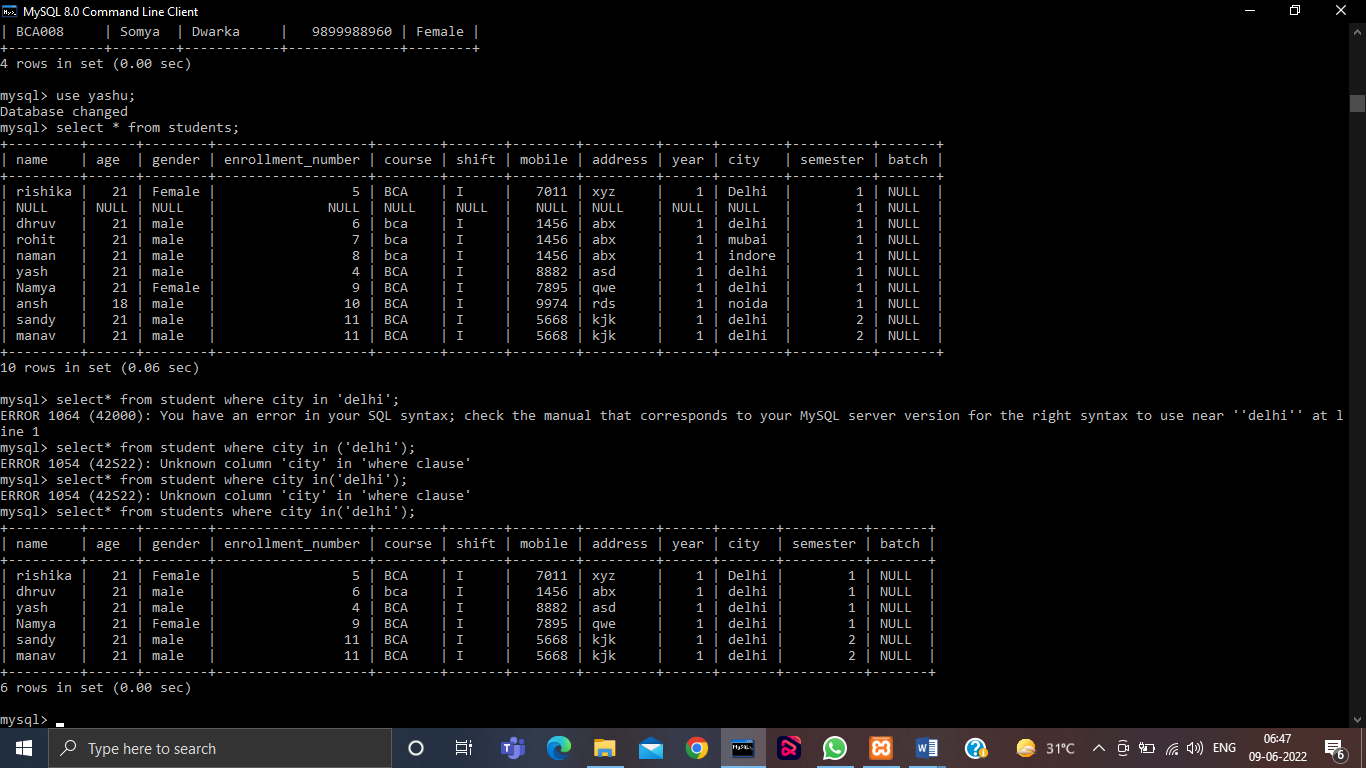
**Between Operator:**

**select \* from student where USN\_number between 'BCA005' and 'BCA008';**



**In Operator:**

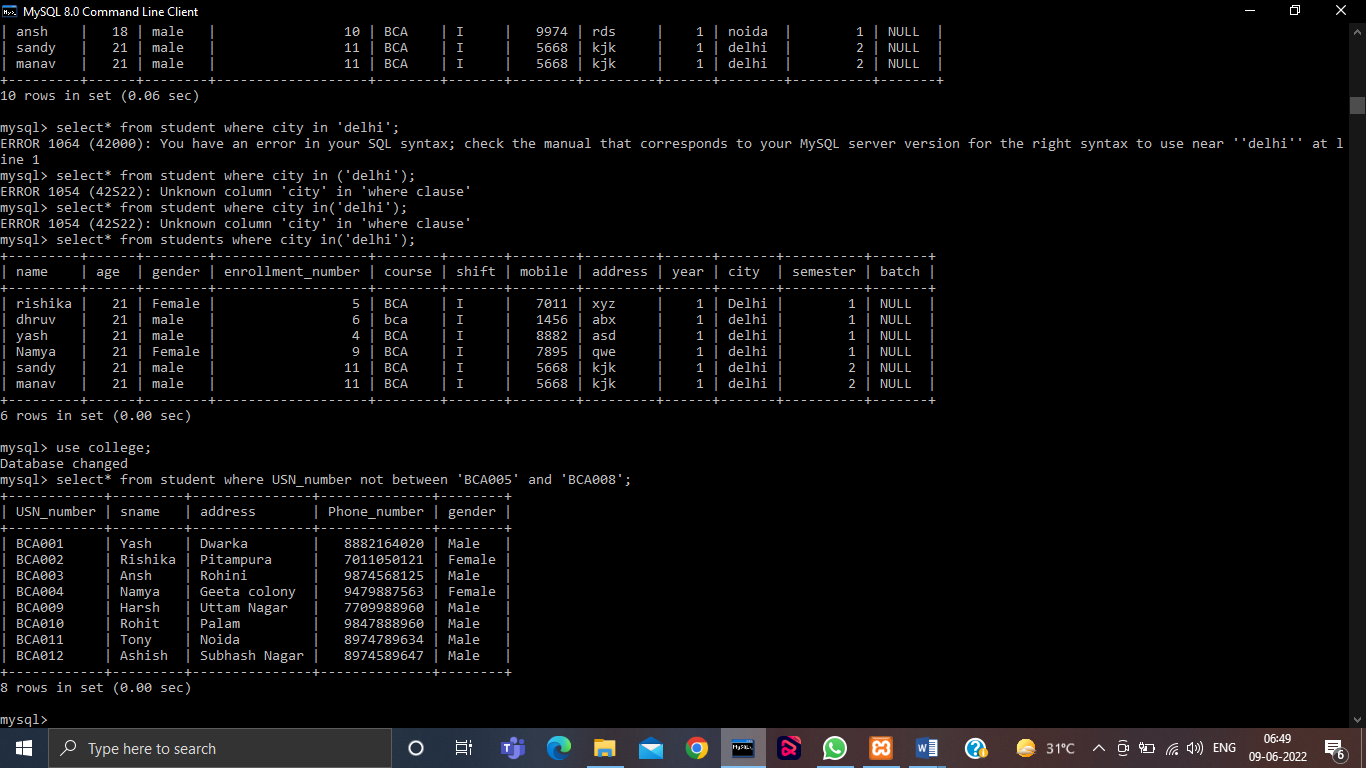
**select \* from students where cityin(‘delhi’);**



**Not Operator:**

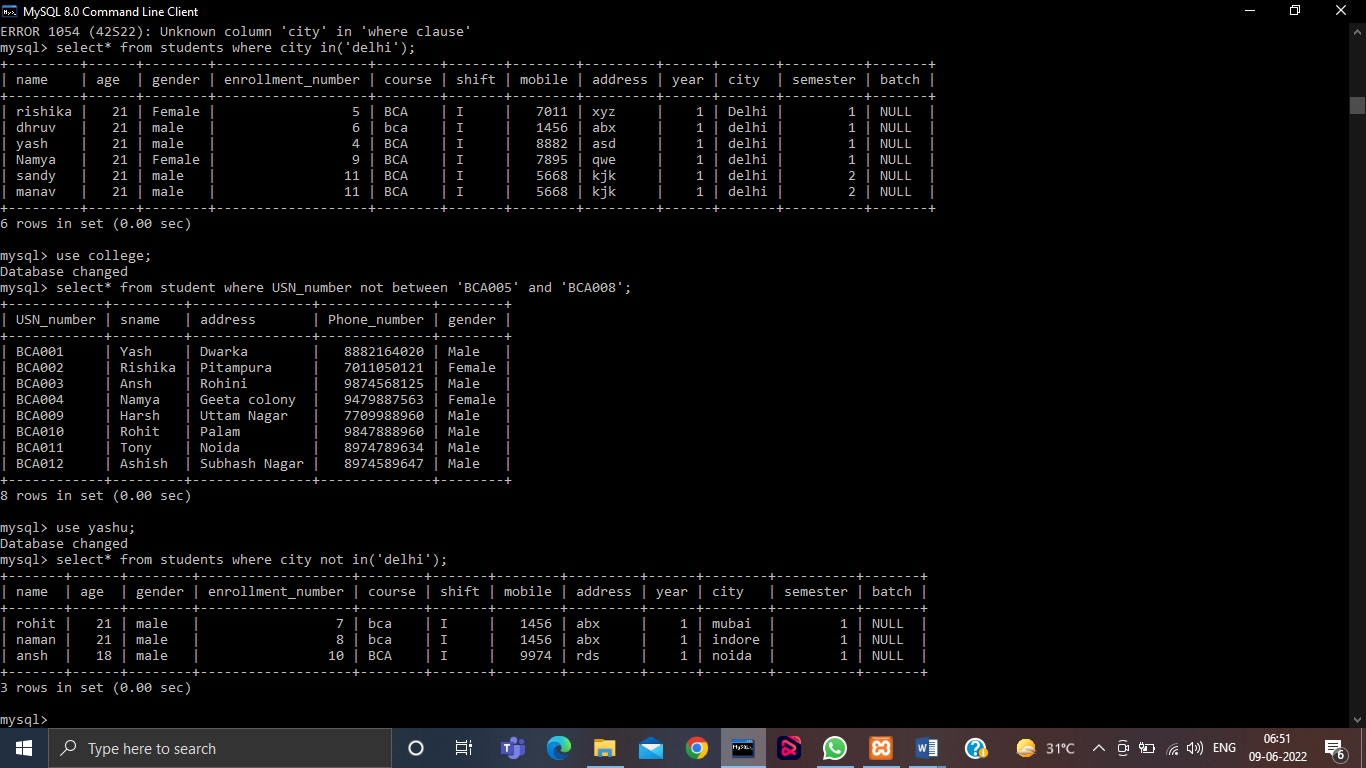
**In Between---**

**select \* from student where USN\_number not between 'BCA005' and 'BCA008';**



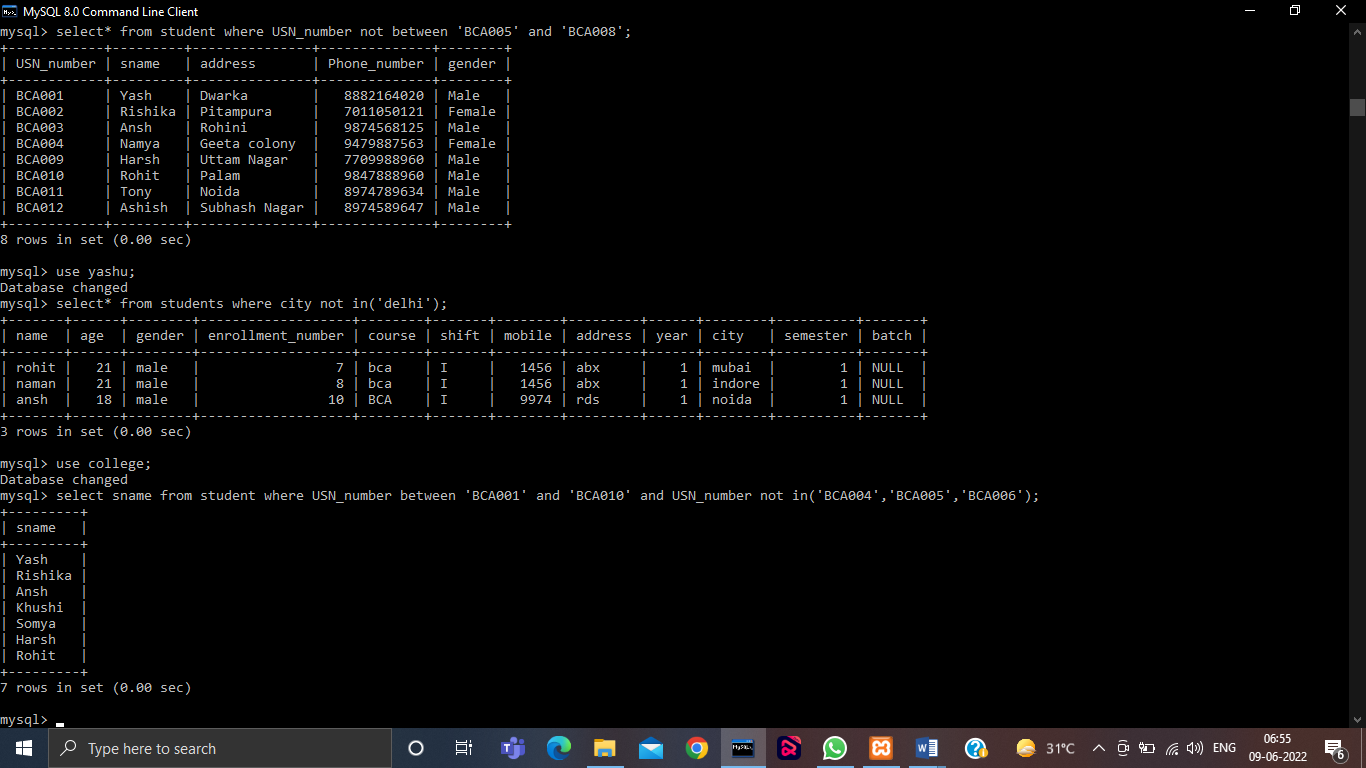
**In----**

**select \* from students where city not in(‘delhi’);**



**Q.Write a query to select all the name with an USN\_Number between 1 and 10 in addition do not show Name with a USN\_number of 4, 5 and 6.**

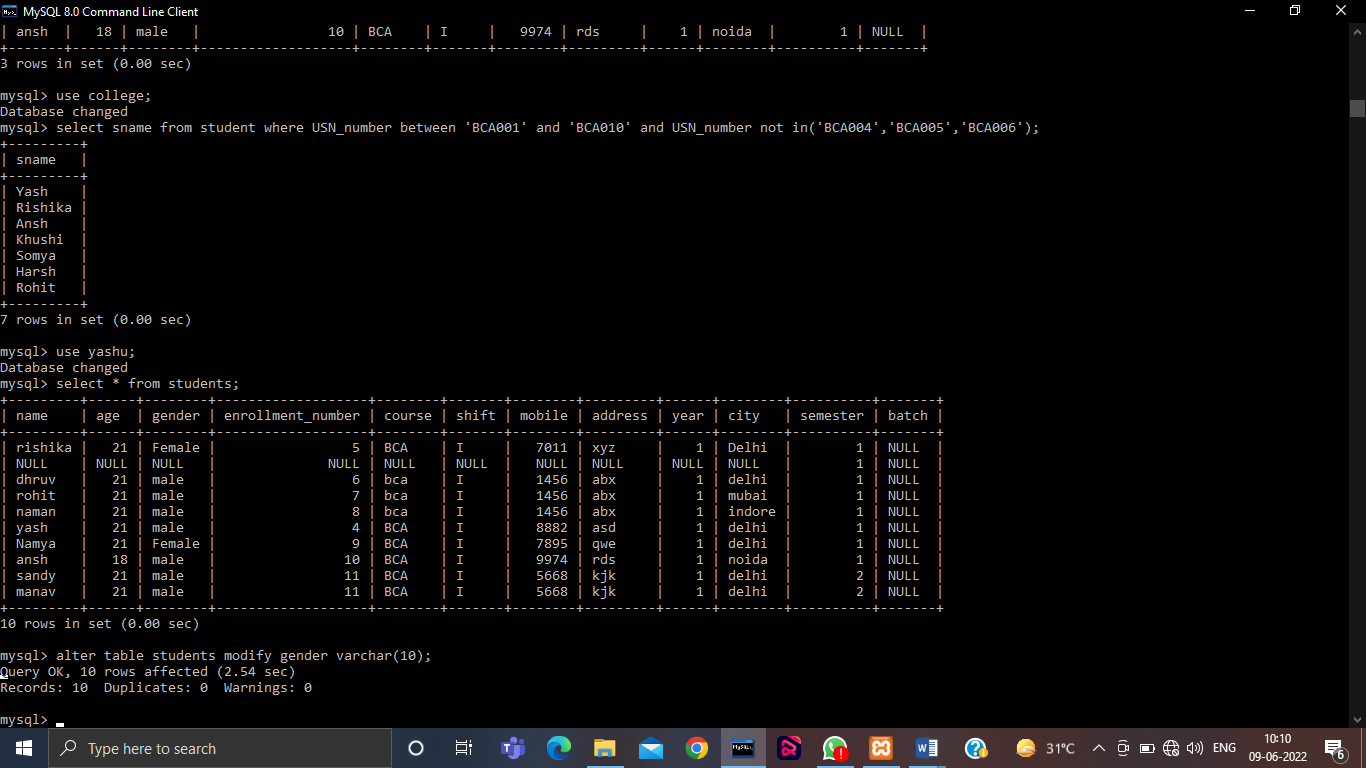
**select sname from student where USN\_Number between 'BCA001' and 'BCA010' and USN\_number not in('BCA004','BCA005','BCA006');**



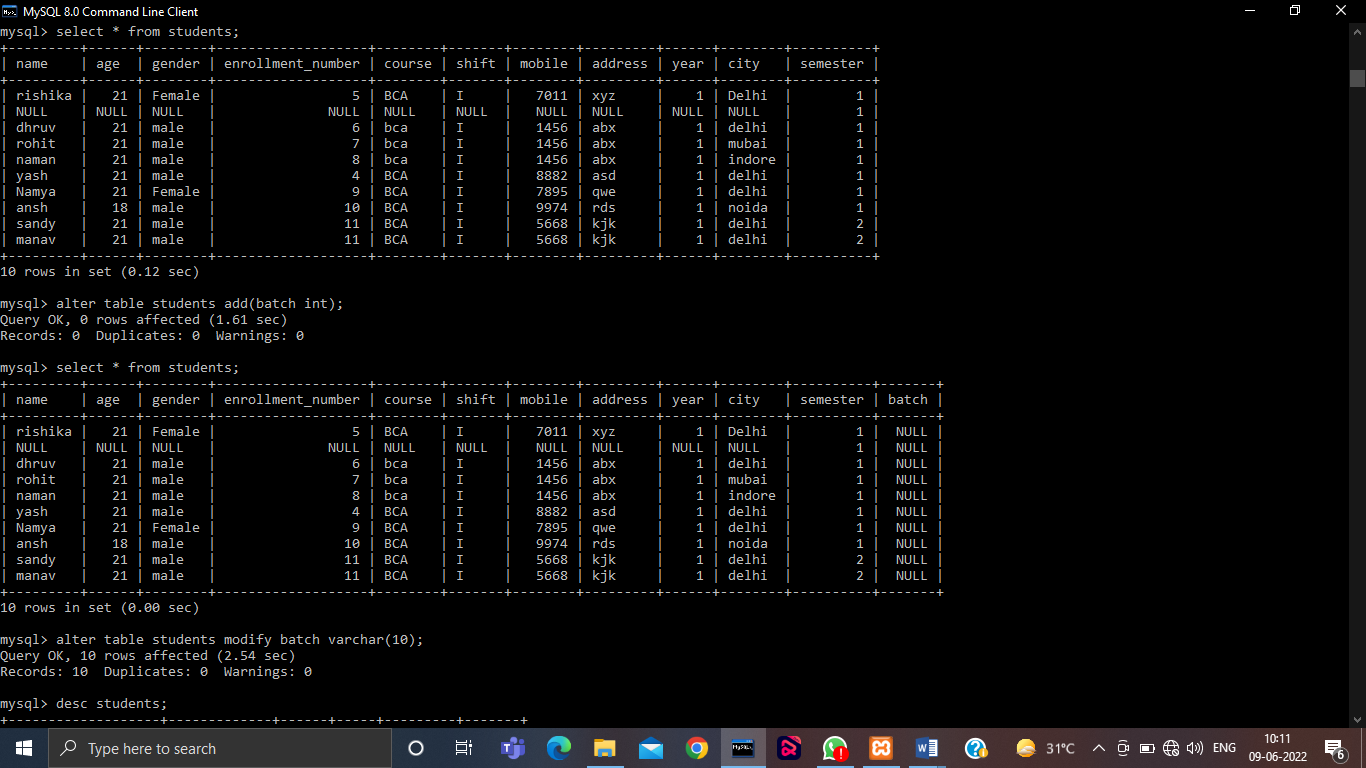
**Q3 Write queries to execute following DDL commands :**

* **CREATE : Create the structure of a table with at least five columns**
* **ALTER: Change the size of a particular column.**
* **Add a new column to the existing table.**
* **Remove a column from the table.**
* **DROP: Destroy the table along with its data.**

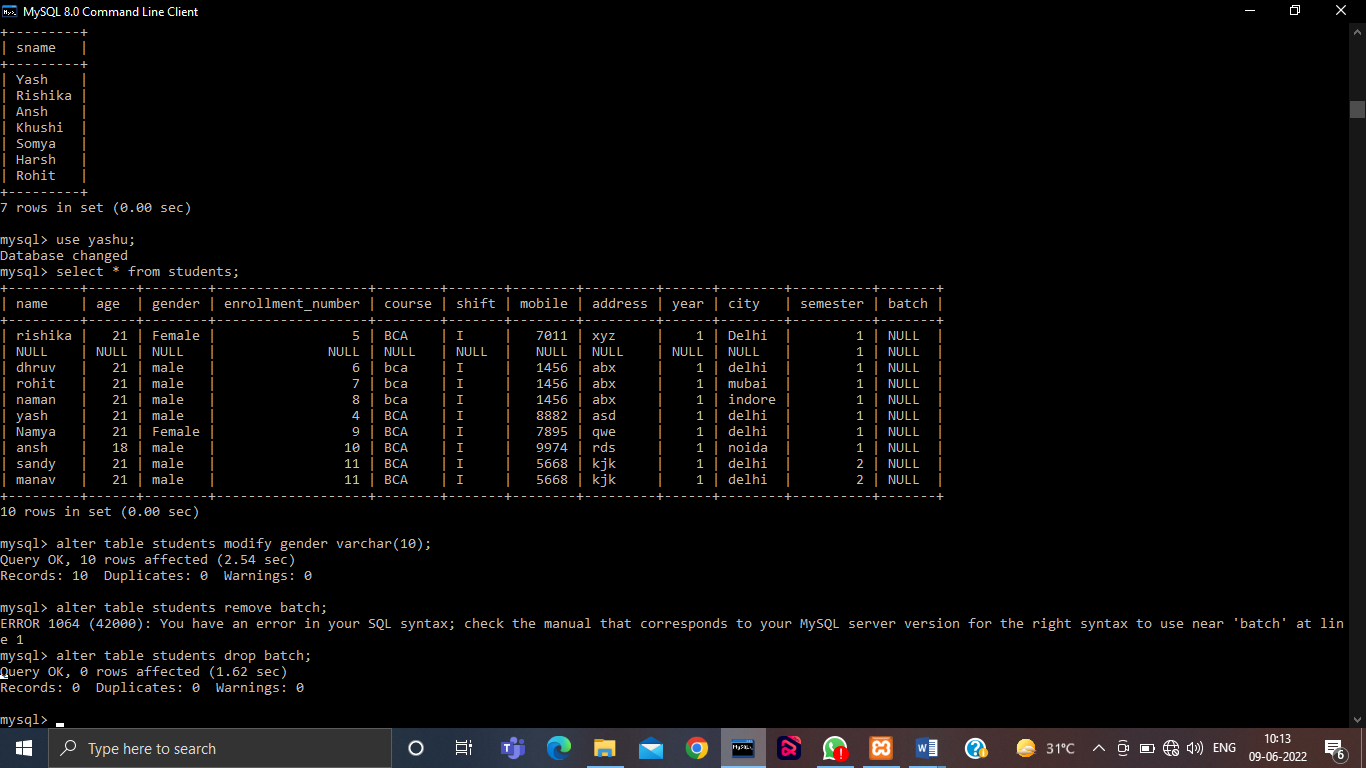
**Change the size of a column**



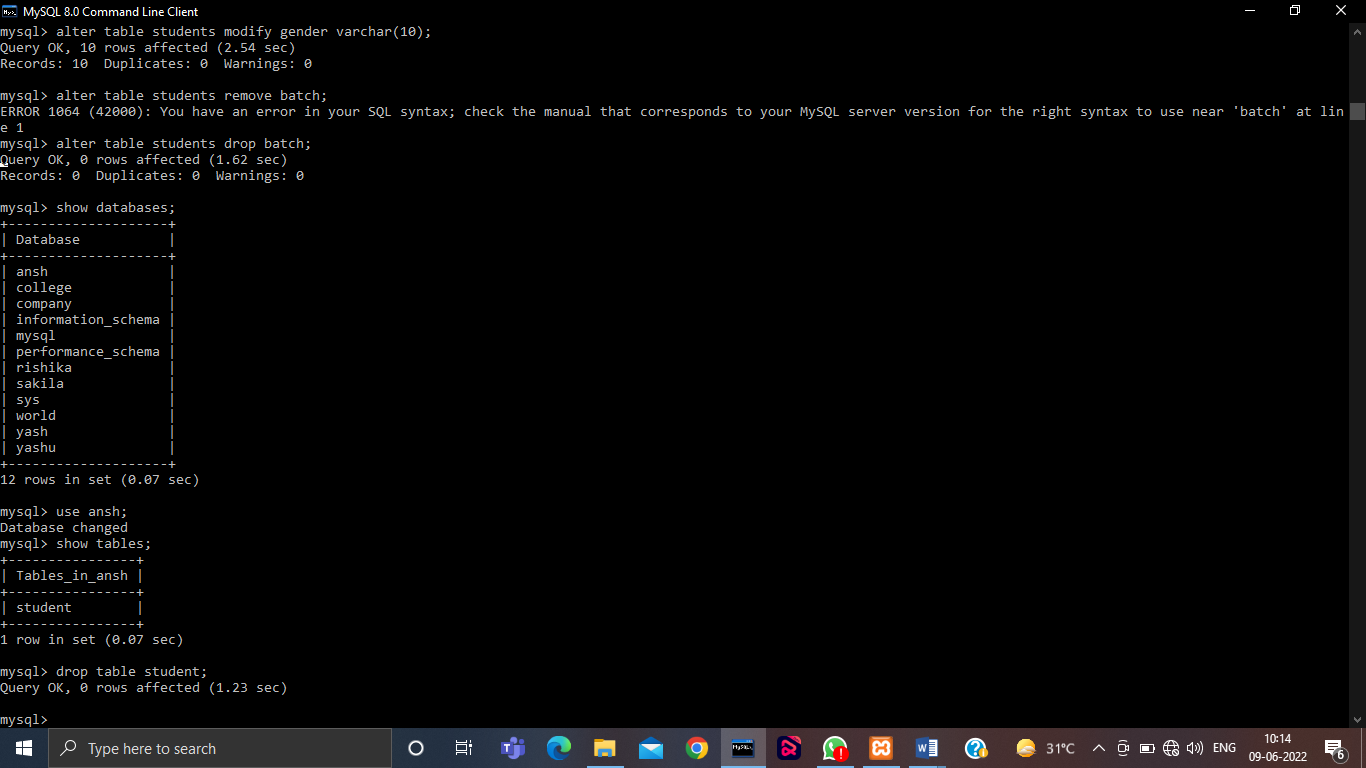
**Add a New Column**

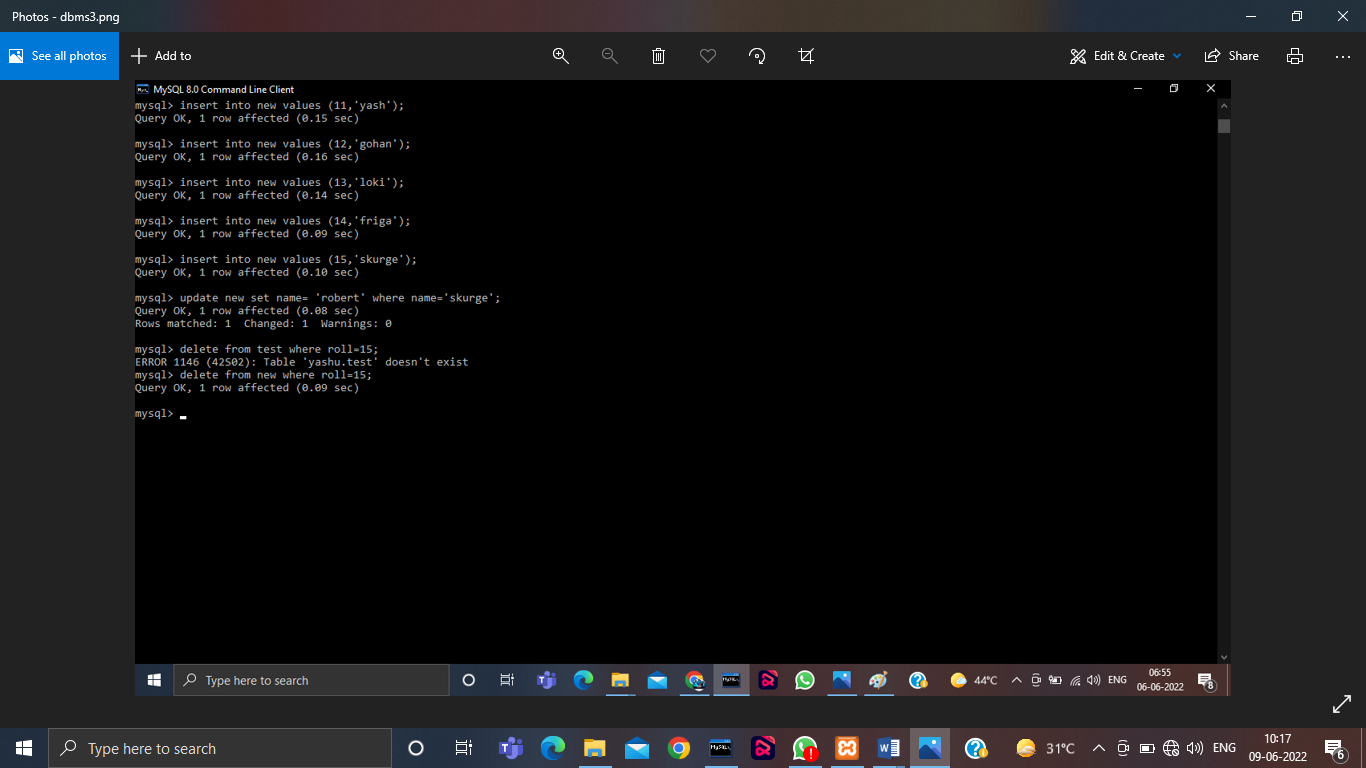


**Remove a Column**



**Drop a Table**



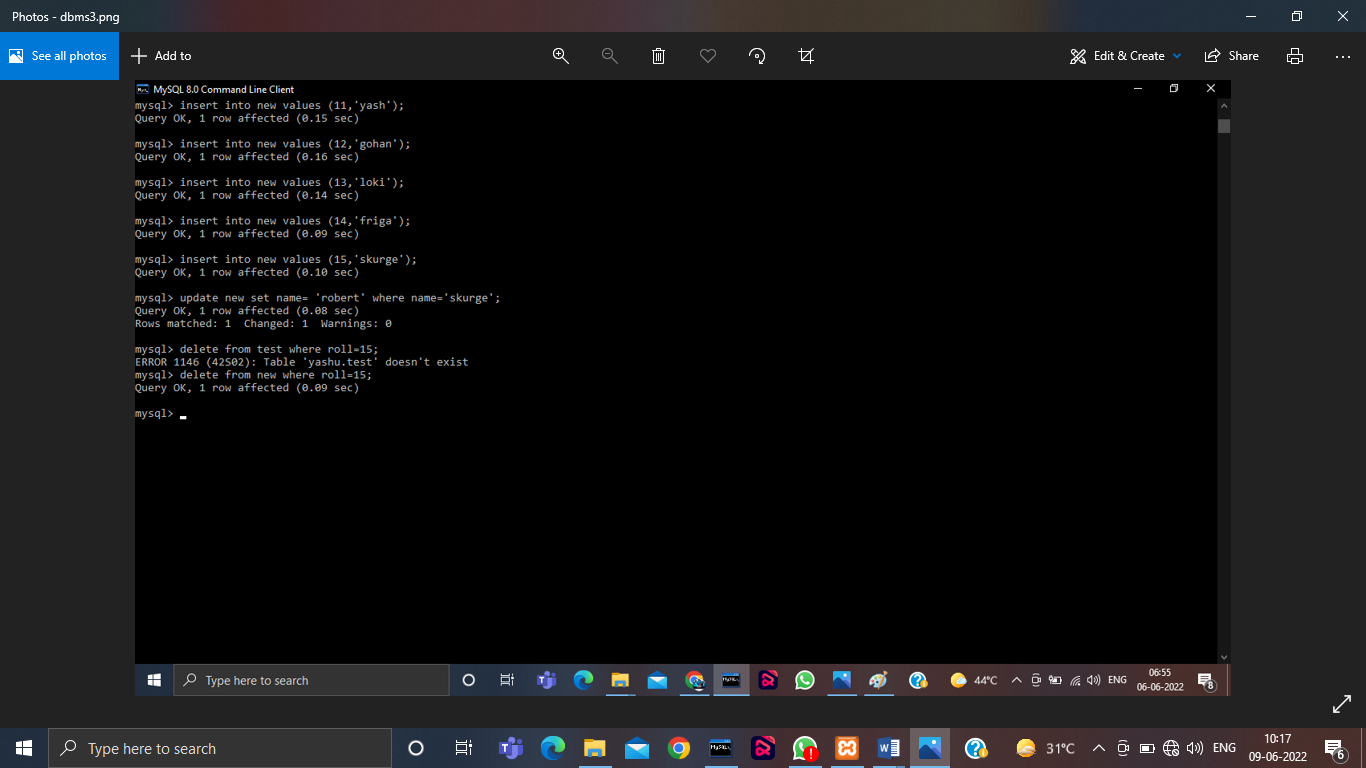


**Q4 Write queries to execute following DML commands :**

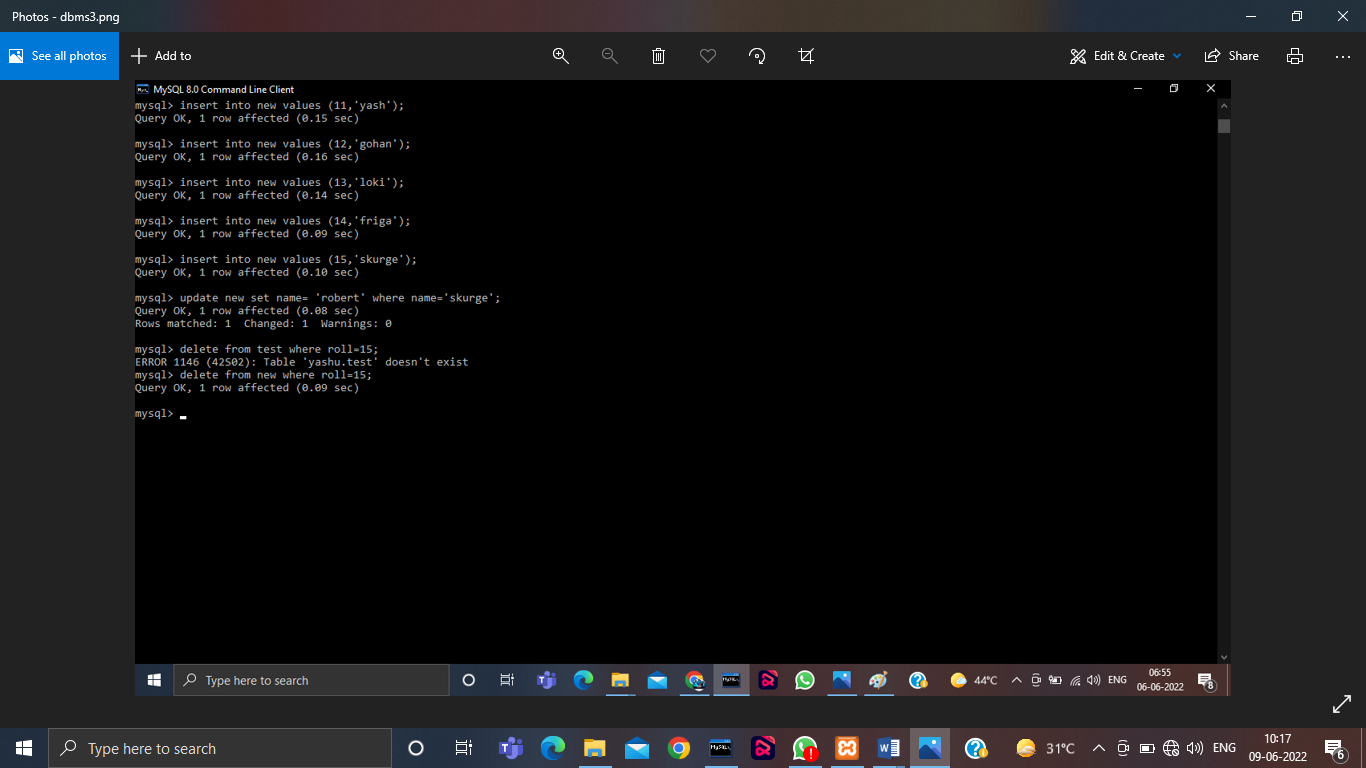
* **INSERT: Insert five records in each table.**
* **UPDATE: Modify data in single and multiple columns in a table**
* **DELETE: Delete selective and all records from a table**

**INSERT**

**UPDATE**



**DELETE**

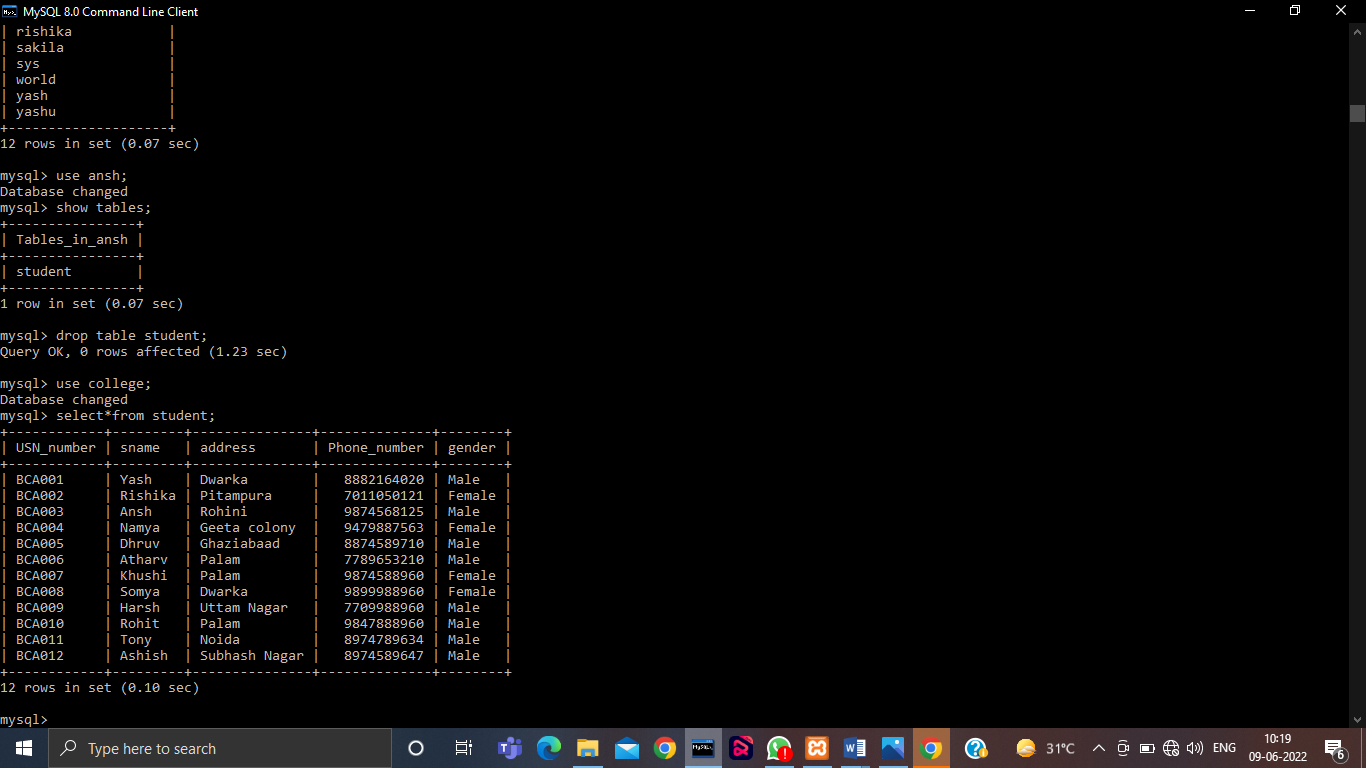


**Q5 Write queries to execute following DML command :**

* **SELECT: Retrieve the entire contents of the table.**
* **Retrieve the selective contents (based on provided conditions) from a table.**
* **Retrieve contents from a table based on various operators i.e. string operators, logical operators and conditional operators,Boolean operators.**
* **Sort the data in ascending and descending order in a table on the basis of one column or more than one column.**
* **To Display the entire contents of the table.**

Query:

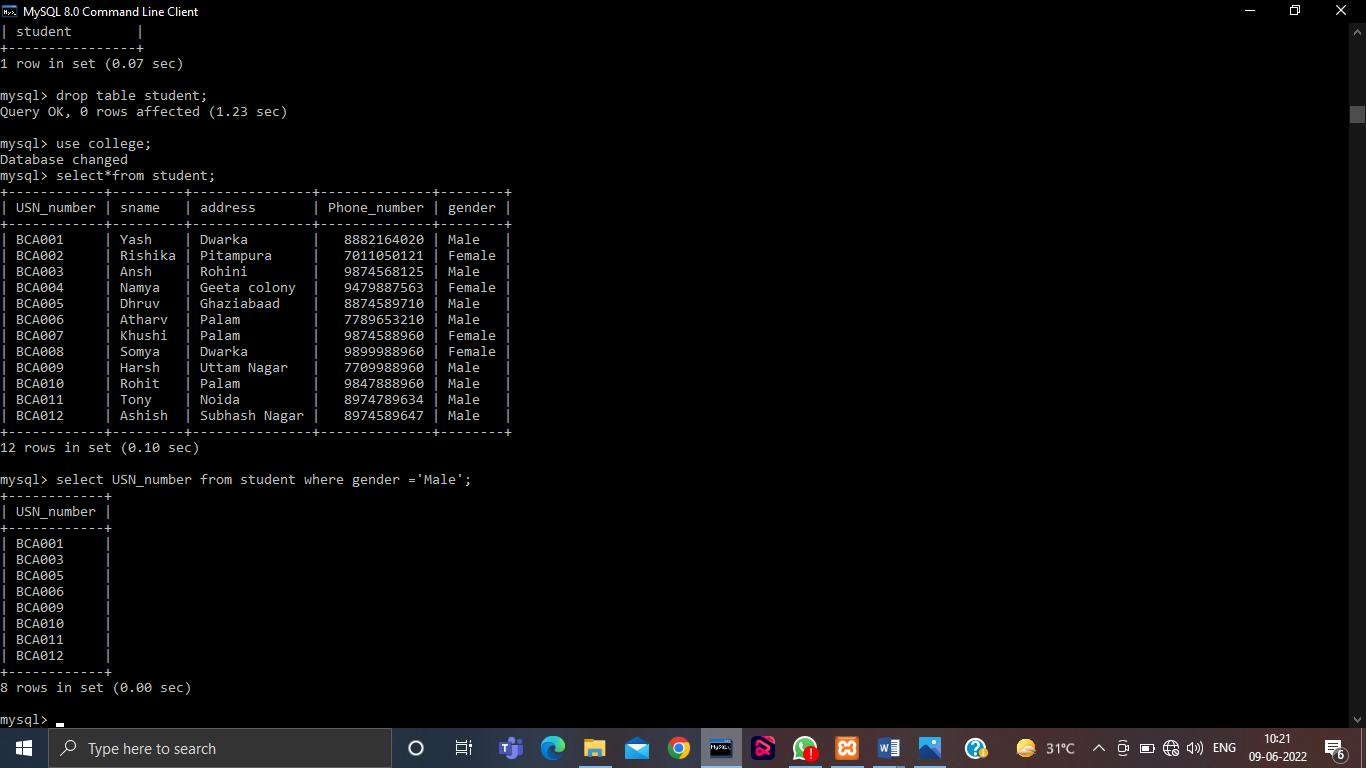
Select \* from <table\_name>;



* **Selective contents..**

Query:

Select <column\_name>,<column\_name> from <table\_name> where <condition>;

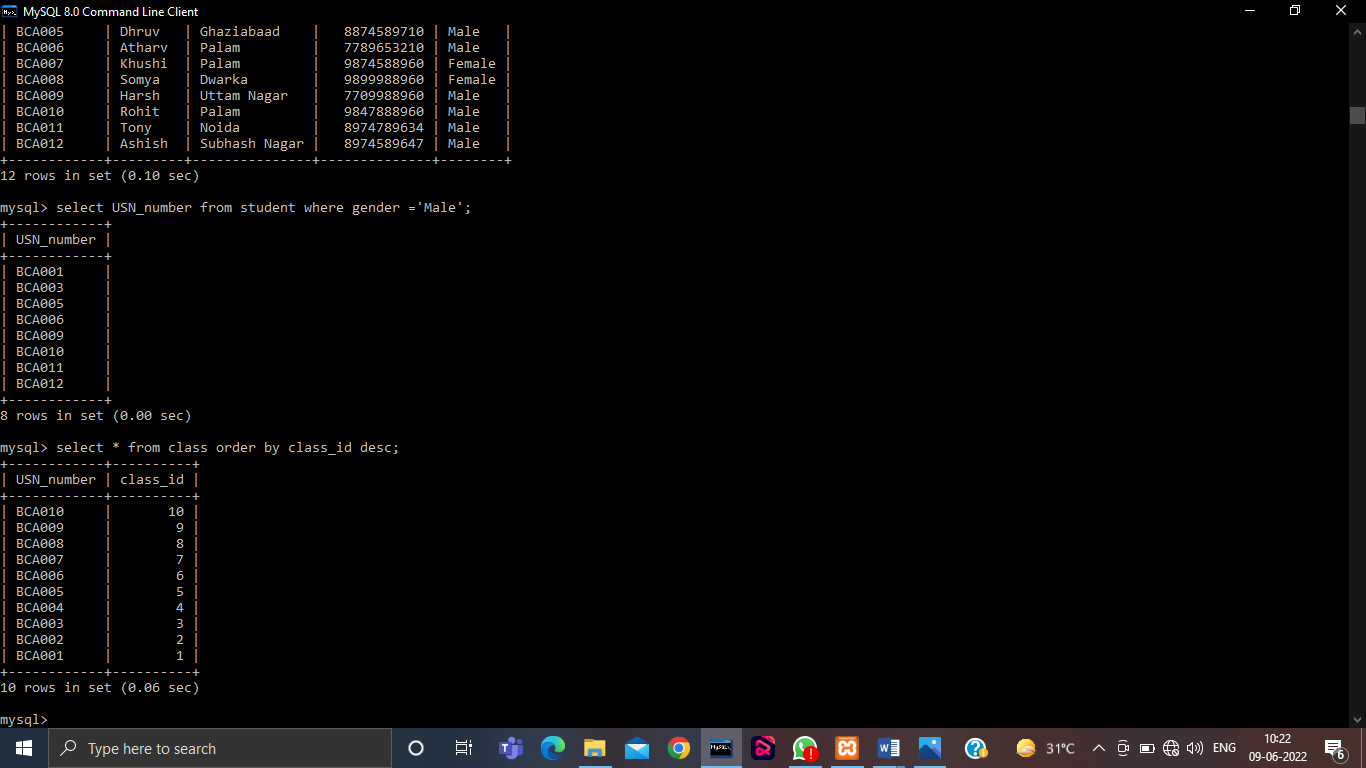


* **Sorting of Data**

Query:

Select \* from <table\_name> order by <column\_name>;

Sorting the Data in descending order.



Sorting the Data in ascending order.



**Q6 Create table using following integrity constraints:**

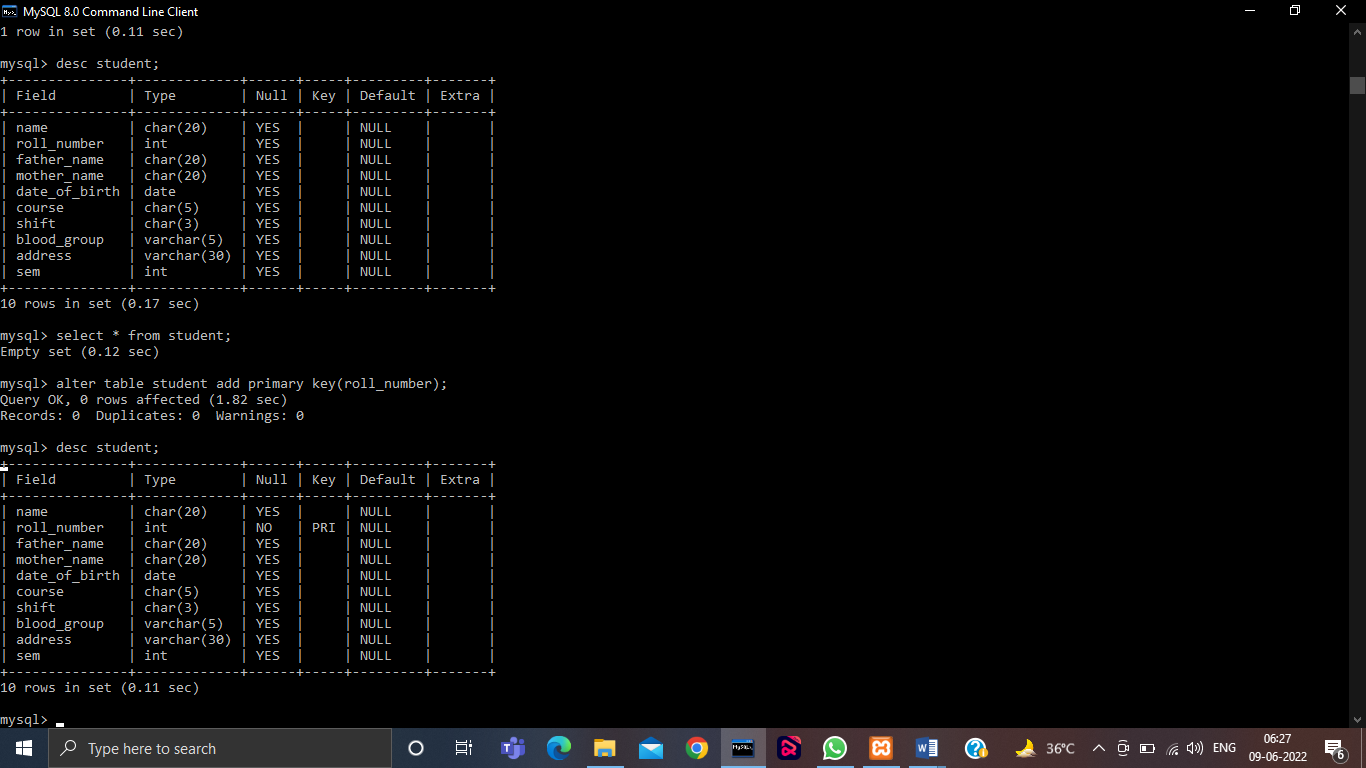
* **Primary Key**
* **Unique Key**
* **Not Null**
* **Check**
* **Default**
* **Foreign Key**

**Primary Key-**

Adding primary key in the existing table.

Query:-

Alter table <table\_name> add primary kay(<column\_name>);

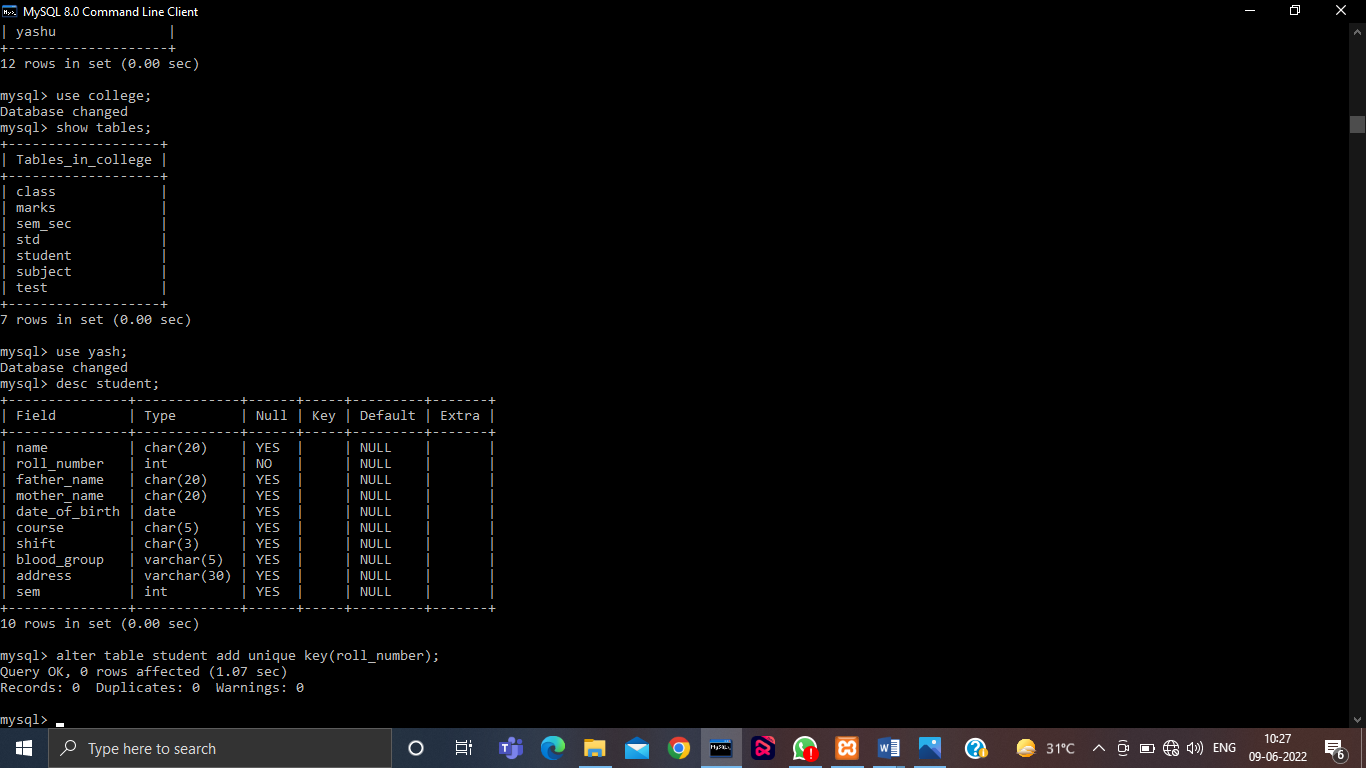


**Unique Key-**

Adding unique key in the existing table.

Query: -

Alter table <table\_name> add unique(<column\_name>);

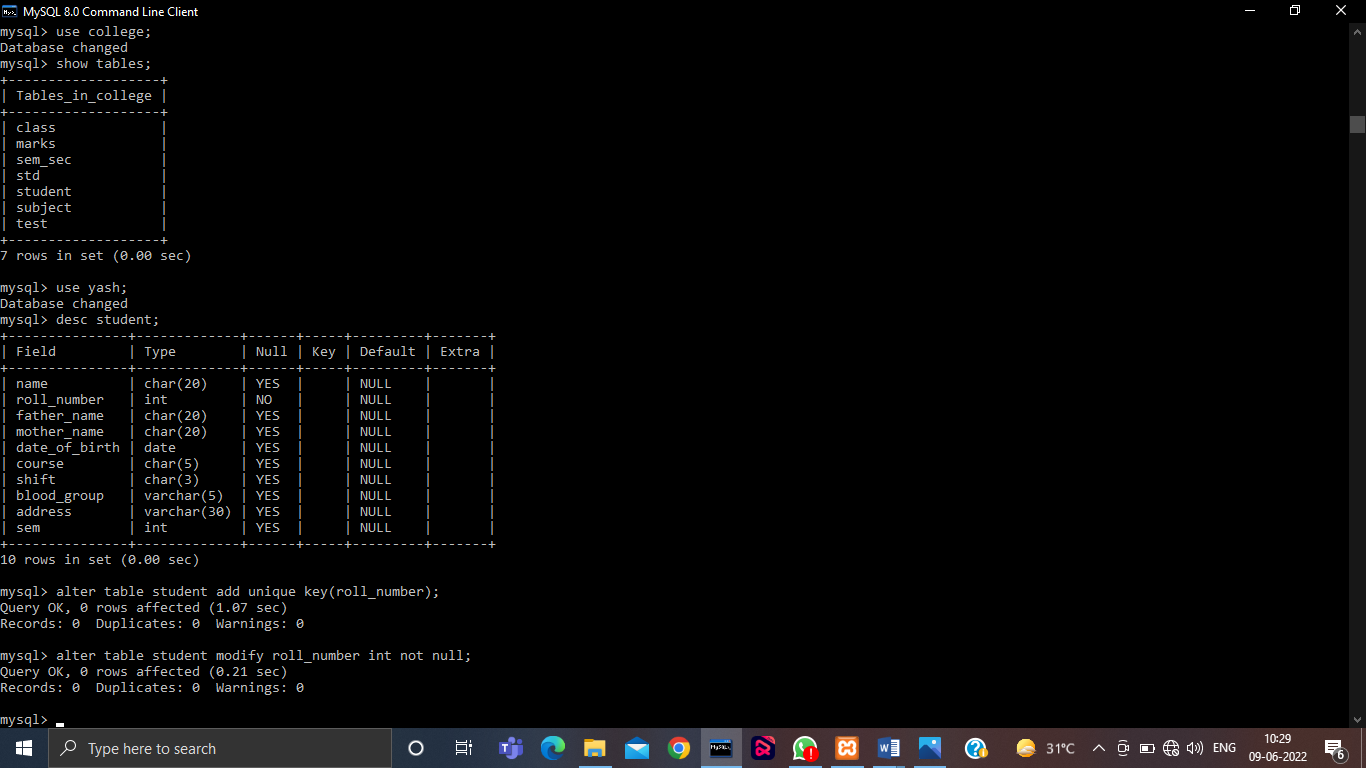


**Not Null-**

Adding not null constraint in the existing table.

Query: -

Alter table <table\_name> modify <column\_name> type not null;

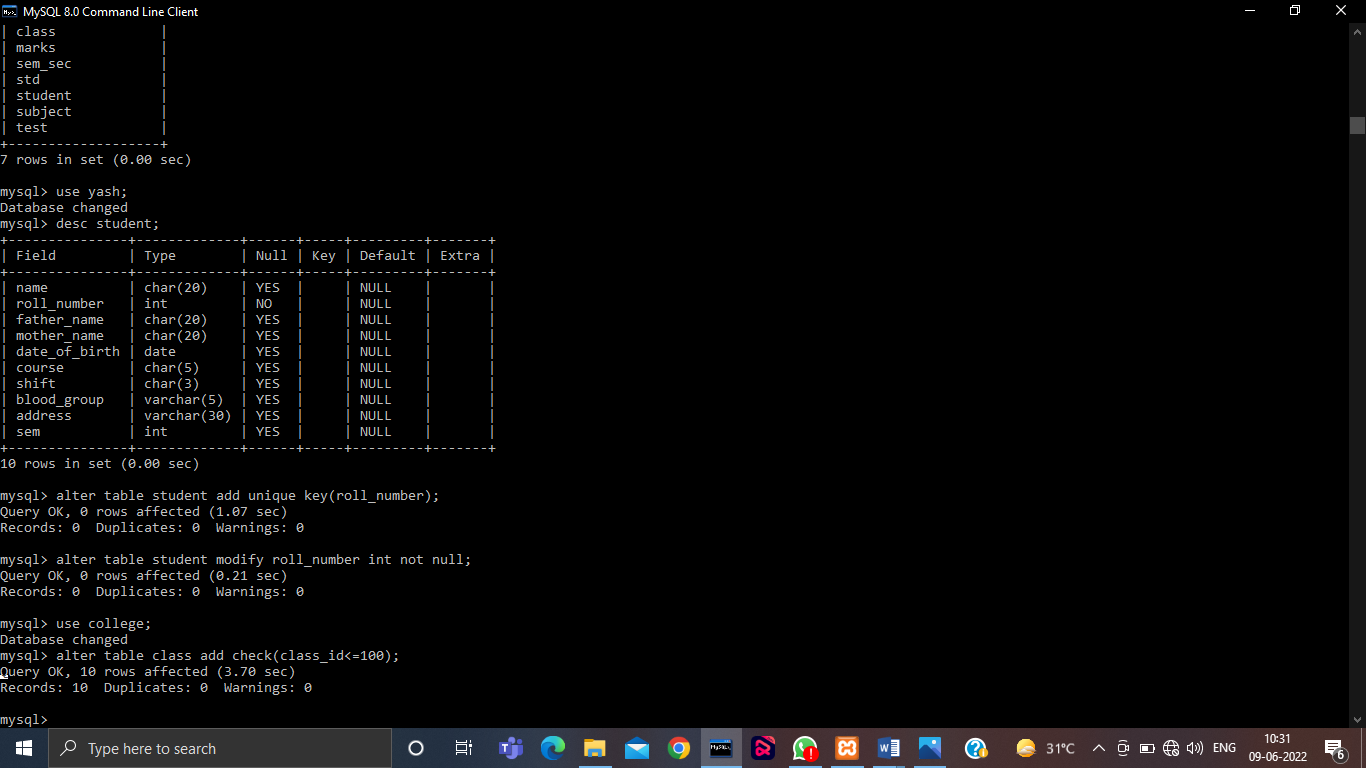


**Check-**

Adding check constraint in the existing table.

Query: -

Alter table <table\_name> add check(condition);

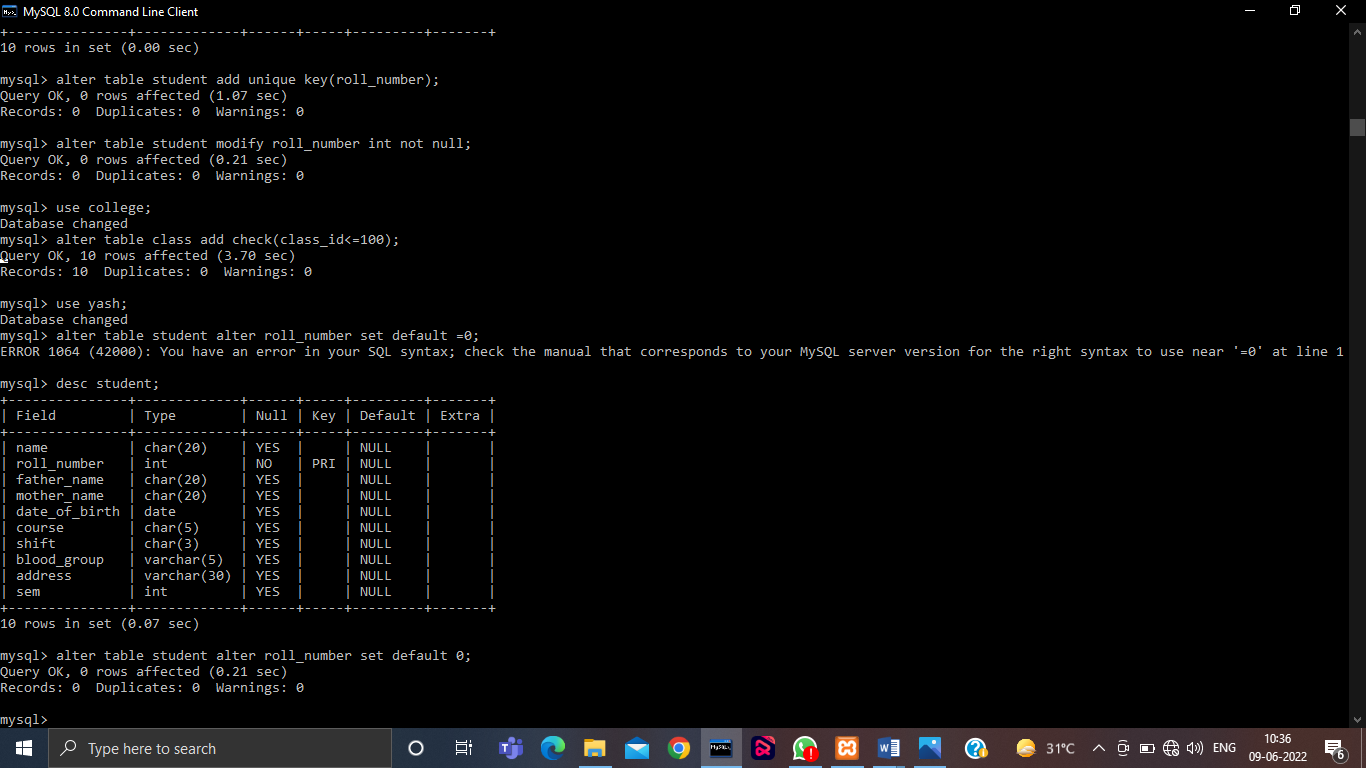


**Default-**

Adding default value in a existing table.

Query: -

Alter table <table\_name> alter <column\_name> set default <value>;

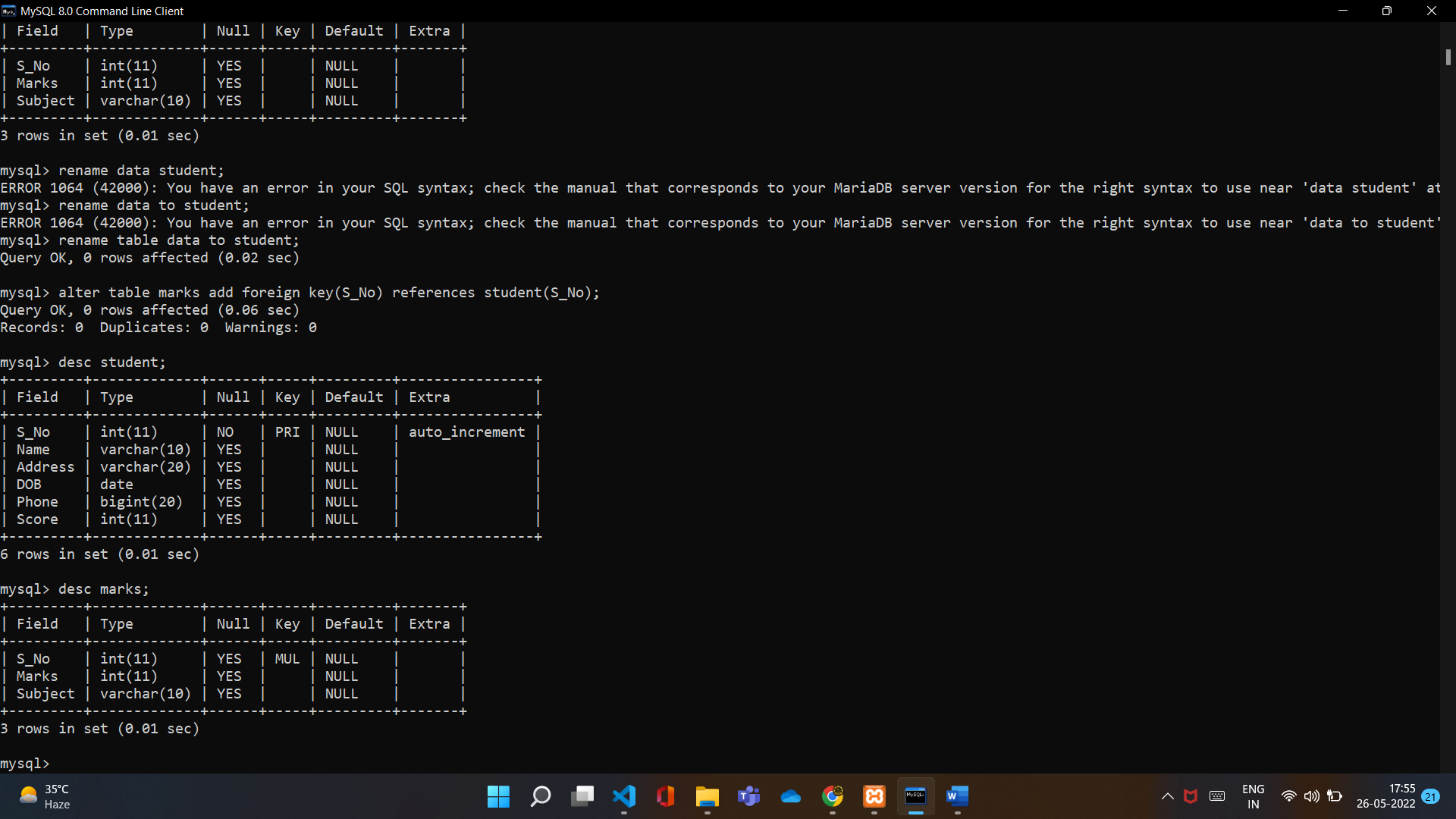


**Foreign Key-**

Adding a relation between 2 existing tables.

Query: -

Alter table <Table\_name> add foreign key(<coumn name>) references <table\_name>(<column>);



**Q7 Write queries to execute following Aggregate functions**

**Sum, Avg, Count, Minimum and Maximum value of a numeric column of a table using aggregate function**

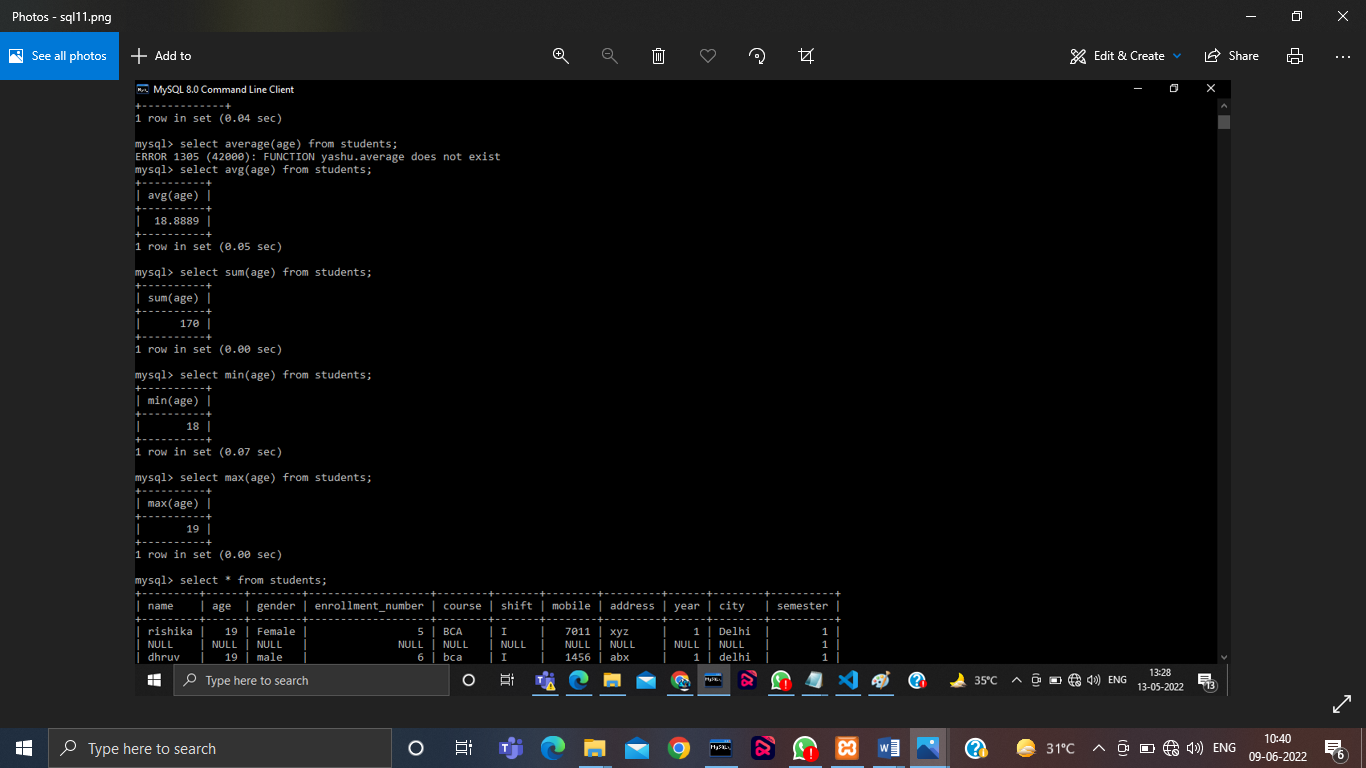
Query:-

Select sum(<column\_name>) from <table\_name>;

Select avg(<column\_name>) from <table\_name>;

Select min(<column\_name>) from <table\_name>;

Select max(<column\_name>) from <table\_name>;



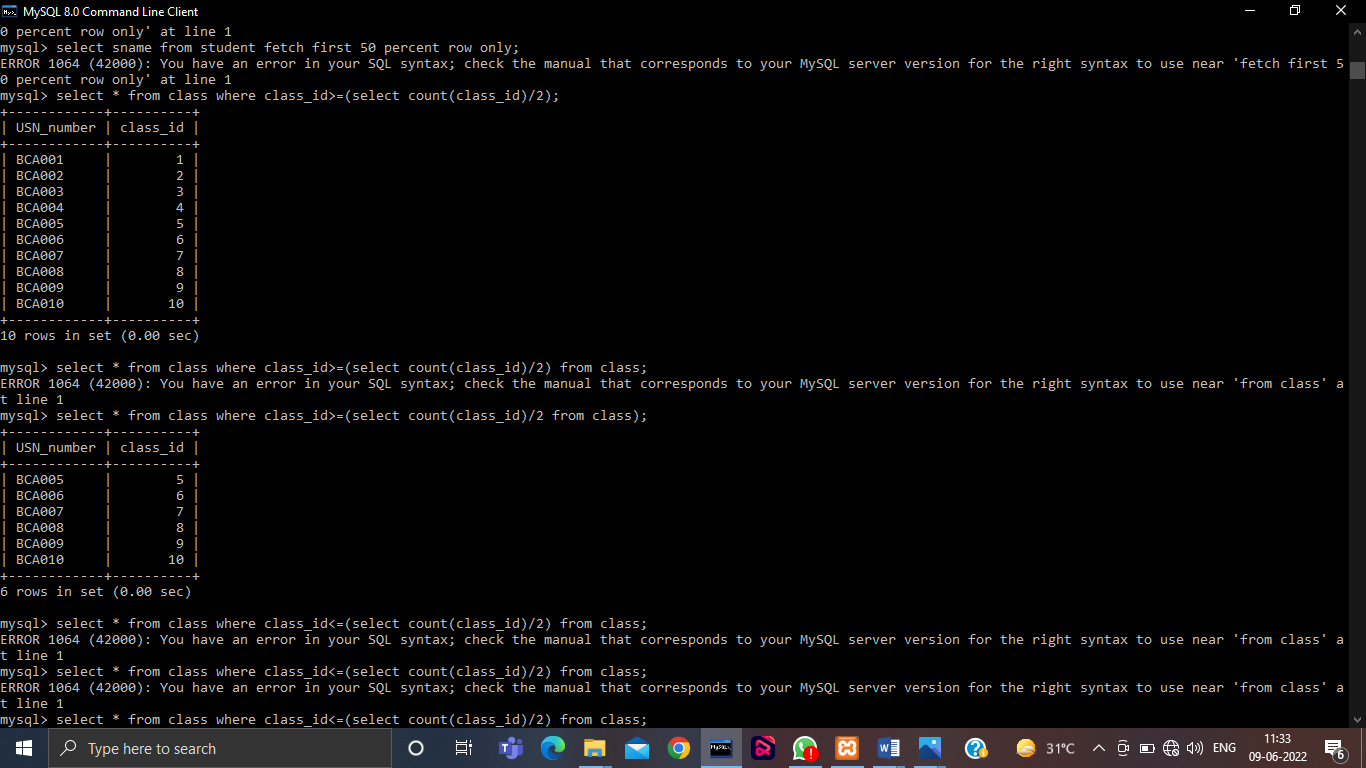
**Q8 Retrieve data from a table using alias names .**

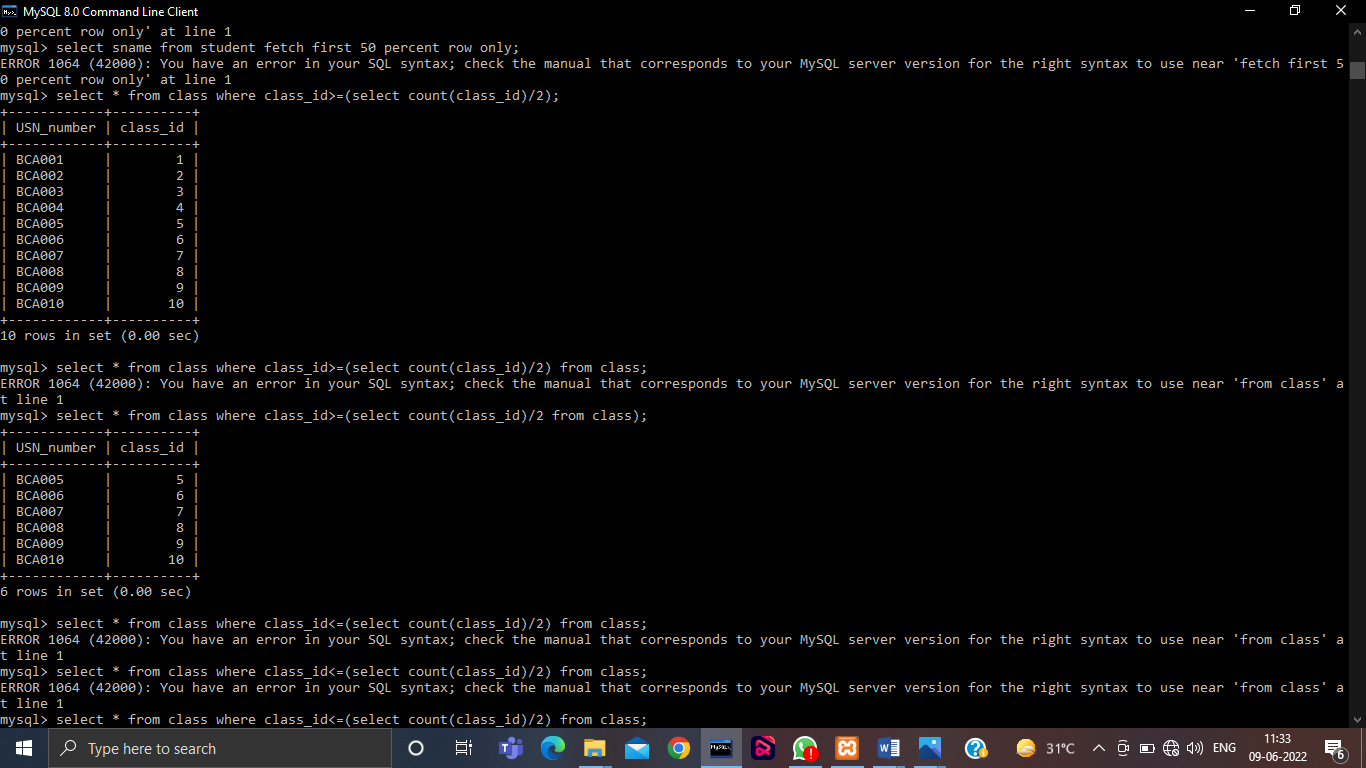
Query: -

Select <column\_name > as ‘Name’ from <table\_name>;



**Q9 Retieve data from a table using nested Queries.**



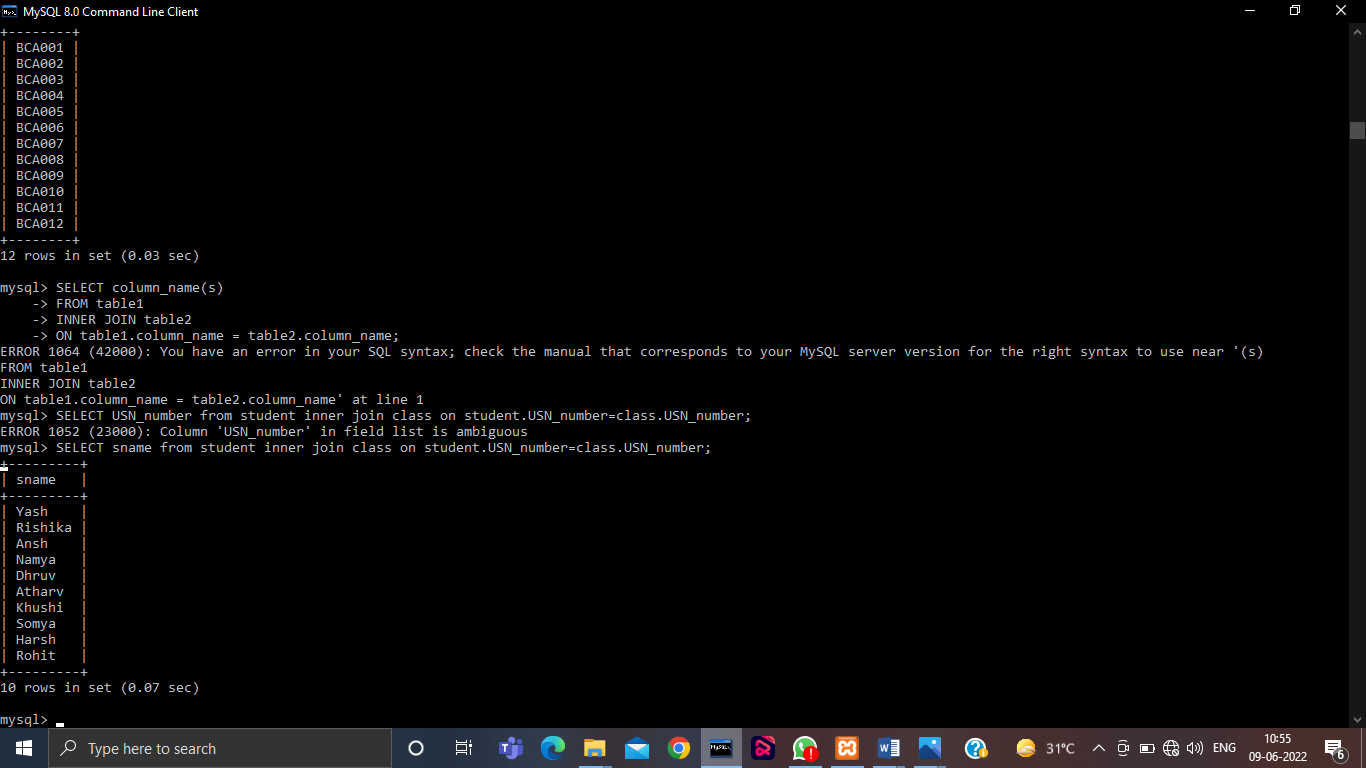


**Q10 Retrieve data from more than one table using inner join, left outer, right outer and full outer joins**

Query:-

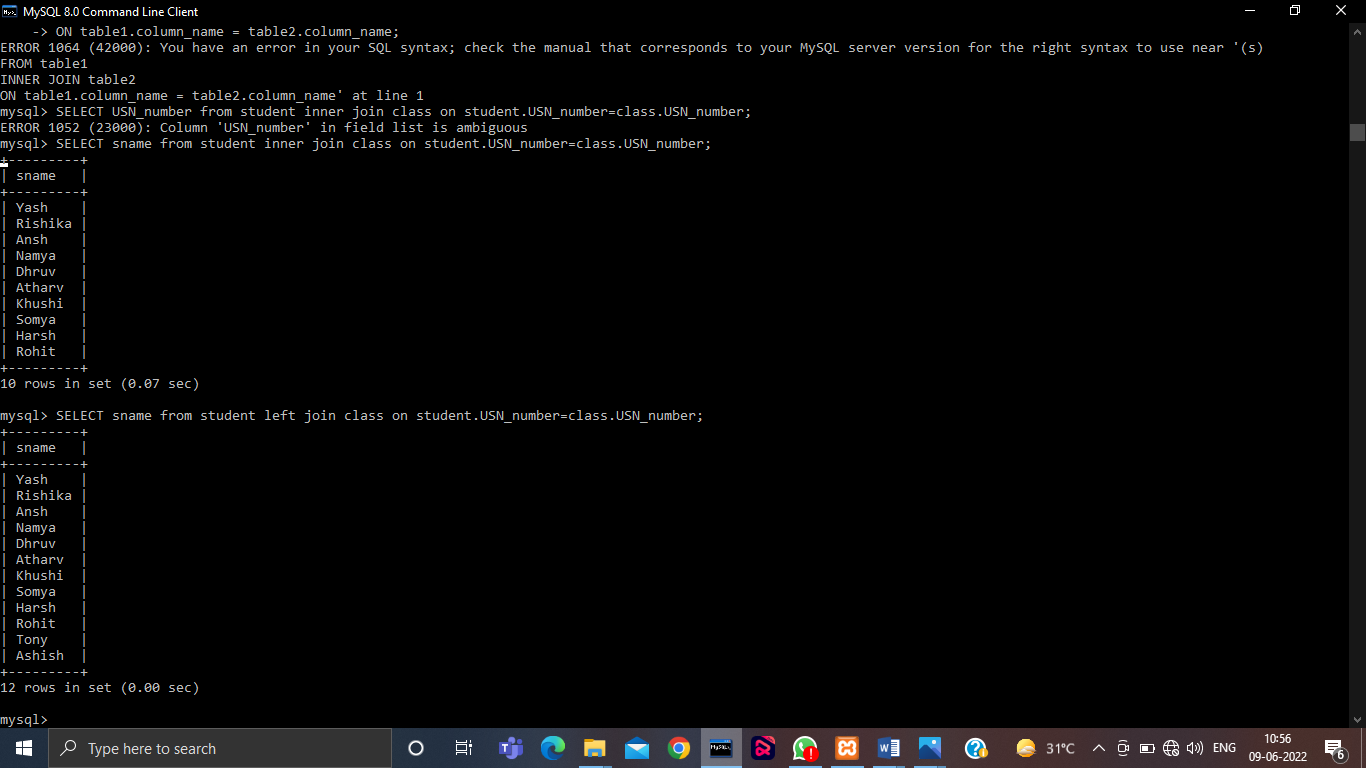
**Inner join**

SjELECT column\_name(s)  
FROM table1  
INNER JOIN table2ON table1.column\_name = table2.column\_name;



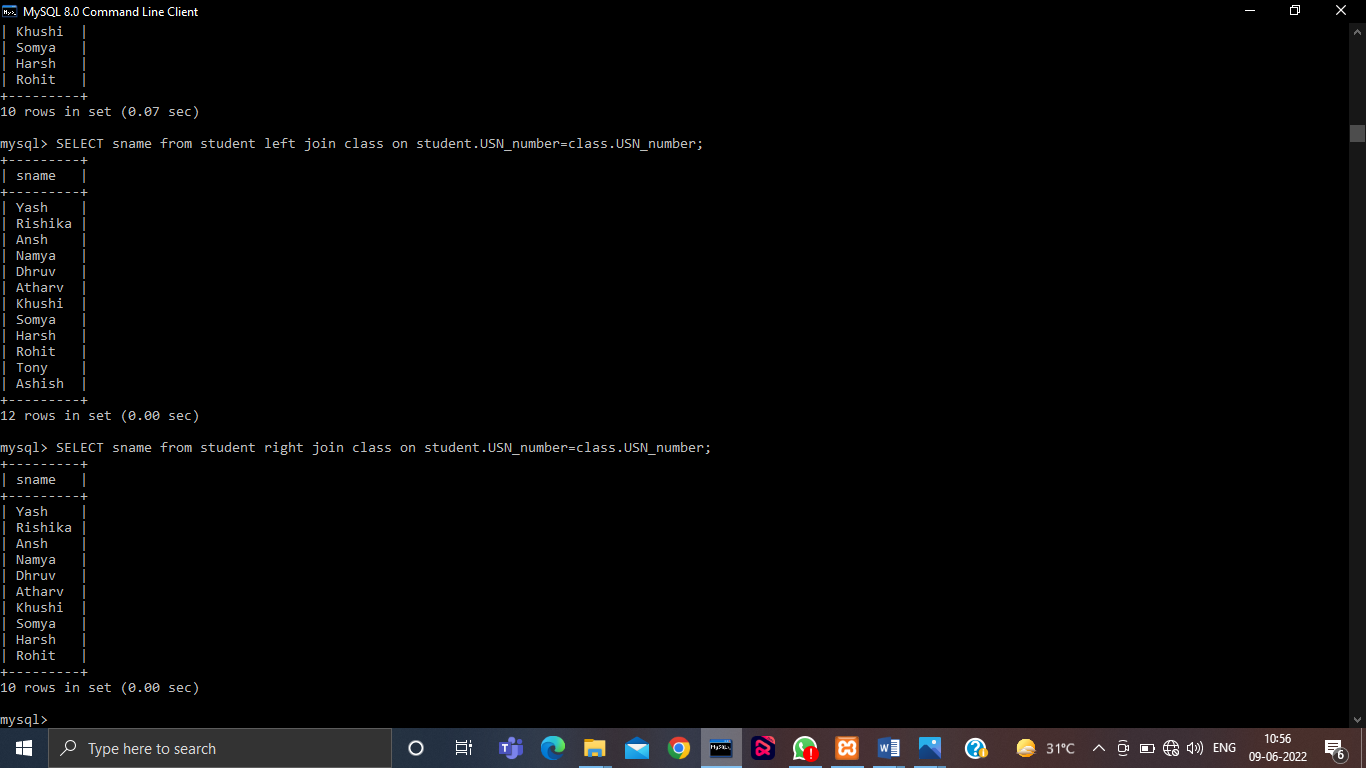
**Left Join**

SELECT column\_name(s)  
FROM table1  
LEFT JOIN table2ON table1.column\_name = table2.column\_name;

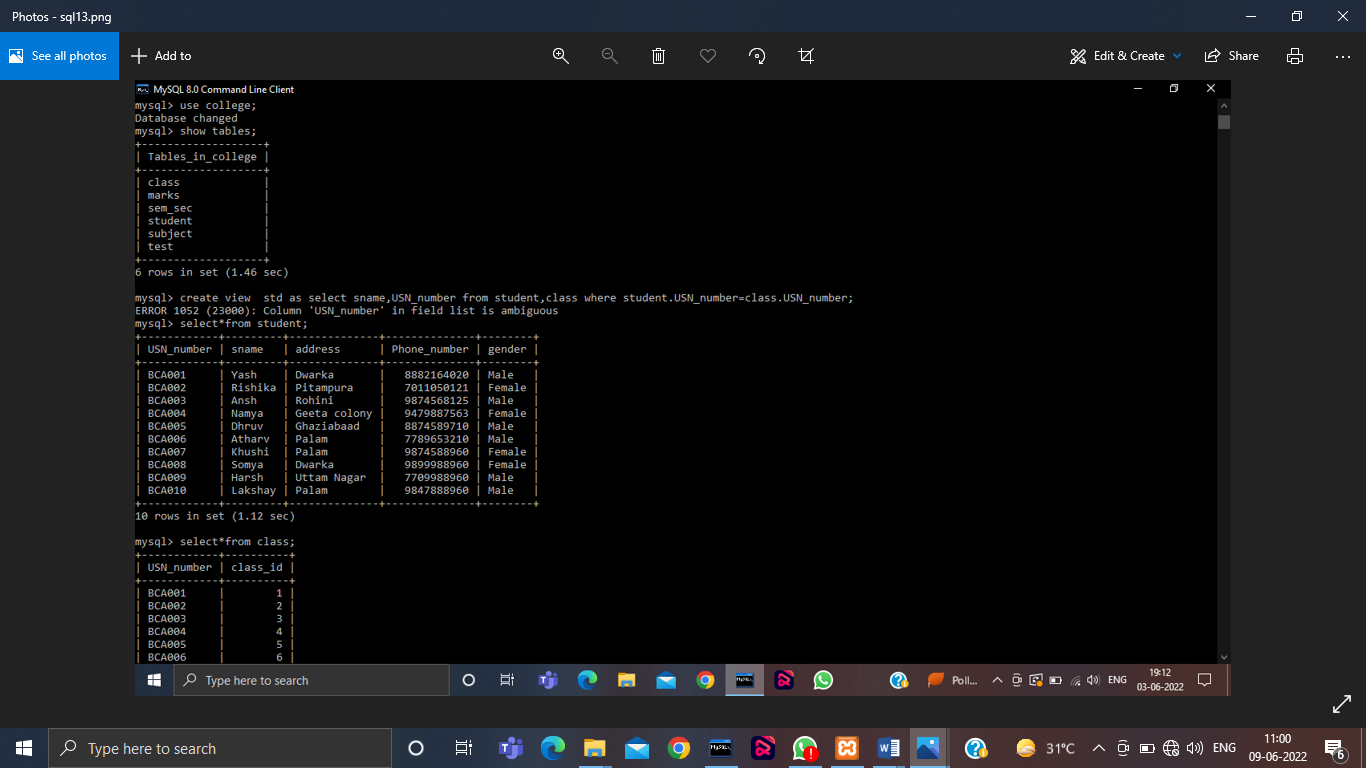


**Right Join**

SELECT column\_name(s)  
FROM table1  
RIGHT JOIN table2ON table1.column\_name = table2.column\_name;



**Q11 Create view from one table and more than one table.**



**Application Based Practicals**

Consider the Insurance company’s Database given below. The primary keys are underlined and the data typesare specified.

* PERSON(driver\_id# : string, name : string, address : string)
* CAR(regno : string, model : string, year : int)
* ACCIDENT(report\_number : int, acc\_date : date, location : string)
* OWNS(driver\_id# : string, regno : string)
* PARTICIPATED(driver\_id# : string, regno : string, report\_number : int, damage\_amount :number(10,2) )

(i) Create the above tables by properly specified the primary key and the foreign key

(ii) Enter at least five tuples for each relation

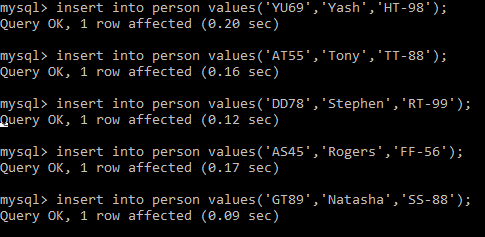
(iii) Demonstrate how you can

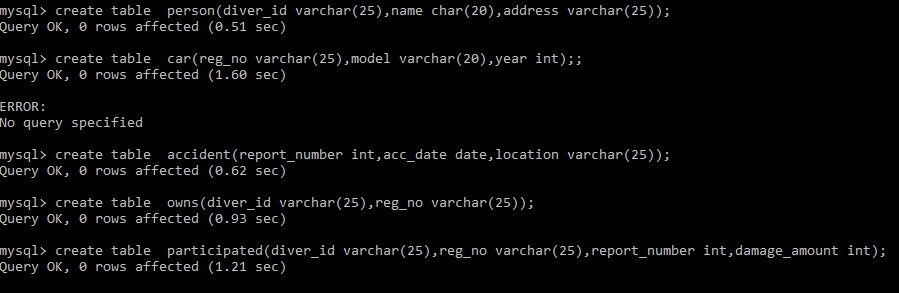
a. Update the damage amount for the car with a specific regno, the accident with report number 12 to 25000.

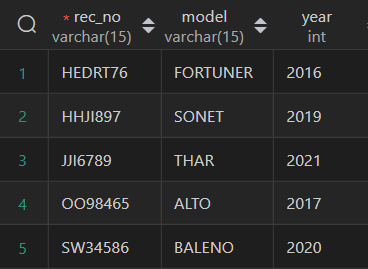
b. Add a new accident to the database.

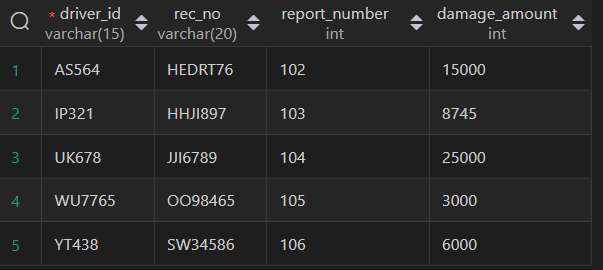
(iv) Find the total number of people who owned cars that were involved in accident in 2002.

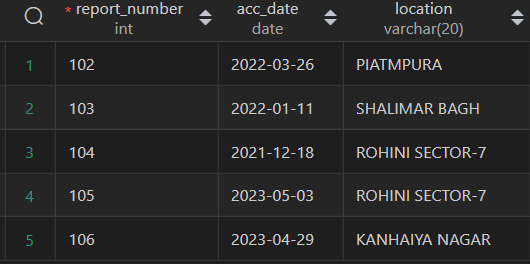
Find the number of accident in which cars belonging to a specific models were involved

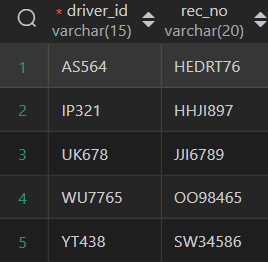




****

****

****

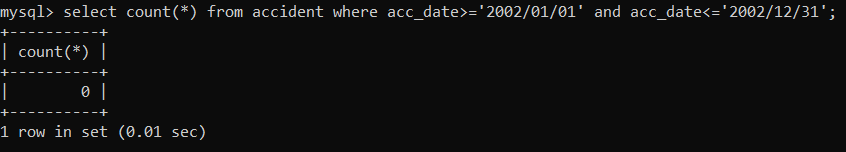
****

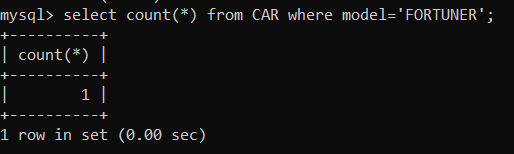
**iii)**

****

****

**iv)**

****

****

**Q 14.Consider the following schema of a library management system.Write the SQL queries for the questions given below;**

**Student(Stud\_no : integer, Stud\_name: string)**

**Membership(Mem\_no: integer, Stud\_no: integer)**

**Book\_(book\_no: integer, book\_name:string, author: string)**

**lss\_rec\_(iss\_no:integer, iss\_date: date, Mem\_no: integer, book\_no: integer)**

**(i) Create the tables with the appropriate integrity constraints**

**(ii) Insert around 10 records in each of the tables**

**(iii)Display all records for all tables**

**(iv) List all the student names with their membership numbers**

**(v) List all the issues for the current date with student and Book names**

**(vi) List the details of students who borrowed book whose author is Elmarsi & Navathe**

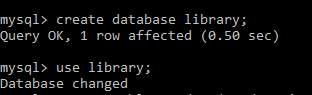
**(vii) Give a count of how many books have been bought by each student**

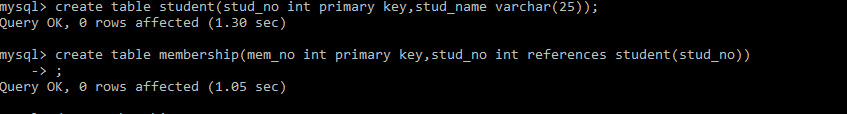
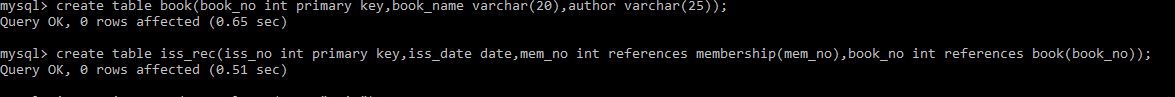
**(viii) Give a list of books taken by student with stud\_no as 1005**

**(ix) Delete the List of books details which are issued as of today**

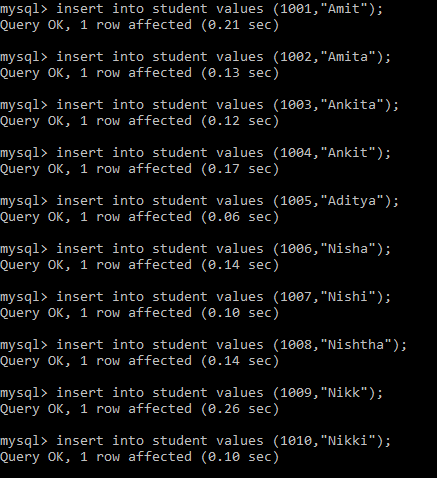
**(x) Create a view which lists out the iss\_no, iss \_date, stud\_name, book name**

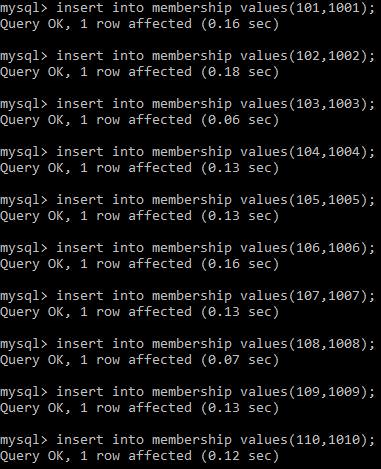
i)

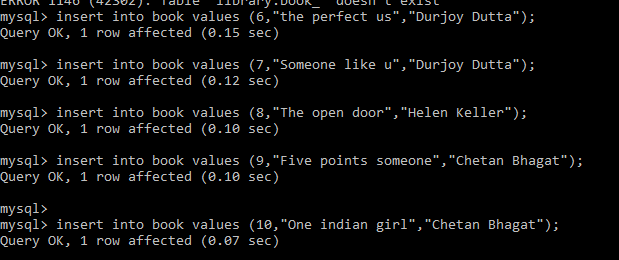


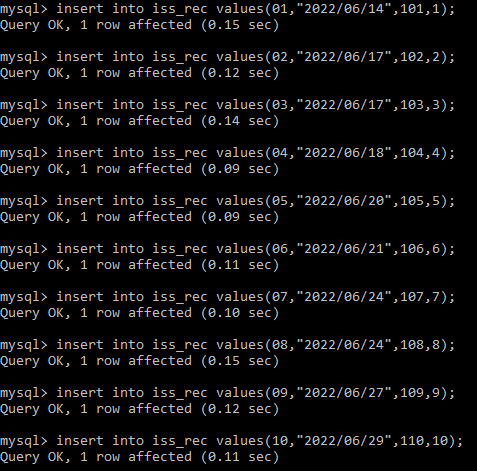
 

ii)

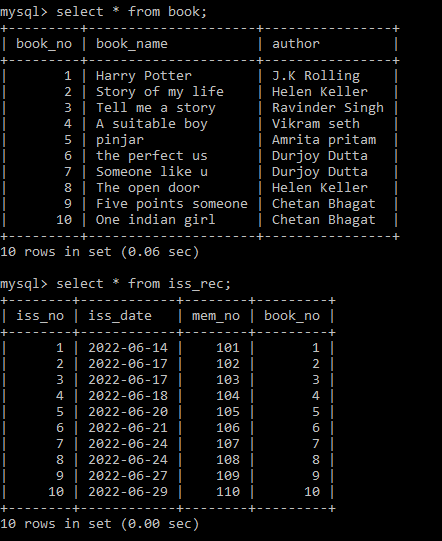


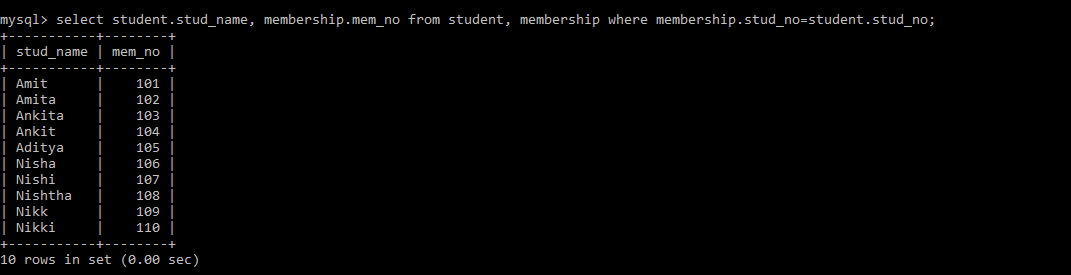


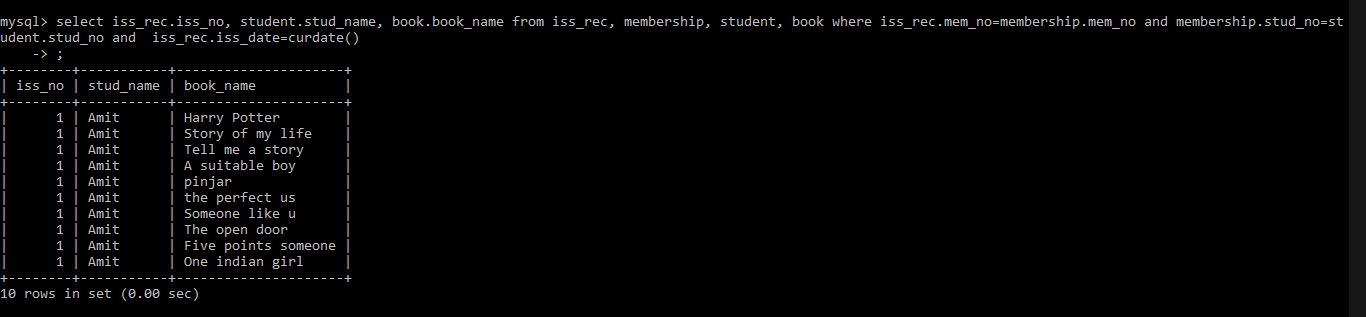


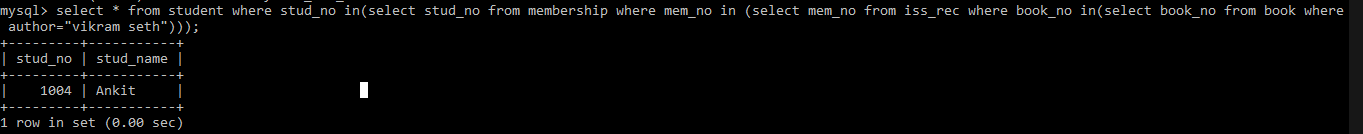
iii)

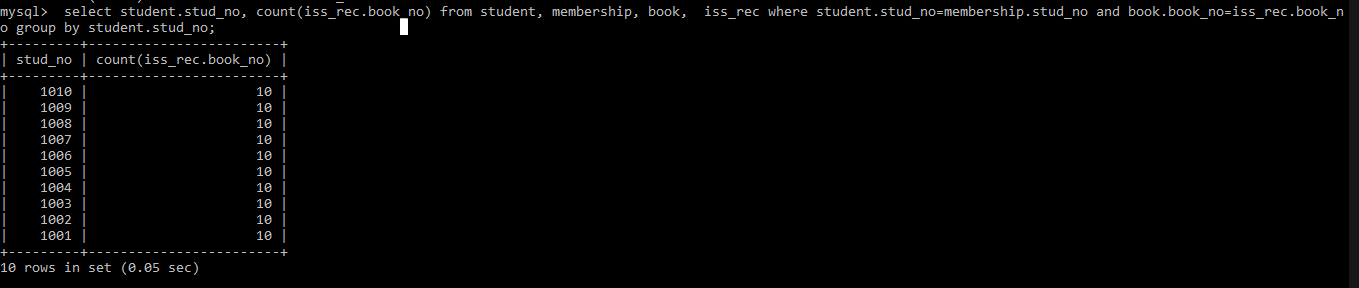
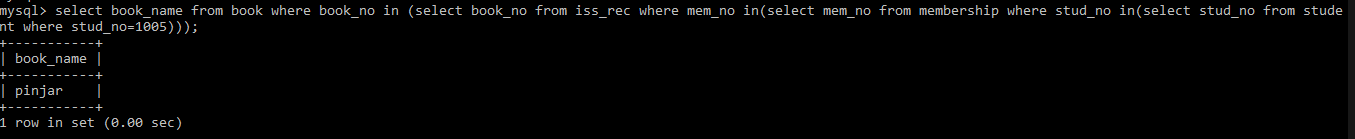
 

iv) 

v)



vi) 

vii) viii) ix) 

x) 