**Manav Rachna International Institute**

**of Research and Studies**

**School of Computer Applications**

**(Department of Computer Applications)**

|  |  |
| --- | --- |
| **Submitted By** | |
| **Student Name** | **Vilohit Barua** |
| **Roll No** | **24/SCA/BCA(AI&ML)/073** |
| **Course** | **Bachelor of Computer Applications (BCA)** |
| **Semester** | **1st Semester** |
| **Section/Group** | **E** |
| **Department** | **Computer Applications** |
| **Batch** | **2024-28** |
|  | |
| **Submitted To** | |
| **Faculty Name** | **Mrs. Aastha** |

|  |  |
| --- | --- |
|  | **SCHOOL OF COMPUTER APPLICATIONS** |

## Experiment No: 1

**Creating 3 Tables Name Student,Employee and courses.**

-- Creating the Student table

CREATE TABLE Student (student\_id INT PRIMARY KEY, student\_name VARCHAR(40) NOT NULL, address VARCHAR(300), gender VARCHAR(15), course VARCHAR(8));

-- Inserting values into the Student table

INSERT INTO Student (student\_id, student\_name, address, gender, course) VALUES

(1, 'John Doe', '123 Elm Street', 'Male', 'MCA'),

(2, 'Jane Smith', '456 Oak Avenue', 'Female', 'BCA'),

(3, 'Mike Brown', '789 Pine Road', 'Male', 'MBA'),

(4, 'Lisa Ray', '321 Maple Lane', 'Female', 'BBA'),

(5, 'Tom White', '654 Cedar Blvd', 'Male', 'PhD');

-- Creating the Course table

CREATE TABLE Course ( course\_id NUMBER(3) PRIMARY KEY, course\_name VARCHAR2(20) NOT NULL, location VARCHAR2(20));

-- Inserting values into the Course table

INSERT INTO Course (course\_id, course\_name, location) VALUES

(101, 'MCA', 'Building A'),

(102, 'BCA', 'Building B'),

(103, 'MBA', 'Building C'),

(104, 'BBA', 'Building D'),

(105, 'PhD', 'Building E');

-- Creating the Employee table

CREATE TABLE Employee ( emp\_id int(4) , emp\_name varchar(40) NOT NULL, department\_id int(3), salary int, commission int);

-- Inserting values into the Employee table

INSERT INTO Employee (emp\_id, emp\_name, department\_id, salary, commission) VALUES

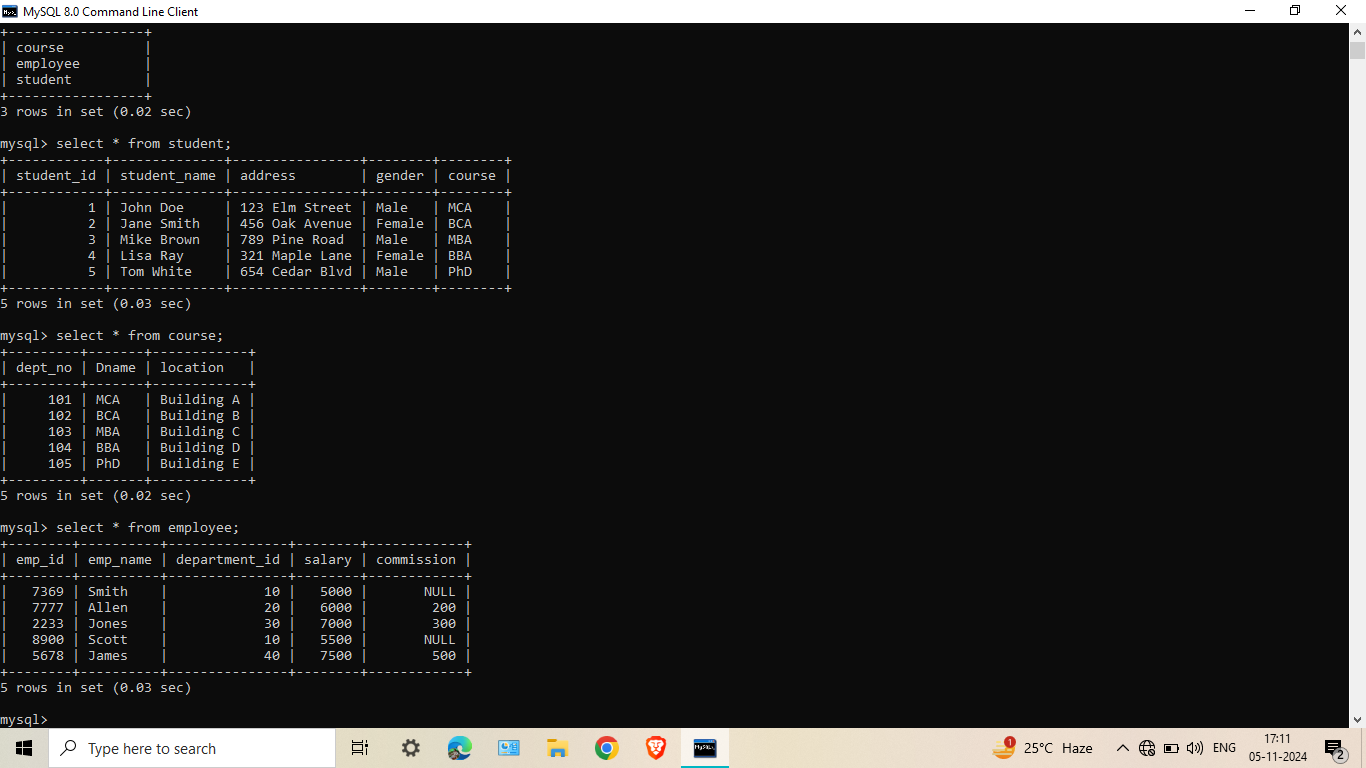
(7369, 'Smith', 10, 5000.00, NULL),

(7777, 'Allen', 20, 6000.00, 200.00),

(2233, 'Jones', 30, 7000.00, 300.00),

(8900, 'Scott', 10, 5500.00, NULL),

(5678, 'James', 40, 7500.00, 500.00);



## Experiment No: 2

**Experiment :-** List all information about all students from the student table

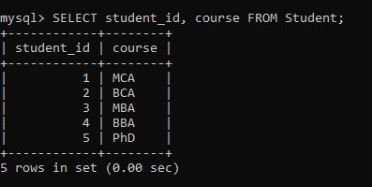
SELECT \* FROM Student;



## Experiment No: 3

**Experiment 3:-** List all student numbers along with their courses

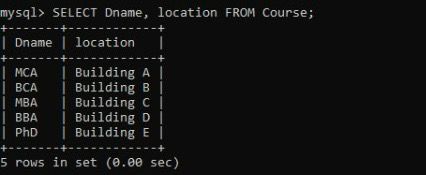
SELECT student\_id, course FROM Student;



**Experiment No: 4**

**Experiment 4:-** List course names and locations from the Course table

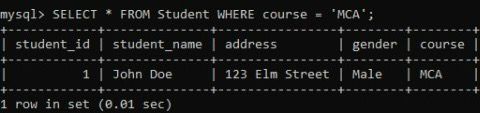
SELECT Dname, location FROM Course;



## Experiment No: 5

**Experiment 5:-** List the details of the students in MCA course

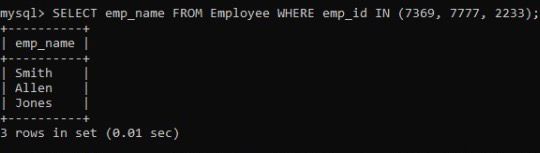
SELECT \* FROM Student WHERE course = 'MCA';



## Experiment No: 6

**Experiment 6:-** List the names of the employees whose employee numbers are 7369, 7777, 2233

SELECT emp\_name FROM Employee WHERE emp\_id IN (7369, 7777, 2233);

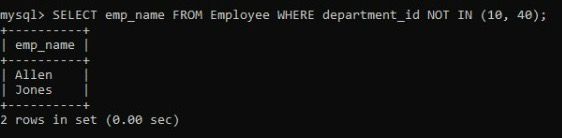


## 

## Experiment No: 7

**Experiment 7:-** List the employee names not belonging to department 10, 40

SELECT emp\_name FROM Employee WHERE department\_id NOT IN (10, 40);

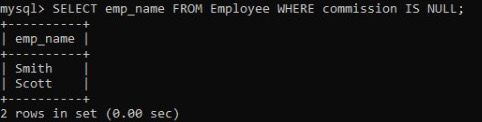


## 

## Experiment No: 8

**Experiment 8:-** List the employee names who are not eligible for commission

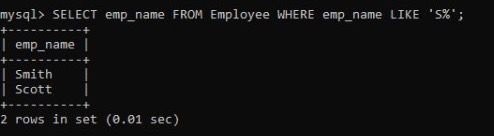
SELECT emp\_name FROM Employee WHERE commission IS NULL;



## Experiment No: 9

**Experiment 9:-** List the employees whose names start with “S” (uppercase only)

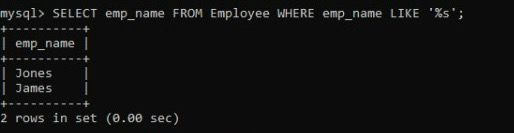
SELECT emp\_name FROM Employee WHERE emp\_name LIKE 'S%';



## Experiment No: 10

**Experiment No. 10:-** List the employees ending with name “s”

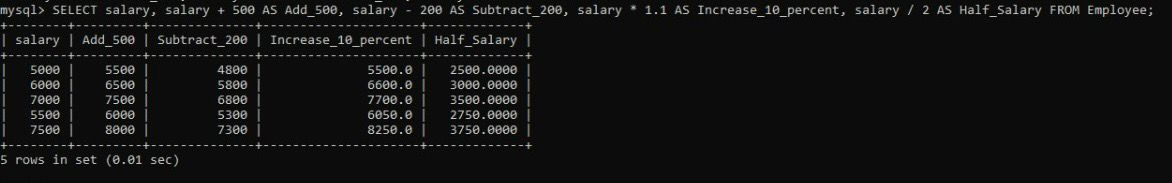
SELECT emp\_name FROM Employee WHERE emp\_name LIKE '%s';



## Experiment No: 11

**Experiment 11:-** Display all the Arithmetic functions used in SQL

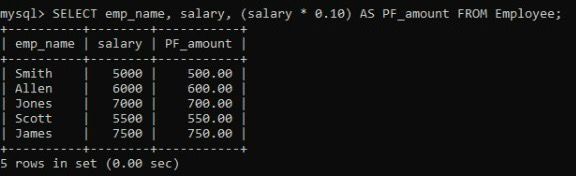
SELECT salary, salary + 500 AS Add\_500, salary - 200 AS Subtract\_200, salary \* 1.1 AS Increase\_10\_percent, salary / 2 AS Half\_Salary FROM Employee;



## Experiment No: 12

**Experiment 12:-** List the names, salary and PF amount of all employees (PF is calculated as 10% of salary)

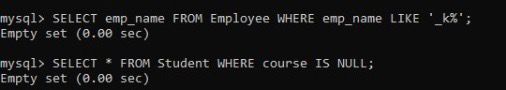
SELECT emp\_name, salary, (salary \* 0.10) AS PF\_amount FROM Employee;



## Experiment No: 13

**Experiment 13:-** List the employee names having “k” as the second character

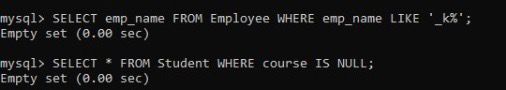
SELECT emp\_name FROM Employee WHERE emp\_name LIKE '\_k%';



## Experiment No: 14

**Experiment 14:-** List the students not assigned to any department

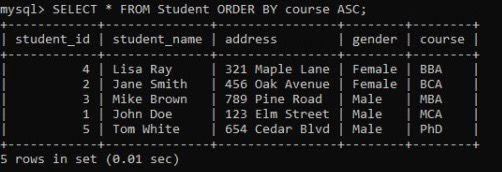
SELECT \* FROM Student WHERE course IS NULL;



## Experiment No: 15

**Experiment 15:-** . List the student details in ascending order of course

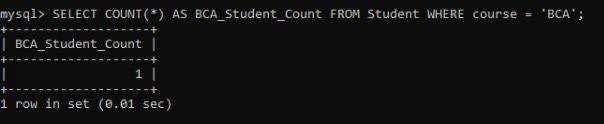
SELECT \* FROM Student ORDER BY course ASC;



## Experiment No: 16

**Experiment 16:-** List the number of students in BCA course

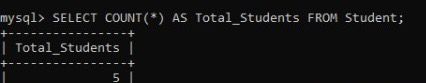
SELECT COUNT(\*) AS BCA\_Student\_Count FROM Student WHERE course = 'BCA';



## Experiment No: 17

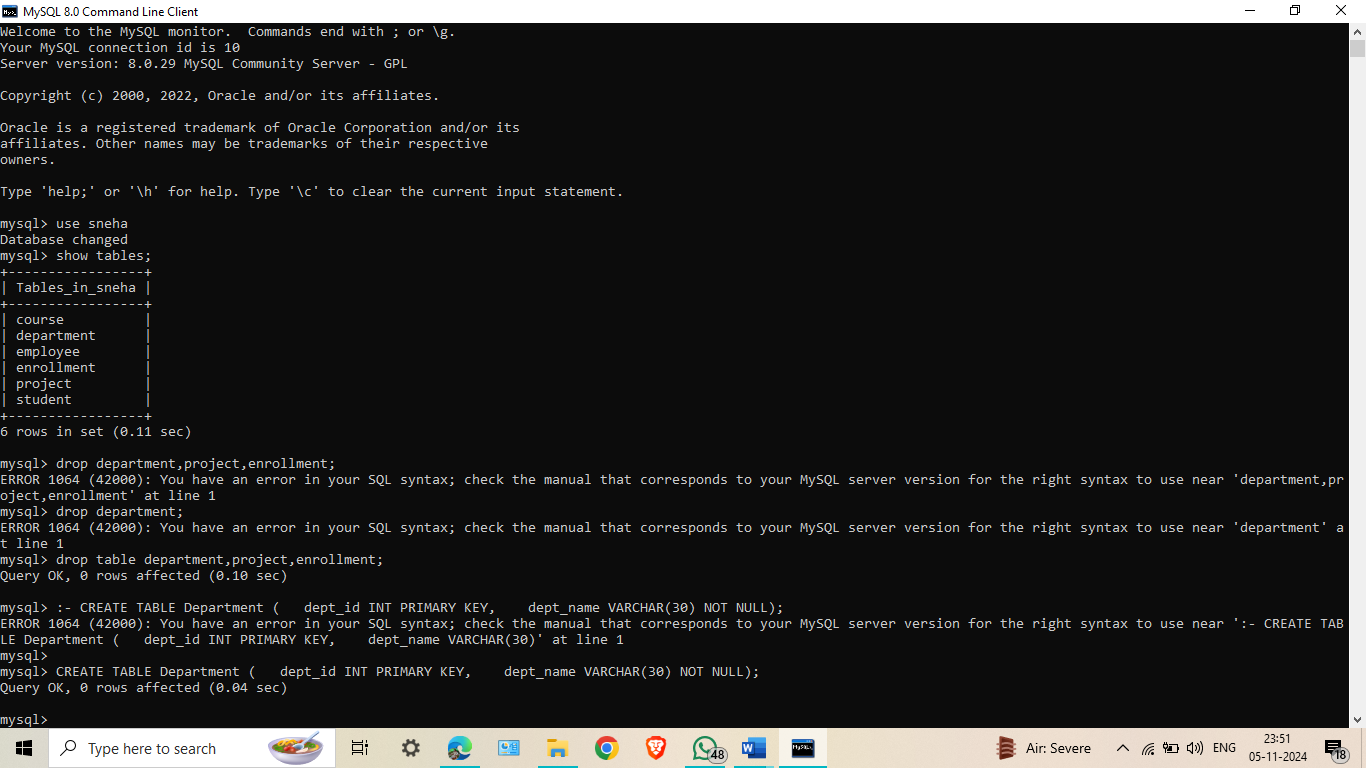
## Experiment 17:- List the number of students available in student table

## SELECT COUNT(\*) AS Total\_Students FROM Student;



## Experiment No: 18

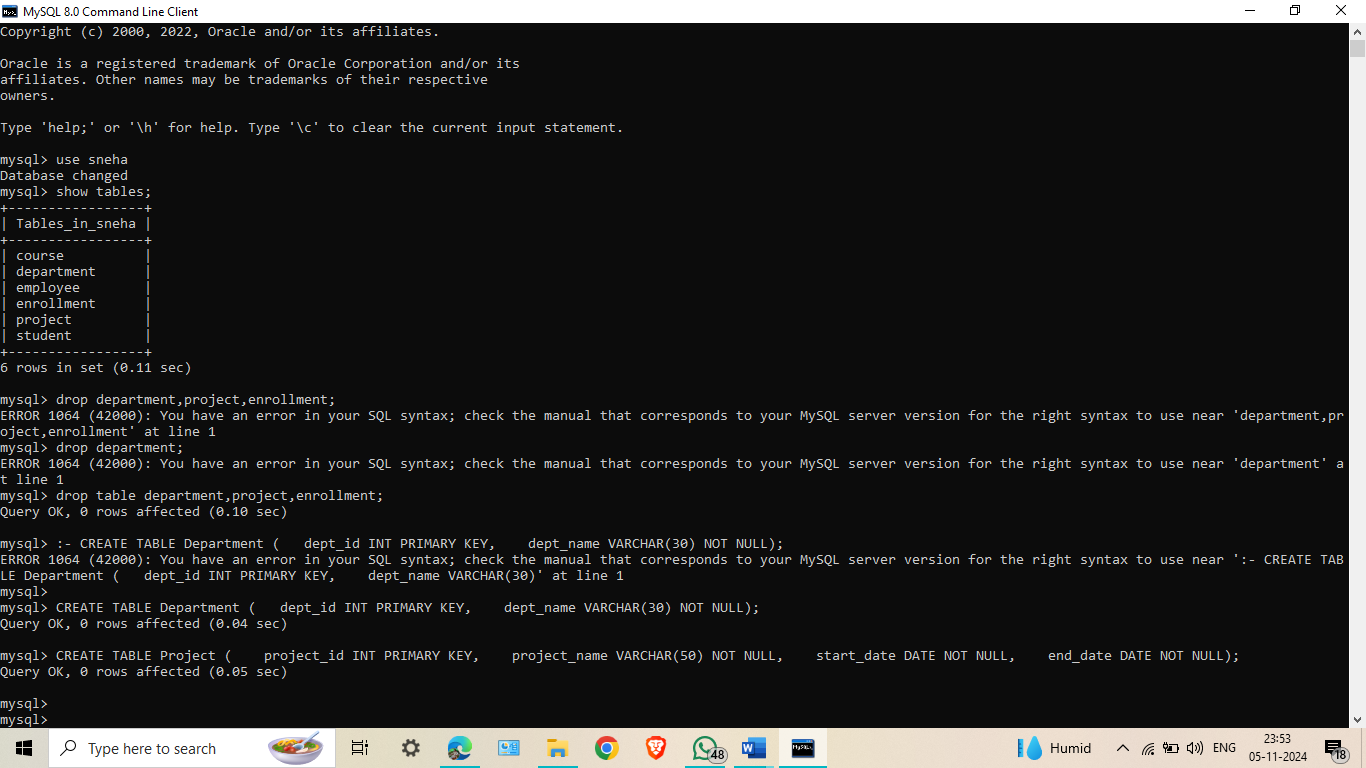
## Experiment 18:- CREATE TABLE Department ( dept\_id INT PRIMARY KEY, dept\_name VARCHAR(30) NOT NULL);



## Experiment No: 19

## Experiment 19:- Create a table with all columns having NOT NULL constraints

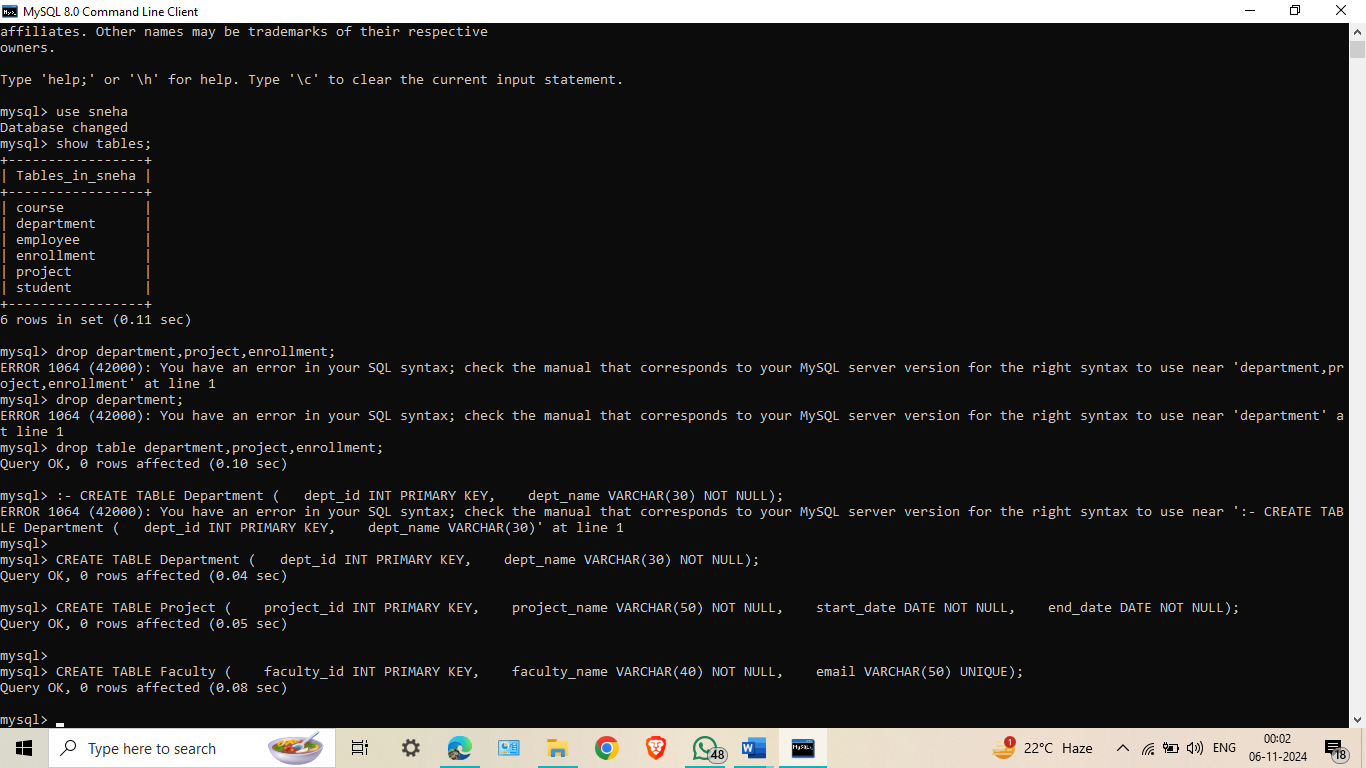
CREATE TABLE Project ( project\_id INT PRIMARY KEY, project\_name VARCHAR(50) NOT NULL, start\_date DATE NOT NULL, end\_date DATE NOT NULL);



## Experiment No: 20

## Experiment 20:- Create a table with a unique key constraint

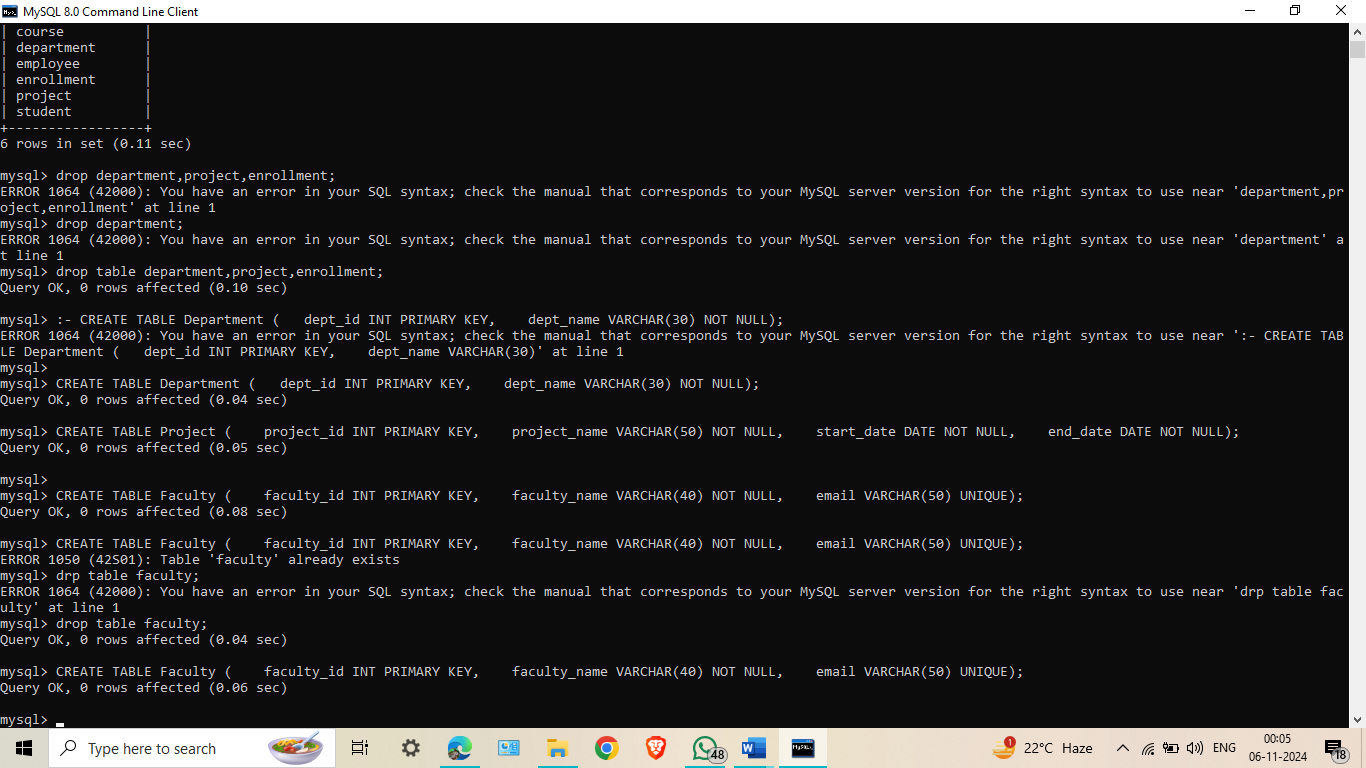
## CREATE TABLE Faculty ( faculty\_id INT PRIMARY KEY, faculty\_name VARCHAR(40) NOT NULL, email VARCHAR(50) UNIQUE);



## Experiment No: 21

## Experiment 21:- Create a table with a unique key constraint

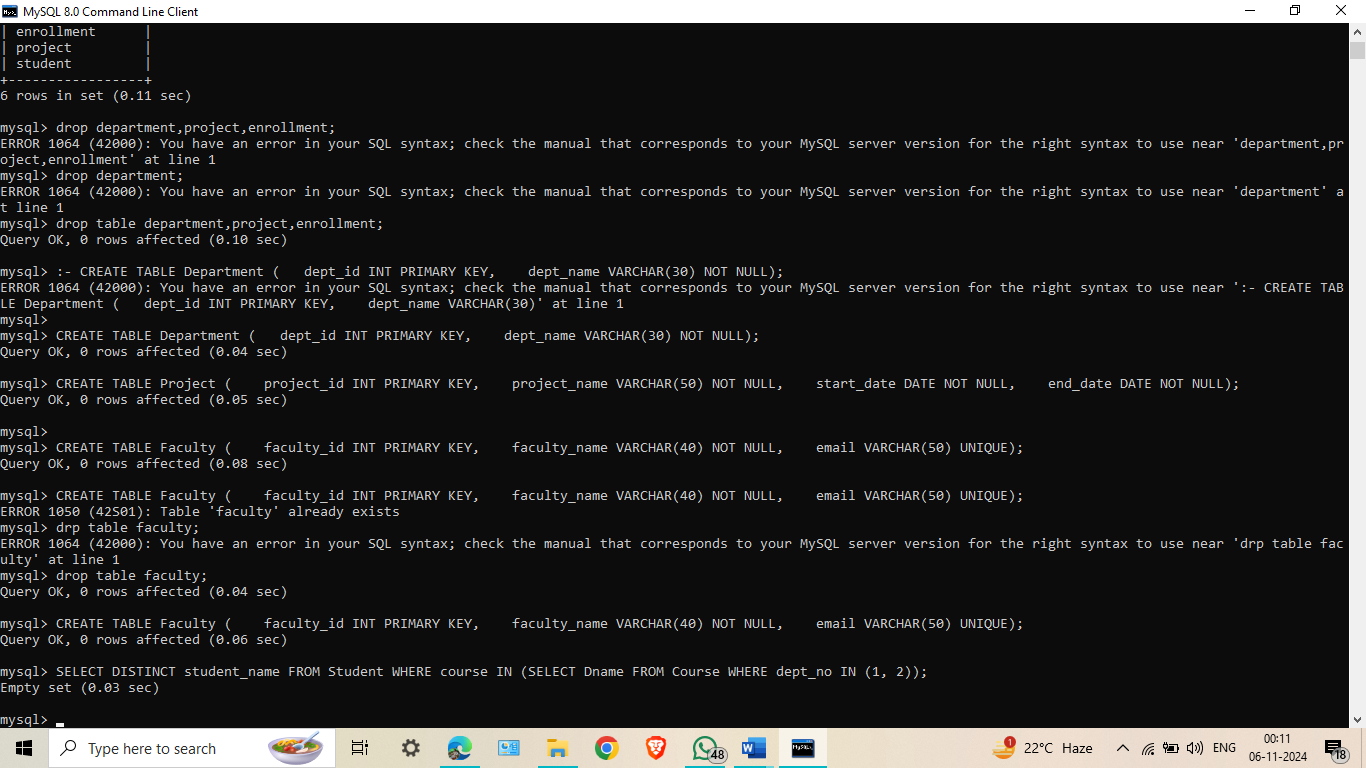
CREATE TABLE Faculty ( faculty\_id INT PRIMARY KEY, faculty\_name VARCHAR(40) NOT NULL, email VARCHAR(50) UNIQUE);



## Experiment No: 22

## Experiment 22:- Display the different students in department 1 and 2

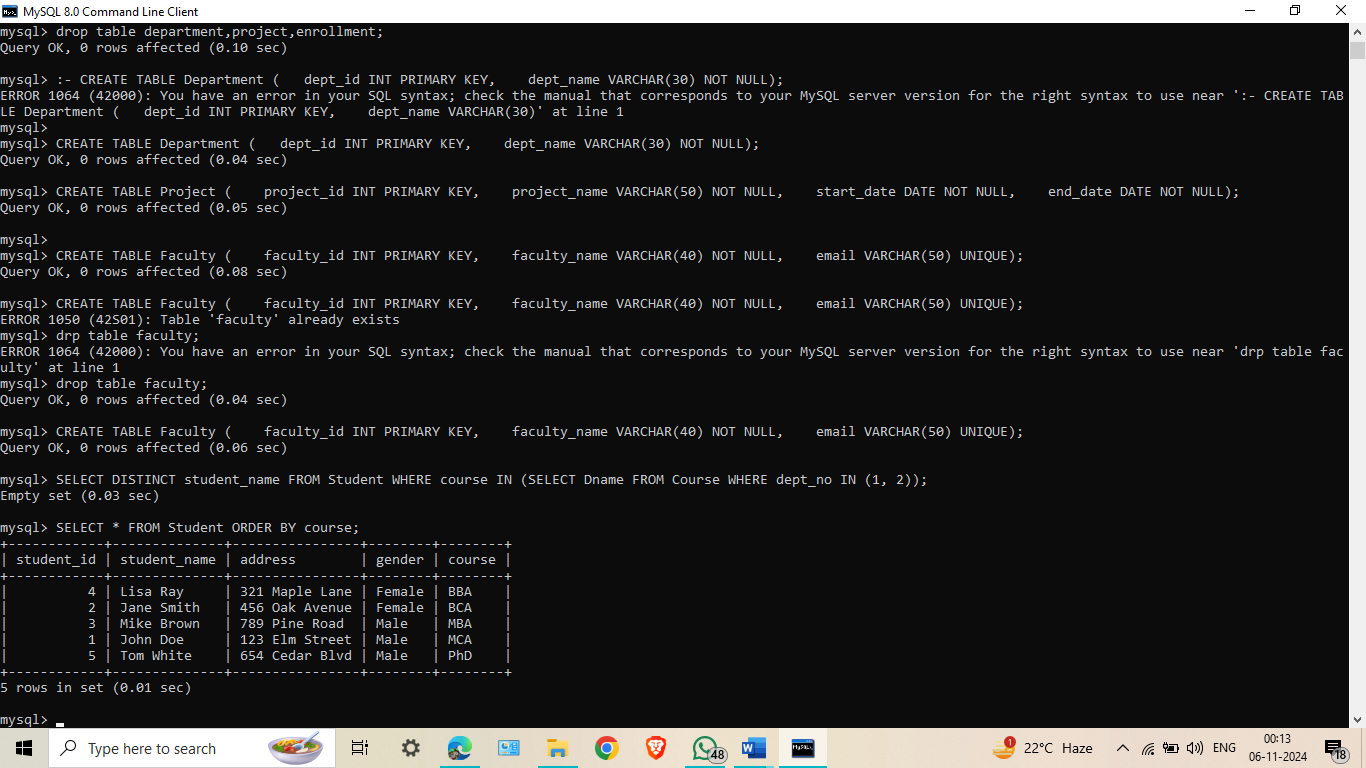
## SELECT DISTINCT student\_name FROM Student WHERE course IN (SELECT Dname FROM Course WHERE dept\_no IN (1, 2));



## Experiment No: 23

## Experiment 23:- Display the list of students ordered by course

## SELECT \* FROM Student ORDER BY course;



## Experiment No: 24

## Experiment 24:- Display the alphabetically sorted list of students

## SELECT \* FROM Student ORDER BY student\_name ASC;

