

PROJECT NAME: COURSE REGISTRATION MANAGEMENT SYSTEM

# GROUP MEMBERS:

|  |  |  |
| --- | --- | --- |
| **NAME** | **ID** | **CONTRIBUTION** |
| .MD AKRAM HOSSAIN | 23-50745-1 | 20% |
| ARPITA ISLAM | 23-50712-1 | 20% |
| MD NAHIDUR RAHMAN | 23-50715-1 | 20% |
| MEHJABIN MOSTAFA | 23-50707-1 | 20% |
| . ESTIAK AHNED AL - NAHIAN | 23-50698-1 | 20% |

COURSE: INTRODUCTION TO DATABASE SECTION: j

# Table contents:

**Introduction ProjectScenario ER-Diagram Normalization SchemaDesign UserCreation TableCreation TableSequences TableInsertion QueryWriting RelationalAlgebra Conclusion**

# INTRODUCTION:

This project involves designing and implementing a Course Registration Management System for a university. Here are some key components and functionalities of the system

1.Student Enrollment • Students can enroll in various courses offered by the university each semester. • Enrollment records include student ID, name, program, and the courses they are enrolled in.

2.Course Details • Each course has specific details such as a course number, title, credits, , and prerequisites. • Courses may have multiple sections offered at different times .

3.Instructors • Instructors teach courses and are identified by their instructor ID and specific department. • Each section of a course is taught by a different instructor.

5.User Interfaces • The system should provide user-friendly interfaces for students to browse available courses, enroll in courses, and view their enrollment status and grades. • Instructors should have interfaces to manage their course sections, submit grades, and view student enrollment.

6.Administration and Management • Administrative users should have access to tools for managing course offerings, instructor assignments, and student records. • The system should support administrative tasks such as adding new courses, assigning instructors to courses, and generating reports on student enrollment and performance.

7.Security and Access Control • The system should implement security measures to protect student and instructor data. • Access control mechanisms should ensure that only authorized users can perform certain actions within the system.

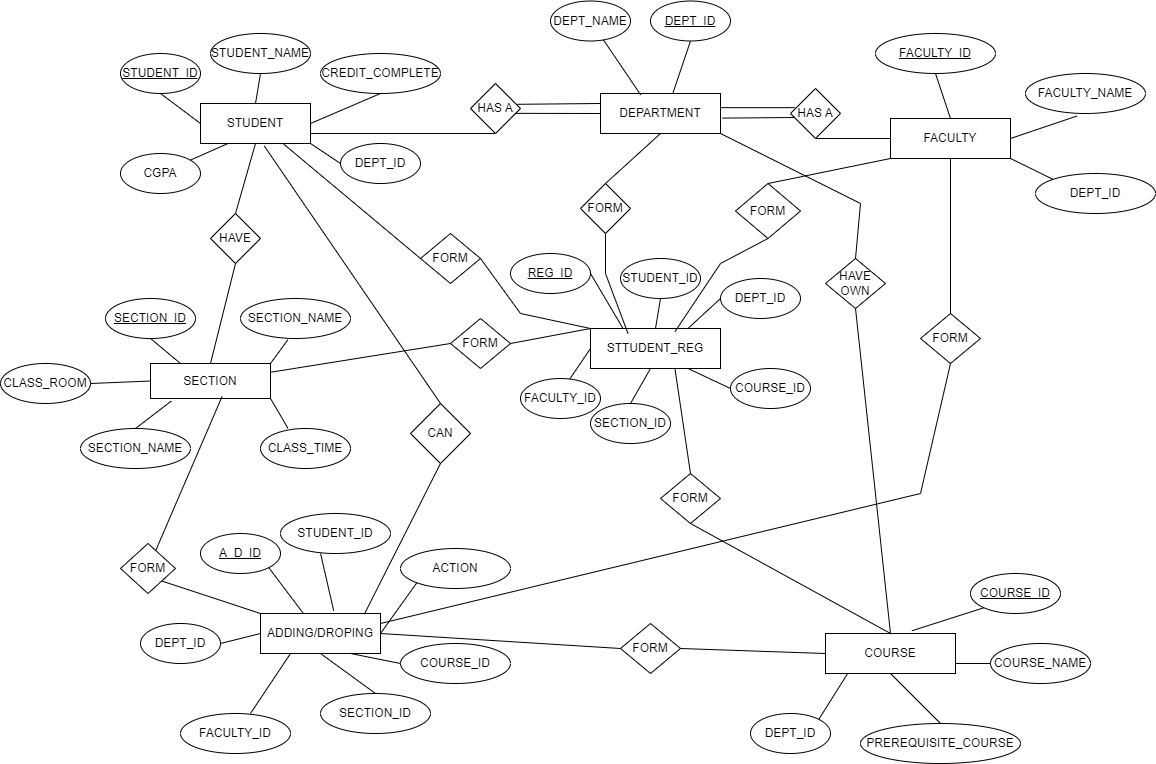
Overall, the Course Registration Management System aims to streamline the process of course enrollment, facilitate effective teaching and learning, and provide administrators with tools for managing academic programs

and student records.

# Project Scenario:

In a University Course Registration Management System, students are enrolled each semester for various courses. Each course has specific details such as a title, credits, syllabus, and pre- requisites and course is identified by course number. Onecourses can be taken by many students. Additionally, courses are offered in different sections, each with its own instructor, course number (foreign key), timing, and classroomnumber. Sections are identified by section’s number. Students enroll in courses, and their enrollment records include their name, and program. Also, a student is identified by a student id. Instructors, identified by their instructor ID which is an unique property and specific departments who teach courses. Different sections are taken by different instructors. Farther, the enrollment of students in courses and grades awarded to students in each course are enrolled for must be appropriately modelled.

# ER-Diagram:



**Normalization: Have own**

UNF

Have own(dept\_id,dept\_name,course\_id,course\_name,prerequisite\_course)

1. NF

There is no multivalued attribute .relation already in 1 NF

1. dept\_id, dept\_name,course\_id, course\_name,prerequisite\_course

1. NF
2. dept\_id, dept\_name
3. course\_id, course\_name,prerequisite\_course
4. NF

There is no transitive dependency , relation already in 3 NF

1. dept\_id, dept\_name

2 . course\_id, course\_name,prerequisite\_course

Table creation

1 dept\_id, dept\_name

2 course\_id, course\_name,prerequisite\_course,dept\_id

# Has a

UNF

Has a(faculty\_id,faculty\_name,dept\_id,dept\_name)

1 NF

There is no multivalued attribute .relation already in 1 NF

1.faculty\_id,faculty\_name,dept\_id,dept\_name

2 NF

1. faculty\_id,faculty\_name

2. dept\_id,dept\_name

3 NF

There is no transitive dependency , relation already in 3 NF

1. faculty\_id,faculty\_name

2. dept\_id,dept\_name

Table creation

1 dept\_id,dept\_name

2 faculty\_id,faculty\_name,dept\_id

**Has a**

UNF

Has a (dept\_id, dept\_name,student\_id,student\_name\_cradit\_complete,cgpa,)

1 NF

There is no multivalued attribute .relation already in 1 NF

1. dept\_id, dept\_name,student\_id,student\_name\_cradit\_complete,cgpa

2 NF

1. dept\_id, dept\_name

2. ,student\_id,student\_name\_cradit\_complete,cgpa

3 NF

There is no transitive dependency , relation already in 3 NF

1 .dept\_id, dept\_name

2 .,student\_id,student\_name\_cradit\_complete,cgpa

Table creation

1.dept\_id, dept\_name

2 .student\_id,student\_name\_cradit\_complete,cgpa,dept\_id

**FORM**

UNF

FORM (student\_id,student\_name\_cradit\_complete,cgpa,dept\_id,dept\_id, dept\_name, faculty\_id,faculty\_name,dept\_id , course\_id, course\_name,prerequisite\_course,dept\_id,section\_id,section\_name,class\_time,class\_rom,a\_d\_id,student\_id,action,course\_id,section\_idfaculty\_id,dept\_id)

1 NF

There is no multivalued attribute .relation already in 1 NF

1. student\_id,student\_name\_cradit\_complete,cgpa,dept\_id,dept\_id, dept\_name, faculty\_id,faculty\_name,dept\_id , course\_id, course\_name,prerequisite\_course,dept\_id,section\_id,section\_name,class\_time,class\_rom,a\_d\_id,student\_id,action,course\_id,section\_idfaculty\_id,dept\_id)

2NF

1. dept\_id, dept\_name

2 . student\_id,student\_name\_cradit\_complete,cgpa,dept\_id,faculty\_id,faculty\_name,dept\_id , course\_id,course\_name,prerequisite\_course,dept\_id,section\_id,section\_name,class\_time,class\_rom,a\_d\_id,student\_id,action,course\_id,section\_id,faculty\_id,dept\_id)

3 NF

1 section\_id,section\_name,class\_time,class\_rom

2.,student\_id,action,course\_id,section\_id,faculty\_id,dept\_id

3.student\_id,.dept\_id,course\_id,section\_id,faculty\_id,

Table creation

1. Section\_id,section\_name,class\_time,class\_rom

2. A\_d\_id,student\_id,action,course\_id,section\_id,faculty\_id,dept\_id

3. Reg\_id,student\_id,.dept\_id,course\_id,section\_id,faculty\_id,

4.course\_name,prerequisite\_course,dept\_id,section\_id,section\_name,class\_time,class

**Can**

UNF

can(A\_d\_id,student\_id,action,course\_id,section\_id,faculty\_id,dept\_id, student\_id,student\_name\_cradit\_complete,cgpa,dept\_id

)

1 NF

There is no multivalued attribute .relation already in 1 NF

1. A\_d\_id,student\_id,action,course\_id,section\_id,faculty\_id,dept\_id,dept\_name,, student\_id,student\_name\_cradit\_complete,cgpa

2NF

1. A\_d\_id,student\_id,action,course\_id,section\_id,faculty\_id,dept\_id,

2 . student\_id,student\_name,cradit\_complete,cgpa,dept\_id

3NF

1. student\_id,student\_name,cradit\_complete,cgpa

2 . dept\_id,dept\_name

3. Course\_id,section\_id,faculty\_id,dept\_id

Table creation

1. dept\_id, dept\_name, student\_id

2 . couse\_id , course\_name ,dept\_id

3.student\_id,student\_name,cradit\_complete,cgpa,dept\_i

# Temporary Tables:

1. dept\_id, dept\_name, student\_id

2 . couse\_id , course\_name ,dept\_id

3.student\_id,student\_name,cradit\_complete,cgpa,dept\_id.

4. Section\_id,section\_name,class\_time,class\_rom

5. A\_d\_id,student\_id,action,course\_id,section\_id,faculty\_id,dept\_id

6. Reg\_id,student\_id,.dept\_id,course\_id,section\_id,faculty\_id,

7. course\_name,prerequisite\_course,dept\_id,section\_id,section\_name,class\_time,class

8 dept\_id,dept\_name

9 dept\_id, dept\_name

# Final table:

1. dept\_id, dept\_name

2. course\_id, course\_name,prerequisite\_course,dept\_id

3. faculty\_id,faculty\_name,dept\_id

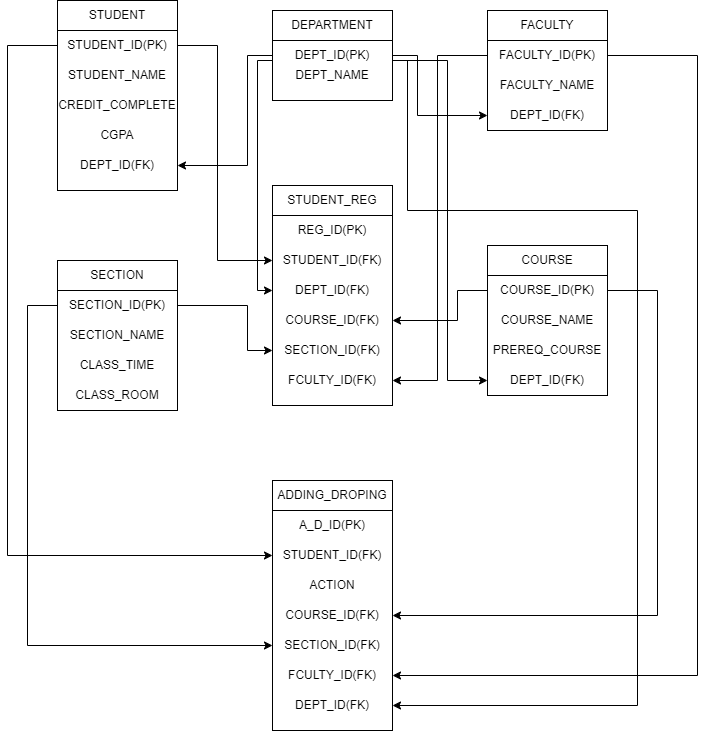
4. student\_id,student\_name\_cradit\_complete,cgpa,dept\_id

5.. Section\_id,section\_name,class\_time,class\_rom

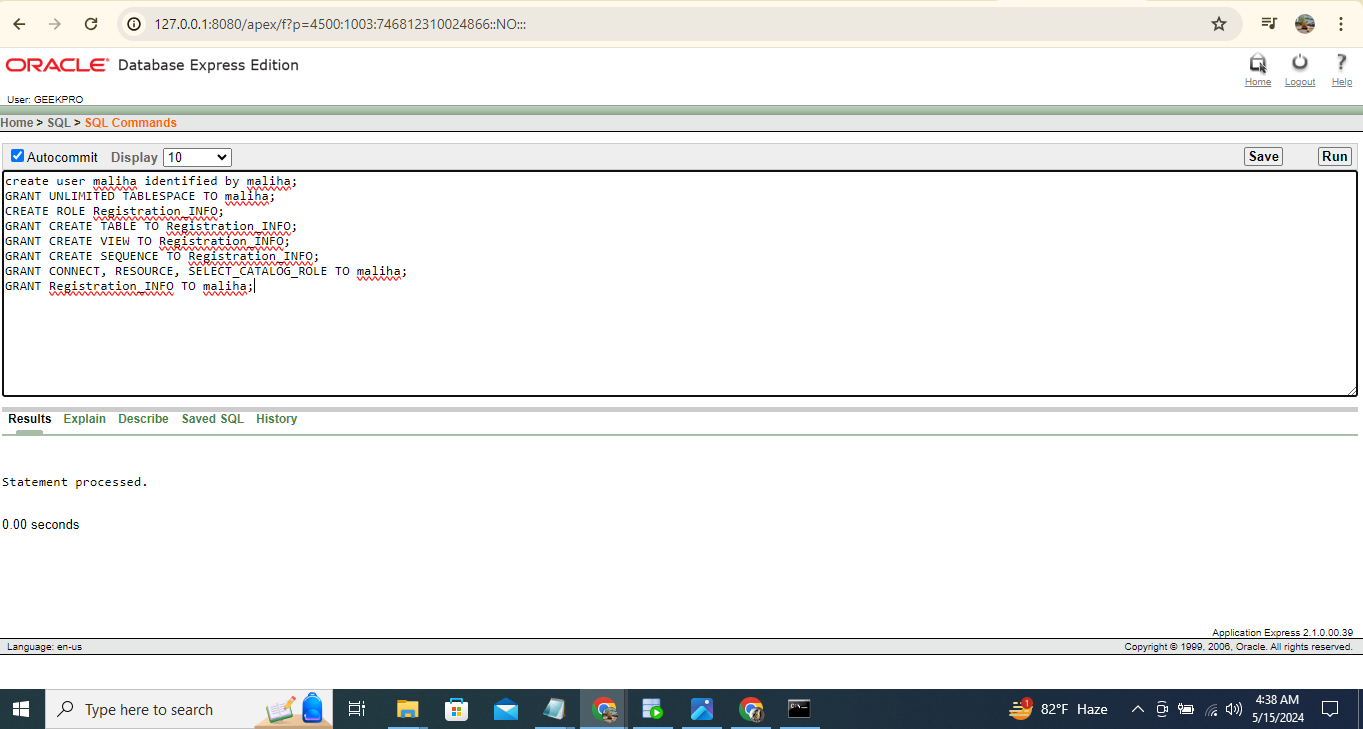
6.. A\_d\_id,student\_id,action,course\_id,section\_id,faculty\_id,dept\_id

7. Reg\_id,student\_id,.dept\_id,course\_id,section\_id,faculty\_id,

# Schema Design:



**User Creation:**



User create

create user maliha identified by maliha;

GRANT UNLIMITED TABLESPACE TO maliha;

CREATE ROLE Registration\_INFO;

GRANT CREATE TABLE TO Registration\_INFO;

GRANT CREATE VIEW TO Registration\_INFO;

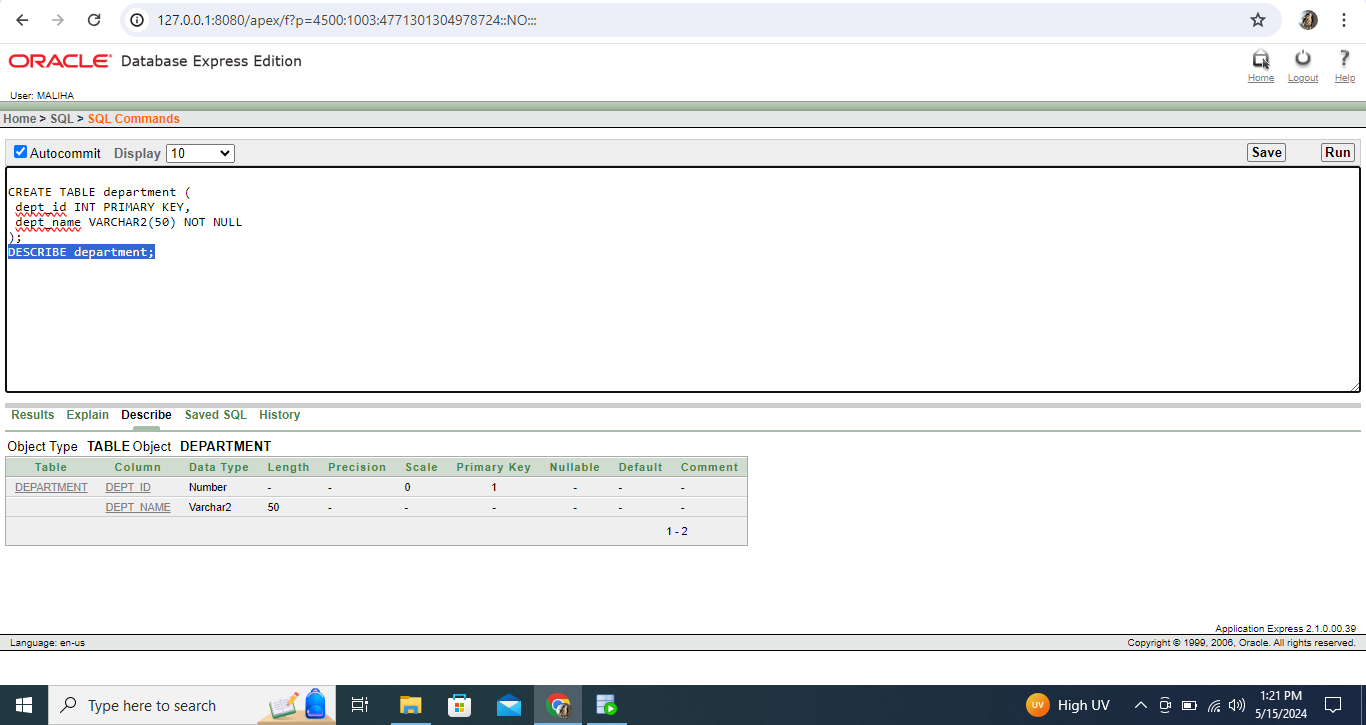
GRANT CREATE SEQUENCE TO Registration\_INFO;

GRANT CONNECT, RESOURCE, SELECT\_CATALOG\_ROLE TO maliha;

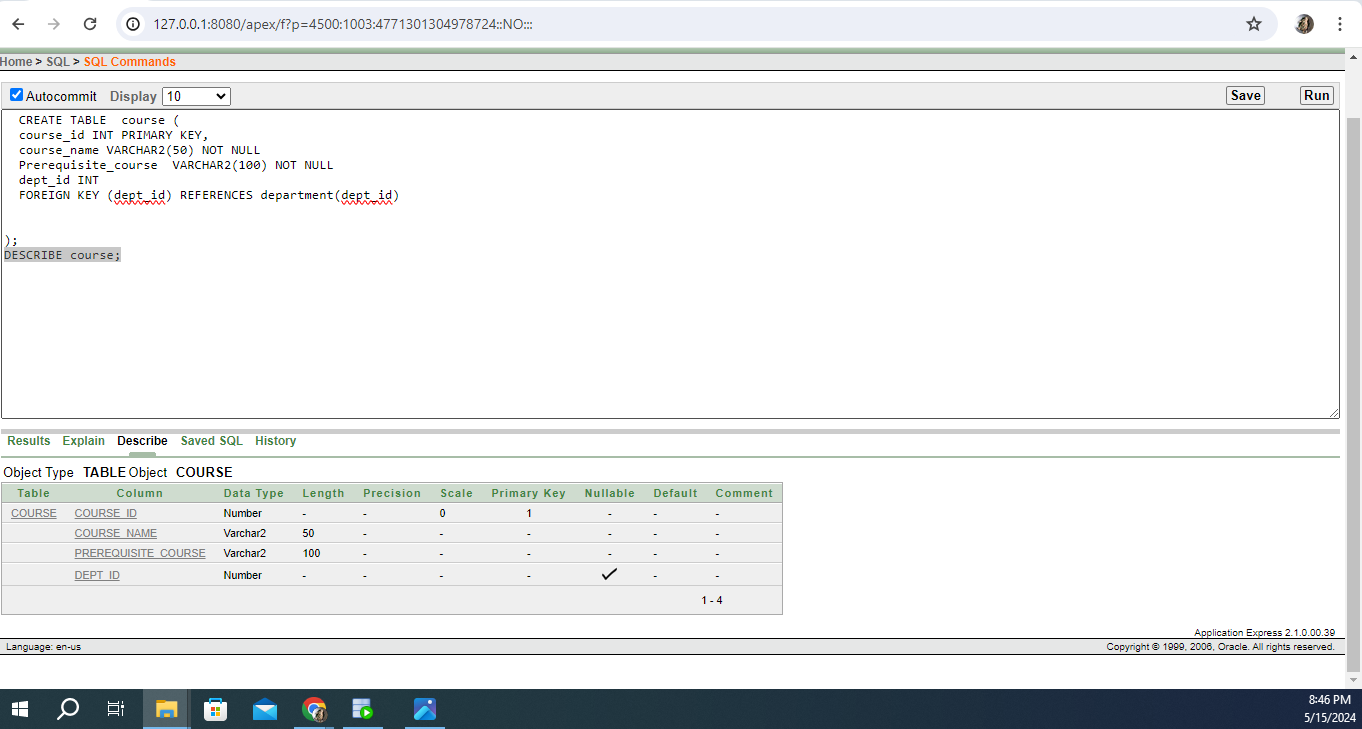
GRANT Registration\_INFO TO maliha;

**Table Creation (Screenshot):**

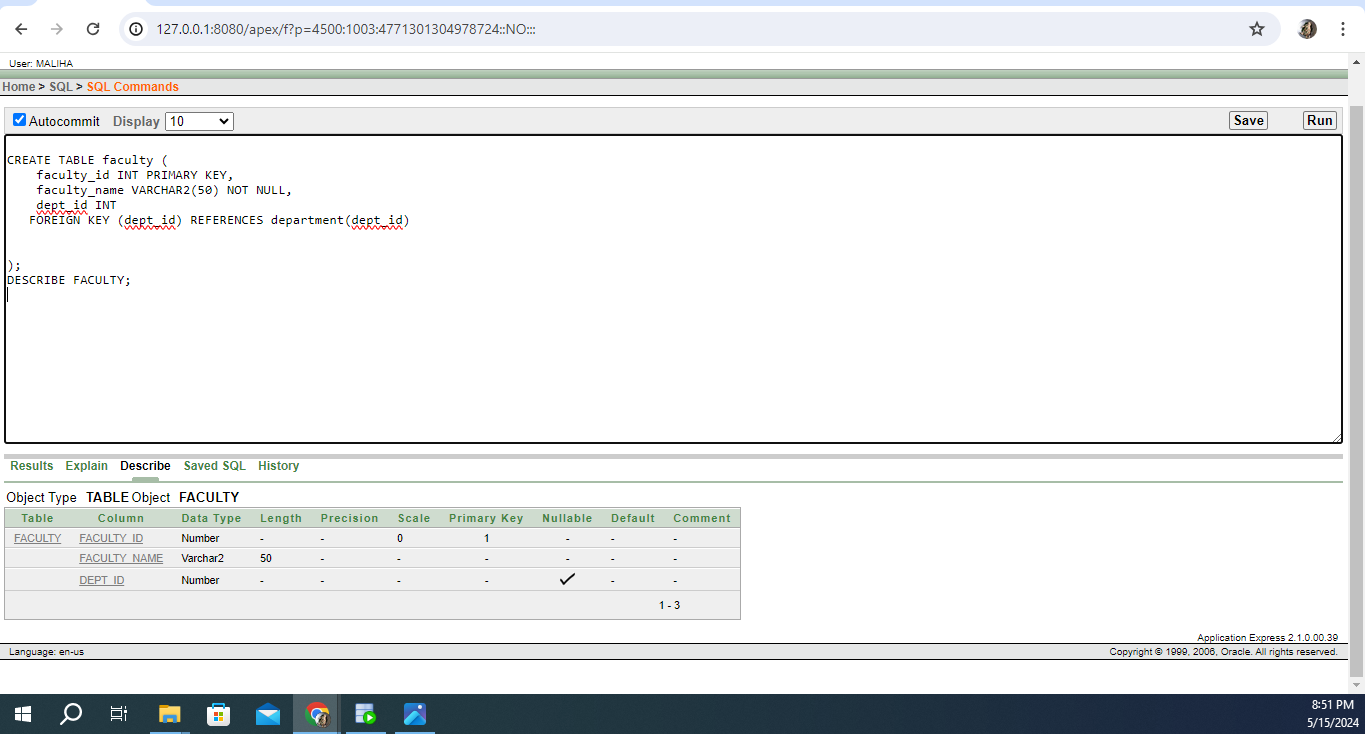
**1.department**



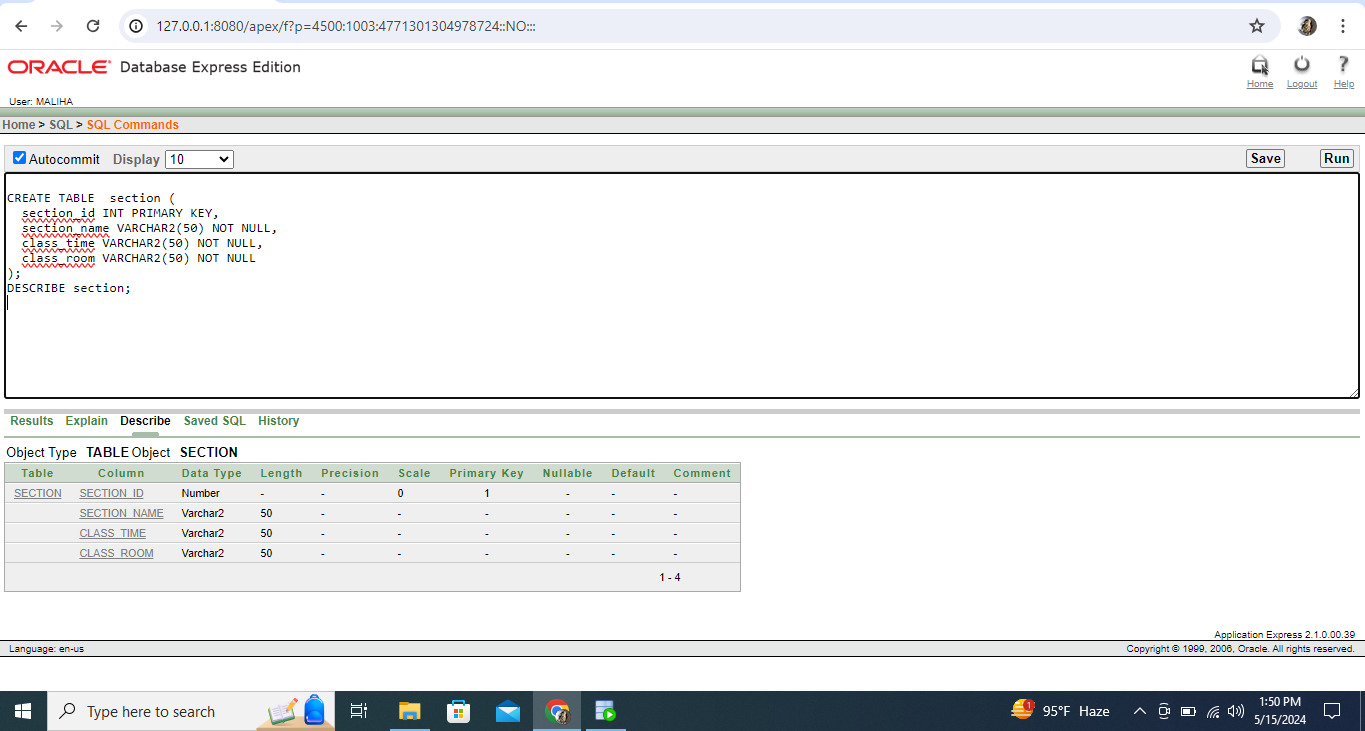
**2.course**



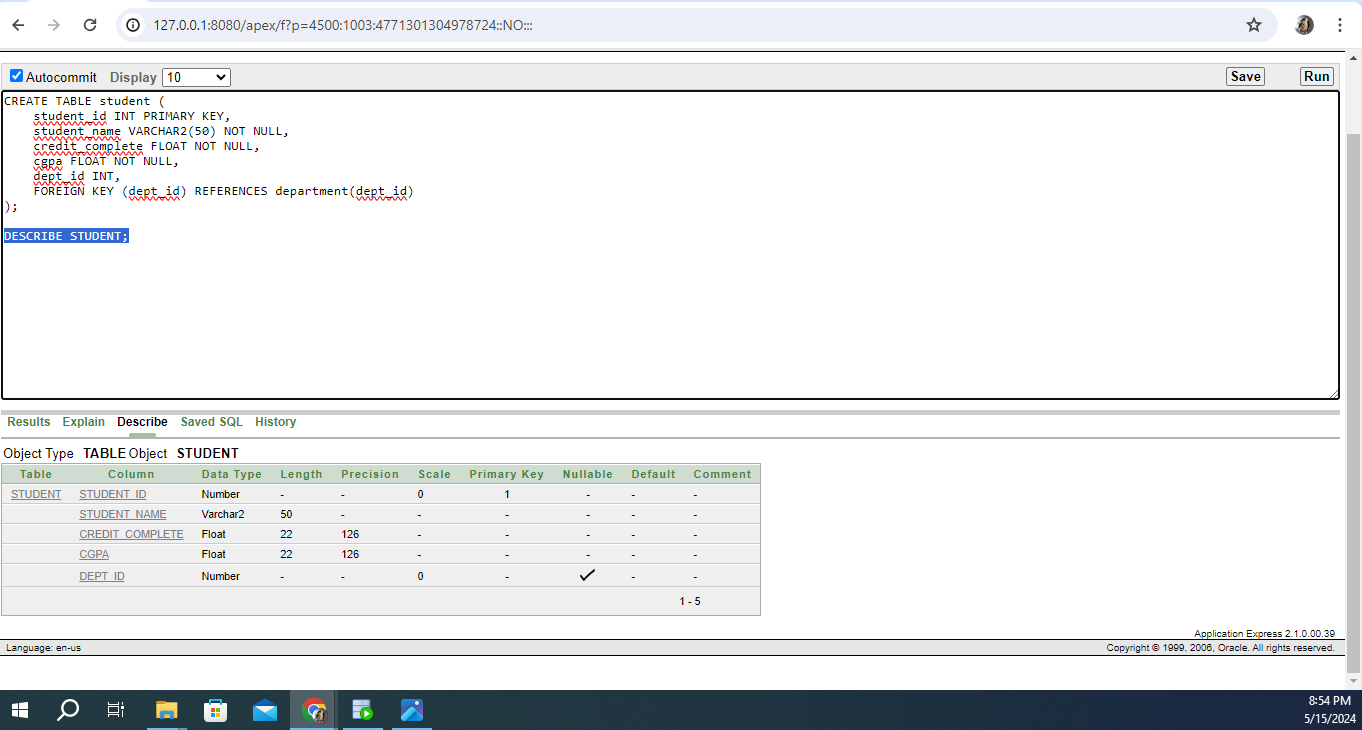
## 3.faculty



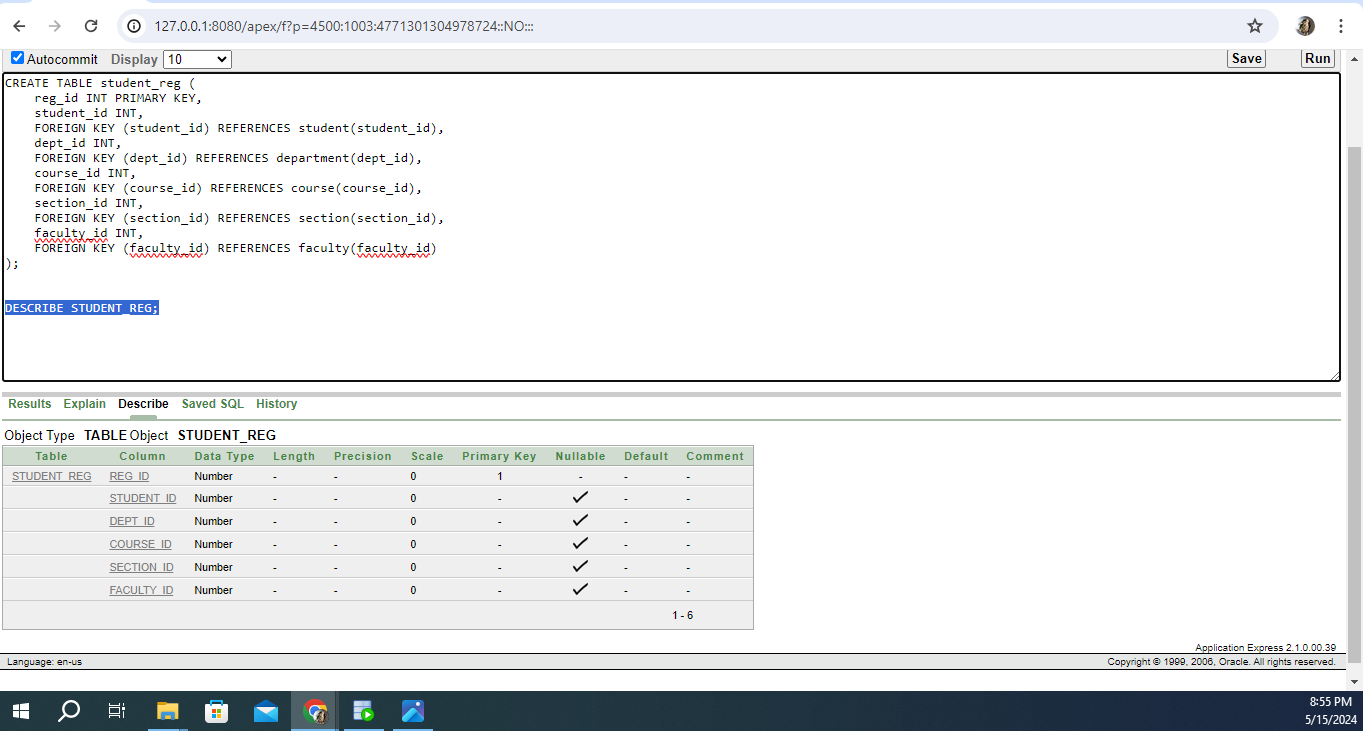
**4.section**



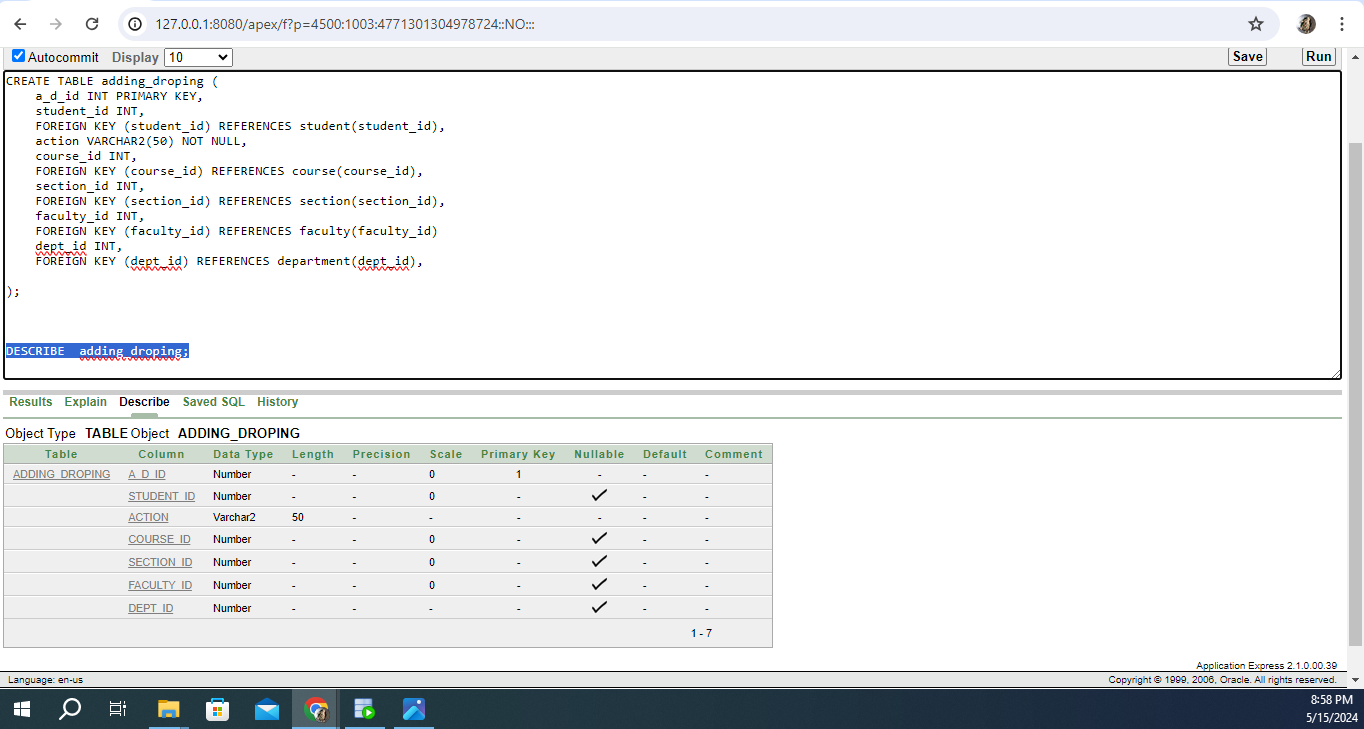
## 5.student



**6.studentreg**



**7.adding/droping**



## Table Creation (Query):

TABLE

CREATE TABLE department (

dept\_id INT PRIMARY KEY,

dept\_name VARCHAR2(50) NOT NULL

);

DESCRIBE student;

CREATE TABLE faculty (

faculty\_id INT PRIMARY KEY,

faculty\_name VARCHAR2(50) NOT NULL,

dept\_id INT

FOREIGN KEY (dept\_id) REFERENCES department(dept\_id)

);

CREATE TABLE course (

course\_id INT PRIMARY KEY,

course\_name VARCHAR2(50) NOT NULL

Prerequisite\_course VARCHAR2(100) NOT NULL

dept\_id INT

FOREIGN KEY (dept\_id) REFERENCES department(dept\_id)

);

DESCRIBE course;

CREATE TABLE section (

section\_id INT PRIMARY KEY,

section\_name VARCHAR2(50) NOT NULL,

class\_time VARCHAR2(50) NOT NULL,

class\_room VARCHAR2(50) NOT NULL

);

DESCRIBE section;

CREATE TABLE student (

student\_id INT PRIMARY KEY,

student\_name VARCHAR2(50) NOT NULL,

credit\_complete FLOAT NOT NULL,

cgpa FLOAT NOT NULL,

dept\_id INT,

FOREIGN KEY (dept\_id) REFERENCES department(dept\_id)

);

DESCRIBE STUDENT;

CREATE TABLE student\_reg (

reg\_id INT PRIMARY KEY,

student\_id INT,

FOREIGN KEY (student\_id) REFERENCES student(student\_id),

dept\_id INT,

FOREIGN KEY (dept\_id) REFERENCES department(dept\_id),

course\_id INT,

FOREIGN KEY (course\_id) REFERENCES course(course\_id),

section\_id INT,

FOREIGN KEY (section\_id) REFERENCES section(section\_id),

faculty\_id INT,

FOREIGN KEY (faculty\_id) REFERENCES faculty(faculty\_id)

);

CREATE TABLE adding\_droping (

a\_d\_id INT PRIMARY KEY,

student\_id INT,

FOREIGN KEY (student\_id) REFERENCES student(student\_id),

action VARCHAR2(50) NOT NULL,

course\_id INT,

FOREIGN KEY (course\_id) REFERENCES course(course\_id),

section\_id INT,

FOREIGN KEY (section\_id) REFERENCES section(section\_id),

faculty\_id INT,

FOREIGN KEY (faculty\_id) REFERENCES faculty(faculty\_id)

dept\_id INT,

FOREIGN KEY (dept\_id) REFERENCES department(dept\_id),

);

DESCRIBE adding\_droping;

## Table Sequences:

CREATE SEQUENCE dept\_id

INCREMENT BY 1

START WITH 1

MAXVALUE 1000

NOCACHE

NOCYCLE;

CREATE SEQUENCE course\_id

INCREMENT BY 1

START WITH 1

MAXVALUE 1000

NOCACHE

NOCYCLE;

CREATE SEQUENCE section\_id

INCREMENT BY 1

START WITH 1

MAXVALUE 1000

NOCACHE

NOCYCLE;

CREATE SEQUENCE student\_id

INCREMENT BY 1

START WITH 1

MAXVALUE 1000

NOCACHE

NOCYCLE;

CREATE SEQUENCE faculty\_id

INCREMENT BY 1

START WITH 1

MAXVALUE 1000

NOCACHE

NOCYCLE;

CREATE SEQUENCE reg\_id

INCREMENT BY 1

START WITH 1

MAXVALUE 1000

NOCACHE

NOCYCLE;

CREATE SEQUENCE a\_d\_id

INCREMENT BY 1

START WITH 1

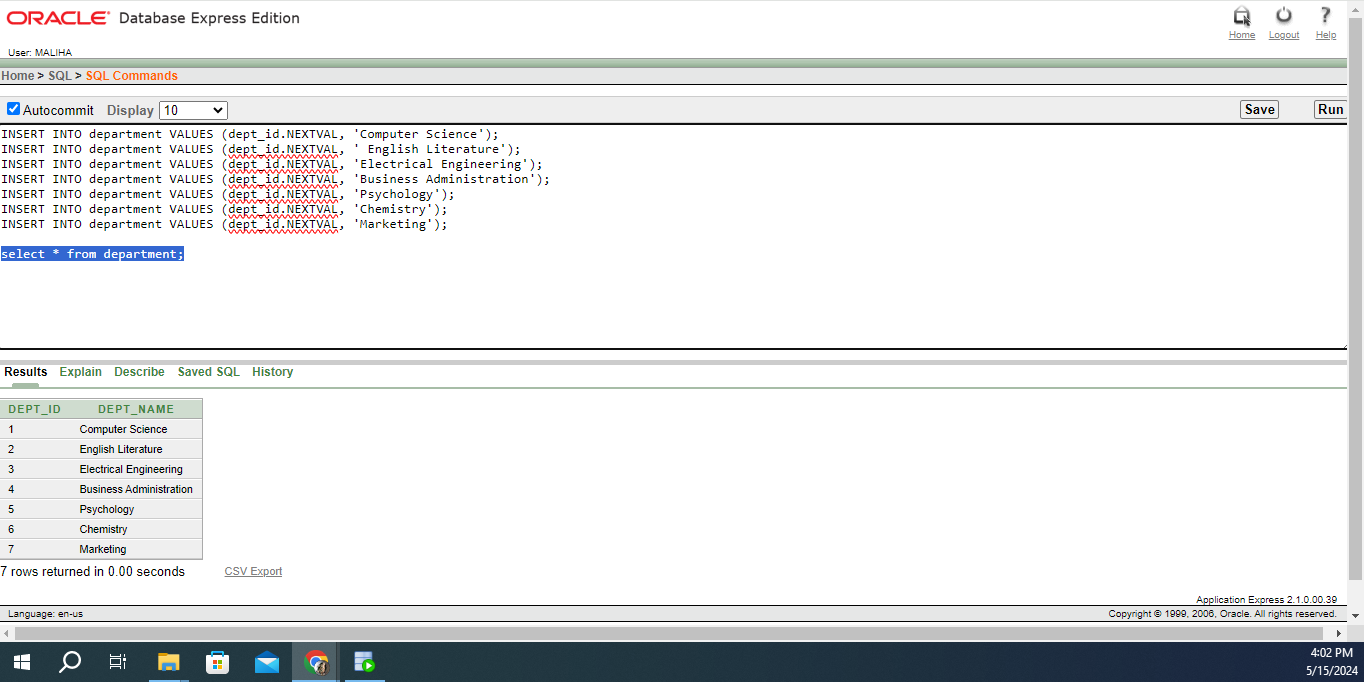
MAXVALUE 1000

NOCACHE

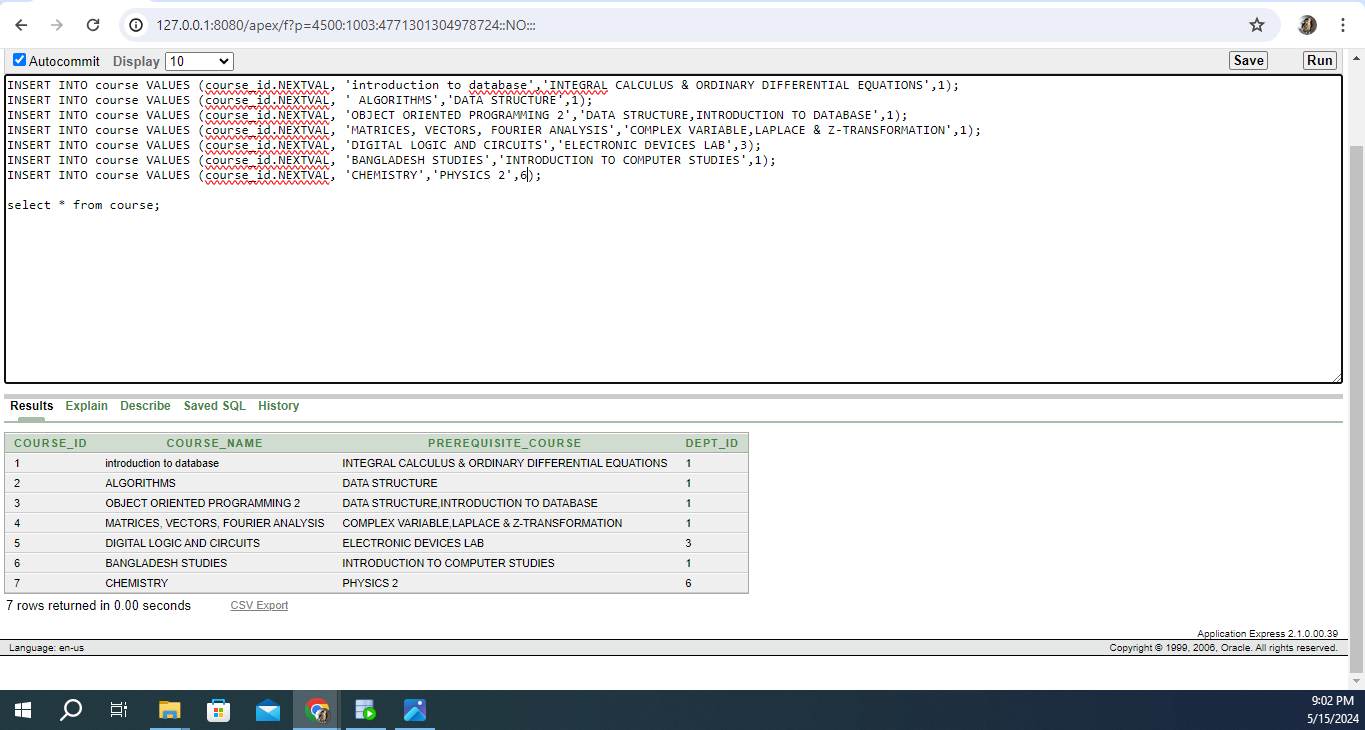
NOCYCLE;

# Table Insertion (Screenshot):

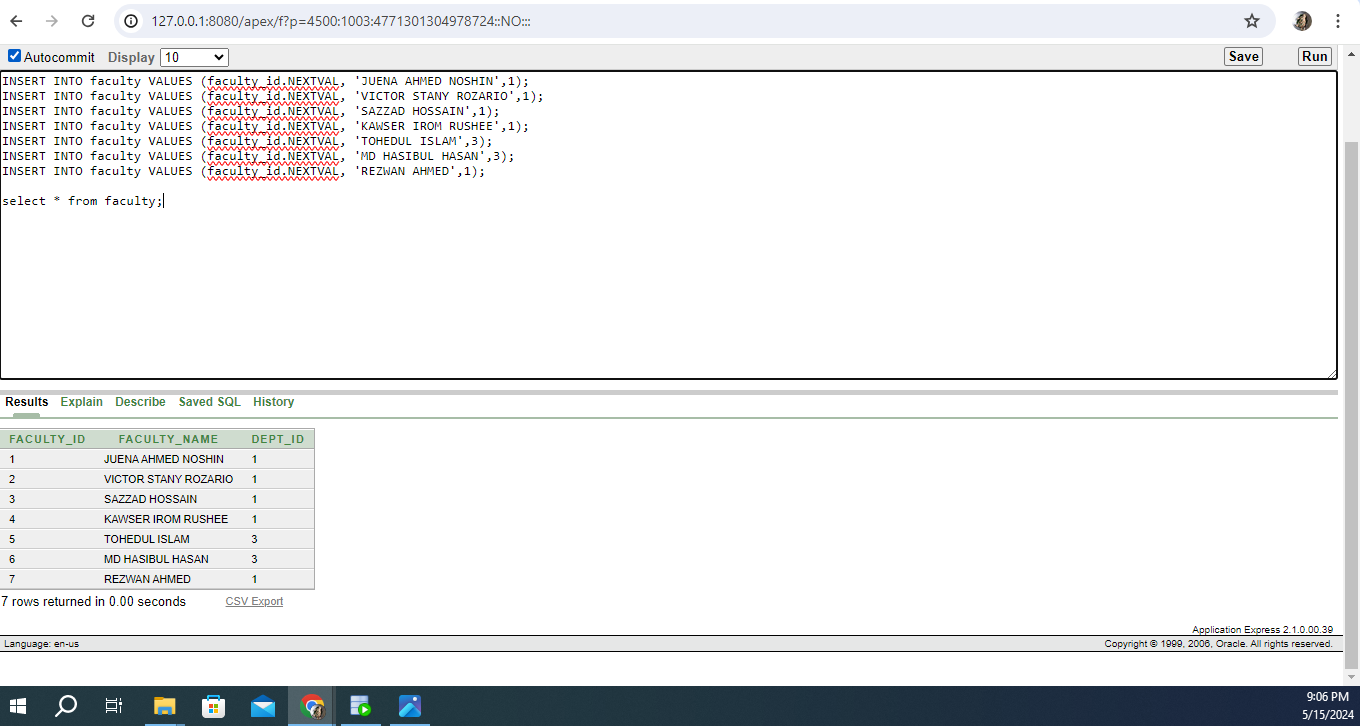
**1.department**



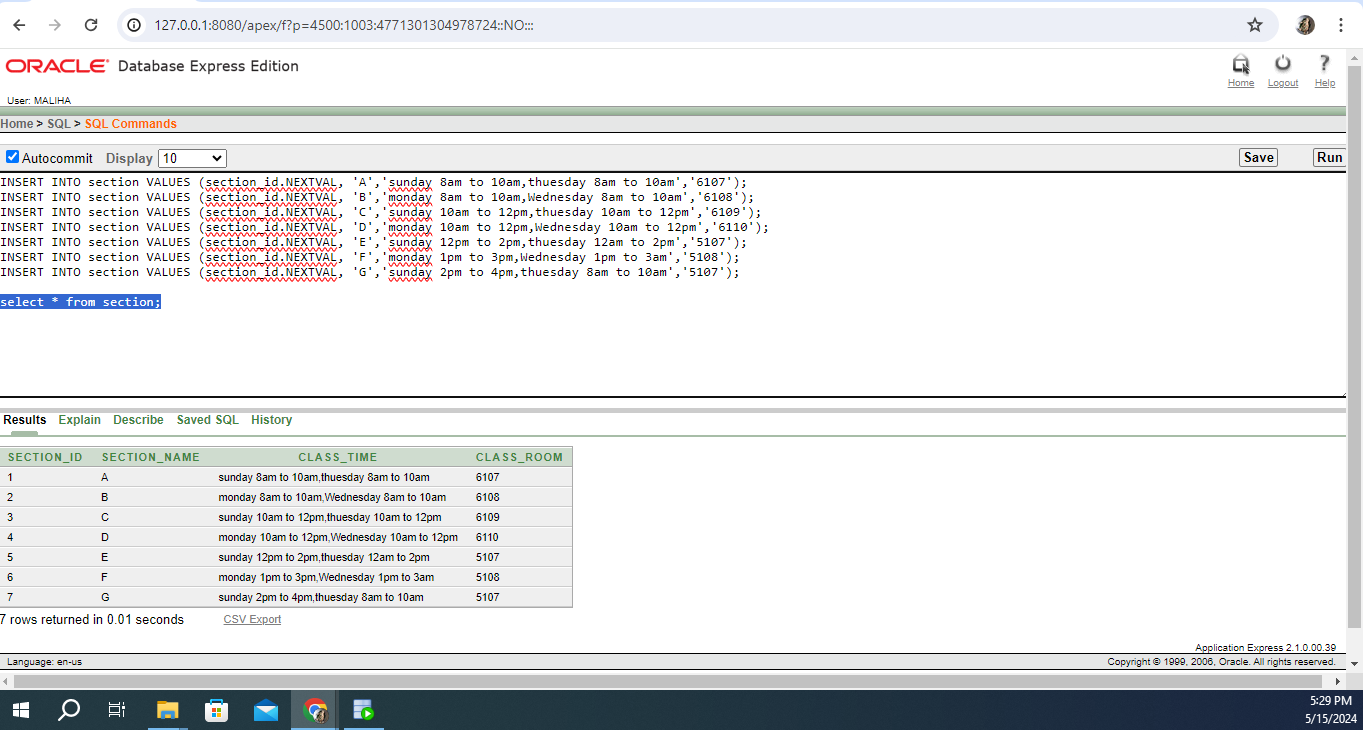
# 2.course



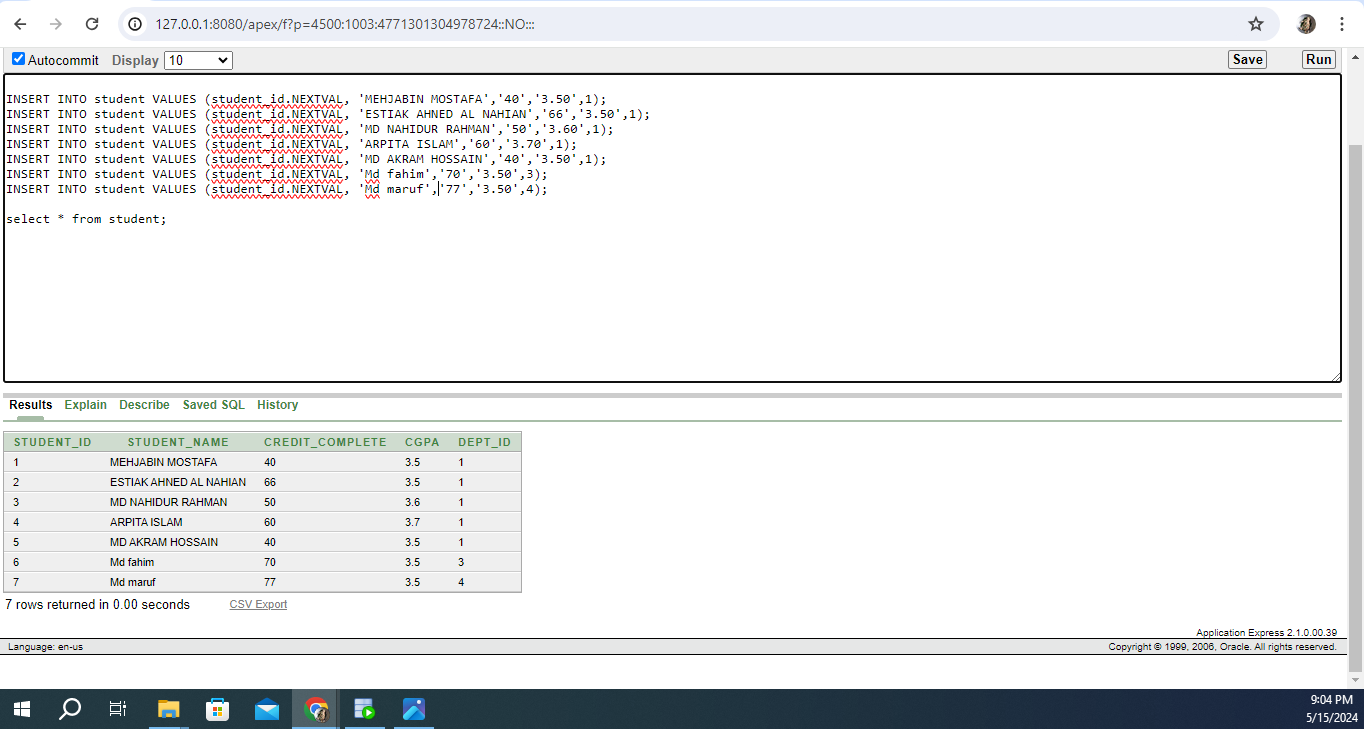
**3.faculty**



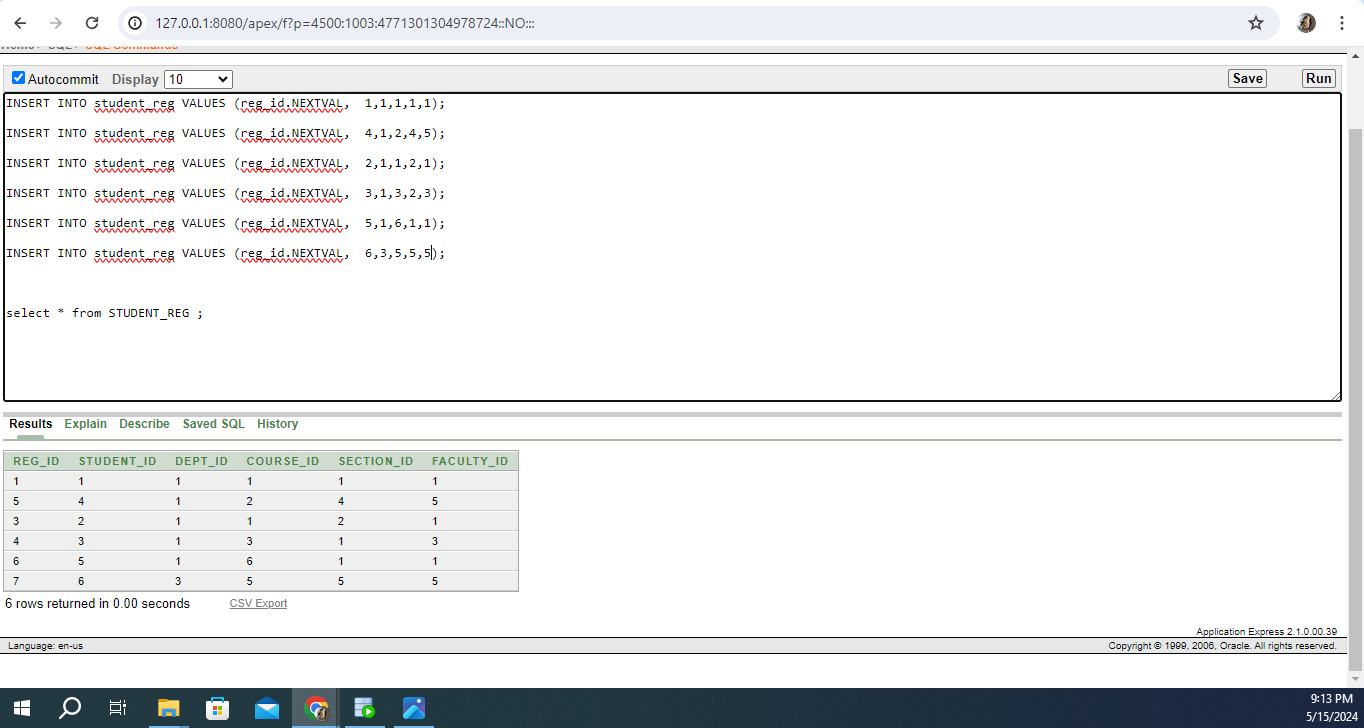
# 4.section



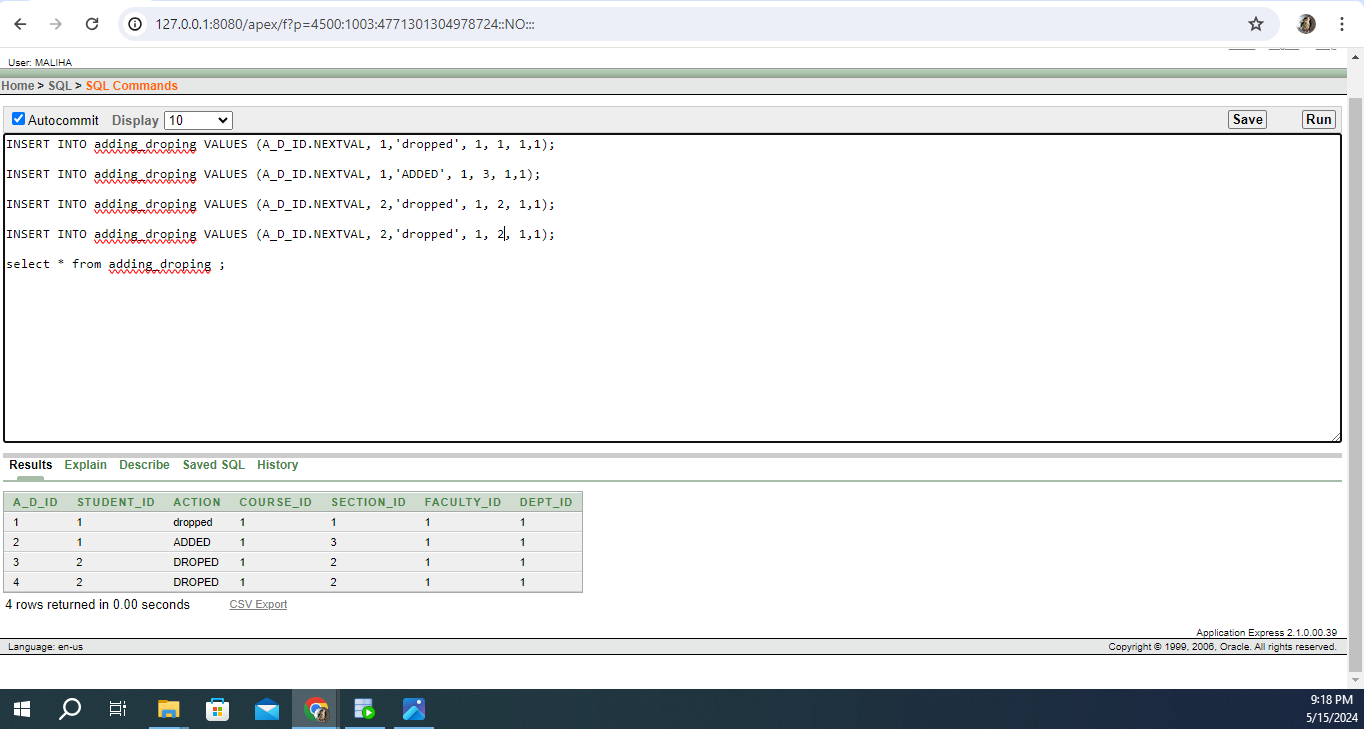
**5.student**



# 6.student\_reg



# 8.adding/dropding



**Table Insertion:**

1.depertment

INSERT INTO department VALUES (dept\_id.NEXTVAL, 'Computer Science');

INSERT INTO department VALUES (dept\_id.NEXTVAL, ' English Literature');

INSERT INTO department VALUES (dept\_id.NEXTVAL, 'Electrical Engineering');

INSERT INTO department VALUES (dept\_id.NEXTVAL, 'Business Administration');

INSERT INTO department VALUES (dept\_id.NEXTVAL, 'Psychology');

INSERT INTO department VALUES (dept\_id.NEXTVAL, 'Chemistry');

INSERT INTO department VALUES (dept\_id.NEXTVAL, 'Marketing');

select \* from department;

2.course

INSERT INTO course VALUES (course\_id.NEXTVAL, 'introduction to database','INTEGRAL CALCULUS & ORDINARY DIFFERENTIAL EQUATIONS',1);

INSERT INTO course VALUES (course\_id.NEXTVAL, ' ALGORITHMS','DATA STRUCTURE',1);

INSERT INTO course VALUES (course\_id.NEXTVAL, 'OBJECT ORIENTED PROGRAMMING 2','DATA STRUCTURE,INTRODUCTION TO DATABASE',1);

INSERT INTO course VALUES (course\_id.NEXTVAL, 'MATRICES, VECTORS, FOURIER ANALYSIS','COMPLEX VARIABLE,LAPLACE & Z-TRANSFORMATION',1);

INSERT INTO course VALUES (course\_id.NEXTVAL, 'DIGITAL LOGIC AND CIRCUITS','ELECTRONIC DEVICES LAB',1);

INSERT INTO course VALUES (course\_id.NEXTVAL, 'BANGLADESH STUDIES','INTRODUCTION TO COMPUTER STUDIES',1);

INSERT INTO course VALUES (course\_id.NEXTVAL, 'CHEMISTRY','PHYSICS 2',1);

select \* from course;

3.section

INSERT INTO section VALUES (section\_id.NEXTVAL, 'A','sunday 8am to 10am,thuesday 8am to 10am','6107');

INSERT INTO section VALUES (section\_id.NEXTVAL, 'B','monday 8am to 10am,Wednesday 8am to 10am','6108');

INSERT INTO section VALUES (section\_id.NEXTVAL, 'C','sunday 10am to 12pm,thuesday 10am to 12pm','6109');

INSERT INTO section VALUES (section\_id.NEXTVAL, 'D','monday 10am to 12pm,Wednesday 10am to 12pm','6110');

INSERT INTO section VALUES (section\_id.NEXTVAL, 'E','sunday 12pm to 2pm,thuesday 12am to 2pm','5107');

INSERT INTO section VALUES (section\_id.NEXTVAL, 'F','monday 1pm to 3pm,Wednesday 1pm to 3am','5108');

INSERT INTO section VALUES (section\_id.NEXTVAL, 'G','sunday 2pm to 4pm,thuesday 8am to 10am','5107');

select \* from section;

4.student

INSERT INTO student VALUES (student\_id.NEXTVAL, 'MEHJABIN MOSTAFA','40','3.50',1);

INSERT INTO student VALUES (student\_id.NEXTVAL, 'ESTIAK AHNED AL NAHIAN','66','3.50',1);

INSERT INTO student VALUES (student\_id.NEXTVAL, 'MD NAHIDUR RAHMAN','50','3.60',1);

INSERT INTO student VALUES (student\_id.NEXTVAL, 'ARPITA ISLAM','60','3.70',1);

INSERT INTO student VALUES (student\_id.NEXTVAL, 'MD AKRAM HOSSAIN','40','3.50',1);

INSERT INTO student VALUES (student\_id.NEXTVAL, 'Md fahim','70','3.50',3);

INSERT INTO student VALUES (student\_id.NEXTVAL, 'Md maruf','77','3.50',4);

select \* from student;

5.faculty

INSERT INTO faculty VALUES (faculty\_id.NEXTVAL, 'JUENA AHMED NOSHIN',1);

INSERT INTO faculty VALUES (faculty\_id.NEXTVAL, 'VICTOR STANY ROZARIO',1);

INSERT INTO faculty VALUES (faculty\_id.NEXTVAL, 'SAZZAD HOSSAIN',1);

INSERT INTO faculty VALUES (faculty\_id.NEXTVAL, 'KAWSER IROM RUSHEE',1);

INSERT INTO faculty VALUES (faculty\_id.NEXTVAL, 'TOHEDUL ISLAM',3);

INSERT INTO faculty VALUES (faculty\_id.NEXTVAL, 'MD HASIBUL HASAN',3);

INSERT INTO faculty VALUES (faculty\_id.NEXTVAL, 'REZWAN AHMED',1);

select \* from faculty;

6.studen\_reg

INSERT INTO student\_reg VALUES (reg\_id.NEXTVAL, 1,1,1,1,1);

INSERT INTO student\_reg VALUES (reg\_id.NEXTVAL, 4,1,2,4,5);

INSERT INTO student\_reg VALUES (reg\_id.NEXTVAL, 2,1,1,2,1);

INSERT INTO student\_reg VALUES (reg\_id.NEXTVAL, 3,1,3,2,3);

INSERT INTO student\_reg VALUES (reg\_id.NEXTVAL, 5,1,6,1,1);

INSERT INTO student\_reg VALUES (reg\_id.NEXTVAL, 6,3,5,5,5);

select \* from STUDENT\_REG ;

7.adding/droping

INSERT INTO adding\_droping VALUES (A\_D\_ID.NEXTVAL, 1,'dropped', 1, 1, 1,1);

INSERT INTO adding\_droping VALUES (A\_D\_ID.NEXTVAL, 1,'ADDED', 1, 3, 1,1);

INSERT INTO adding\_droping VALUES (A\_D\_ID.NEXTVAL, 2,'dropped', 1, 2, 1,1);

INSERT INTO adding\_droping VALUES (A\_D\_ID.NEXTVAL, 2,'dropped', 1, 2, 1,1);

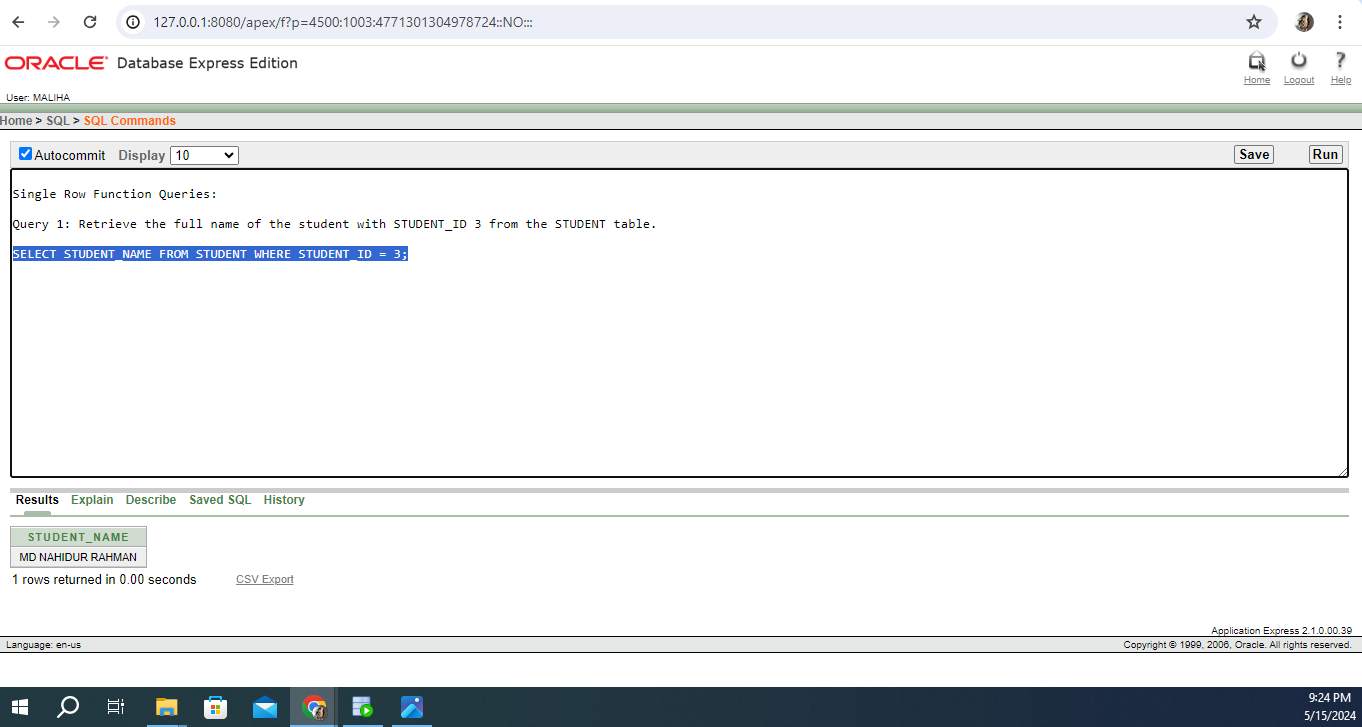
select \* from adding\_droping ;

# Query Writing:

SINGLE ROW FUNCTION

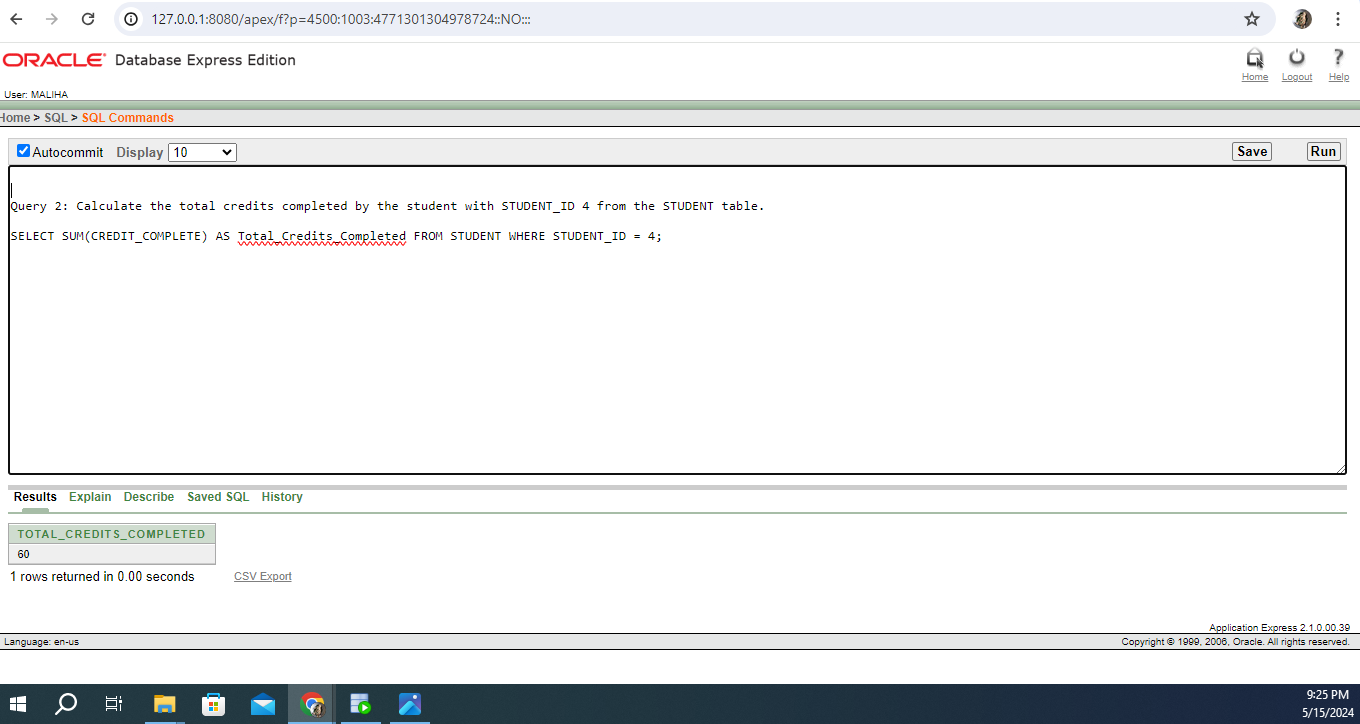
1. Retrieve the full name of the student with STUDENT\_ID 3 from the STUDENT table.:

SELECT STUDENT\_NAME FROM STUDENT WHERE STUDENT\_ID = 3;



1. Calculate the total credits completed by the student with STUDENT\_ID 4 from the STUDENT table.

SELECT SUM(CREDIT\_COMPLETE) AS Total\_Credits\_Completed FROM STUDENT WHERE STUDENT\_ID = 4;

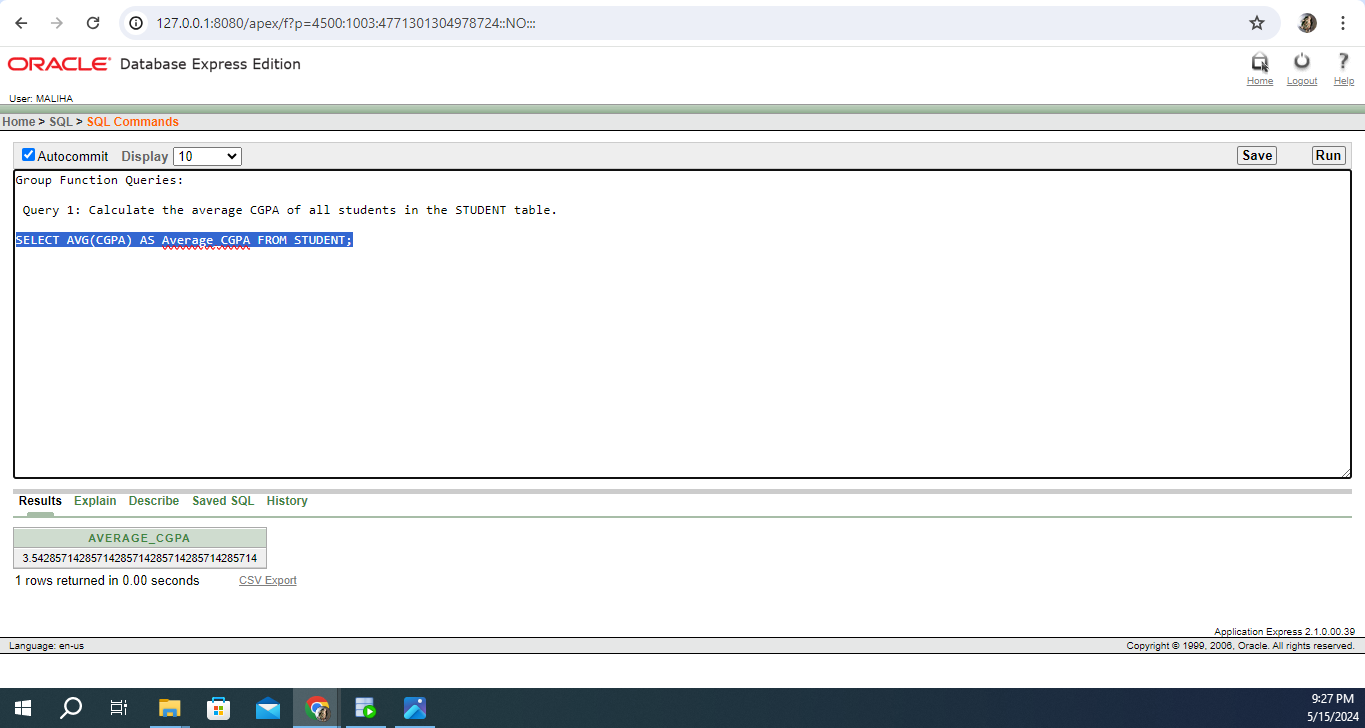


**2. GROUP FUNCTION**

1. Calculate the average CGPA of all students in the STUDENT table.

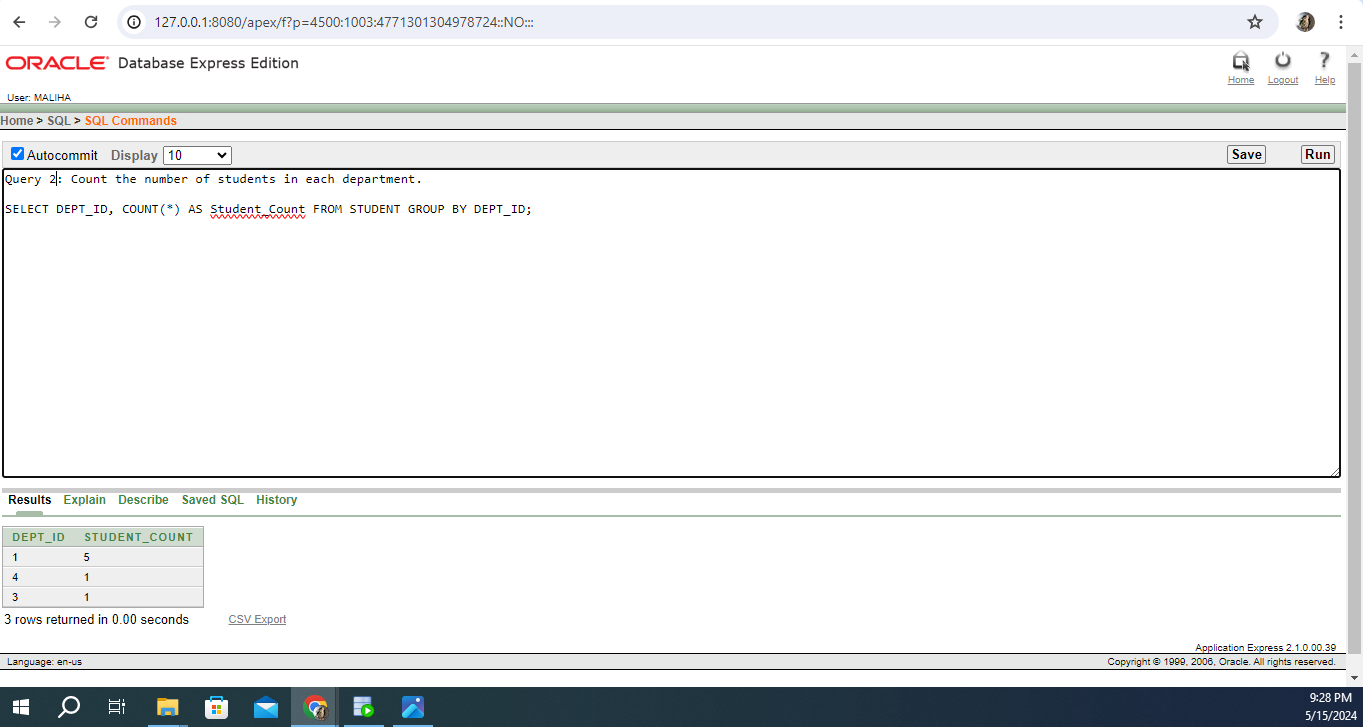
SELECT AVG(CGPA) AS Average\_CGPA FROM STUDENT;

1.



2.Count the number of students in each department.

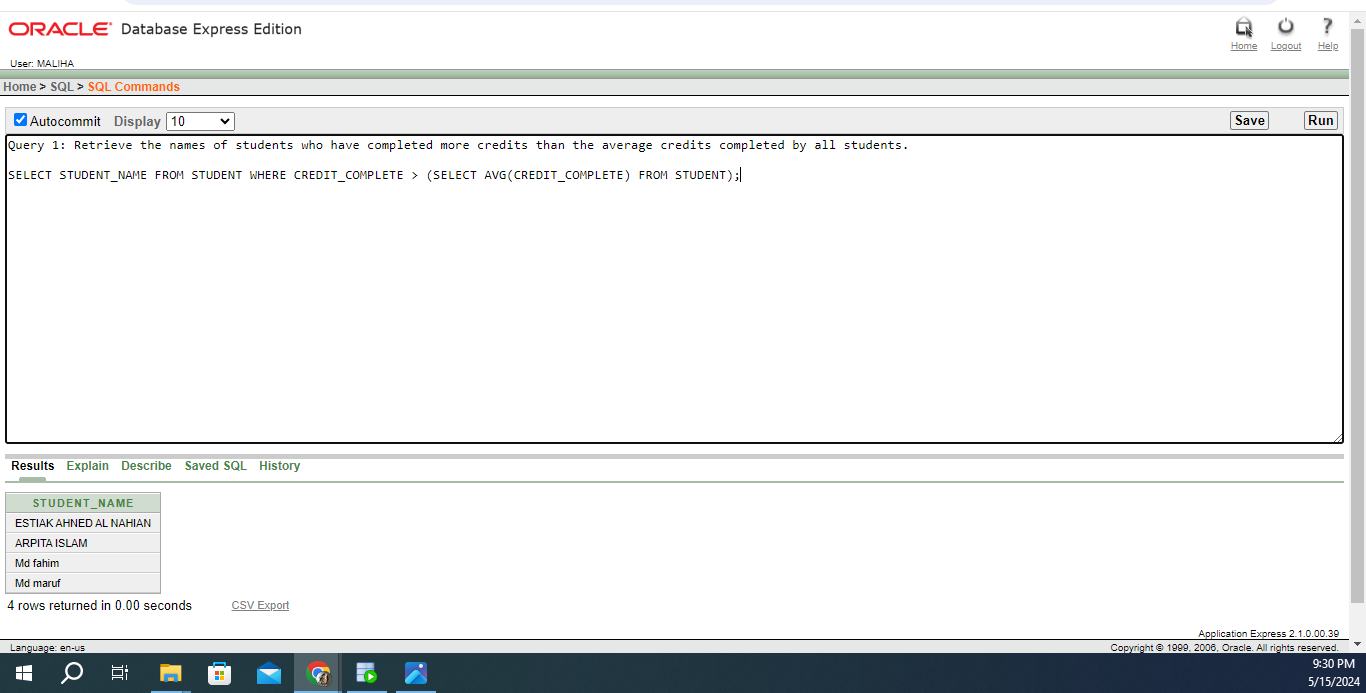
SELECT DEPT\_ID, COUNT(\*) AS Student\_Count FROM STUDENT GROUP BY DEPT\_ID;



**subquery :**

1. Retrieve the names of students who have completed more credits than the average credits completed by all students.

SELECT STUDENT\_NAME FROM STUDENT WHERE CREDIT\_COMPLETE > (SELECT AVG(CREDIT\_COMPLETE) FROM STUDENT);

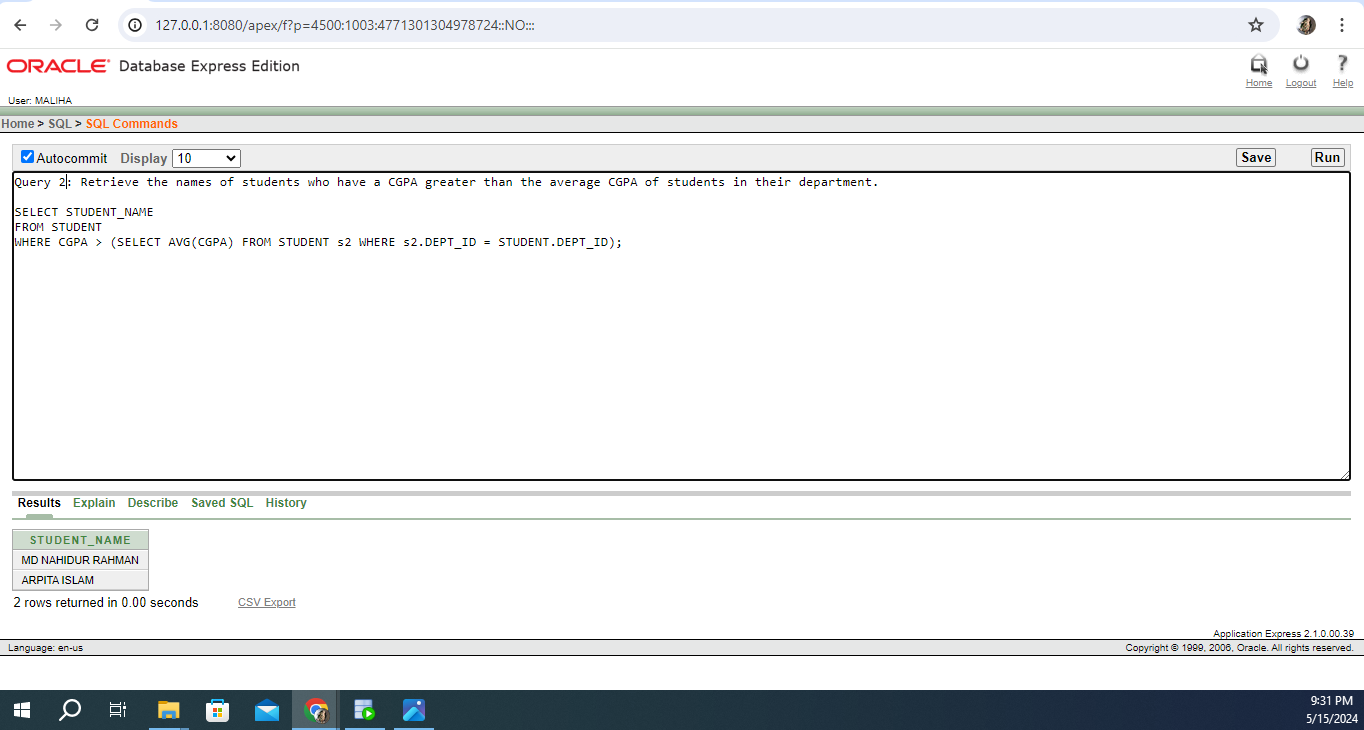


1. Retrieve the names of students who have a CGPA greater than the average CGPA of students in their department..

SELECT STUDENT\_NAME

FROM STUDENT

WHERE CGPA > (SELECT AVG(CGPA) FROM STUDENT s2 WHERE s2.DEPT\_ID = STUDENT.DEPT\_ID);



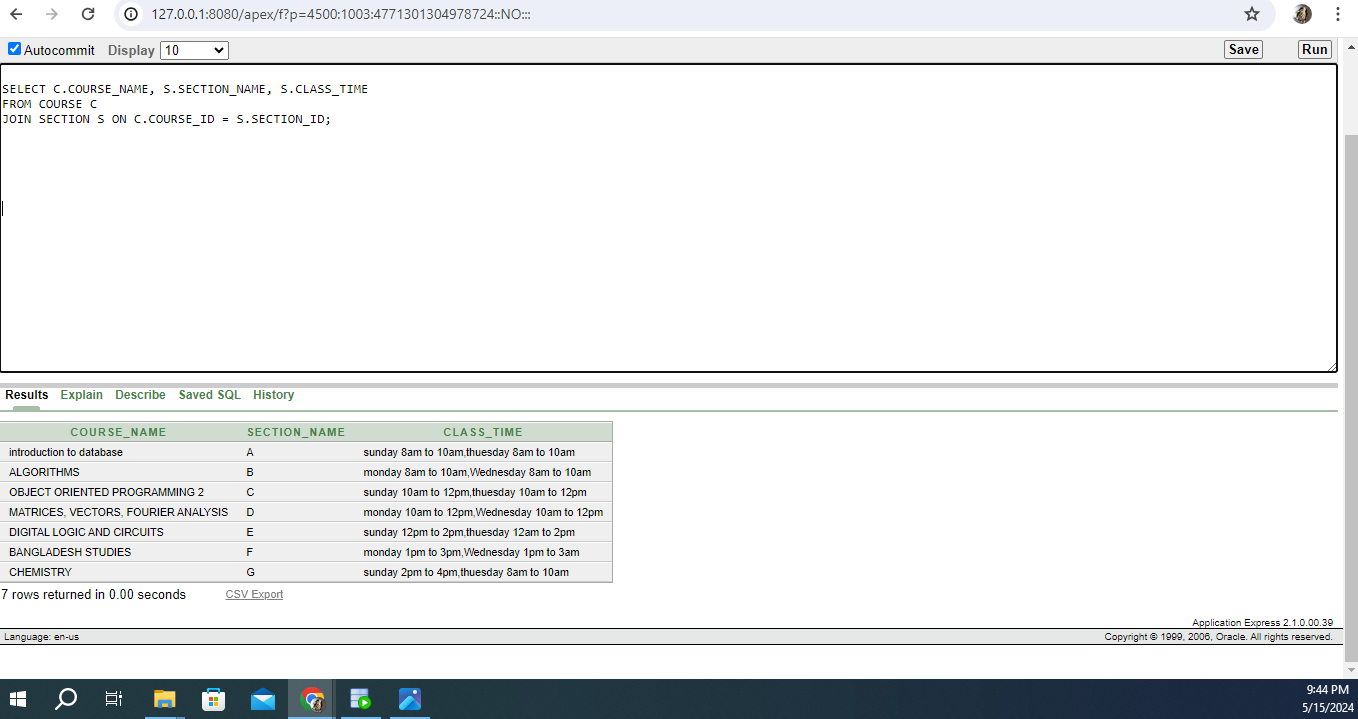
JOINING :

1.Retrieve the course name, section name, and class time for each course.

SELECT C.COURSE\_NAME, S.SECTION\_NAME, S.CLASS\_TIME

FROM COURSE C

JOIN SECTION S ON C.COURSE\_ID = S.SECTION\_ID;



2.To retrieve student registration information along with course name, department name, faculty name, class time, and room number.

SELECT

SR.STUDENT\_ID,

S.STUDENT\_NAME,

C.COURSE\_NAME,

D.DEPT\_NAME,

F.FACULTY\_NAME,

SEC.CLASS\_TIME,

SEC.CLASS\_ROOM

FROM

STUDENT\_REG SR

JOIN

STUDENT S ON SR.STUDENT\_ID = S.STUDENT\_ID

JOIN

COURSE C ON SR.COURSE\_ID = C.COURSE\_ID

JOIN

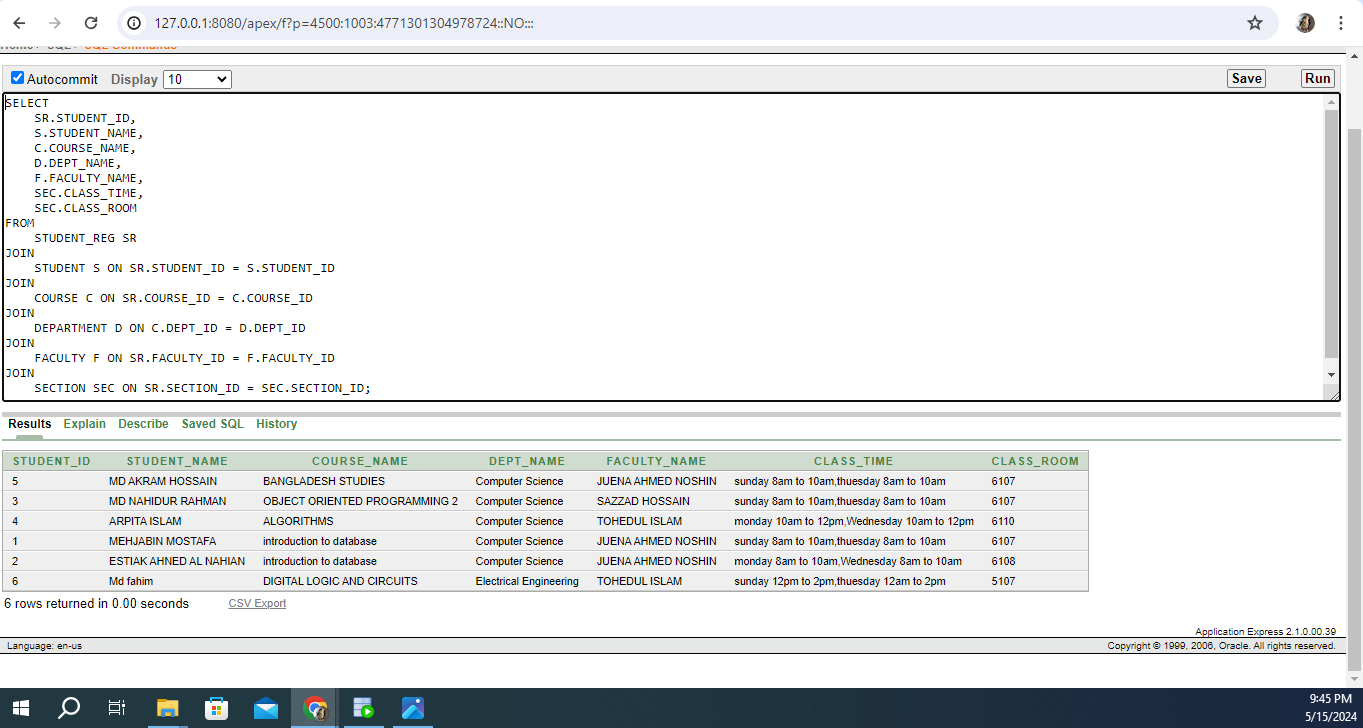
DEPARTMENT D ON C.DEPT\_ID = D.DEPT\_ID

JOIN

FACULTY F ON SR.FACULTY\_ID = F.FACULTY\_ID

JOIN

SECTION SEC ON SR.SECTION\_ID = SEC.SECTION\_ID;



**view:**

1.This view will provide a summary of student enrollment, showing the number of students enrolled in each course.

CREATE VIEW Course\_Enrollment AS

SELECT

C.COURSE\_ID,

C.COURSE\_NAME,

COUNT(SR.STUDENT\_ID) AS Enrollment\_Count

FROM

STUDENT\_REG SR

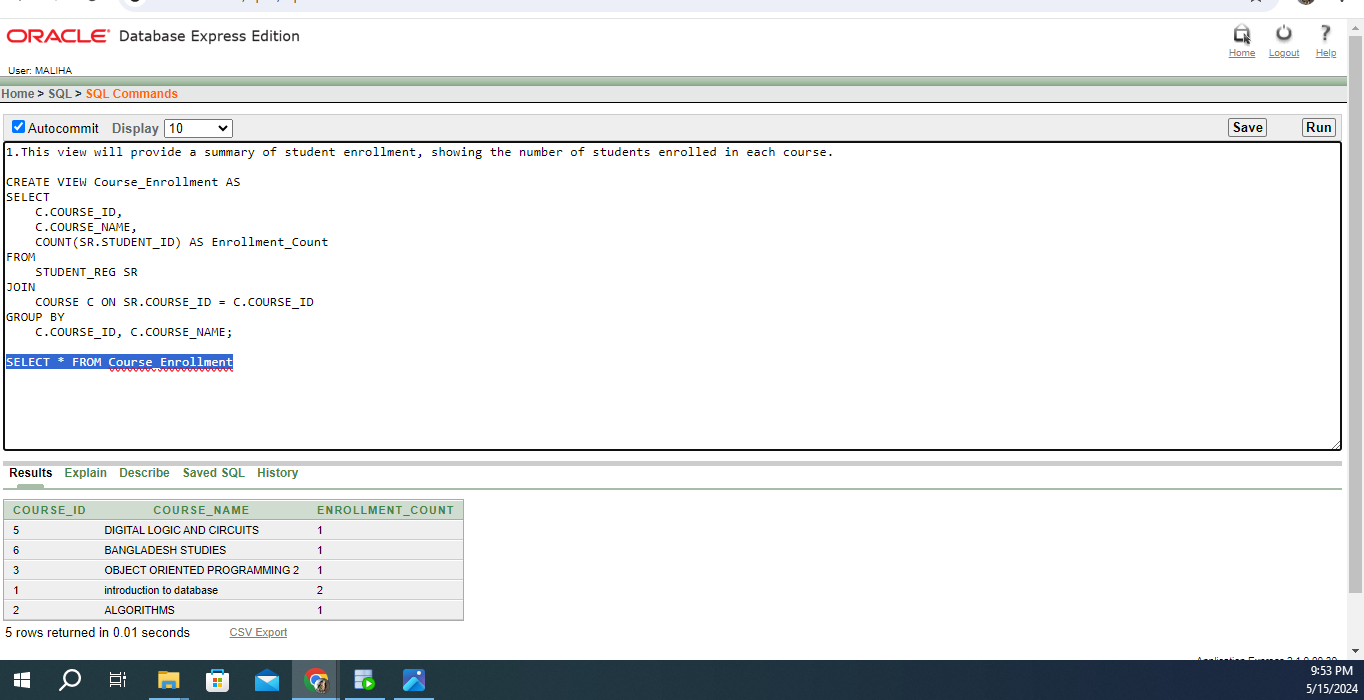
JOIN

COURSE C ON SR.COURSE\_ID = C.COURSE\_ID

GROUP BY

C.COURSE\_ID, C.COURSE\_NAME;

SELECT \* FROM Course\_Enrollment



2.To create a view that includes information about students' course enrollment and any actions related to adding or dropping courses

CREATE VIEW Student\_Enrollment\_Info AS

SELECT

SR.STUDENT\_ID,

S.STUDENT\_NAME,

C.COURSE\_NAME,

D.DEPT\_NAME,

F.FACULTY\_NAME,

SEC.CLASS\_TIME,

SEC.CLASS\_ROOM,

AD.ACTION

FROM

STUDENT\_REG SR

JOIN

STUDENT S ON SR.STUDENT\_ID = S.STUDENT\_ID

JOIN

COURSE C ON SR.COURSE\_ID = C.COURSE\_ID

JOIN

DEPARTMENT D ON C.DEPT\_ID = D.DEPT\_ID

JOIN

FACULTY F ON SR.FACULTY\_ID = F.FACULTY\_ID

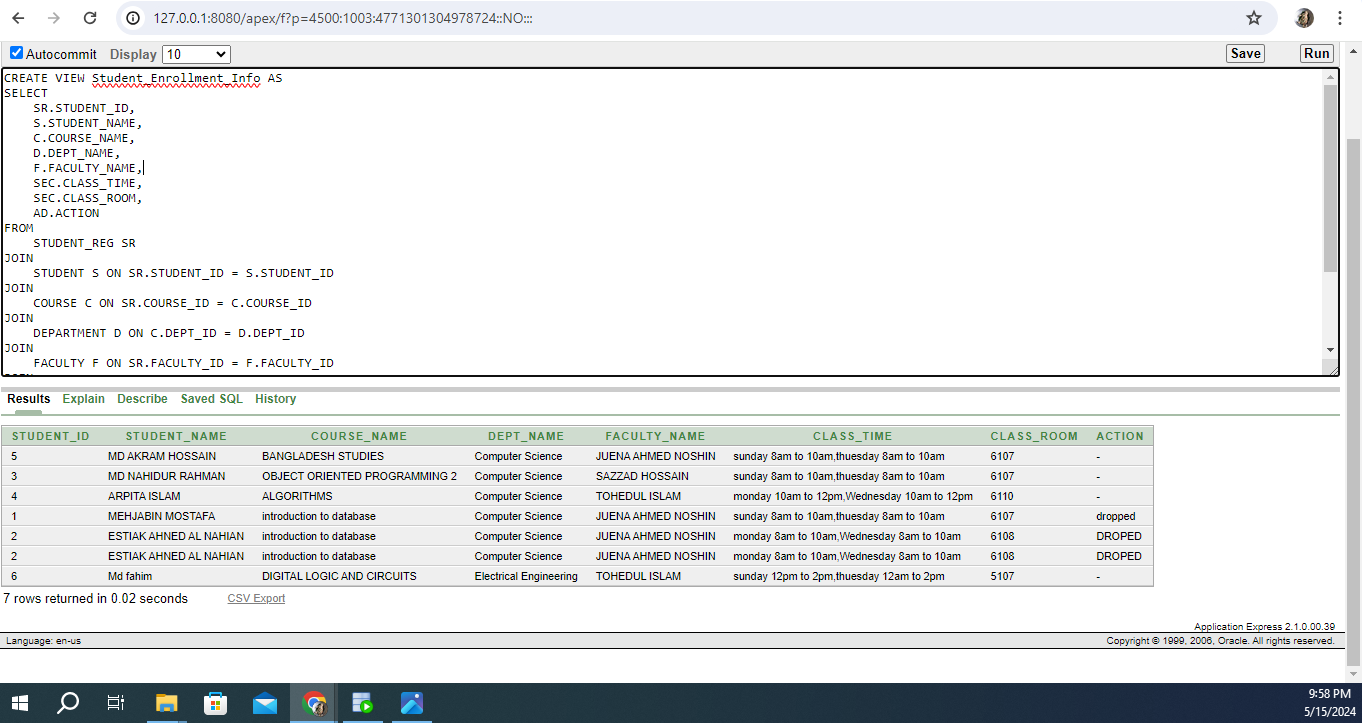
JOIN

SECTION SEC ON SR.SECTION\_ID = SEC.SECTION\_ID

LEFT JOIN

ADDING\_DROPING AD ON SR.STUDENT\_ID = AD.STUDENT\_ID AND SR.COURSE\_ID = AD.COURSE\_ID AND SR.SECTION\_ID = AD.SECTION\_ID;

SELECT \* FROM Student\_Enrollment\_Info



# Relational Algebra:

**1. Projecting Student Names and CGPA:**

**Π**(*STUDENT\_NAME, CGPA*(**STUDENT**))

2. Selecting Courses Offered by Computer Science Department:

**𝜎**(*DEPT\_ID=1*, **COURSE**)

3. Joining Student Registration with Course and Faculty:

**Π**(*STUDENT\_REG.STUDENT\_ID, STUDENT\_REG.COURSE\_ID, COURSE.COURSE\_ID, COURSE.DEPT\_ID, FACULTY.FACULTY\_ID)*

(**STUDENT\_REG ⋈ COURSE) ⋈ FACULTY**

4. Union of Computer Science and Electrical Engineering Courses:

**Π**(*COURSE\_ID, COURSE\_NAME, DEPT\_ID)* (**𝜎**(*DEPT\_ID=1,* **COURSE**) ∪ **𝜎**(*DEPT\_ID=3,* **COURSE**))

5. Projecting Student Names and Their Enrolled Course Names:

**Π**(*STUDENT.STUDENT\_NAME, COURSE.COURSE\_NAME*)

(**STUDENT ⋈ STUDENT\_REG) ⋈ COURSE**

# Conclusion:

The Course Registration Management System is developed to streamline course enrollment, enhance teaching and learning, and provide efficient administrative management of academic programs and student records.

Key functionalities include student enrollment, course details management, instructor assignments, user-friendly interfaces for students and instructors, administrative tools for course management and reporting, and implementation of security measures to protect sensitive data.

Ultimately, the system aims to optimize the course enrollment process, foster academic success, and empower administrators with the necessary tools for effective management of academic programs and student records.