

Project Title: University Management System

|  |  |  |
| --- | --- | --- |
| Student Name | Student Id | Contribution |
| ASHRAFUL ISLAM | 20-42010-1 | Introduction, Project proposal, Class Diagram, Use Case Diagram, Activity Diagram, ER diagram, User Interface |
| KAZI MUNTASHIR FAHAD | 20-43834-2 | Normalization, Up to 3rd NF, Schema Diagram, Scenario  Description,Table Creation, trigger. |
| MT. ZUBIDA KHATUN | 19-41677-3 | Table Creation, Data Insertion, User Interface. |
| SUMAIA MAHMUDA | 19-41622-3 | Query Writing, Diagram Cheaking, Conclusion |

**Course Name: ADVANCE DATABASE MANAGEMENT**

**SYSTEM.**

**Section: B**

**Contents**

Cover page 1

Introduction 3

Project Proposal 3

Class diagram 4

Use case diagram 5

Activity diagram 5

User Interface……………………………………………………………………...6-7

Scenario description…………................………………………………………..8

ER diagram 9

Normalization 10-17

Schema diagram 18

Table creation and insertion 19-36

Query writing 37-57

Conclusion 58

**Introduction:**

University management system project is a web-based solution for colleges, universities, and schools. It was created for the university and its affiliated institutions to conduct, monitor, and analyze complicated activities such as student admission, examinations, and much more. It is a program that manages the complete student life cycle up to degree completion.

we have computers with large computing power and almost every business is going to take the advantages of using those technologies. But nowadays digital certifications itself become an essential component for every business infrastructure. Because it provides security and it can identify every unique individual. Besides, it also provides confidential communications to the users.

This university management system we developed it to solve the problems of the universities and make work smart instead of using papers now you can you use this system to do all the work, every university or educational institution has challenges to overcome and manage the information of students, faculties, registrations courses and staff at the management level, this system designed to assist strategic planning and will help to ensure that the university can meet the minimum standard of the university.

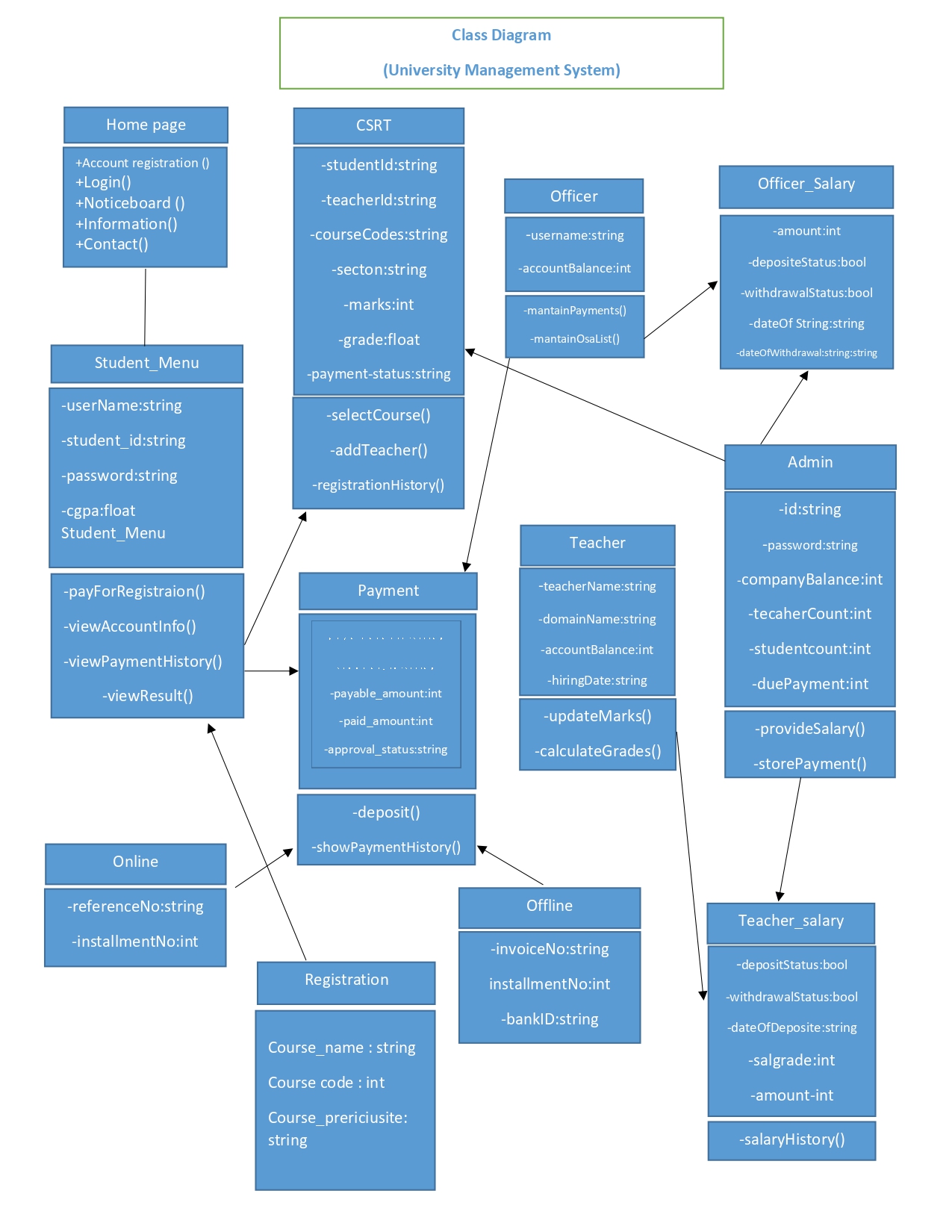
The main aim of this Project University Management System is to manage student registration as well as staff registration so that the teachers can submit student exam results online and take a class attendance online during class sessions.

**Project Proposal:**

We have developed a database system, which aims to replicate an online university portal / management system.

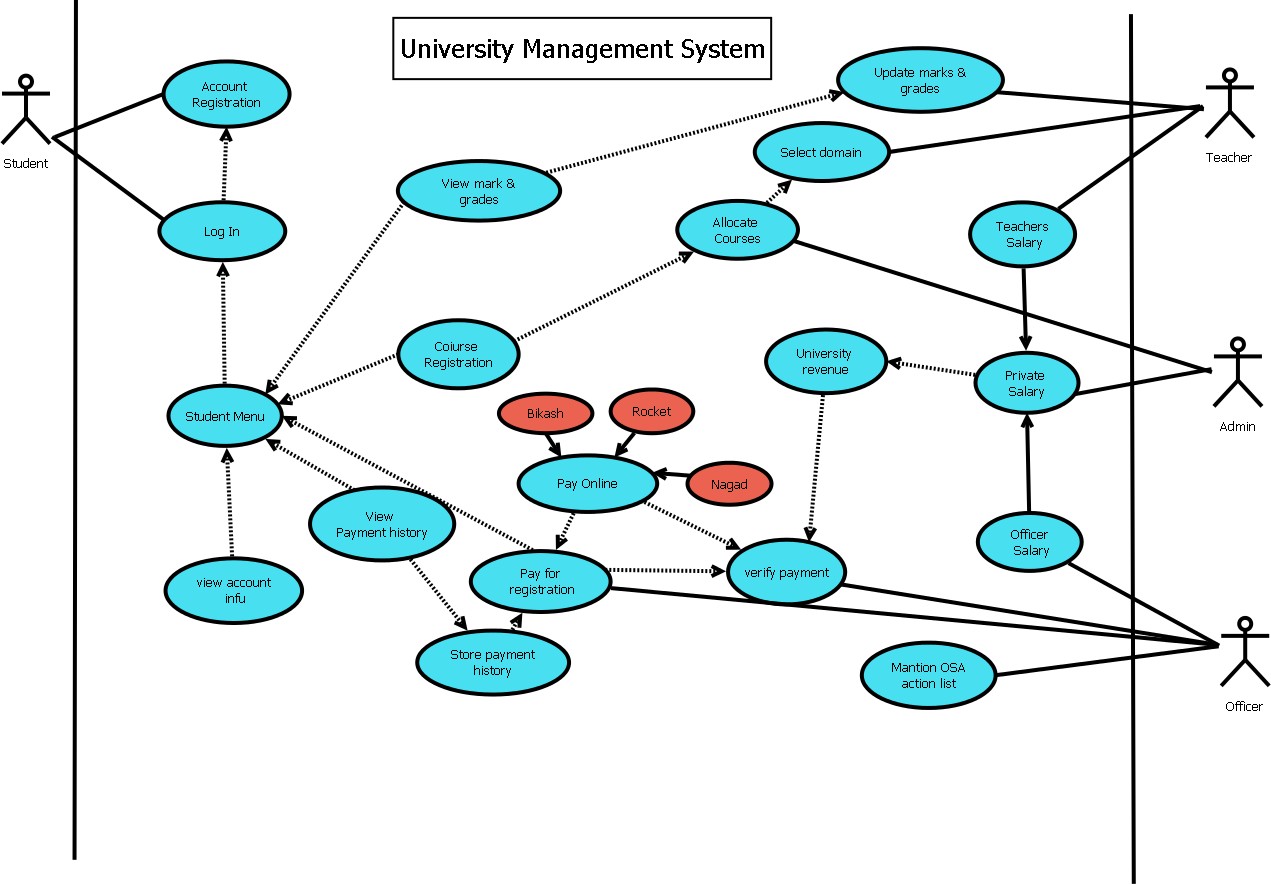
Here we have user types such as students, teachers, admin and officers. Students will be able to see their information, results and apply for registering courses. Teachers will be able to evaluate students, update their marks and grades. The admin will allocate courses for students who have applied for registration, store and update various attributes of the student account and provide salary to the teachers. The officer will maintain the payment system for the university and create records relating to order and discipline.

**Class Diagram:** This is the class diagram of our project.

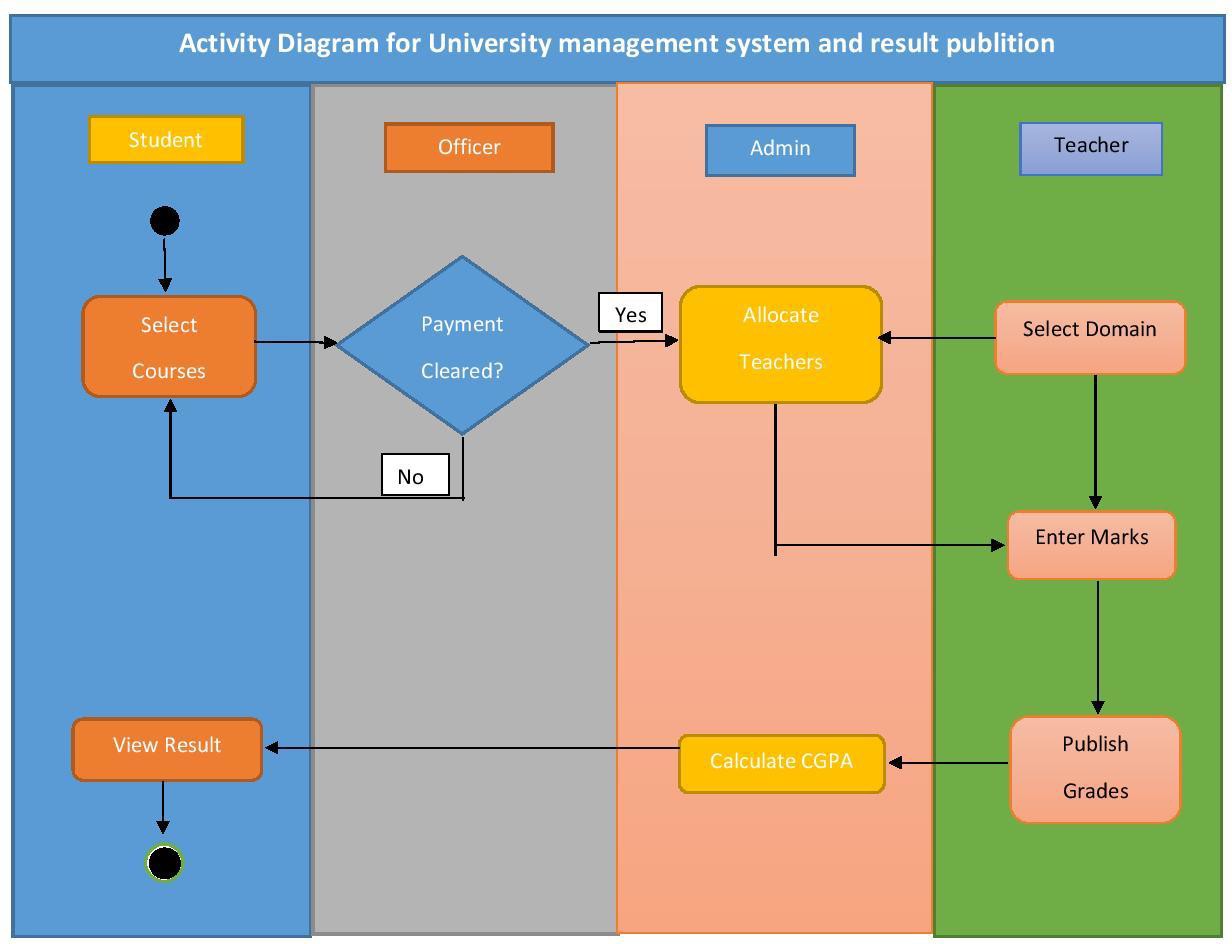
****

**Fig 1: Class Diagram.**

**Use case Diagram:** This is the use case diagram for our project.



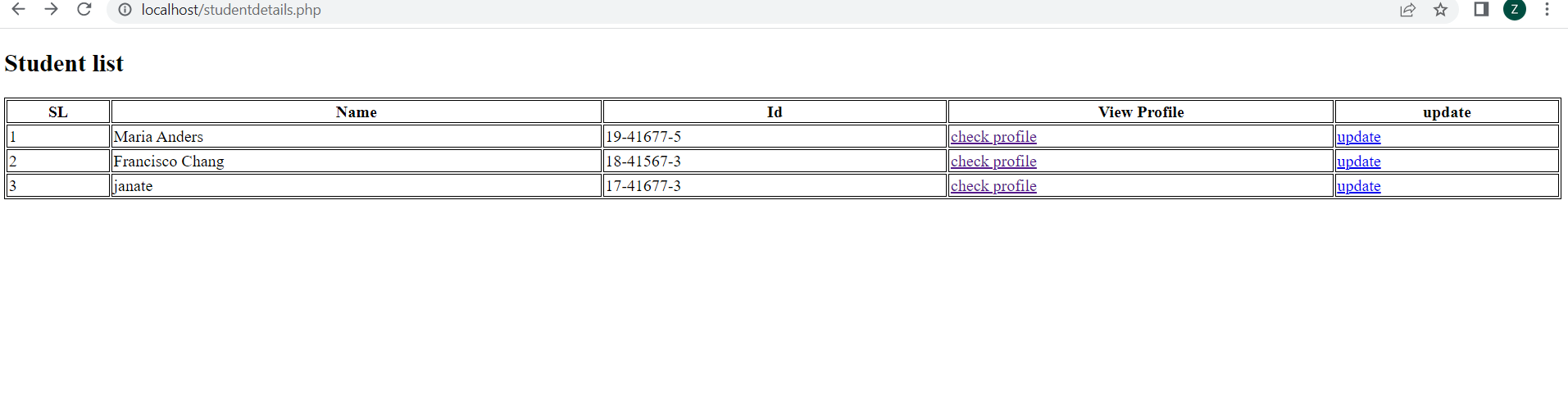
## **Fig 2: Use case diagram.**

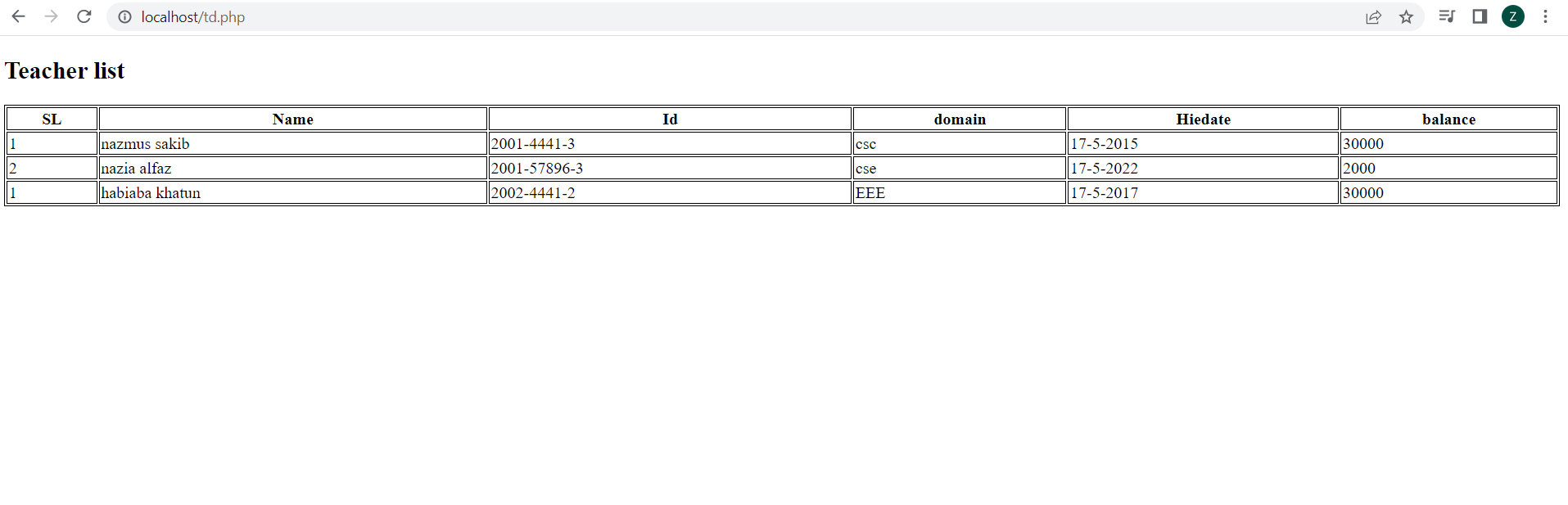
**Activity Diagram:** This is the activity diagram of our project

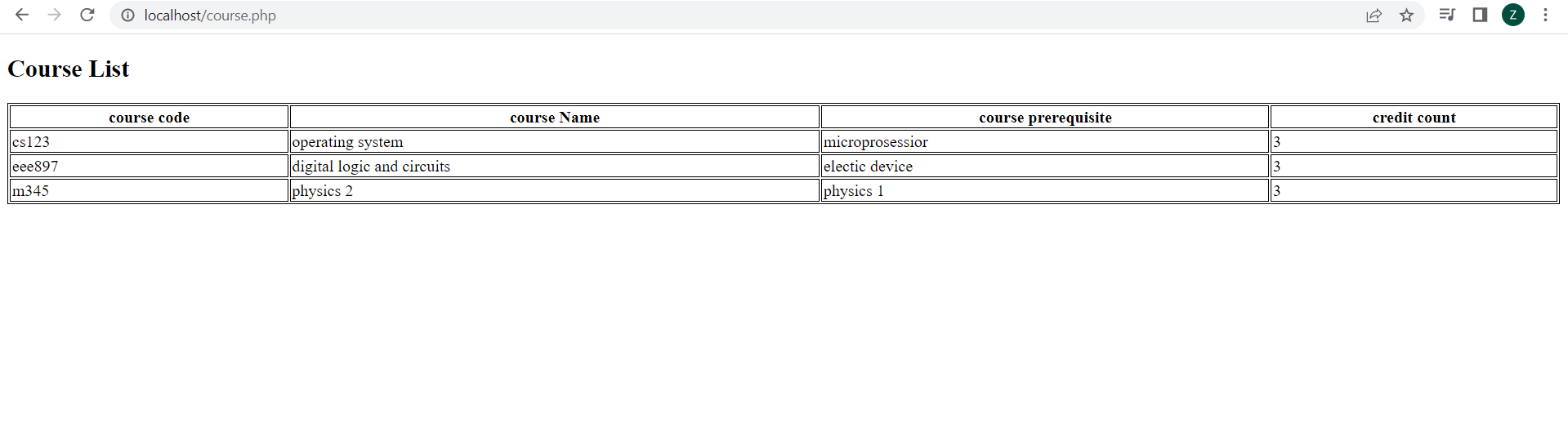
**Fig 3: Activity Diagram**

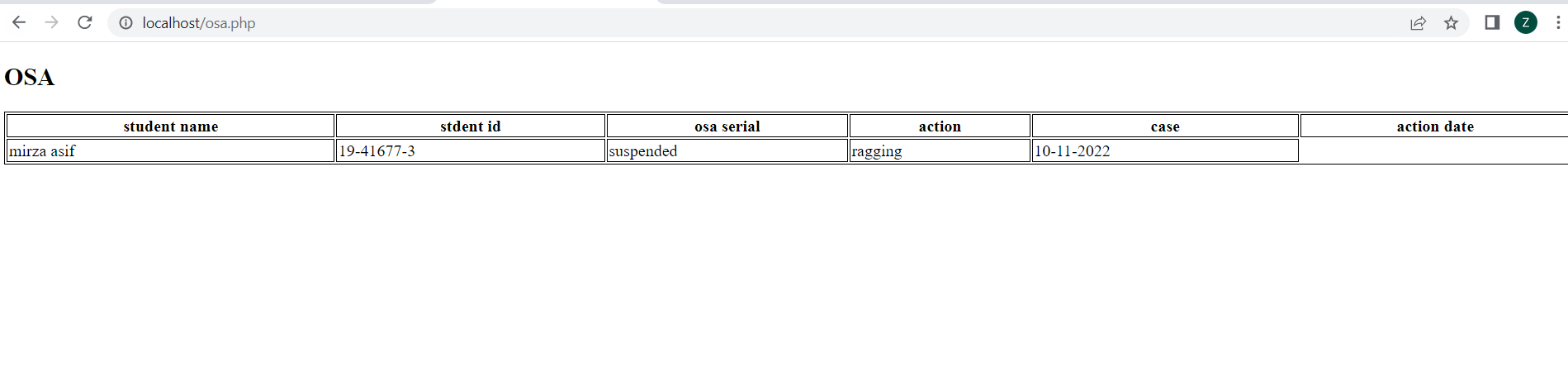
**User Interface:**

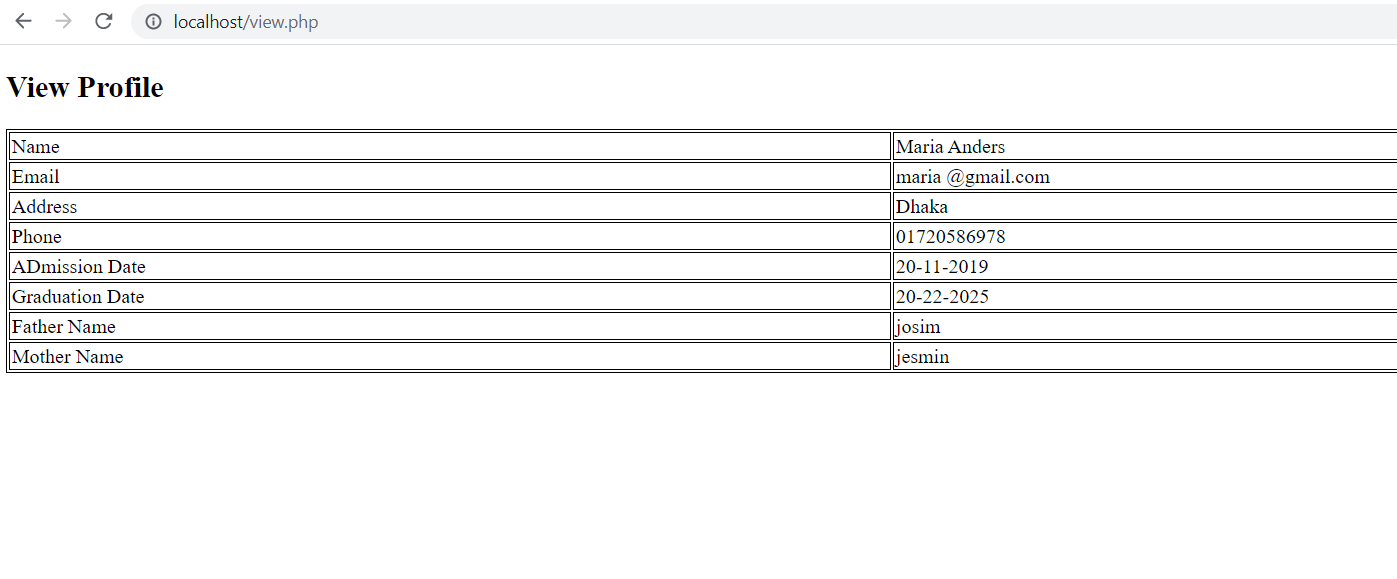
****

****

****

****

****

****

## Scenario Description:

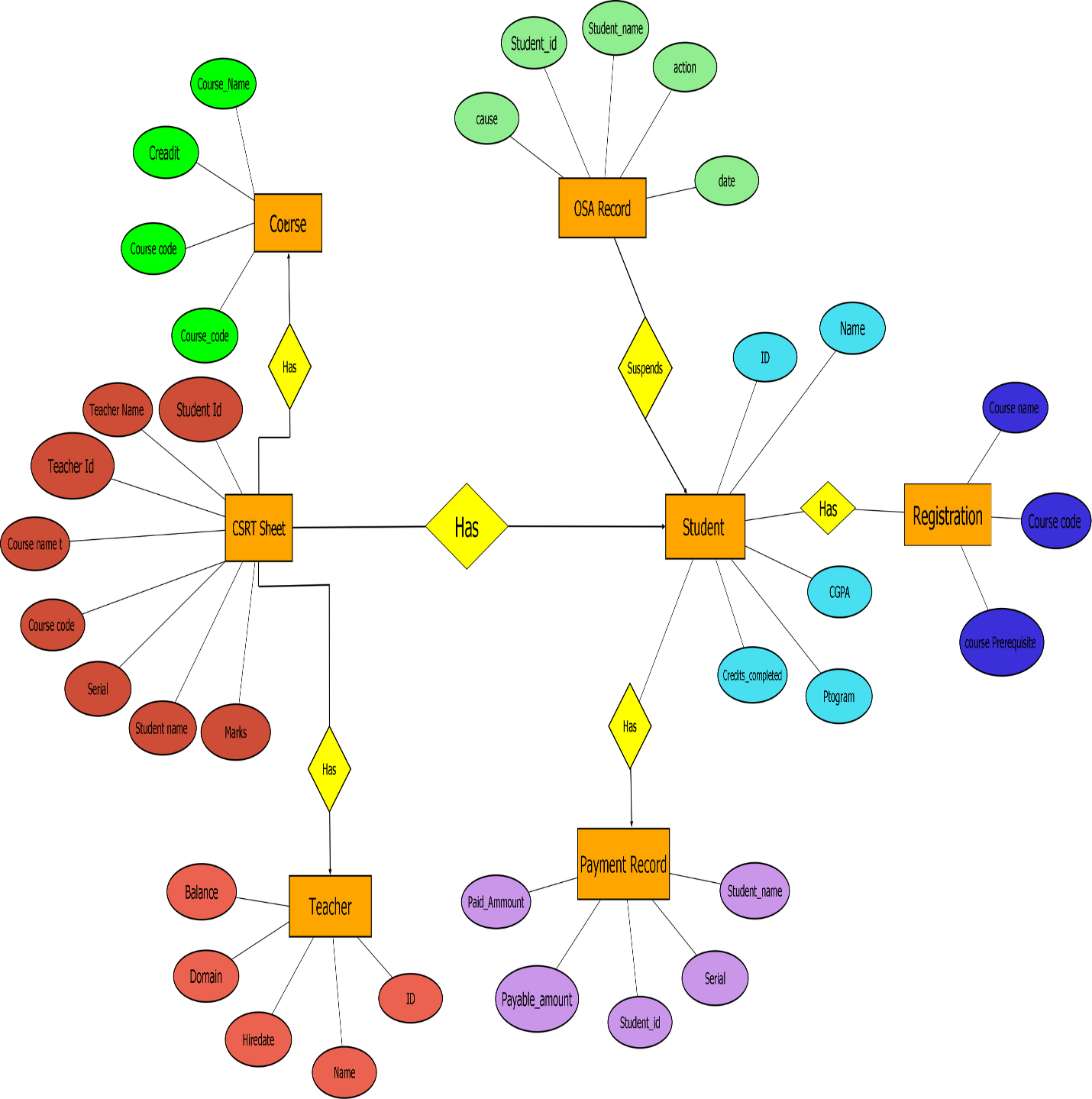
We have developed a database system which aims to replicate an university management system. Here we have user types such as students, teachers, admin for Office of Student Affair (OSA) and payment record officers.

In this management system student have their own information. They have their own ID, name, CGPA, program and also how much credit he or she has completed. Students will be able to see their information, results and apply for registering courses.Teachers have their own information. They have their own ID, name, hiredate, domain of their study and also balance information. Teachers will be able to evaluate students, update their marks and grades.

Students and Teachers has been identify by their unique ID. The admin will allocate courses for students who have applied for registration, store and update various attributes of the student account and also provide salary to the teachers. Payment record will also capture by an officer. Who will have information about paid amount, payable amount.

Officer also the information of the student id, student name for identity a student. Serial is unique for the Payment Record officer.

**ER Diagram:** This is the ER diagram of our project

****

**Fig 4: ER Diagram**

**Normalization:** This is the basic normalization that required.

**Has:**

#### **UNF:**

Has (course\_code, course\_prerequisite\_code, credit\_count, course\_name, ID, name, hiredate, domain, balance)

**1NF:** There is no multilevel attribute.

#### course\_code, course\_prerequisite\_code, credit\_count, course\_name, ID, name, hiredate, domain, balance

#### **2NF:**

course\_code, course\_prerequisite\_code, credit\_count, course\_name ID, name, hiredate, domain, balance

**3NF:**  There is no transitive dependency.

course\_code, course\_prerequisite\_code, credit\_count, course\_name ID, name, hiredate, domain, balance

**Table Creation:**

#### course\_code, course\_prerequisite\_code, credit\_count, course\_name

#### ID, name, hiredate, domain, balance, course\_code

**Suspends:**

#### **UNF:**

#### Suspends (student\_id, student\_name, action, cause, date, ID, name, CGPA, program, credits\_completed)

#### **1NF:** There is no multilevel attribute.

#### student\_id, student\_name, action, cause, date, ID, name, CGPA, program, credits\_completed

#### **2NF:**

#### student\_id, student\_name, action, cause, date ID, name, CGPA, program, credits\_completed

#### **3NF:** There is no transitive dependency.

#### student\_id, student\_name, action, cause, date

#### ID, name, CGPA, program, credits\_completed

**Table Creation:**

#### student\_id, student\_name, action, cause, date

#### ID, name, CGPA, program, credits\_completed, student\_id

**Has:**

#### **UNF:**

#### Has (ID, name, CGPA, program, credits\_completed, serial, paid\_amount, payable\_amount, student\_id, student\_name)

#### **1NF:** There is no multilevel attribute.

#### ID, name, CGPA, program, credits\_completed, serial, paid\_amount, payable\_amount, student\_id, student\_name

#### **2NF:**

#### ID, name, CGPA, program, credits\_completed

#### serial, paid\_amount, payable\_amount, student\_id, student\_name

#### **3NF:** There is no transitive dependency.

#### ID, name, CGPA, program, credits\_completed

#### serial, paid\_amount, payable\_amount, student\_id, student\_name

**Table Creation:**

#### ID, name, CGPA, program, credits\_completed

#### serial, paid\_amount, payable\_amount, student\_id, student\_name, ID

**Has:**

#### **UNF:**

Has (course\_code, course\_prerequisite\_code, credit\_count, course\_name, ID, name, CGPA, program, credits\_completed)

**1NF:** There is no multilevel attribute.

course\_code, course\_prerequisite\_code, credit\_count, course\_name, ID, name, CGPA, program, credits\_completed

#### **2NF:**

course\_code, course\_prerequisite\_code, credit\_count, course\_name ID, name, CGPA, program, credits\_completed

**3NF:** There is no transitive dependency.

course\_code, course\_prerequisite\_code, credit\_count, course\_name ID, name, CGPA, program, credits\_completed

**Table Creation:**

course\_code, course\_prerequisite\_code, credit\_count, course\_name ID, name, CGPA, program, credits\_completed, course\_code

**Has:**

#### **UNF:**

Has (course\_code, course\_prerequisite\_code, credit\_count, course\_name, ID, name, CGPA, program, credits\_completed)

**1NF:** There is no multilevel attribute.

course\_code, course\_prerequisite\_code, credit\_count, course\_name, ID, name, CGPA, program, credits\_completed

#### **2NF:**

course\_code, course\_prerequisite\_code, credit\_count, course\_name ID, name, CGPA, program, credits\_completed

**3NF:** There is no transitive dependency.

course\_code, course\_prerequisite\_code, credit\_count, course\_name ID, name, CGPA, program, credits\_completed

**Table Creation:**

course\_code, course\_prerequisite\_code, credit\_count, course\_name ID, name, CGPA, program, credits\_completed, course\_code

**Has:**

#### **UNF:**

#### Has (ID, name, CGPA, program, credits\_completed,course\_name, course\_code, course prerequisite)

#### **1NF:** There is no multilevel attribute.

#### ID, name, CGPA, program, credits\_completed,course\_name, course\_code, course prerequisite

#### **2NF:**

ID, name, CGPA, program, credits\_completed

course\_name, course\_code, course prerequisite

#### **3NF:** There is no transitive dependency.

ID, name, CGPA, program, credits\_completed

course\_name, course\_code, course prerequisite

**Table Creation:**

ID, name, CGPA, program, credits\_completed

course\_name, course\_code, course prerequisite

**Has:**

#### **UNF:**

Has (student\_name, student\_id, teacher\_name, teacher\_id, course\_name, course\_code, serial, marks, course\_name, course\_code,credit)

**1NF:** There is no multilevel attribute.

student\_name, student\_id, teacher\_name, teacher\_id, course\_name, course\_code, serial, marks, course\_name, course\_code,credit

#### **2NF:**

student\_name, student\_id, teacher\_name, teacher\_id

course\_name, course\_code, serial, marks, course\_name, course\_code,credit

**3NF:** There is no transitive dependency.

student\_name, student\_id, teacher\_name, teacher\_id

course\_name, course\_code, serial, marks, course\_name, course\_code,credit

**Table Creation:**

student\_name, student\_id, teacher\_name, teacher\_id

course\_name, course\_code, serial, marks, course\_name, credit

**Temparary Table:**

course\_code, course\_prerequisite\_code, credit\_count, course\_name ID, name, hiredate, domain, balance, **course\_code**

student\_id, student\_name, action, cause, date

~~ID, name, CGPA, program, credits\_completed,~~ **~~student\_id~~**~~,~~

~~ID, name, CGPA, program, credits\_completed~~

serial, paid\_amount, payable\_amount, student\_id, student\_name, **ID**

~~course\_code, course\_prerequisite\_code, credit\_count, course\_name~~

ID, name, CGPA, program, credits\_completed, **course\_code**, **student\_id**,

~~ID, name, CGPA, program, credits\_completed~~

~~course\_name, course\_code, course prerequisite~~

student\_name, student\_id, teacher\_name, teacher\_id

course\_name, course\_code, serial, marks, course\_name

**Final Table:**

course\_code, course\_prerequisite\_code, credit\_count, course\_name ID, name, hiredate, domain, balance, **course\_code**

student\_id, student\_name, action, cause, date

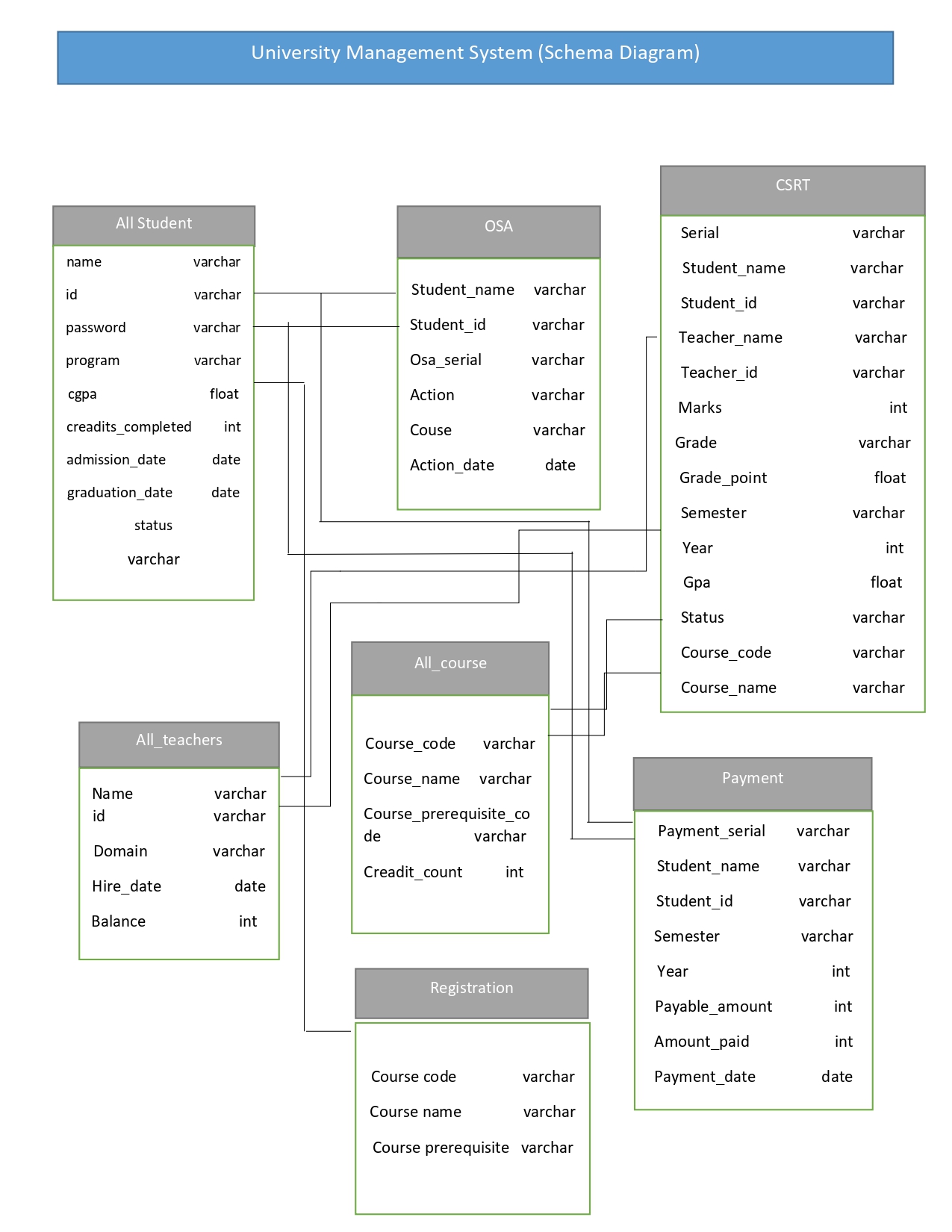
serial, paid\_amount, payable\_amount, student\_id, student\_name, **ID**

ID, name, CGPA, program, credits\_completed, **course\_code**, **student\_id**

course\_name, course\_code, **serial**, marks, course\_name

course\_name, course\_code, serial, marks, course\_name

**Schema diagram:** This is the schema diagram of our project

****

**Fig 5: Schema diagram**

**Table Creation & Data Insertion:**

**Student Table:**

all\_student

create table all\_student (

name varchar(50) not null,

id int constraint pk\_all\_student primary key, password varchar(25) not null,

program varchar(10), cgpa float, credits\_completed int, admission\_date date,

graduation\_date date, status varchar(20));

create sequence all\_student\_id

increment by 1

start with 1 nocycle

nocache;

CREATE INDEX student\_idx ON all\_student(name);

**Insertion:**

insert into all\_student values

('Waskiqure',all\_student\_id.nextval,'0161','EEE', 3.50, 19,to\_date('10-dec-22','dd-mm-yyyy'), null,'Studying');

insert into all\_student values

('Shahriar Zaman',all\_student\_id.nextval,'1234','CSE', 3.84, 16,to\_date('19-dec- 21','dd-mm-yyyy'), null,'Studying');

insert into all\_student values

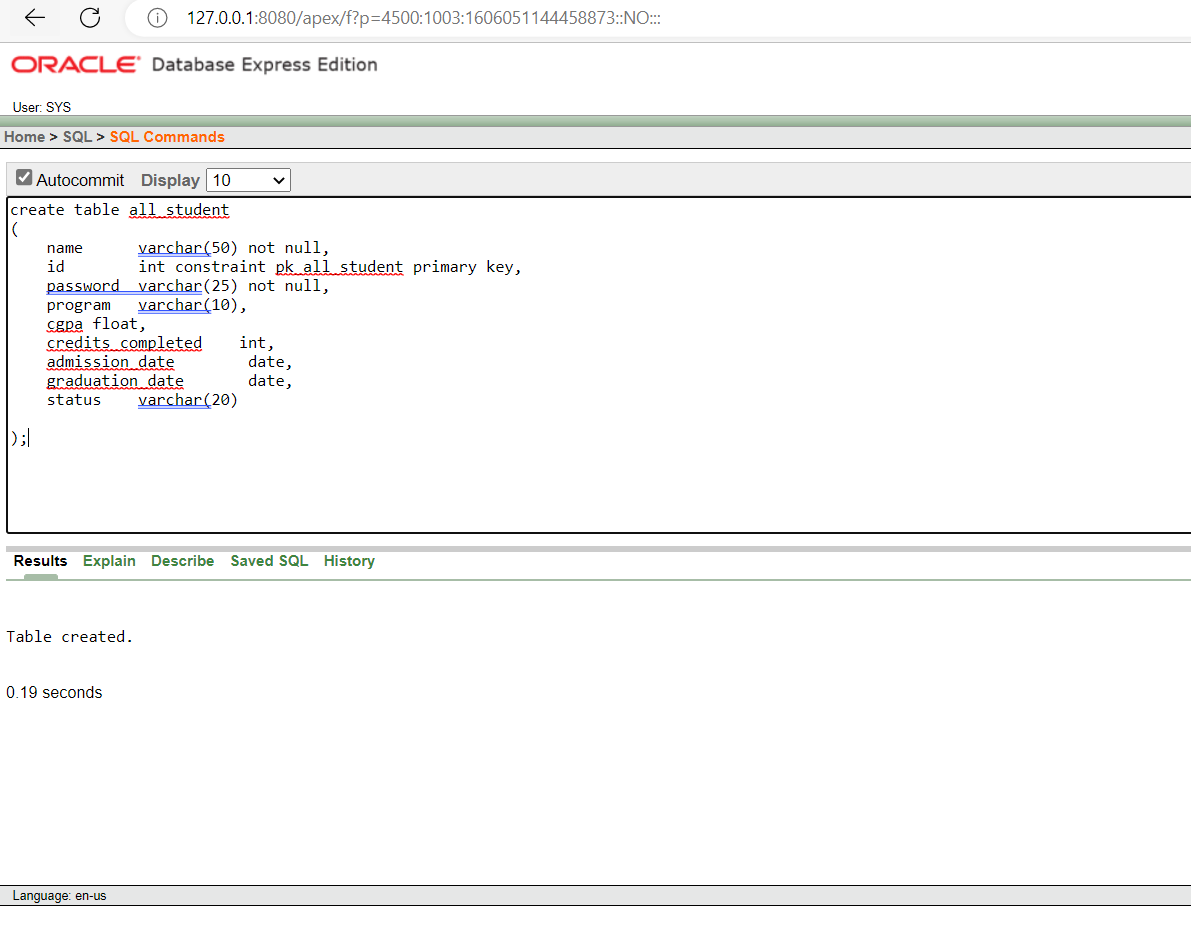
('sumsur nahar',all\_student\_id.nextval,'0159','EEE', 3.25, 15,to\_date('19-dec- 22','dd-mm-yyyy'), null,'Studying');

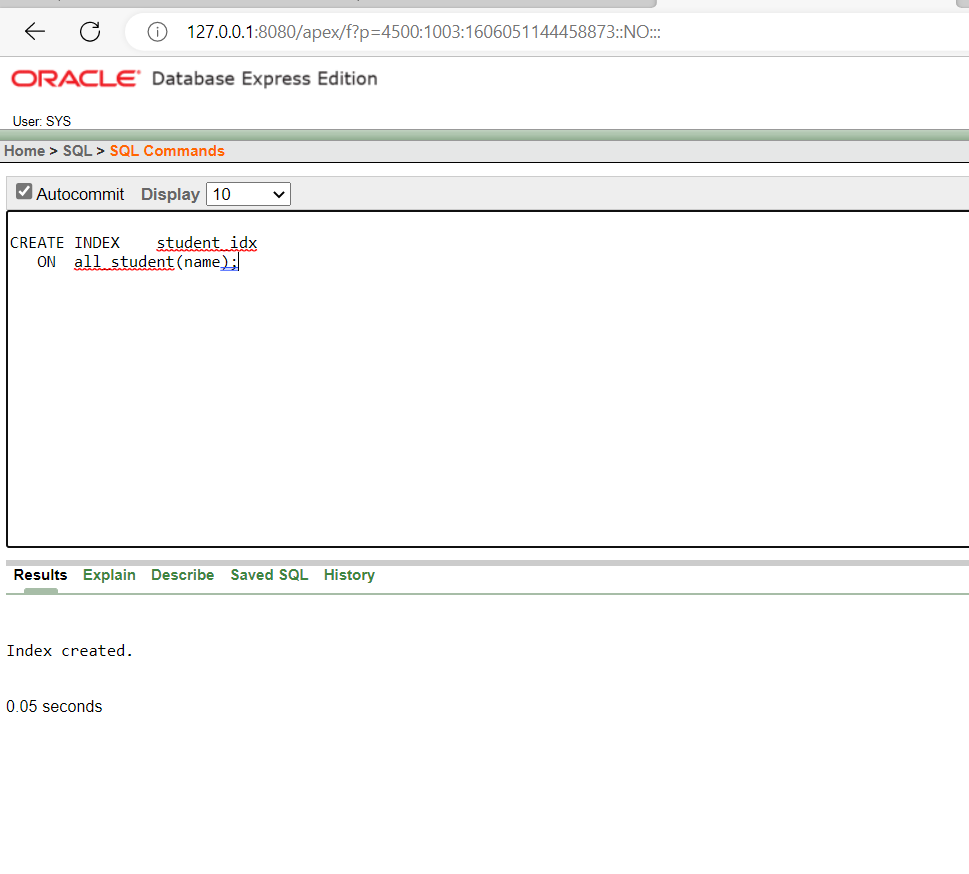
insert into all\_student values

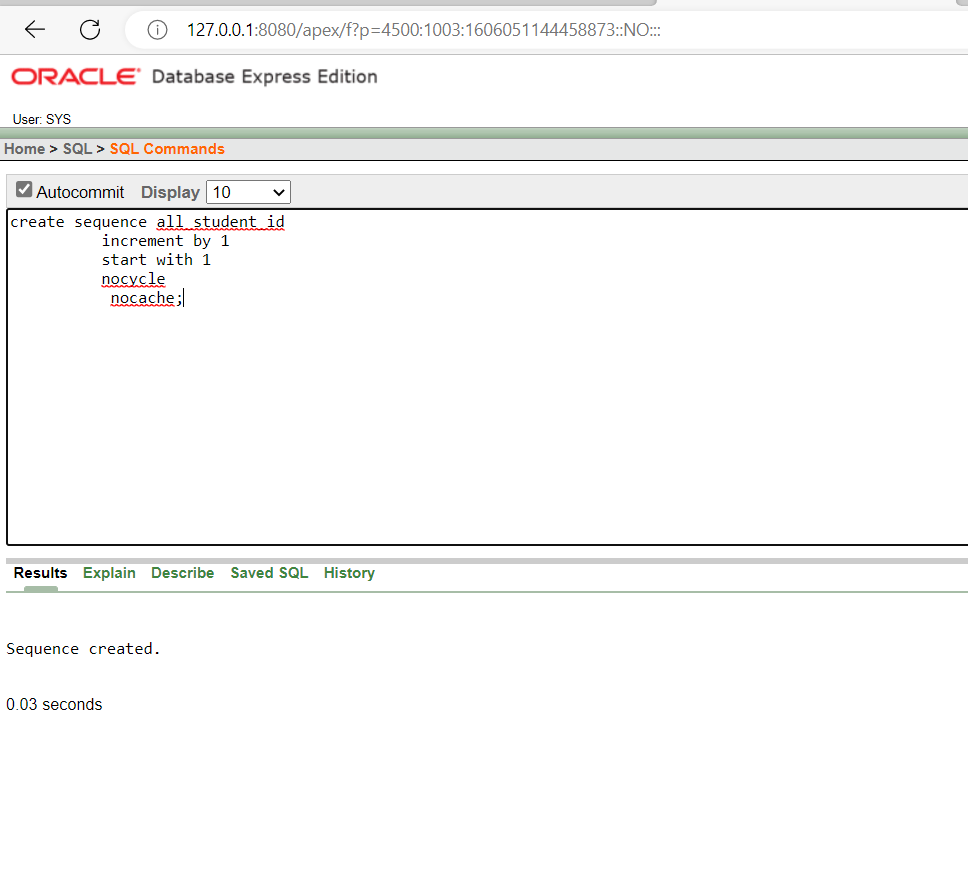
('AMANULLAH',all\_student\_id.nextval,'0160','CSE', 2.98, 15,to\_date('19-dec-21','dd-mm-yyyy'), null,'Studying');

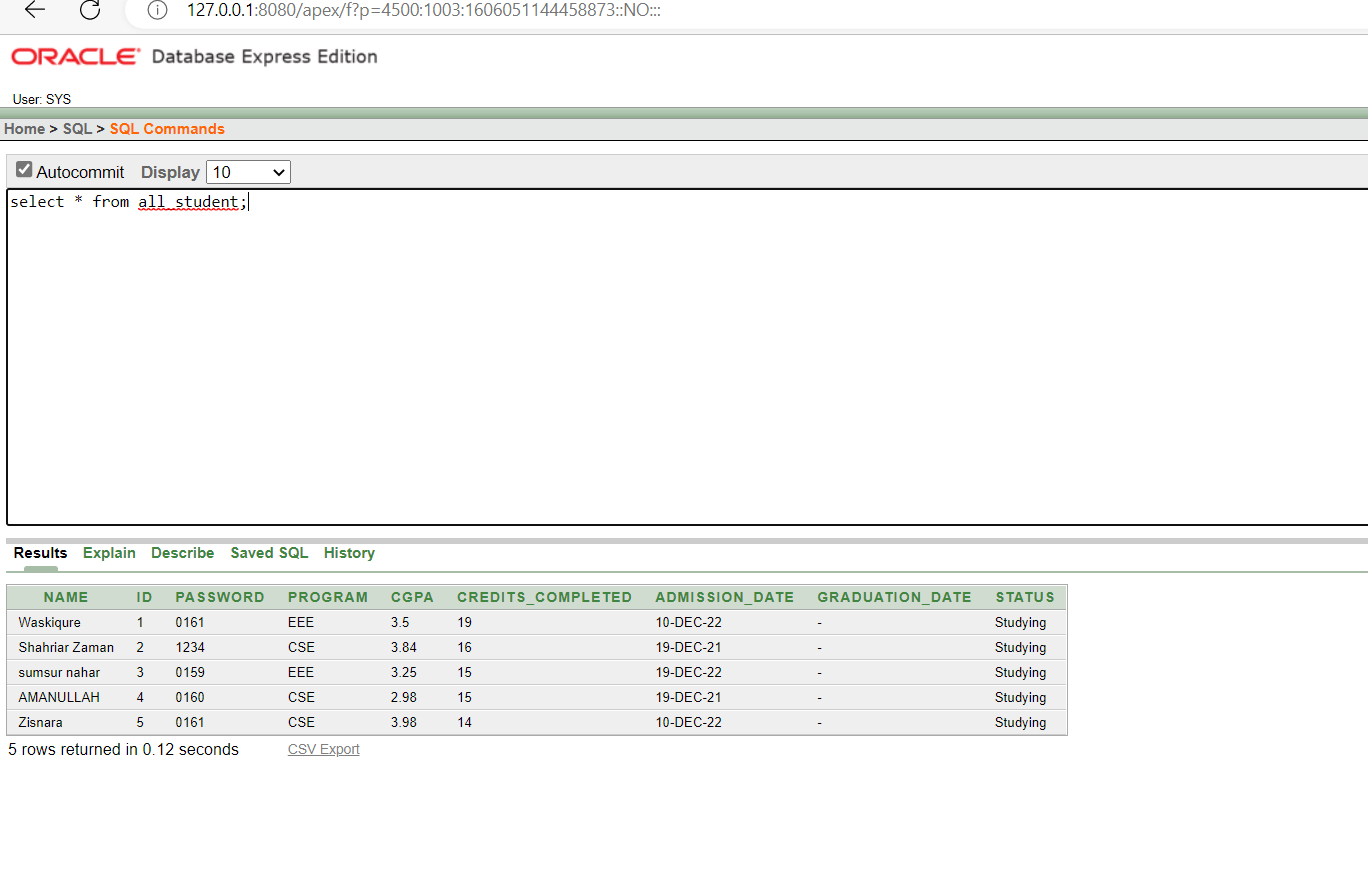
insert into all\_student values

('Zisnara',all\_student\_id.nextval,'0161','CSE', 3.98, 14,to\_date('10-dec-22','dd-mm-yyyy'), null,'Studying');









**Teachers Table:**

create teacher

create table all\_teachers ( name varchar(50) not null,

id int constraint pk\_all\_teachers primary key, domain varchar(10),

hire\_date date, balance int);

create sequence all\_teachers\_id

increment by 1

start with 1 nocycle

nocache;

CREATE INDEX teacher\_idx ON all\_teachers(name,id);

**Insertion:**

insert into all\_teachers values

('Nazmul Hossain',all\_teachers\_id.nextval,'CSC',to\_date('17-11-2022','dd-mm- yyyy'),30000 );

insert into all\_teachers values

('kuwsur imron',all\_teachers\_id.nextval,'CSC',to\_date('5-10-2022','dd-mm- yyyy'),30000 );

insert into all\_teachers values

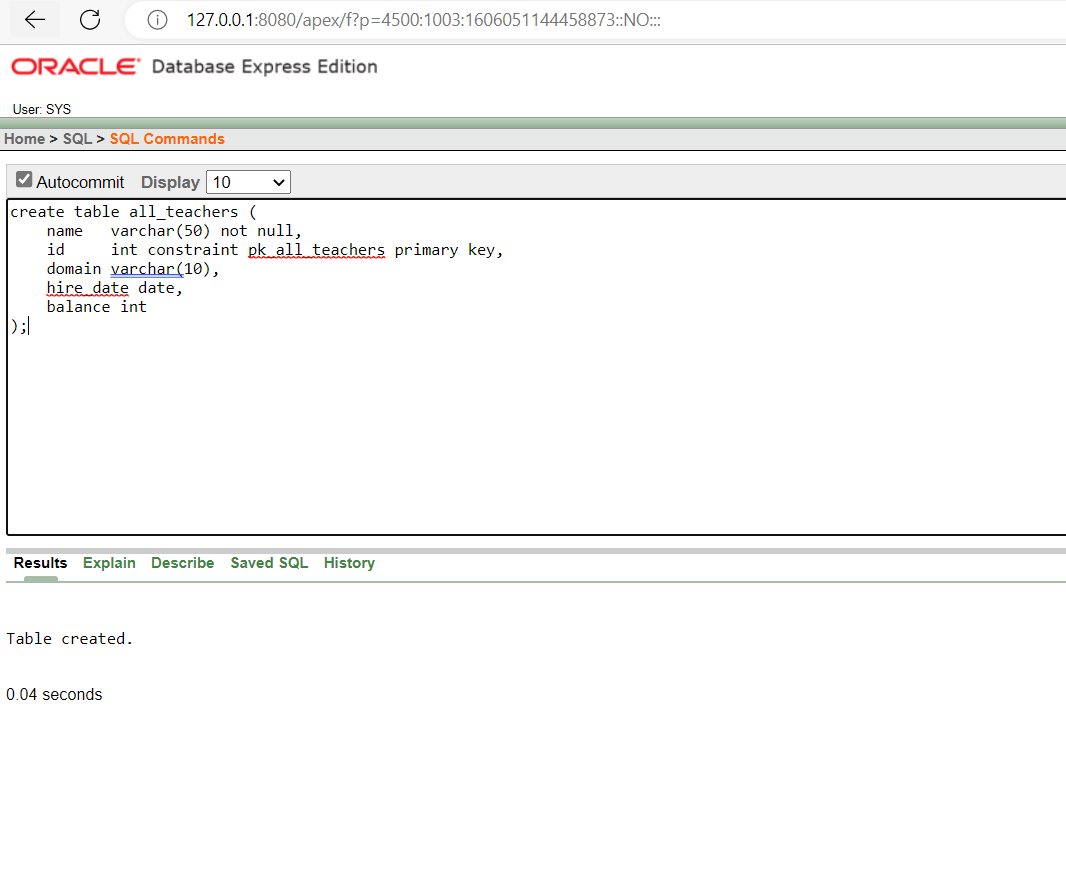
('nazia alfaz',all\_teachers\_id.nextval,'CSC',to\_date('5-1-2022','dd-mm- yyyy'),50000 );

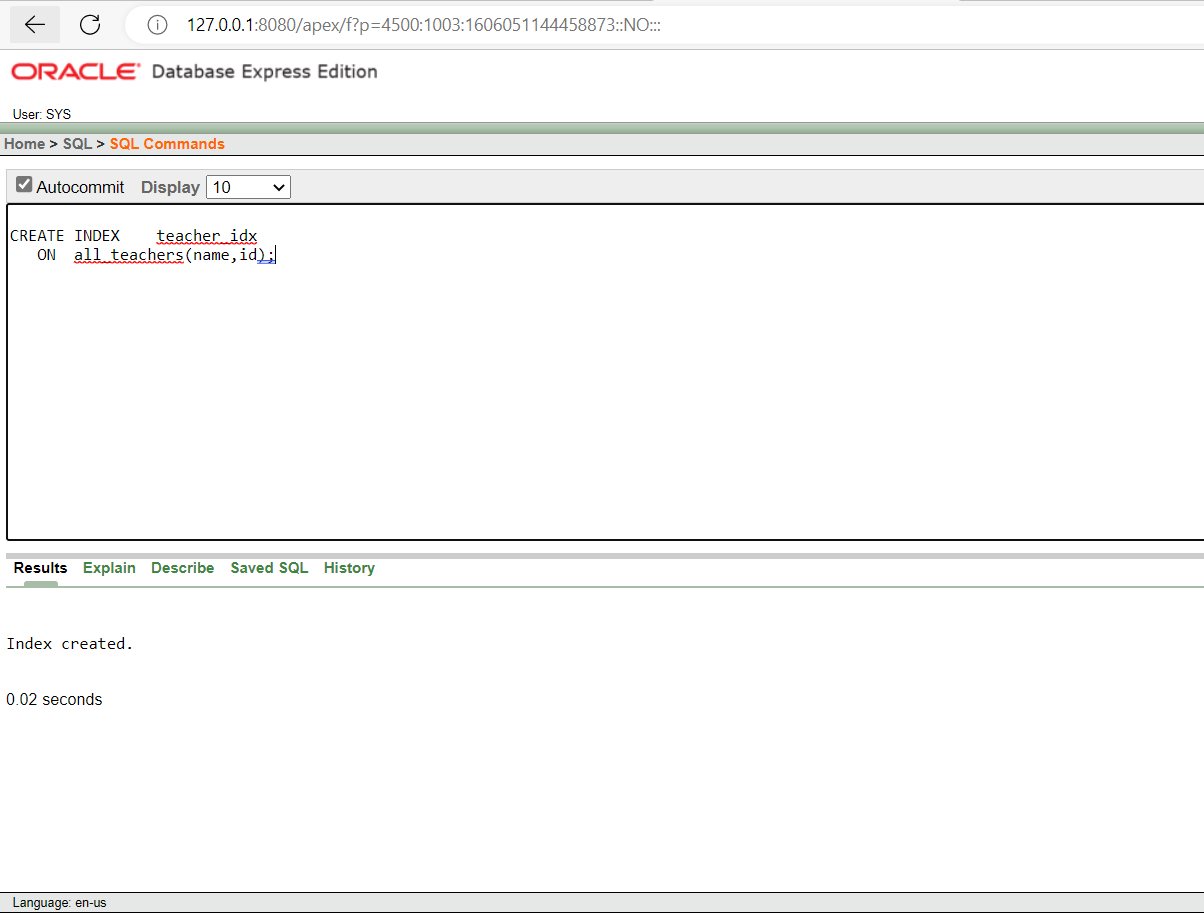
insert into all\_teachers values

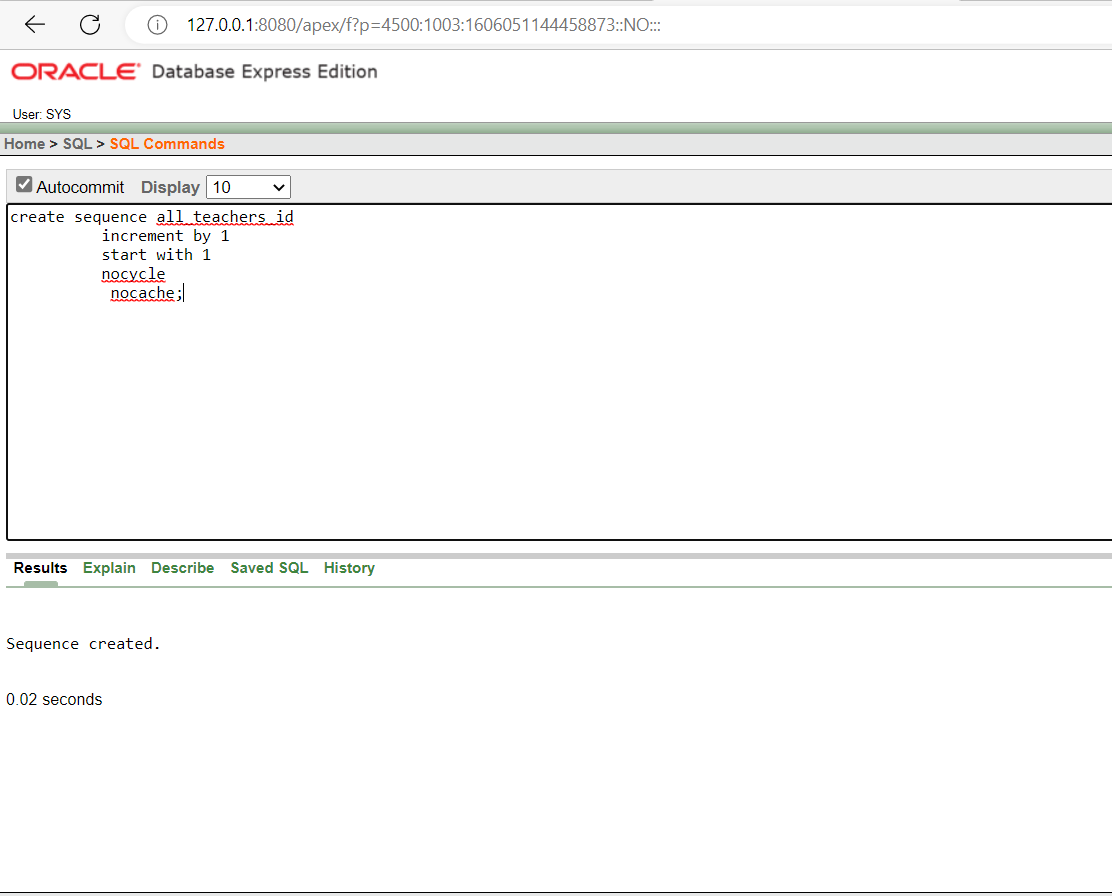
('tanjil amin',all\_teachers\_id.nextval,'EEE',to\_date('2-8-2021','dd-mm- yyyy'),50000 );

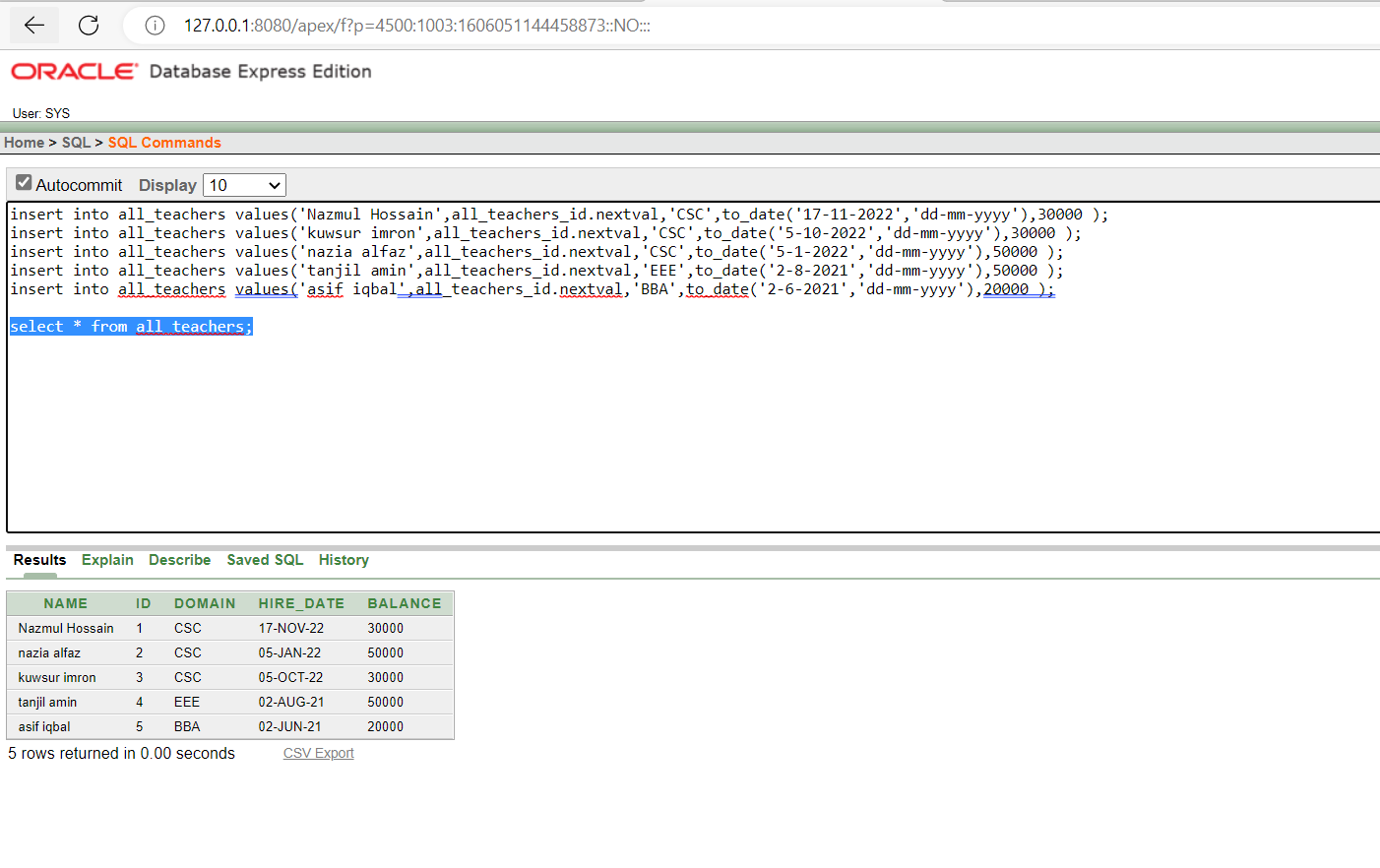
insert into all\_teachers values

('asif iqbal',all\_teachers\_id.nextval,'BBA',to\_date('2-6-2021','dd-mm- yyyy'),20000 );









**Course Table:**

create table all\_course (

course\_code int constraint pk\_all\_course primary key, course\_name varchar(50) not null, course\_prerequisite\_code varchar(10),

credit\_count int);

create sequence all\_course\_id

increment by 1

start with 1 nocycle

nocache;

CREATE INDEX course\_idx

ON all\_course(course\_name);

**Insertion:**

insert into all\_course values

(all\_course\_id.nextval,'Economy',null,3);

insert into all\_course values

(all\_course\_id.nextval,'physics',null,3);

insert into all\_course values

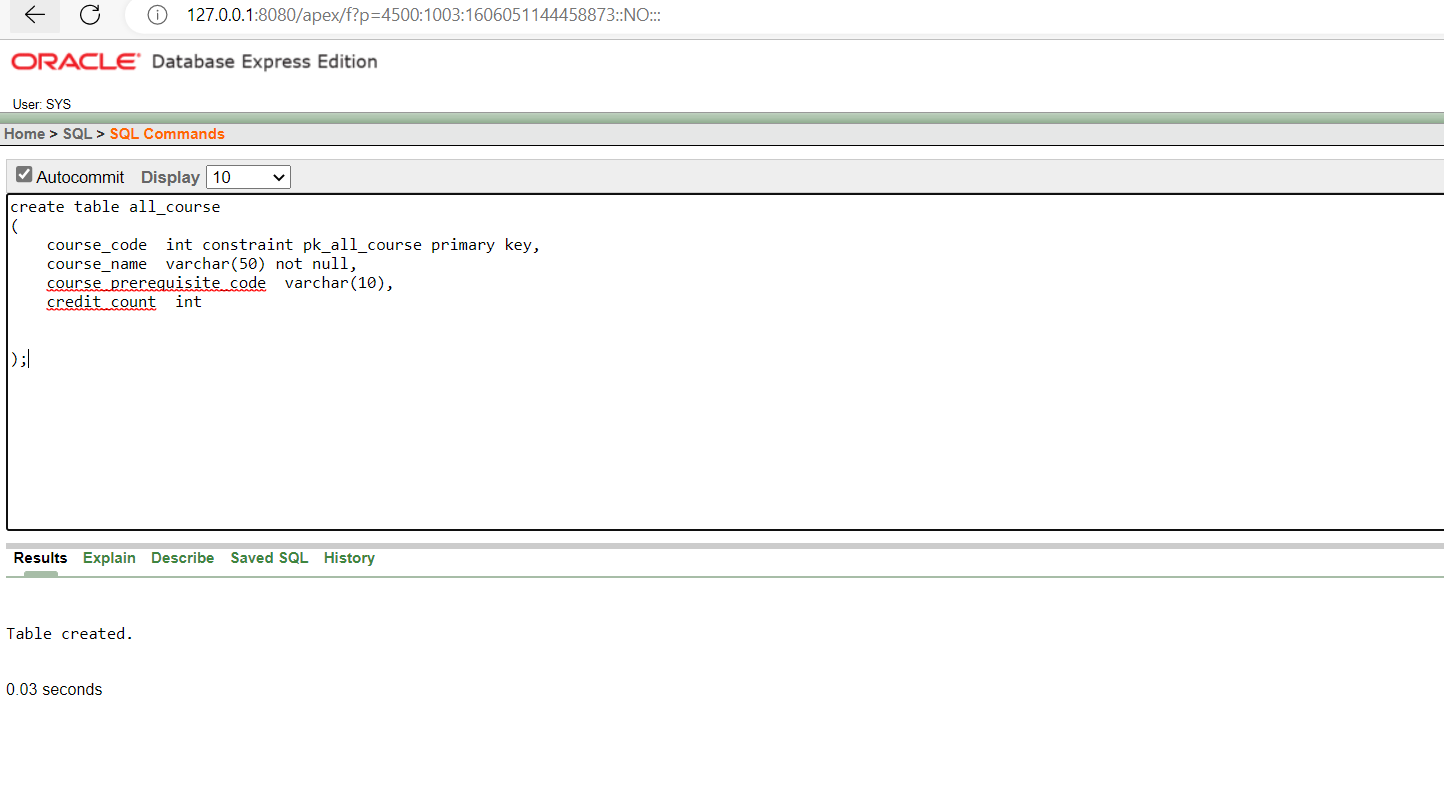
(all\_course\_id.nextval,'computer graphis',null,3);

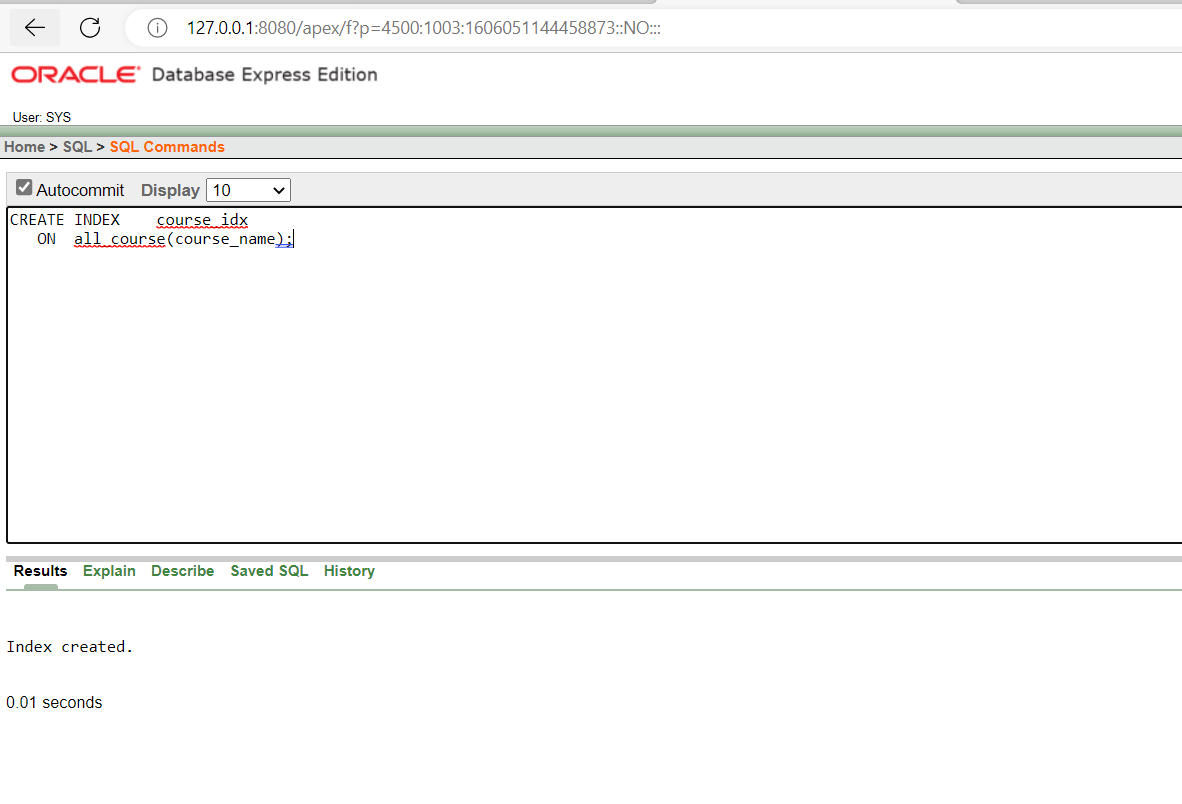
insert into all\_course values

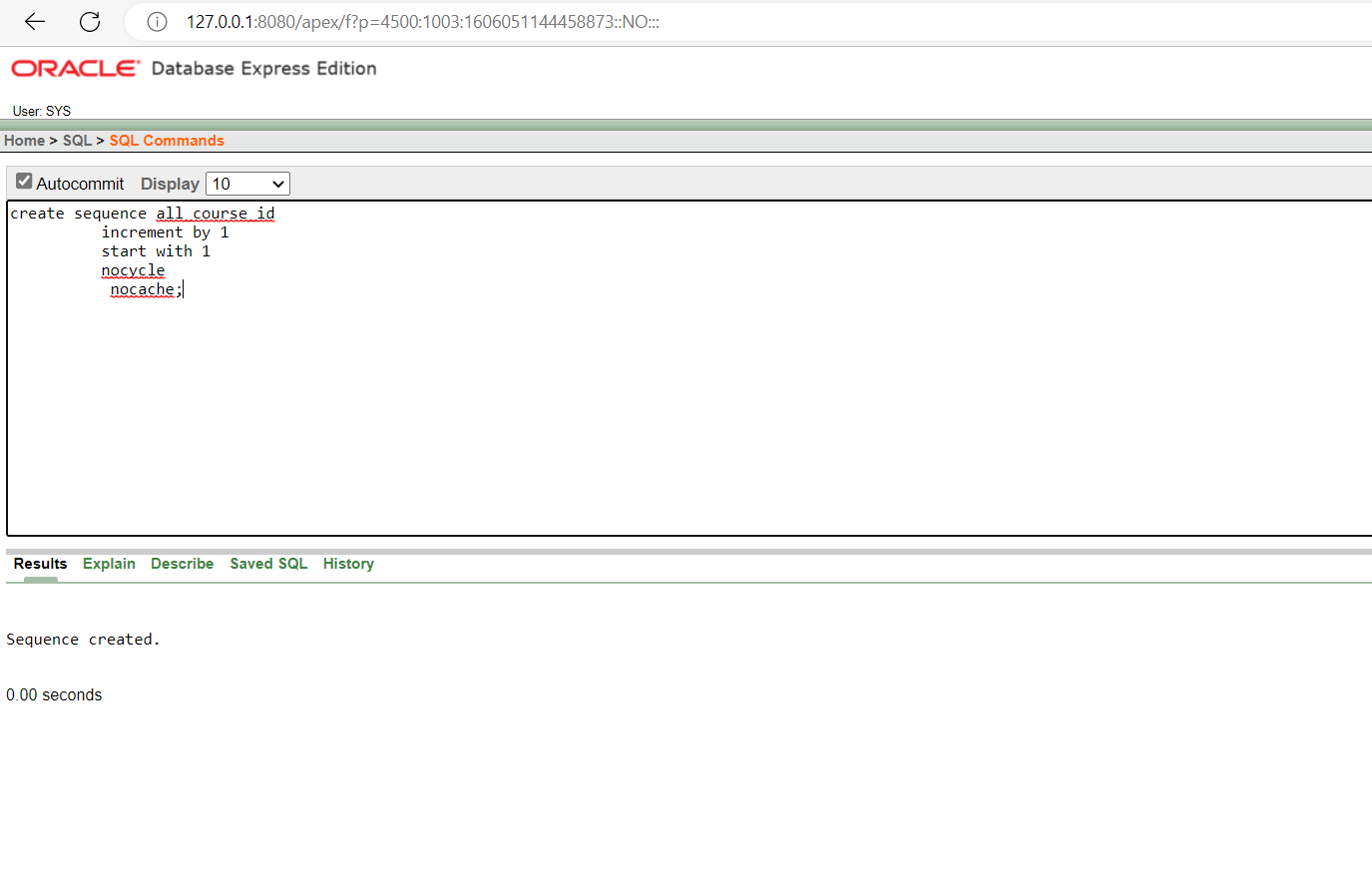
(all\_course\_id.nextval,'software enginnering',null,3);

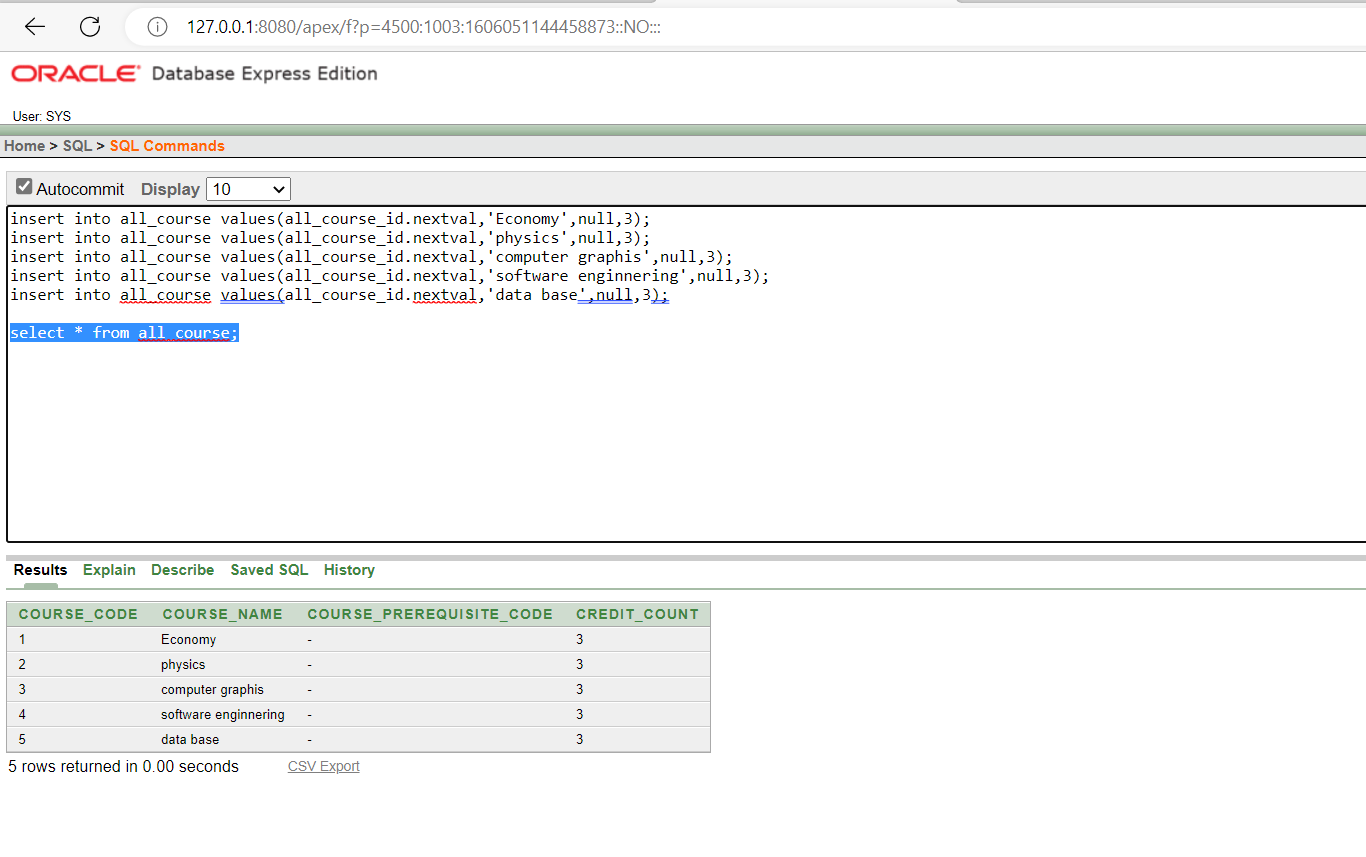
insert into all\_course values

(all\_course\_id.nextval,'data base',null,3);









**OSA Table:**

create table OSA

( student\_name varchar(50),

student\_id int constraint fk\_id references all\_student, osa\_serial int constraint pk\_OSA primary key,

action varchar(30),

cause varchar(100),

action\_date date);

create sequence osa\_serial

increment by 1

start with 1 nocycle

nocache;

CREATE INDEX osa\_idx

ON osa(action);

**Insertion:**

insert into osa values

#### ('Minhaz Ayon','1',osa\_serial.nextval,'Suspended','Ragging',to\_date('19-JUL- 22','dd-mm-yyyy'));

insert into osa values

#### ('Minhaz Ayon','1',osa\_serial.nextval,'Suspended','Ragging',to\_date('19-JUL- 22','dd-mm-yyyy'));

insert into osa values

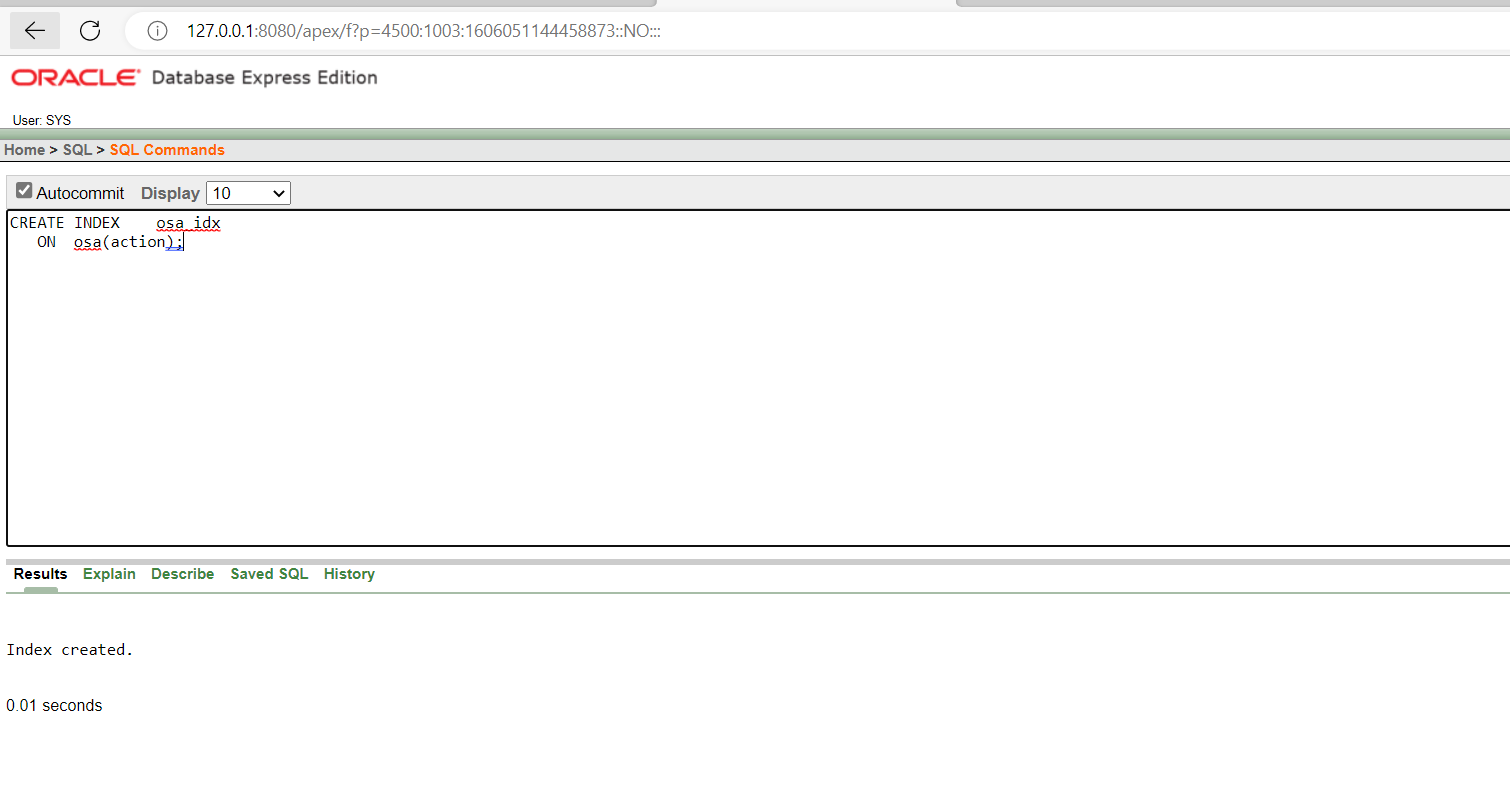
#### ('Prokhor Roy','2',osa\_serial.nextval,'Suspended','Longoverduepayments',to\_date( '19-JUL-22','dd-mm-yyyy'));

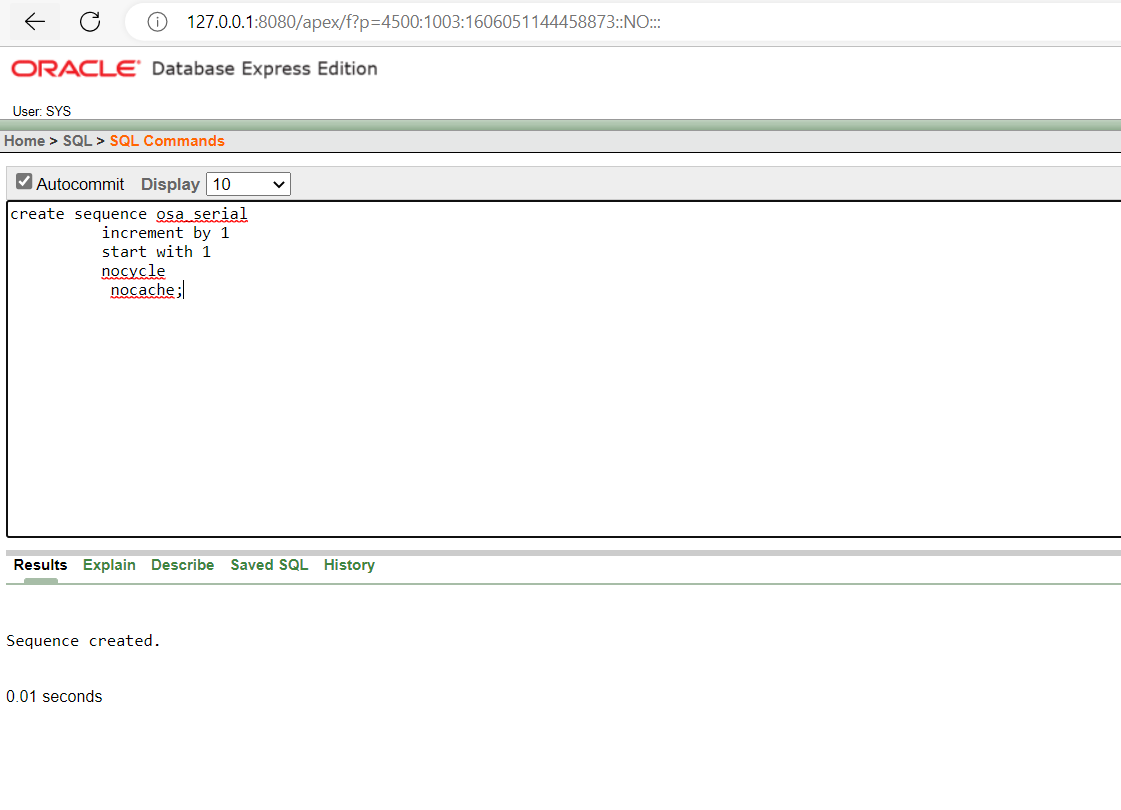
insert into osa values

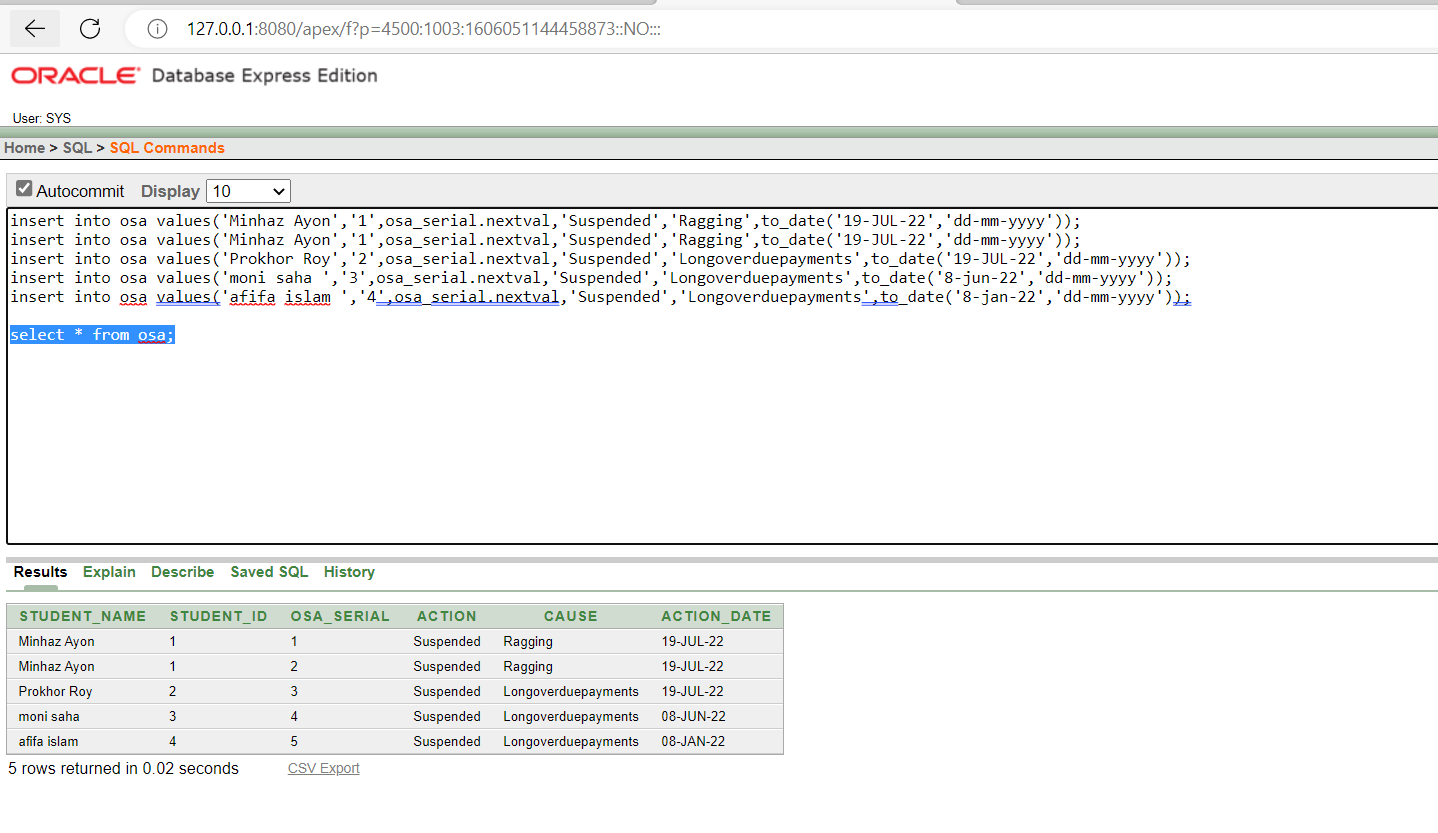
#### ('moni saha ','3',osa\_serial.nextval,'Suspended','Longoverduepayments',to\_date('8- jun-22','dd-mm-yyyy'));

insert into osa values

#### ('afifa islam ','4',osa\_serial.nextval,'Suspended','Longoverduepayments',to\_date('8- jan-22','dd-mm-yyyy'));







**Payment Table:**

create table Payment(

student\_name varchar(50) not null,

s\_id varchar2(8) constraint fk\_cid references all\_student(id), payment\_serial int constraint pk\_Payment primary key,

semester varchar(20), year int, payable\_amount int, amount\_paid int, payment\_date date

);

create sequence pay

increment by 1

start with 1 nocycle

nocache;

CREATE INDEX pay\_idx ON payment(semester);

**Insertion:**

insert into payment values

#### ( 'Shahriar Zaman', '2',pay.nextval,'Spring',2022,86400,86400,to\_date('19-NOV-21','dd-mm- yyyy'));

insert into payment values

( 'Waskiqure', '1',pay.nextval,'Spring',2022,86400,86400,to\_date('19-NOV-21','dd-mm- yyyy'));

insert into payment values

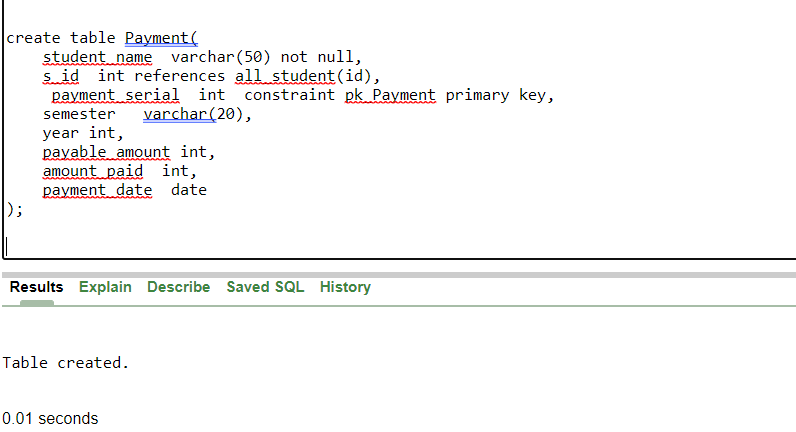
#### ( 'samsur nahar','3',pay.nextval,'Spring',2022,86400,86400,to\_date('22-NOV-21','dd-mm- yyyy'));

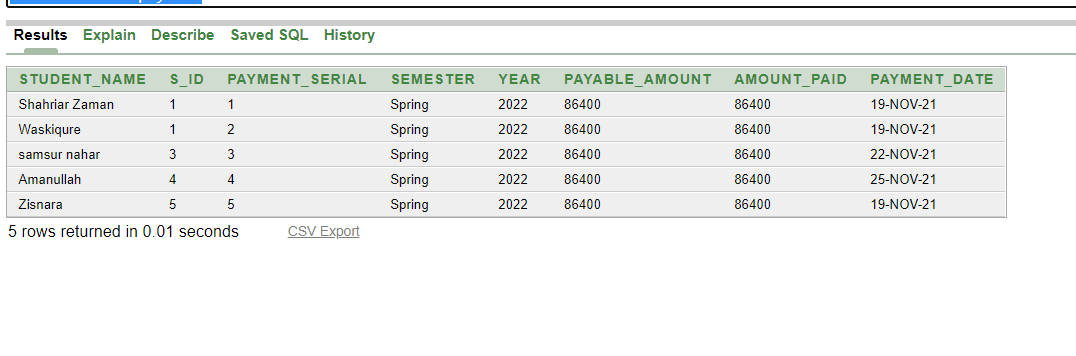
insert into payment values

#### ( 'Amanullah',' 4',pay.nextval,'Spring',2022,86400,86400,to\_date('25-NOV-21','dd-mm- yyyy'));

insert into payment values

( 'Zisnara', '5',pay.nextval,'Spring',2022,86400,86400,to\_date('19-NOV-21','dd-mm- yyyy'));





**Registration Table:**

create table registration

(

course\_code varchar(10) not null primary key,

course\_name varchar(50) not null,

course\_prerequisite\_code varchar(10),

);

**Insertion:**

insert into registration values('PHY1102', 'Physics-1 Lab', null);

insert into registration values('ENG1101', 'English-1', null);

insert into registration values('CSC1101', 'ICS', null);

insert into registration values('CSC1102', 'C++', null);

insert into registration values('CSC1103', 'C++ Lab', null);

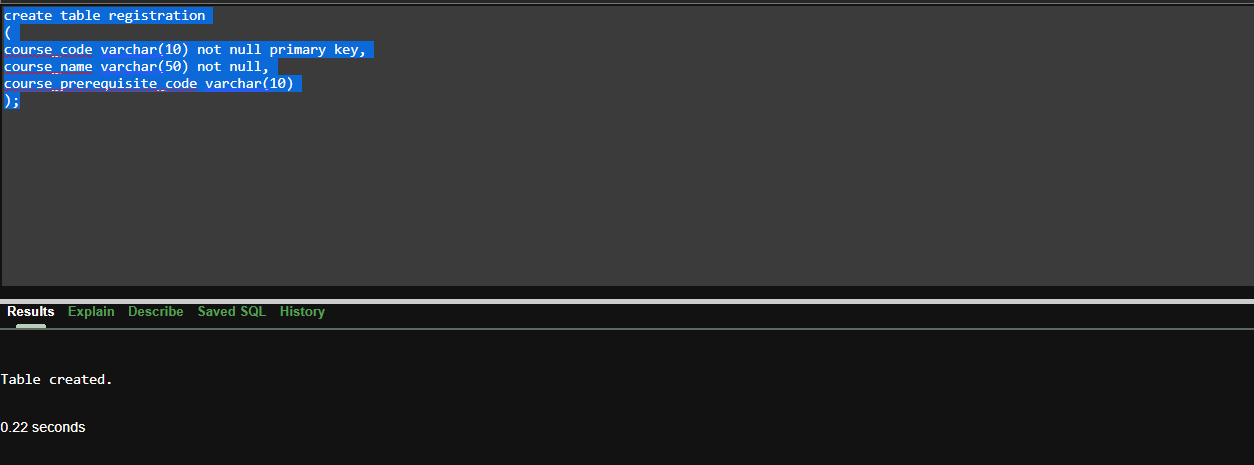
insert into registration values('CSC1204', 'Discrete Math', 'CSC1102');

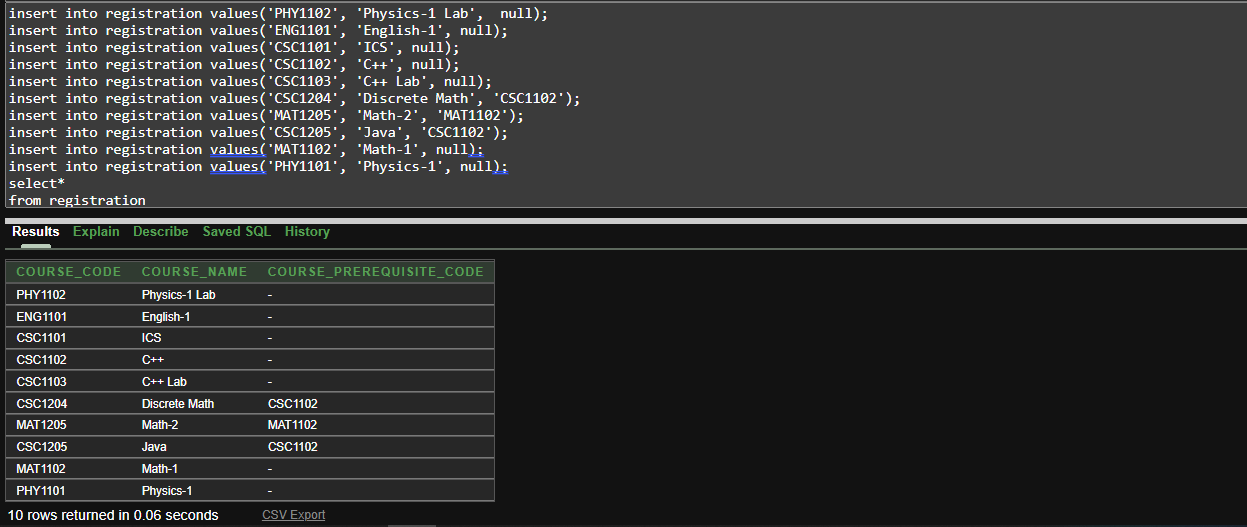
insert into registration values('MAT1205', 'Math-2', 'MAT1102');

insert into registration values('CSC1205', 'Java', 'CSC1102');

insert into registration values('MAT1102', 'Math-1', null);

insert into registration values('PHY1101', 'Physics-1', null);





**CSRT Table:**

create table CSRT (

serial int constraint pk\_CSRT primary key,

course\_code int references all\_course(course\_code), course\_name varchar(50),

teacher\_id int references all\_teachers(ID), teacher\_name varchar(50),

student\_id int not null references all\_student(id), student\_name varchar(50),

marks int

);

create sequence cst

increment by 1

start with 1

nocycle

nocache;

**Insertion:**

insert into CSRT values

(1, 1, 'economy', 1, 'Nazmul Hossain', 1, 'waskiqure', 90);

insert into CSRT values

(2, 2, 'physics', 2, 'Najia Alfaz', 2, 'shahriar zaman', 80);

insert into CSRT values

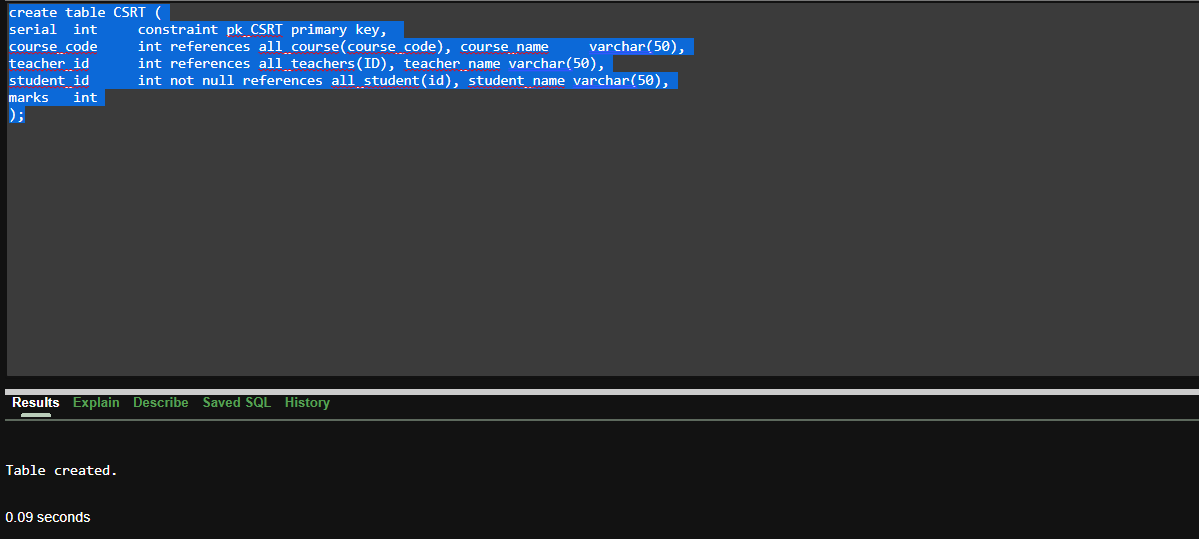
(3, 3, 'computer graphics', 3, 'kuwsur imran', 3, 'sumsun nahar', 85);

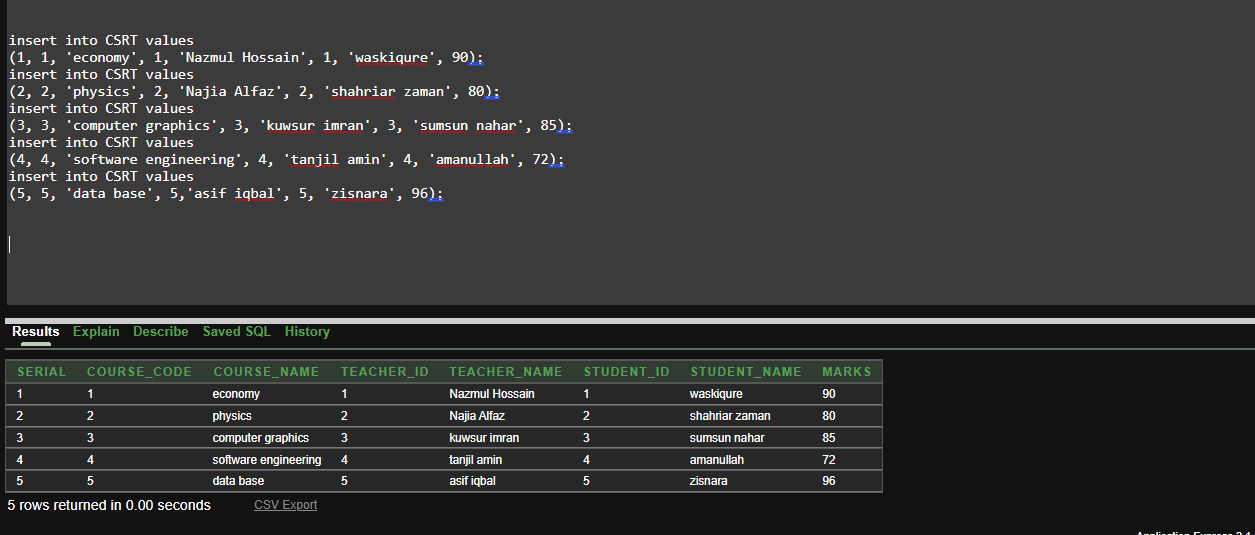
insert into CSRT values

(4, 4, 'software engineering', 4, 'tanjil amin', 4, 'amanullah', 72);

insert into CSRT values

(5, 5, 'data base', 5,'asif iqbal', 5, 'zisnara', 96);





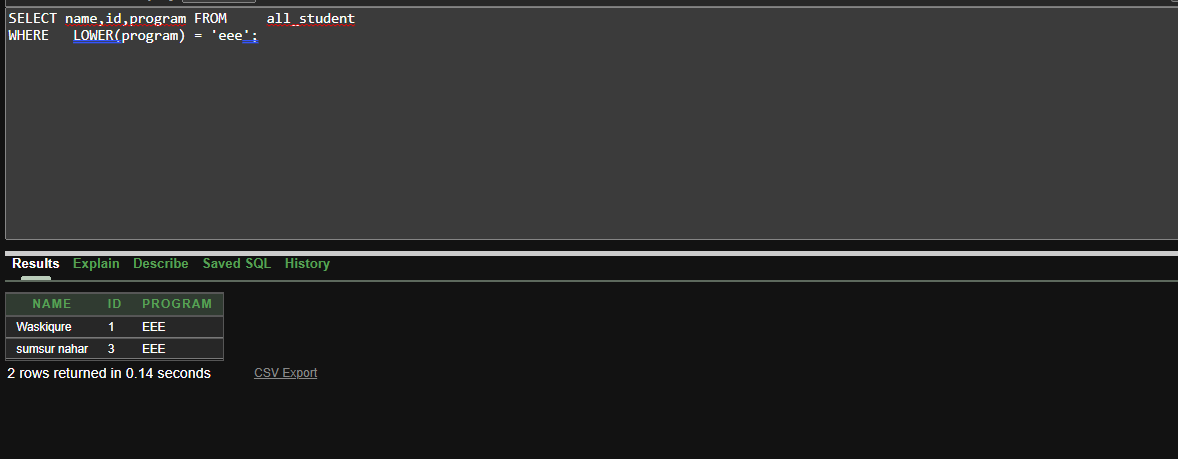
**Query Writing (Write down the question and also the answer)**

# **Single-row function**

##### **Display student name,id and program for program eee.**

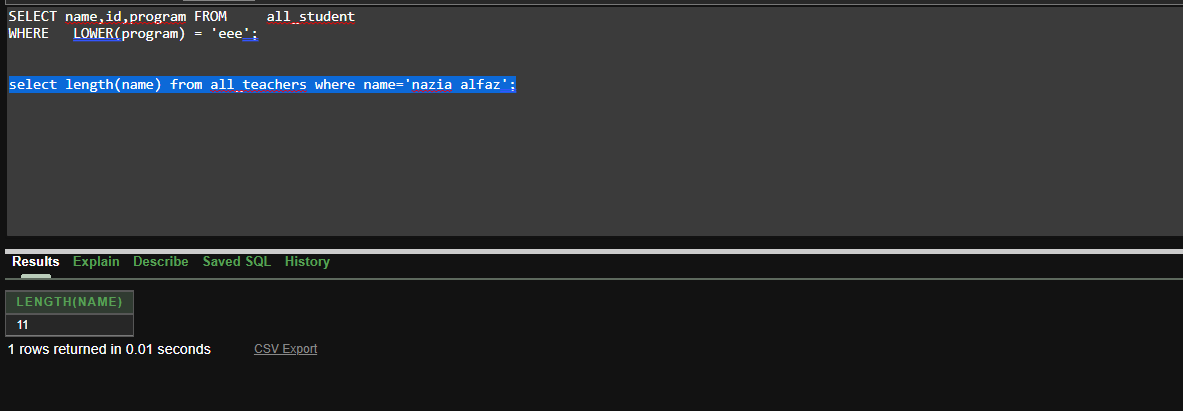
SELECT name,id,program FROM all\_student

WHERE LOWER(program) = 'eee';



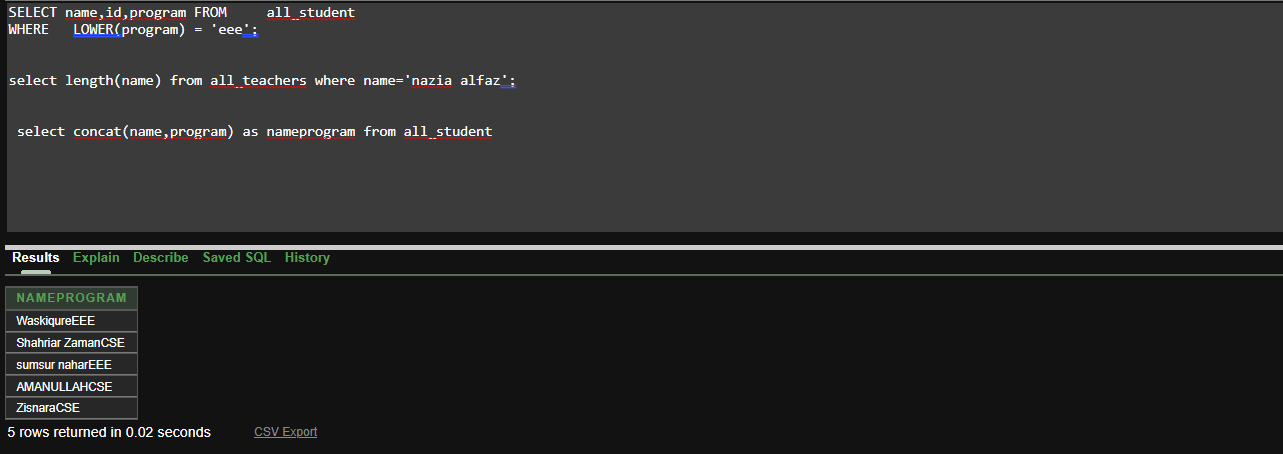
##### **Find the length for name nazia alfaz from teacher table**

select length(name) from all\_teachers where name='nazia alfaz';



1. **concat name,program as nameprogram from all\_student**

select concat(name,program) as nameprogram from all\_student



**Group-row function**

### **Find maximum salary for teacher which domain is csc**

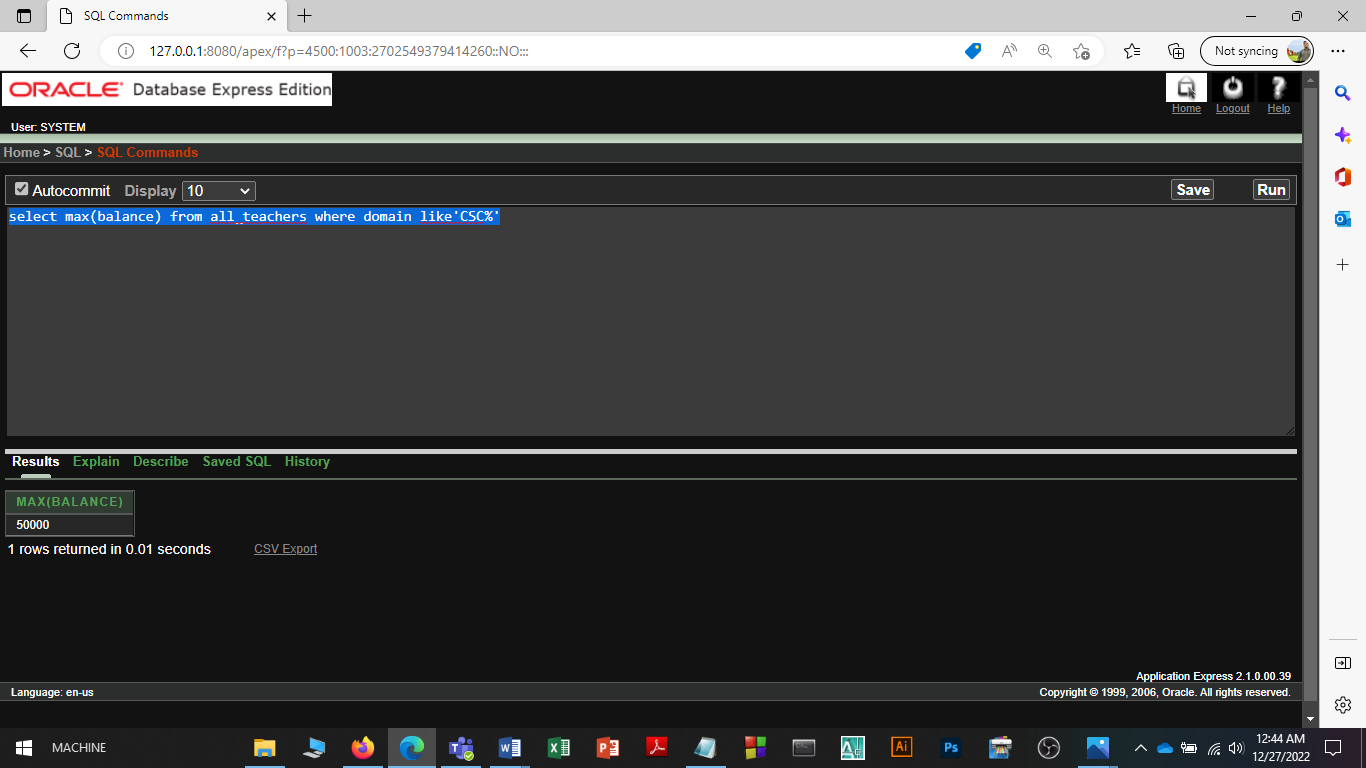
select max(balance) from all\_teachers where domain like'CSC%'

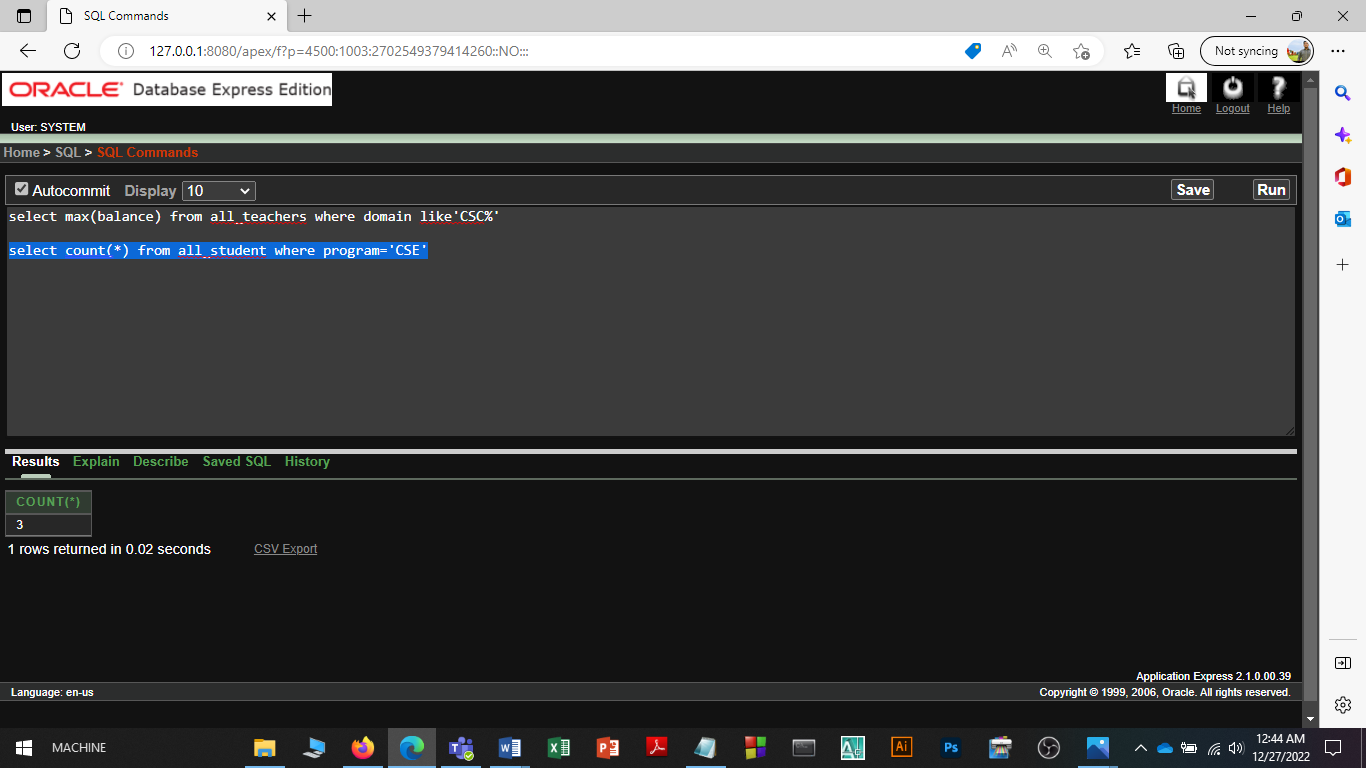
### **How many row for cse program in all\_student table**

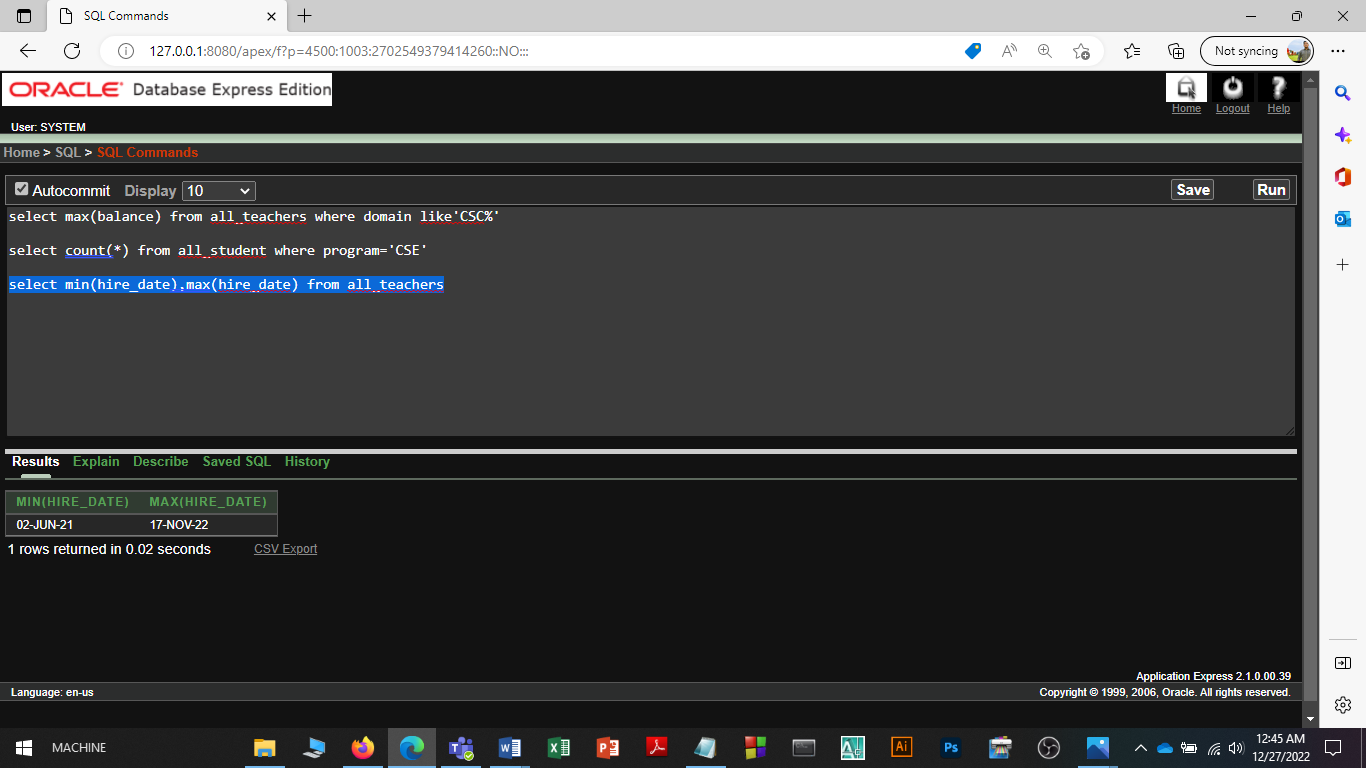
select count(\*) from all\_student where program='CSE'

##### **Find the minimum and maximum hire\_date from all\_teachers table**

select min(hire\_date),max(hire\_date) from all\_teachers



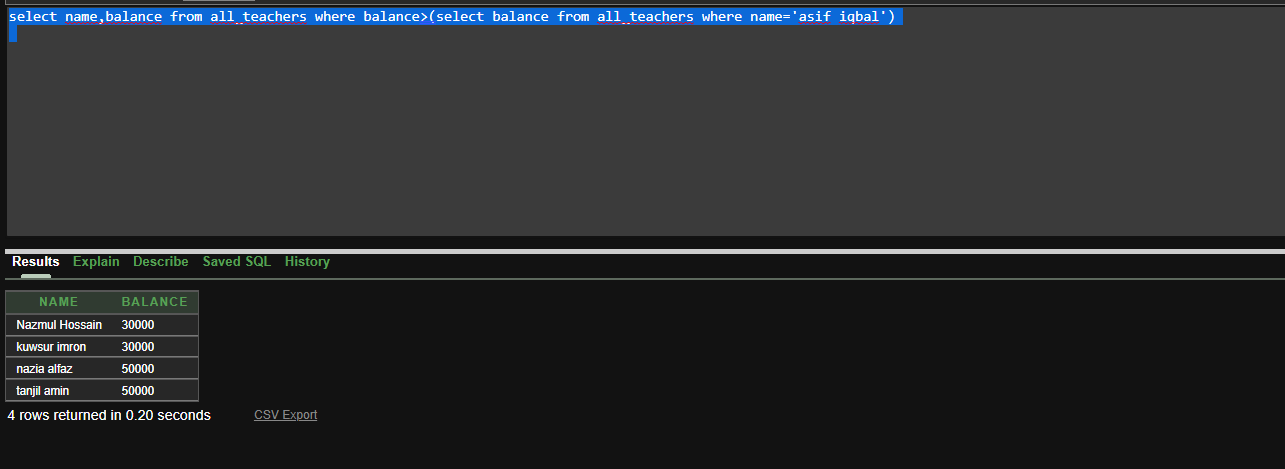




## Subquery

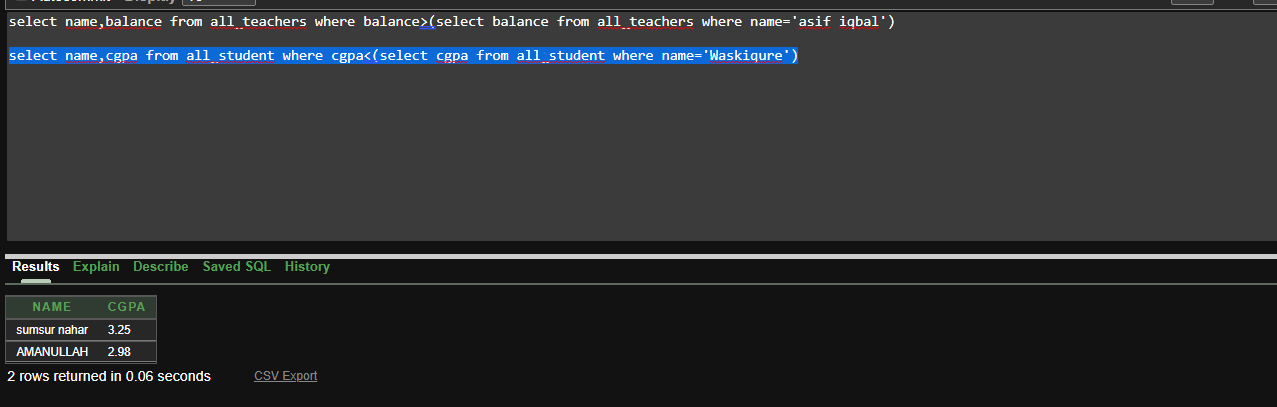
##### **Which teachers have a salary greater than 'asif iqbalâ€™ salary?**

select name,balance from all\_teachers where balance>(select balance from all\_teachers where name='asif iqbal')



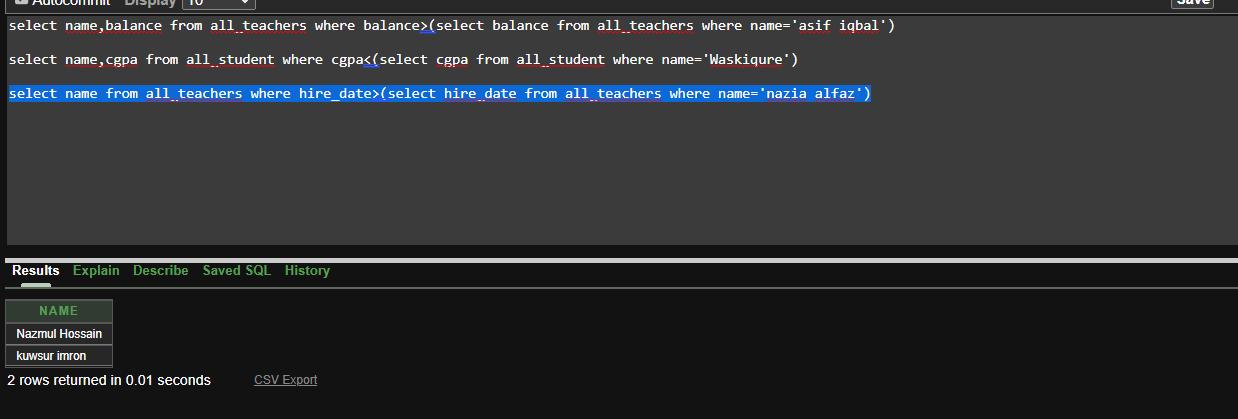
### **which student have a cgpa less than 'waskiqure' cgpa?**

select name,cgpa from all\_student where cgpa<(select cgpa from all\_student where name='Waskiqure')



### **display name who join after nazia alfaz**

select name from all\_teachers where hire\_date>(select hire\_date from all\_teachers where name='nazia alfaz')



## **Joining**

##### **dispaly name,id,payable\_amount of each student**

select s.name,s.id,p.payable\_amount from all\_student s ,payment p where s.id=p.s\_id

##### **Write a query to display the name,**

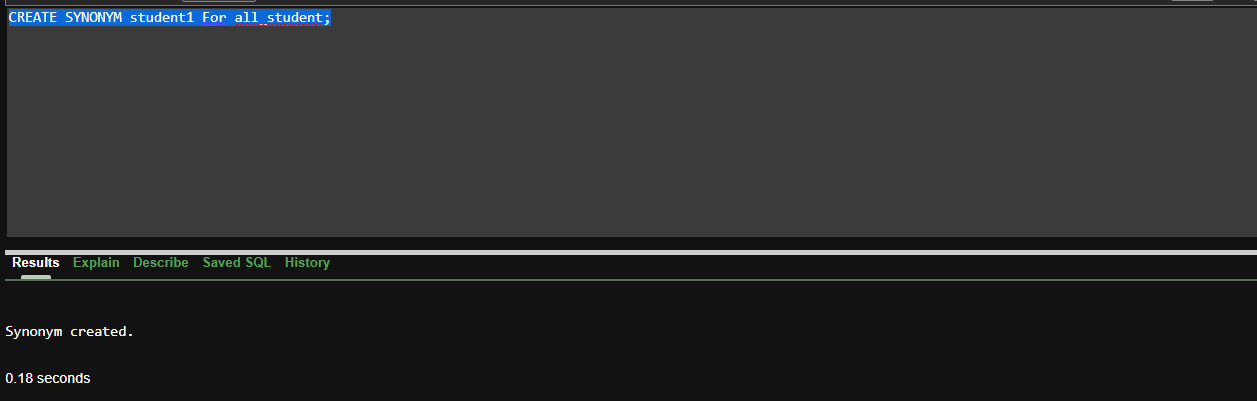
**payment\_serial, semester and payment\_date for all student.**

select s.name,p.payment\_serial,p.semester,p.payment\_date from all\_student s ,payment p where s.id=p.s\_id

## **Synonym**

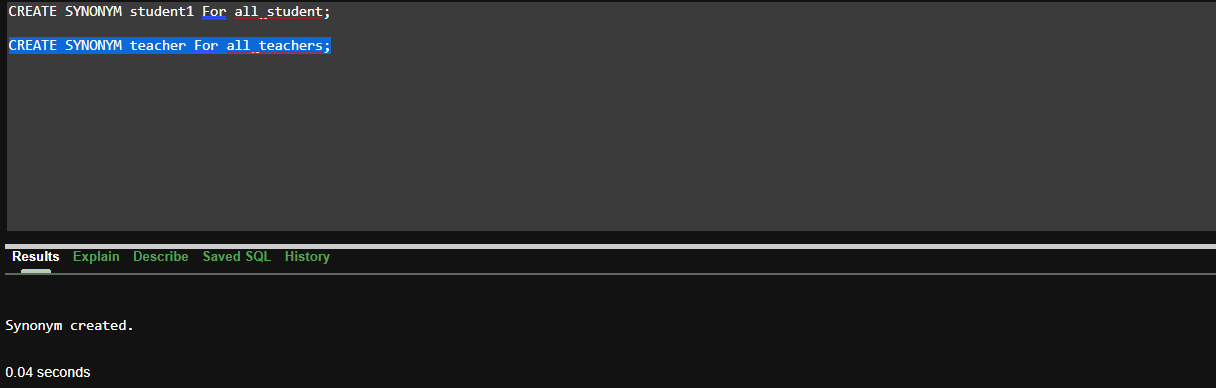
### **Create a shortened name for the all\_student view**

CREATE SYNONYM student1 For all\_student;



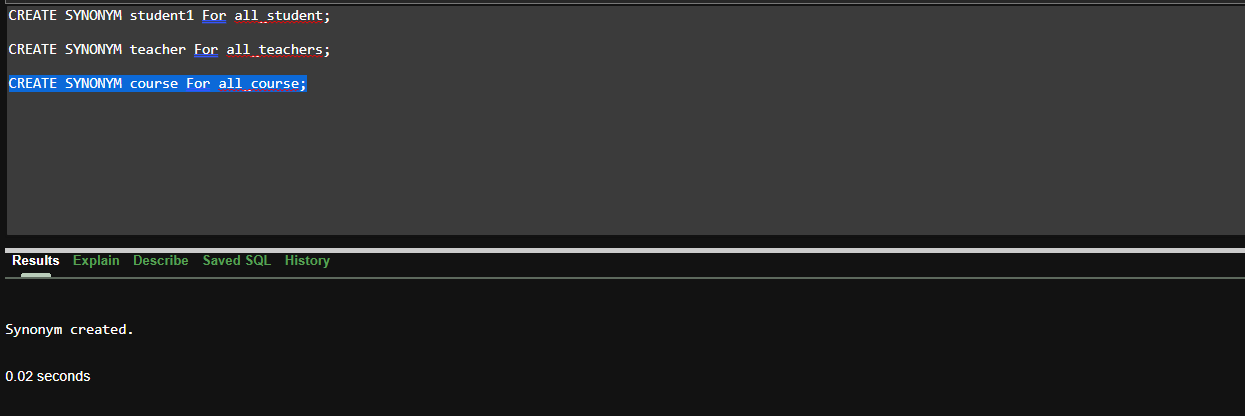
### **Create a shortened name for the all\_teachers view**

CREATE SYNONYM teacher For all\_teachers;



### **Create a shortened name for the all\_course view**

CREATE SYNONYM course For all\_course;



**PL/SQL**

**Function**

1**.Create a function that returns the total number of student.**

CREATE OR REPLACE FUNCTION totalstudent

RETURN number AS

total number(2) := 0;

BEGIN

SELECT count(\*) into total

FROM all\_student;

RETURN total;

END;

/

DECLARE

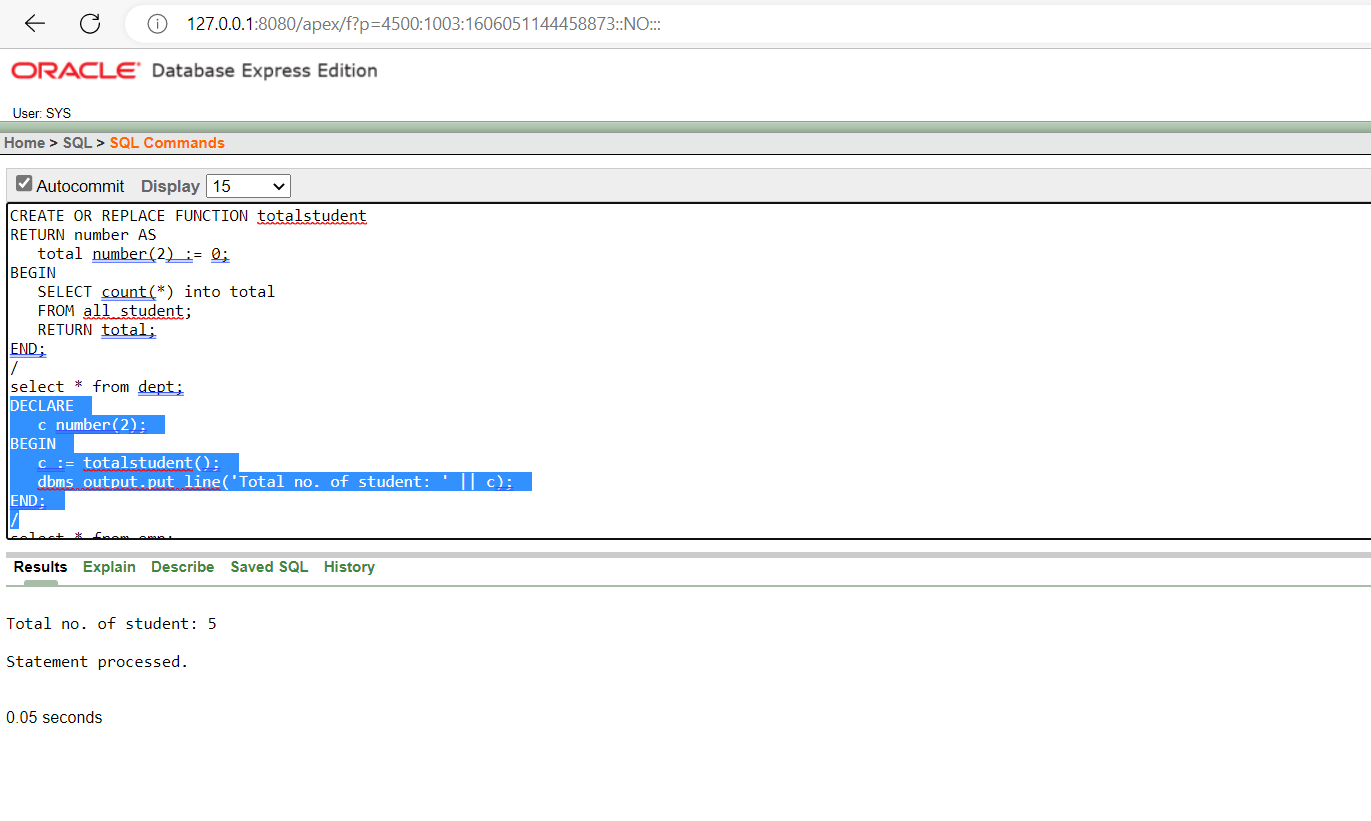
c number(2);

BEGIN

c := totalstudent();

dbms\_output.put\_line('Total no. of student: ' || c);

END; /



2.**Create a function that returns the total number of teacher.**

CREATE OR REPLACE FUNCTION totalteacher

RETURN number AS

total number(2) := 0;

BEGIN

SELECT count(\*) into total

FROM all\_teachers;

RETURN total;

END;

/

DECLARE

c number(2);

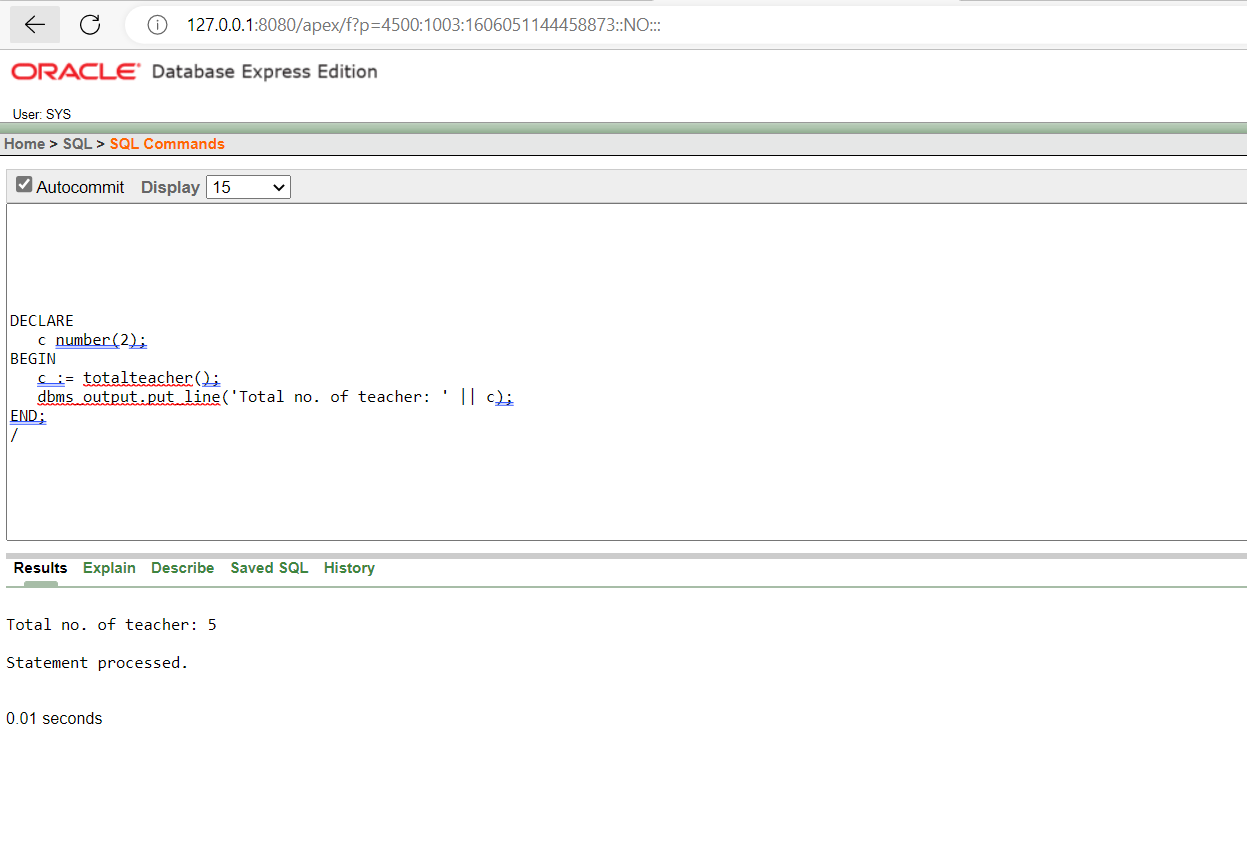
BEGIN

c := totalteacher();

dbms\_output.put\_line('Total no. of teacher: ' || c);

END;

/



3.**Create a function that returns the total number of suspended student.**

CREATE OR REPLACE FUNCTION totalstudent

RETURN number AS

total number(2) := 0;

BEGIN

SELECT count(\*) into total

FROM osa where action='Suspended';

RETURN total;

END;

/

DECLARE

c number(2);

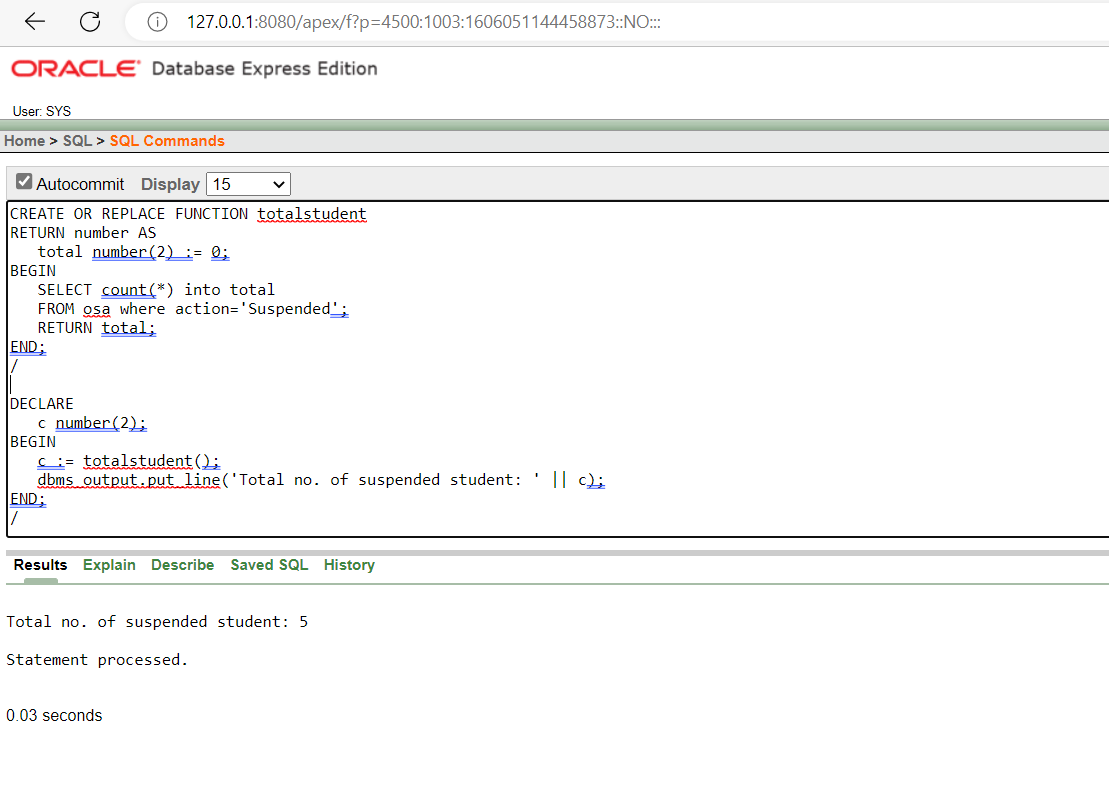
BEGIN

c := totalstudent();

dbms\_output.put\_line('Total no. of suspended student: ' || c);

END;

/



**Record:**

**1.Create a record that can output the id, name, cgpa, and admission date of the student whose Id is 4**

declare

stu\_rec all\_student%rowtype;

begin

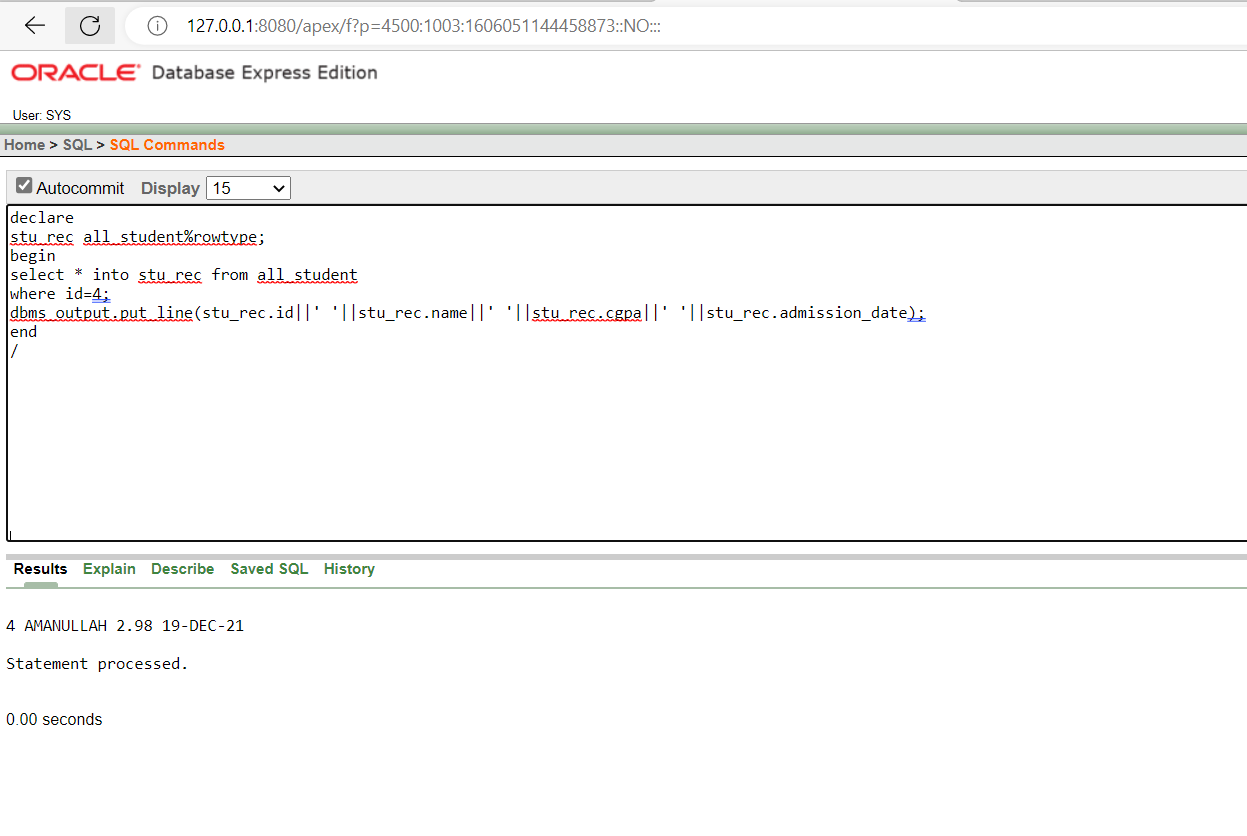
select \* into stu\_rec from all\_student

where id=4;

dbms\_output.put\_line(stu\_rec.id||' '||stu\_rec.name||' '||stu\_rec.cgpa||' '||stu\_rec.admission\_date);

end

/



**2.Create a record that can output the id, name, domain, and hire date of the teacher whose Id is 2**

declare

teacher\_rec all\_teachers%rowtype;

begin

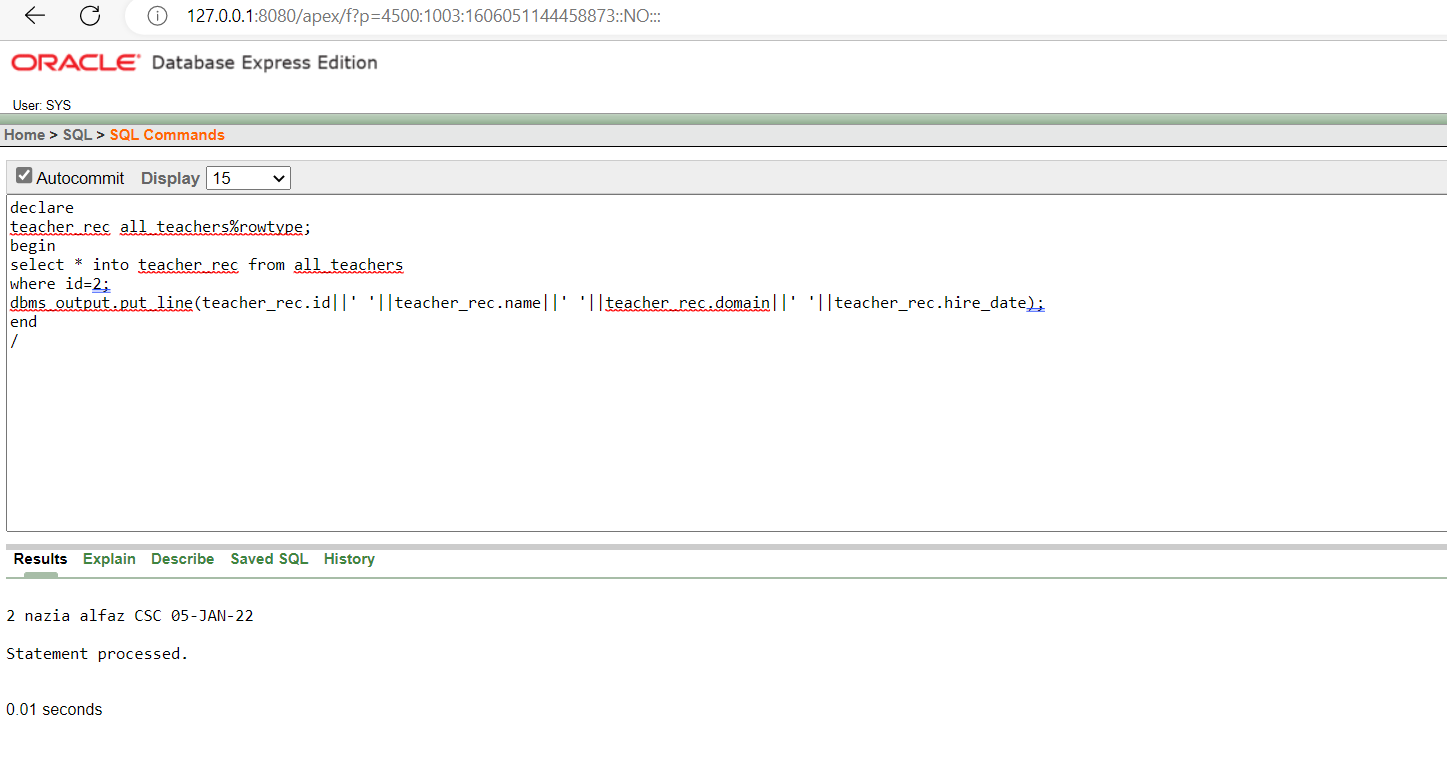
select \* into teacher\_rec from all\_teachers

where id=2;

dbms\_output.put\_line(teacher\_rec.id||' '||teacher\_rec.name||' '||teacher\_rec.domain||' '||teacher\_rec.hire\_date);

end

/



**3.Create a record that can output the id, name, domain, and hire date of the course whose Id is 2**

declare

b all\_course%rowtype;

begin

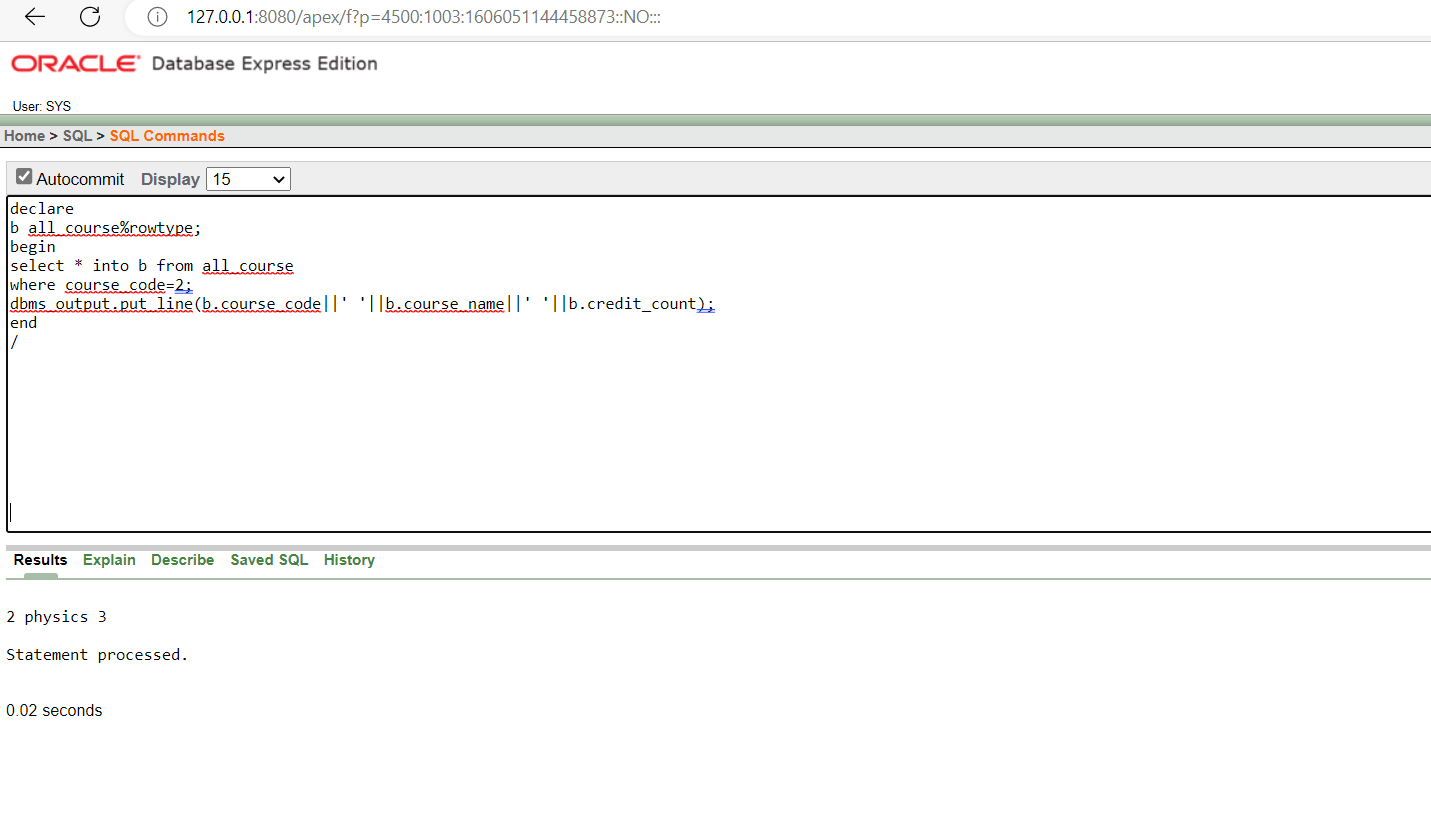
select \* into b from all\_course

where course\_code=2;

dbms\_output.put\_line(b.course\_code||' '||b.course\_name||' '||b.credit\_count);

end

/



**Package:**

1.**Create a package that contains a procedure which can display the name,cgpa whose**

**id is passed as its parameter**

CREATE OR REPLACE PACKAGE BODY stu\_pack AS

PROCEDURE display\_name(e all\_student.id%TYPE) IS

name all\_student.id%TYPE;

BEGIN

SELECT name INTO name

FROM all\_student

WHERE id=e;

dbms\_output.put\_line('student Name: '|| name);

END display\_name;

PROCEDURE display\_cgpa(e1 all\_student.id%TYPE) IS

cgpa all\_student.id%TYPE;

BEGIN

SELECT cgpa INTO cgpa

FROM all\_student

WHERE id=e1;

dbms\_output.put\_line('student cgpa: '|| cgpa);

END display\_cgpa;

END stu\_pack;

/

CREATE OR REPLACE PACKAGE stu\_pack AS

PROCEDURE display\_name(e all\_student.id%type);

PROCEDURE display\_cgpa(e1 all\_student.id%type);

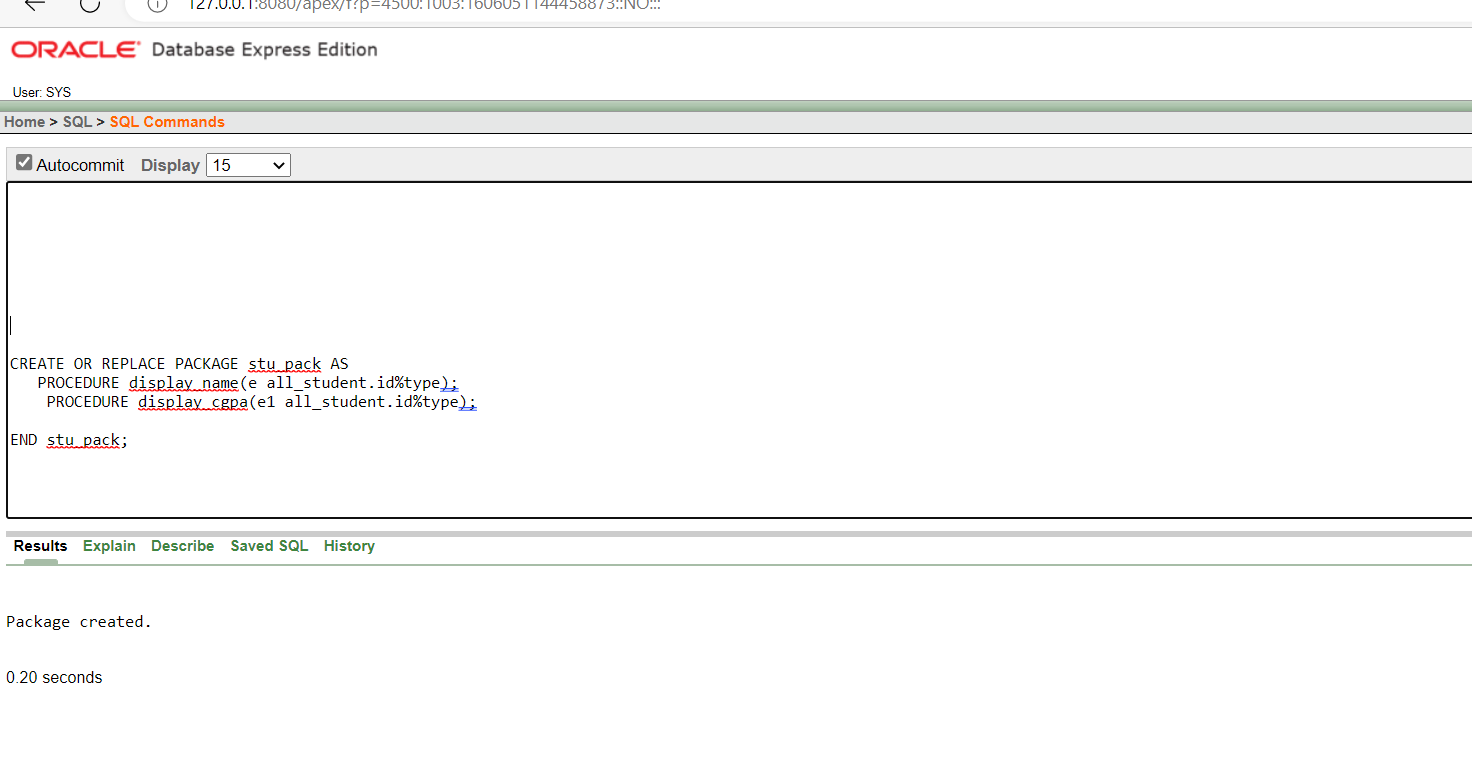
END stu\_pack;

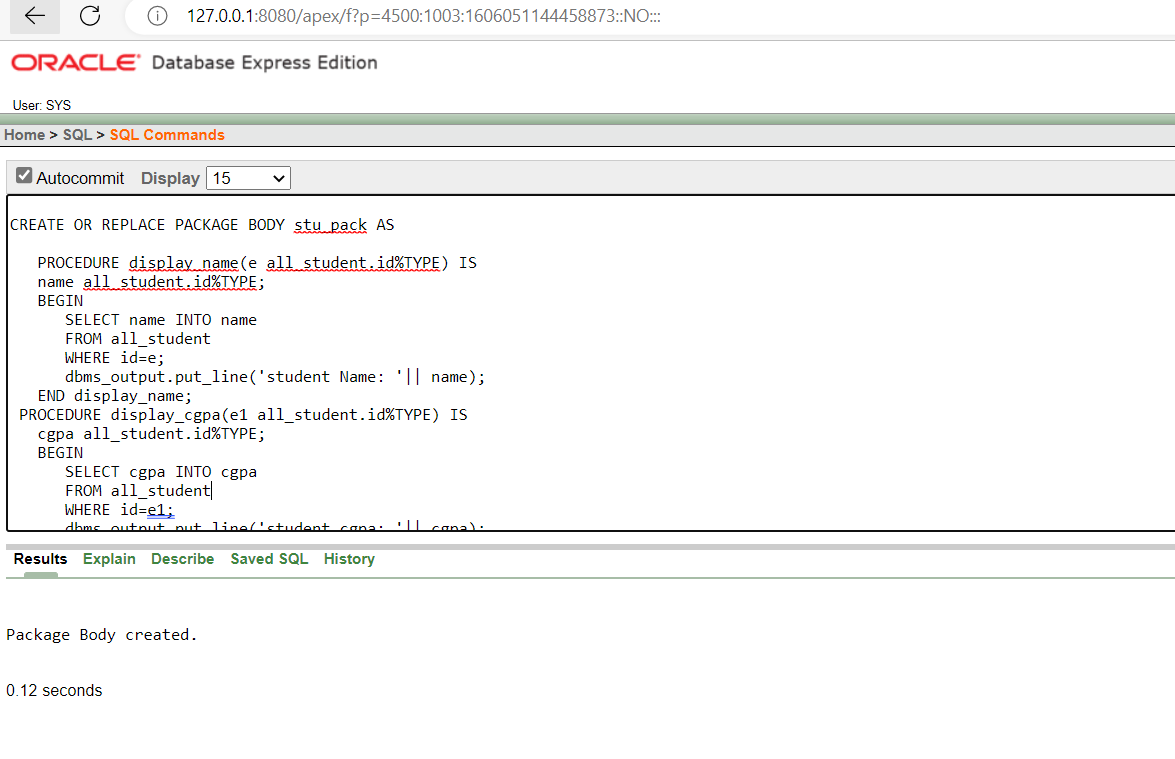
begin

stu\_pack.display\_name('2');

stu\_pack.display\_cgpa('2');

end





2**.Create a package that contains a procedure which can display the name,domain whose**

**id is passed as its parameter from teacher table**

CREATE OR REPLACE PACKAGE BODY tea\_pack AS

PROCEDURE display\_name(e all\_teachers.id%TYPE) IS

name all\_teachers.name%TYPE;

BEGIN

SELECT name INTO name

FROM all\_teachers

WHERE id=e;

dbms\_output.put\_line('teachers Name: '|| name);

END display\_name;

PROCEDURE display\_domain(e1 all\_teachers.id%TYPE) IS

domain all\_teachers.domain%TYPE;

BEGIN

SELECT domain INTO domain

FROM all\_teachers

WHERE id=e1;

dbms\_output.put\_line('teachers domain: '|| domain);

END display\_domain;

END tea\_pack;

/

CREATE OR REPLACE PACKAGE tea\_pack AS

PROCEDURE display\_name(e all\_teachers.id%type);

PROCEDURE display\_domain(e1 all\_teachers.id%type);

END tea\_pack;

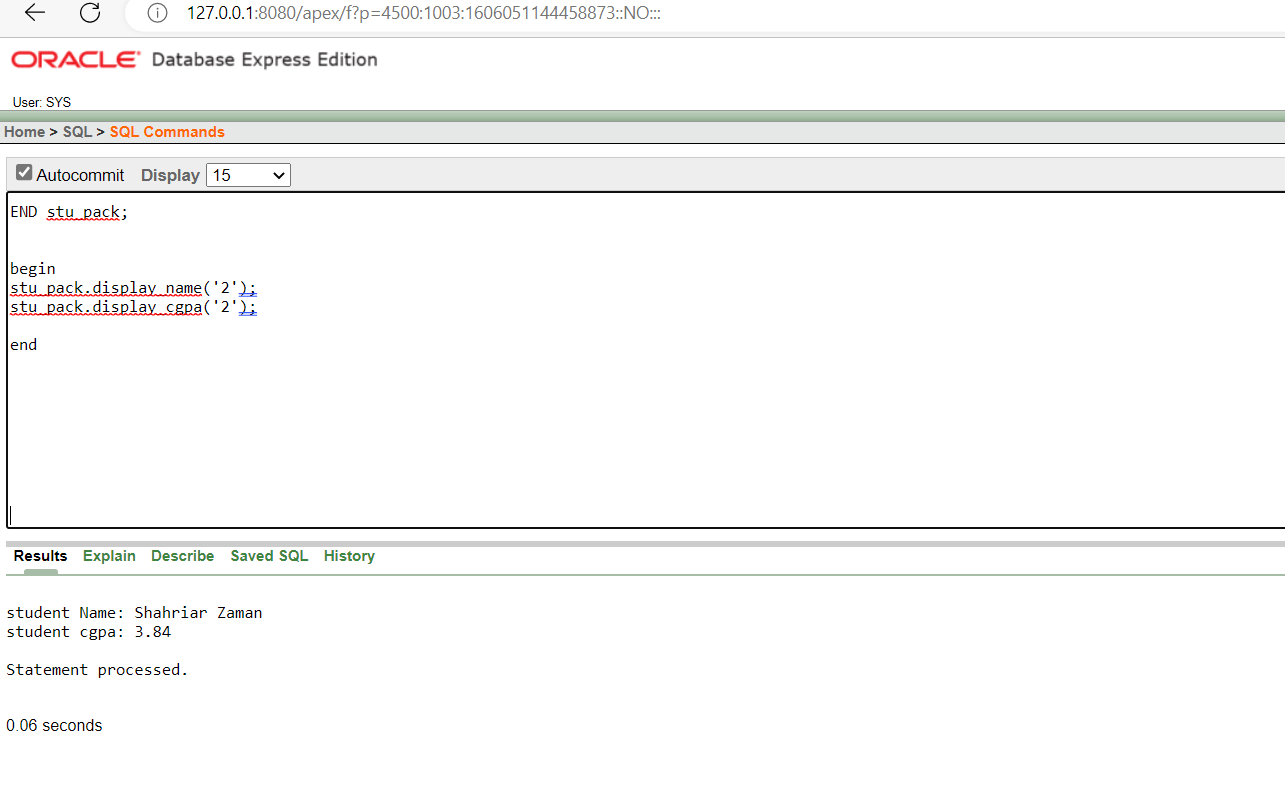
select \* from all\_teachers

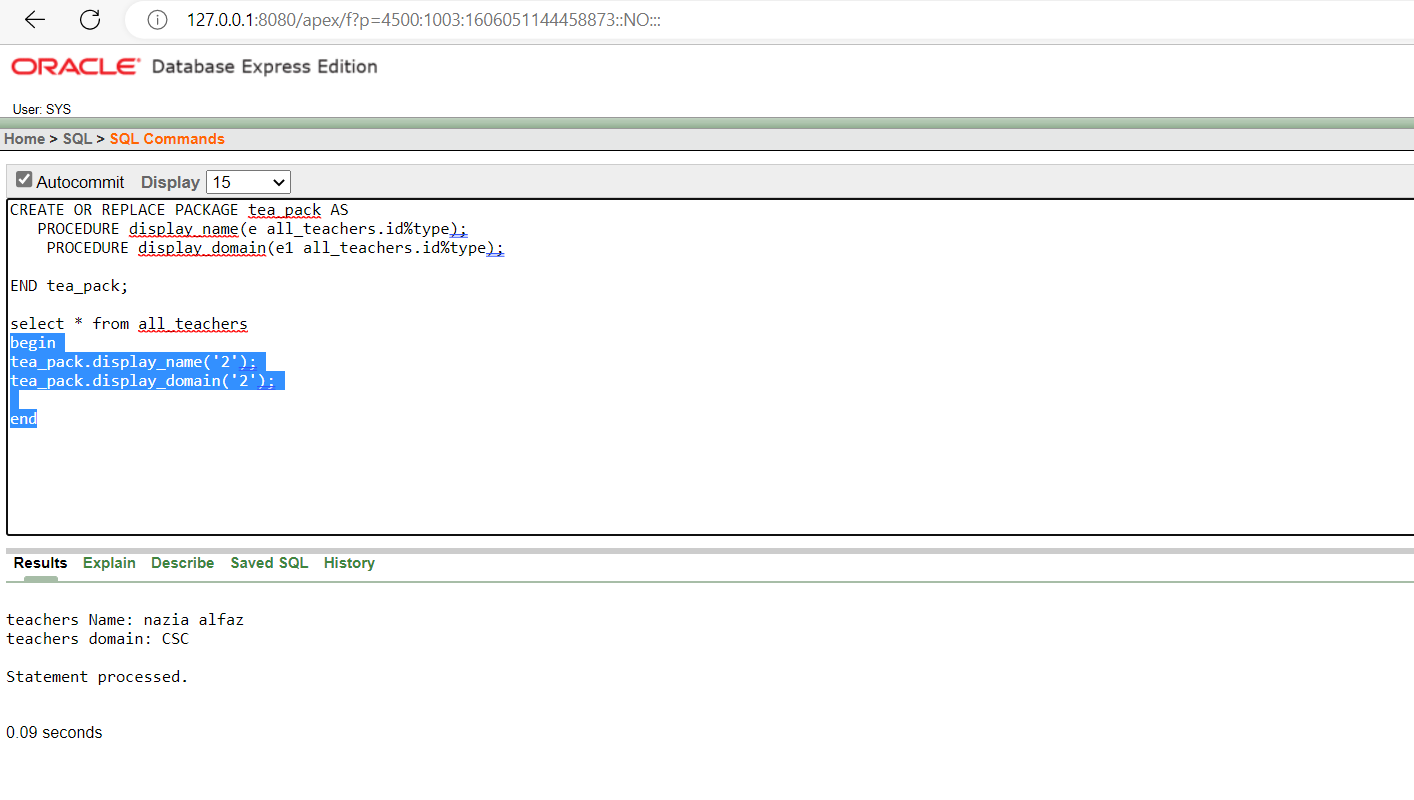
begin

tea\_pack.display\_name('2');

tea\_pack.display\_domain('2');

end





**Trigger:**

1. **Display a trigger that give reminder when banlance is 0.**

create or replace trigger expenditure\_recorder

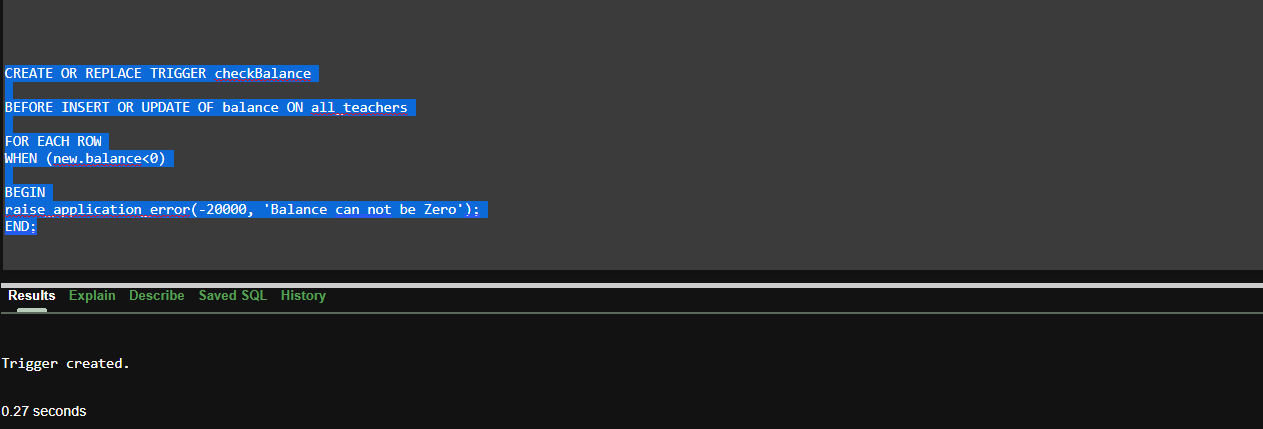
before update on all\_teachers

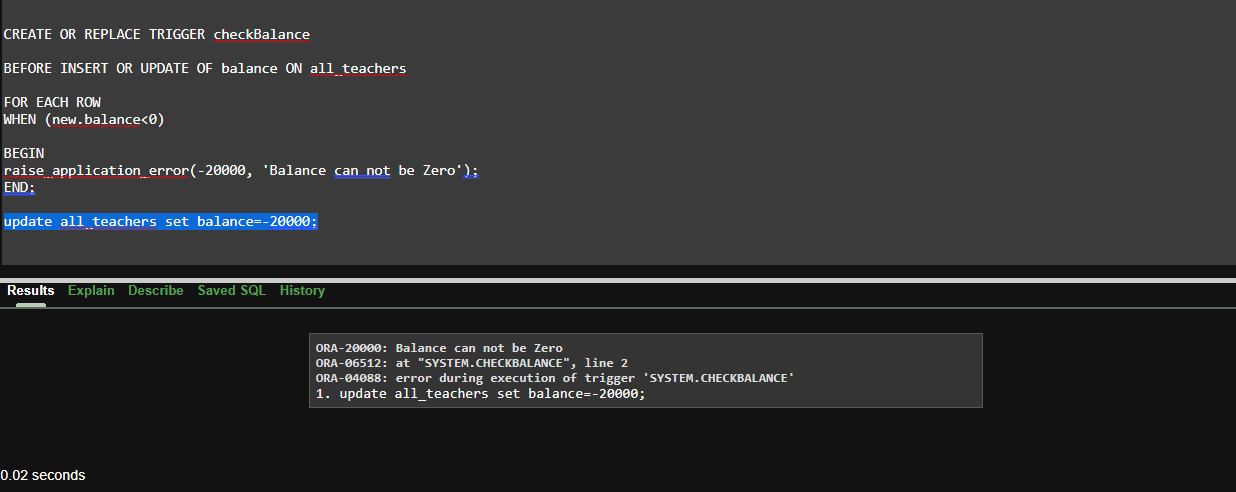
for each row

begin

update this\_semester\_report set expenditure=expenditure+:new.balance-:old.balance;

end;





1. **CGPA is between 0 to 4**

create or replace trigger marks\_value\_fixer

before update of marks on csrt

for each row

begin

if :new.marks>100 then

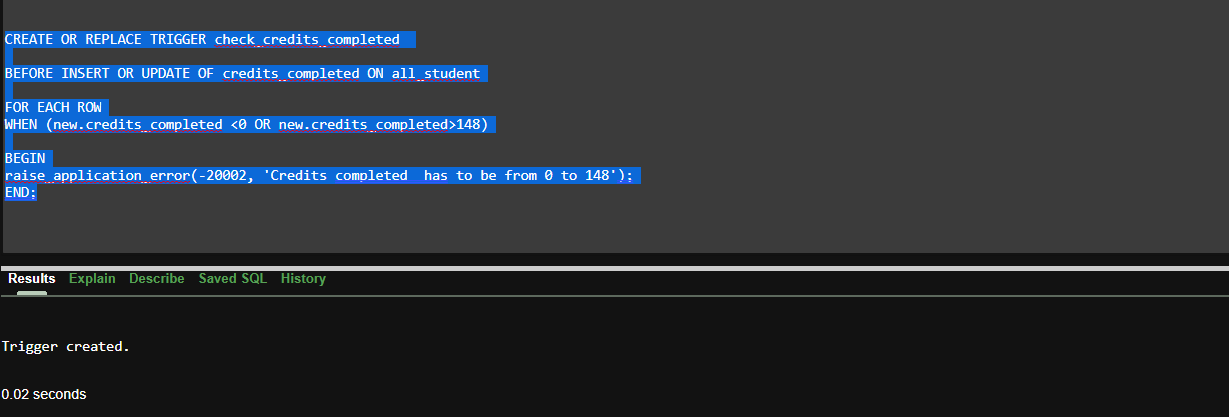
:new.marks:=100;

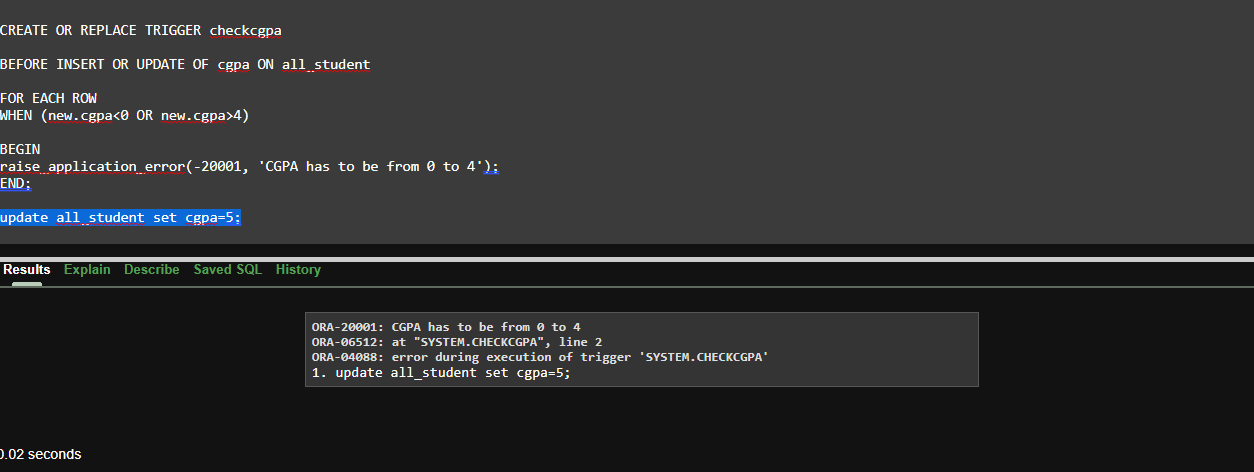
elsif :new.marks<0 then

:new.marks:=0;

end if;

end;





1. **If credit complete is lower or higher than higher than 0 and 148.**

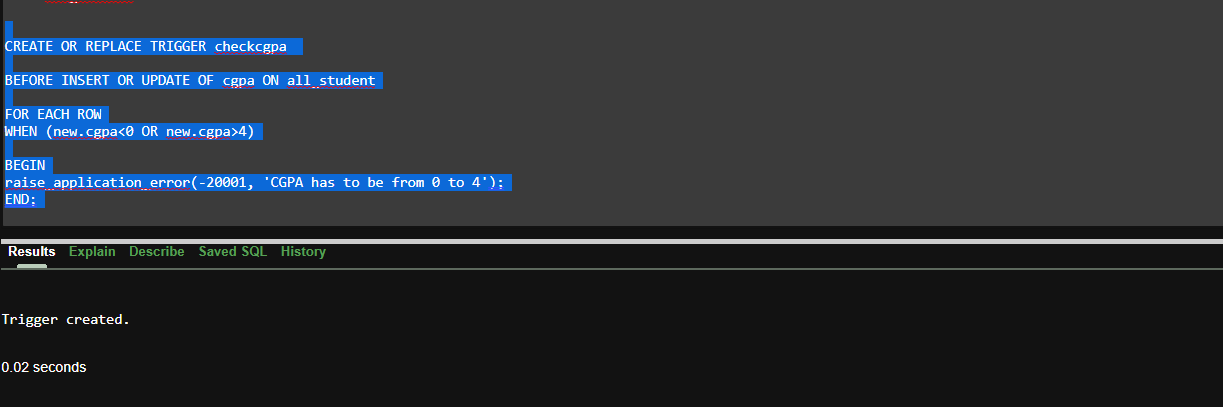
create or replace trigger result\_log

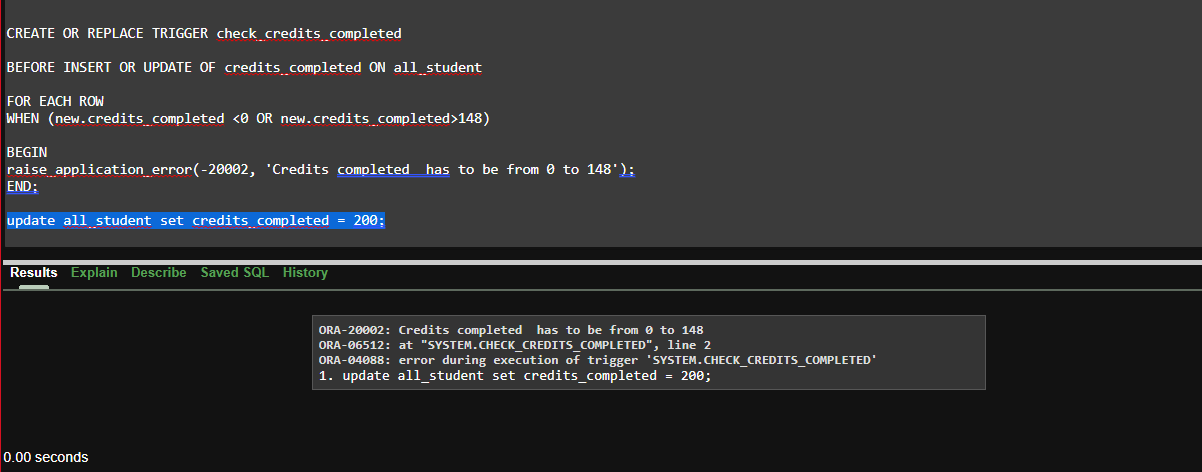
after update on csrt

begin

insert into task\_log values((select id from loggedinTable),'Updating results',sysdate,SYSTIMESTAMP);

end;





**Conclusion:** In our project, there might be some minor problem. We will try to configure them if needed and we will provide more correct database for the software that we are going to done. In final term we implemented the PL/SQL in the project which we have implemented in our mid term. Here, we cannot provide a fully connected user interface for some sort of reason but we provide a interface which is also looks like same which we planed. A project always need to be upgrade. If change needed we can do that. There will be more changes in our project by following our upcoming plans if needed. We try to make more data to entry in our project and give a perfect output.