

**INT – 306**

**DATABASE MANAGEMENT SYSTEM**

**PROJECT**

**STUDENT DATABASE MANAGEMENT SYSTEM**

* STUDENT NAME: SUDHANSHU MODI
* SECTION: E2003
* ROLL NUMBER: RE2003A04
* REGISTRATION NUMBER: 12010612
* SUBMITTED TO: DR. BALRAJ SINGH SIR

**LOVELY PROFESSIONAL UNIVERSITY, PHAGWARA**

**INTRODUCTION**

I have been working on the topic Student database management system using Database management system (DBMS). A database management system is a collection of inter-related data and a set of programs to access that data. This project basically helps the management to keep the correct records safely and securely, while it should be easy to access. I have tried to create the data with minimal redundancy and inconsistency. Integrity of the data is one of the most important and essential factors in creating a successful database, where I have been trying to create this data with minimum integrity problems.

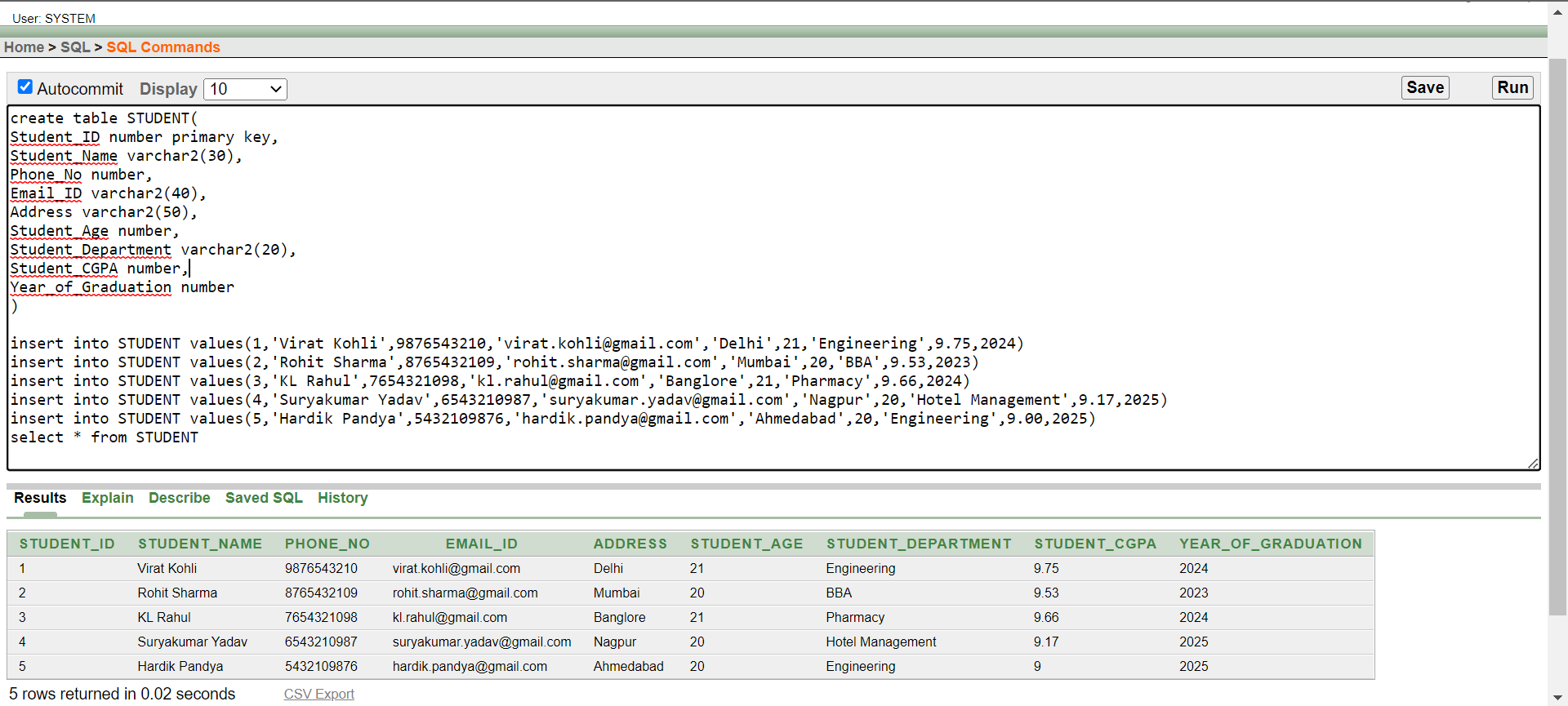
What does my student database management system include?

* A Student, Courses, Fees, Specialization, Exams, Faculty and Management tables which specifies very specific information with 5 entries in every table.
* The student table gives all the information such as Student ID, Name, contact details, Home city, Age, Field of study, their grades and the year in which they are going to graduate.
* The course table includes the course IDs, course names based on the IDs, Head of Departments of the courses and the student course ID which states that which student is studying which course (for e.g., Student with ID=1 studies engineering).
* The Fees table includes the fee type, total fees of the student, amount the student paid and the amount that is remaining.
* The Specialization table includes a specialization ID, specialization name and the specialization HOD for each student.
* The Exam table again includes the Student ID, the Exam ID of that particular exam, exam name and the type of exam.
* The faculty table includes the faculty ID, faculty name, department of the faculty person, the domain/specialization of that particular faculty person and the annual salary.
* And the last table i.e., the Management table includes Student ID login, Student Username, Student password, their last login and the mentor faculty ID.
* Here Student ID, faculty ID is set as primary key. Student course ID, Fee s

**SCREENSHOTS**

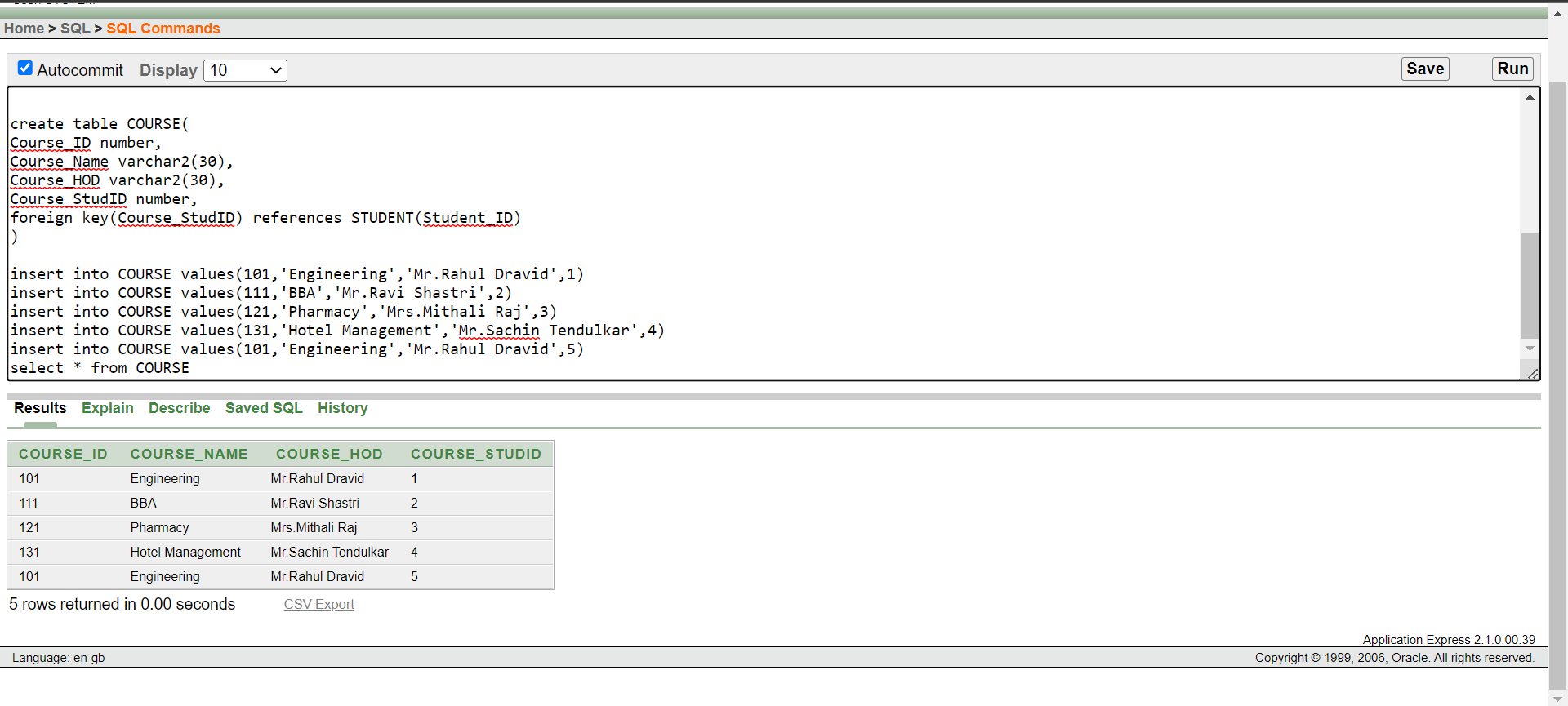
1. **STUDENT TABLE:**

Student ID is set as primary key.



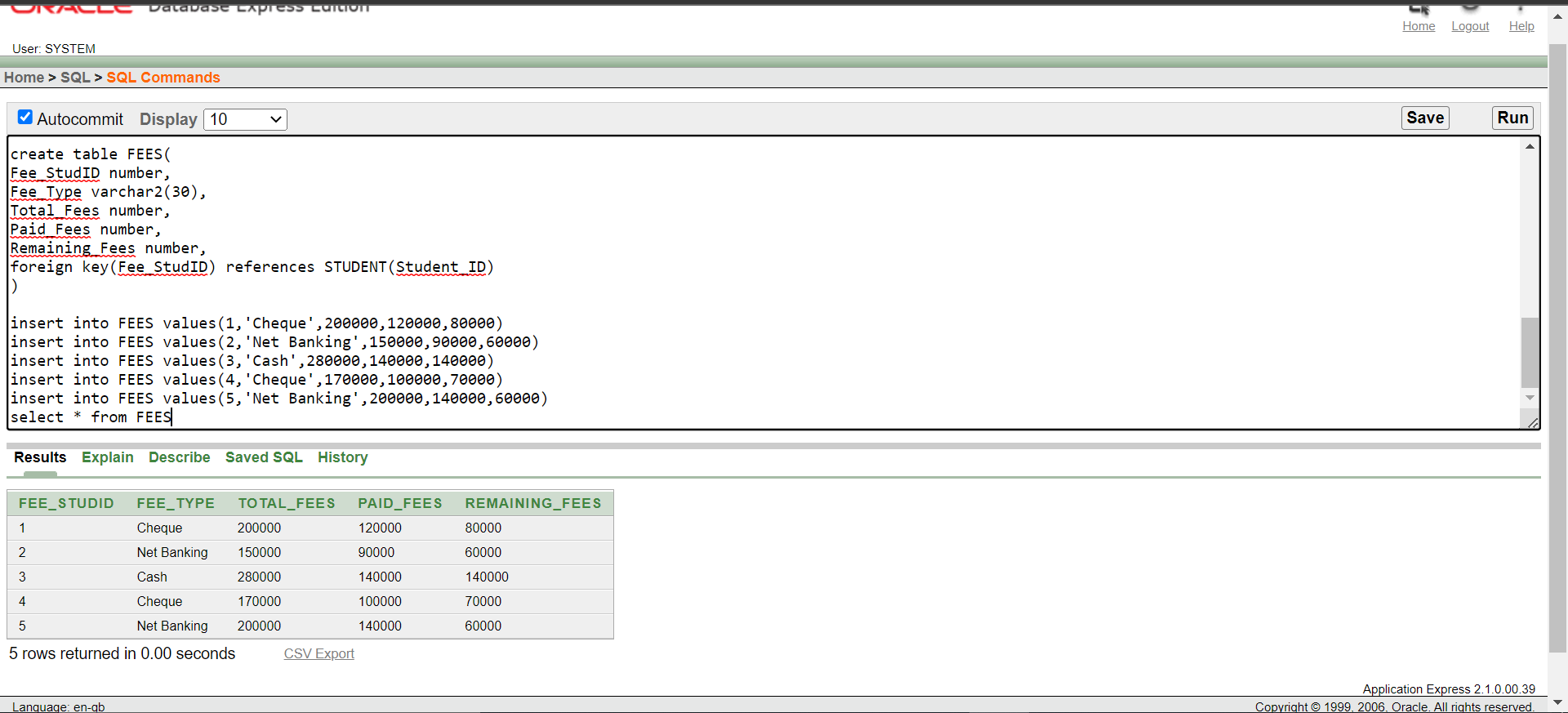
1. **COURSES TABLE:**

Student Course ID is set as foreign key.



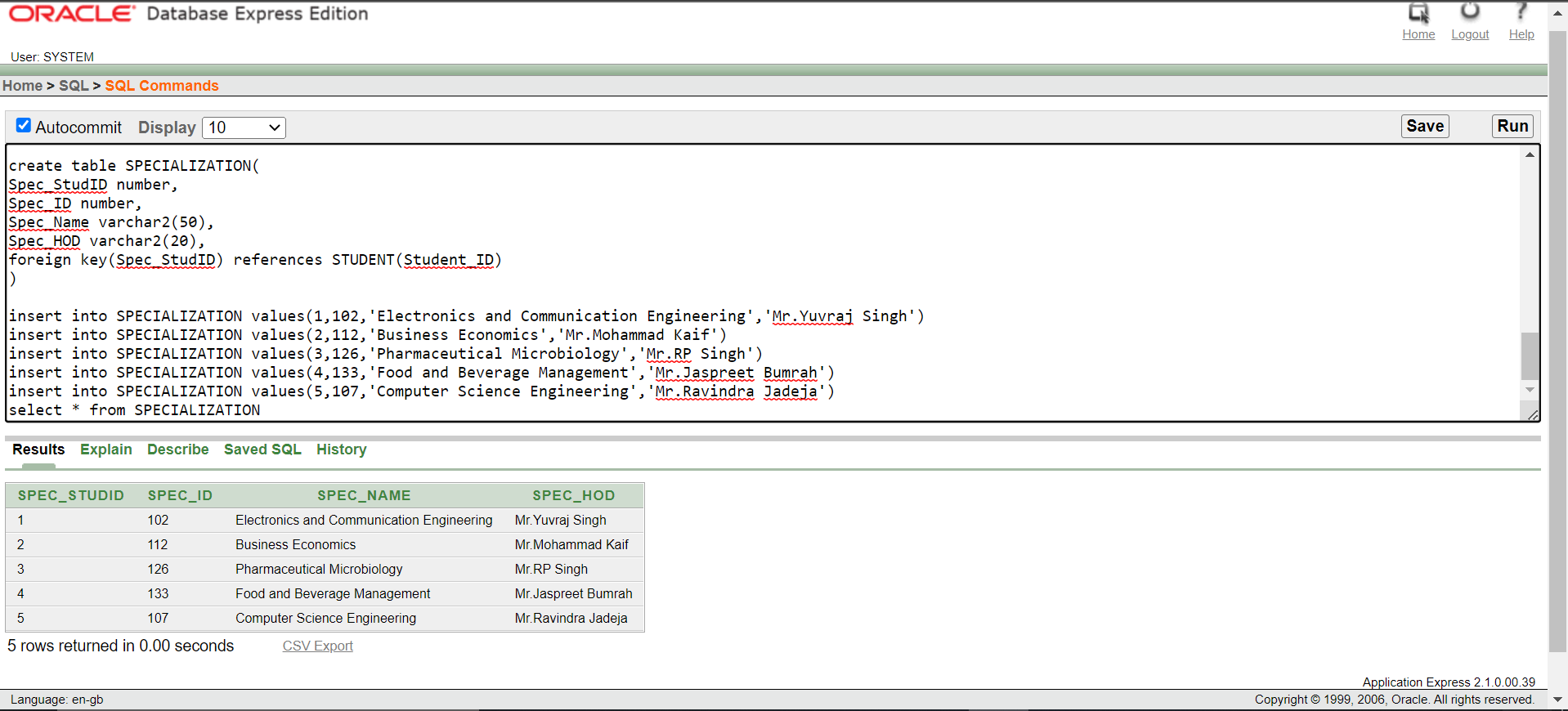
1. **FEES TABLE:**

Student Fees ID is set as foreign key.



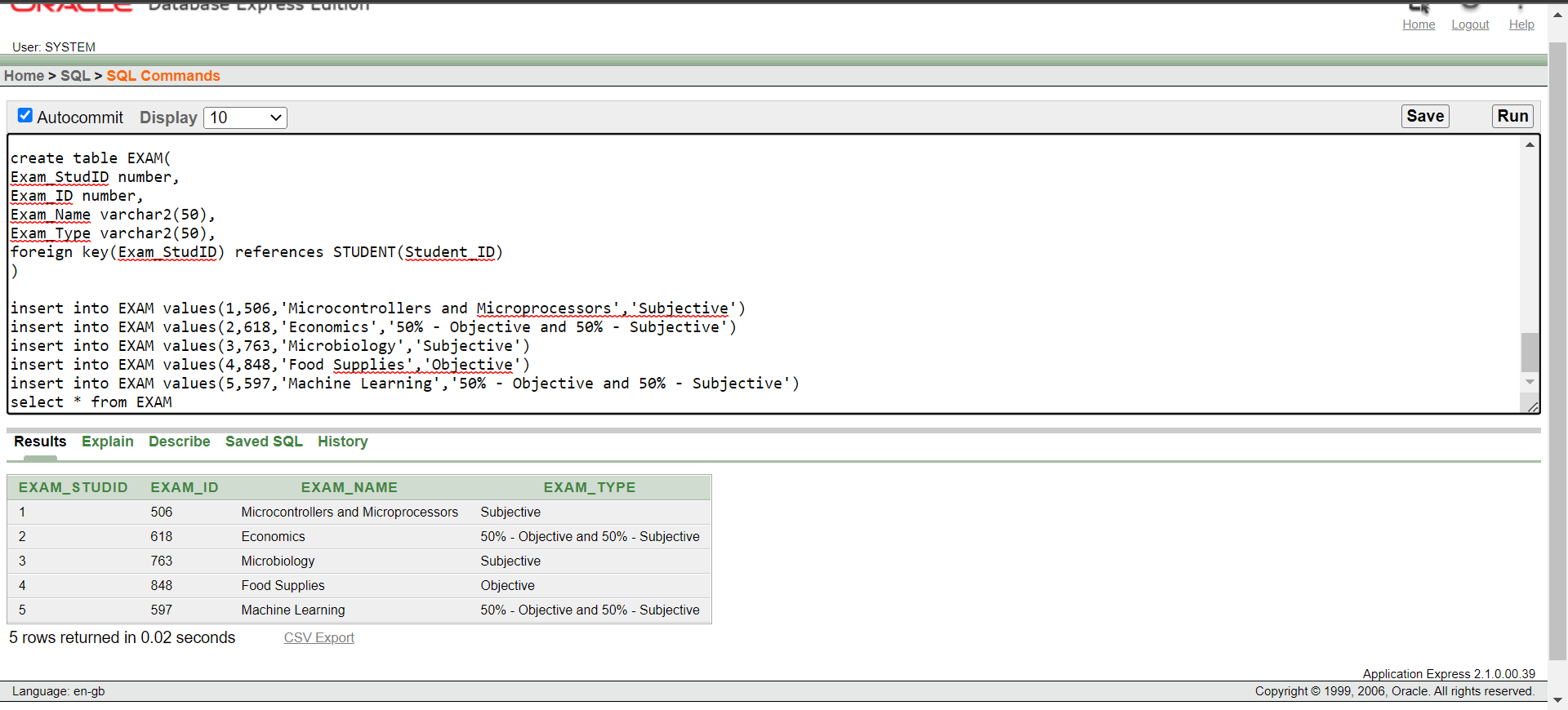
1. **SPECIALIZATION TABLE:**

The Student specialization ID is set as foreign key.



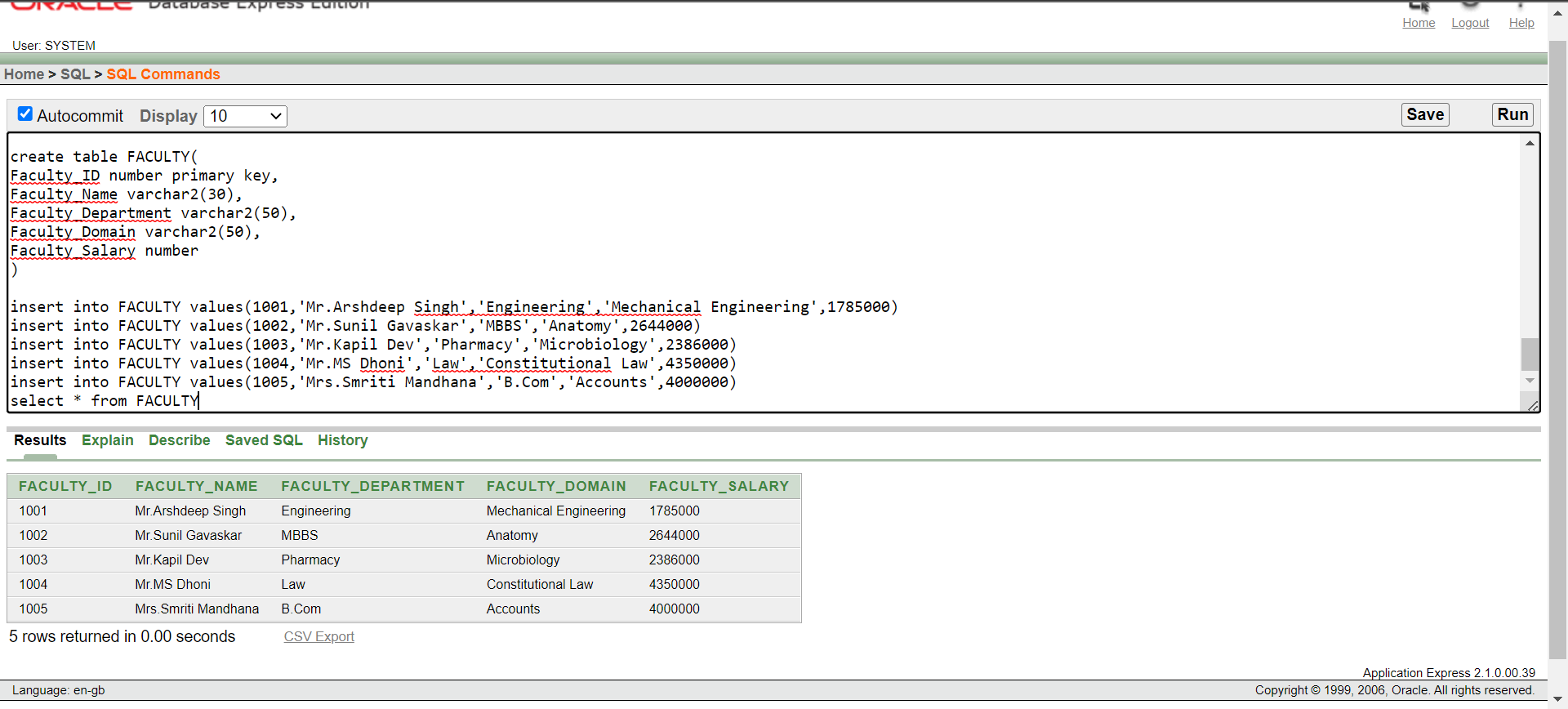
1. **EXAM TABLE:**

Student Exam ID is set as foreign key.



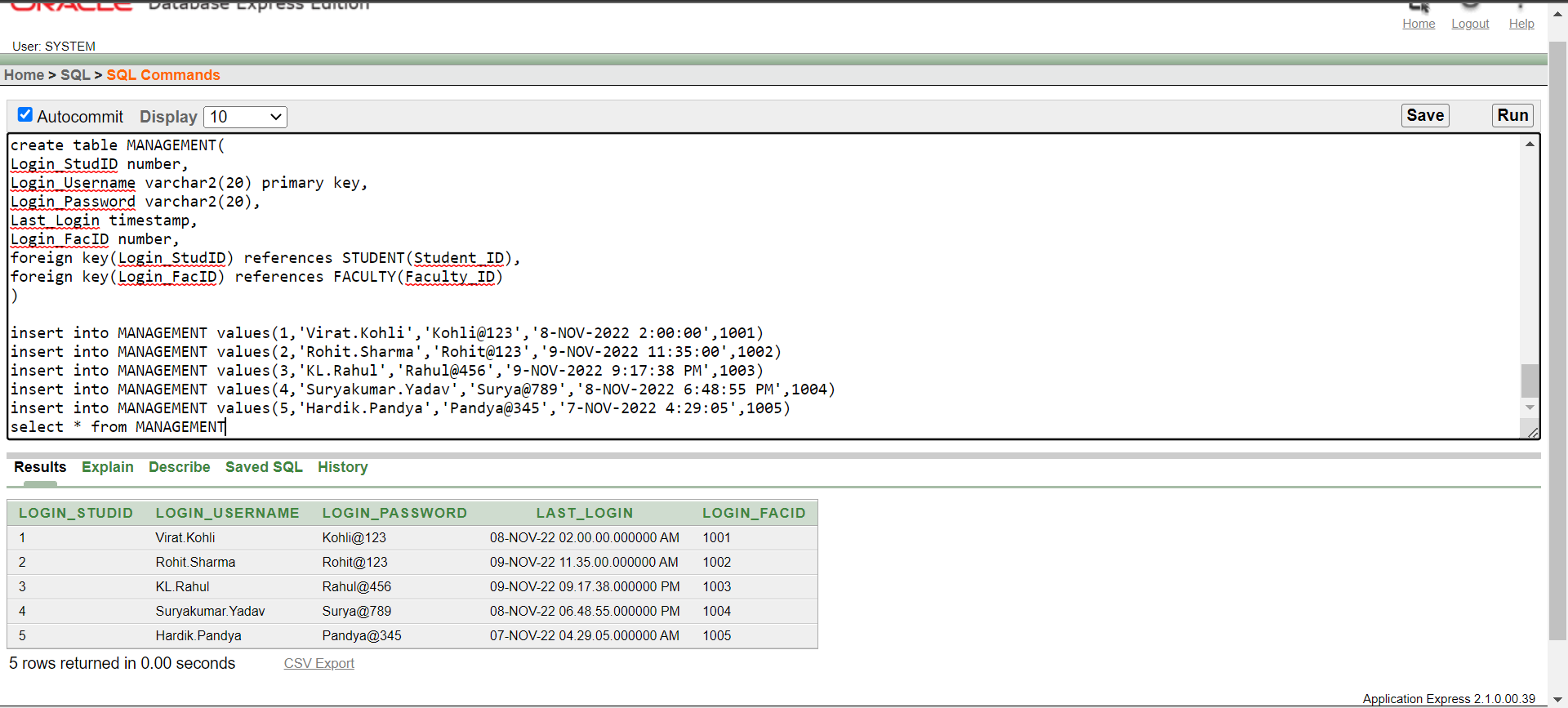
1. **FACULTY TABLE:**

Faculty ID is set as the primary key.

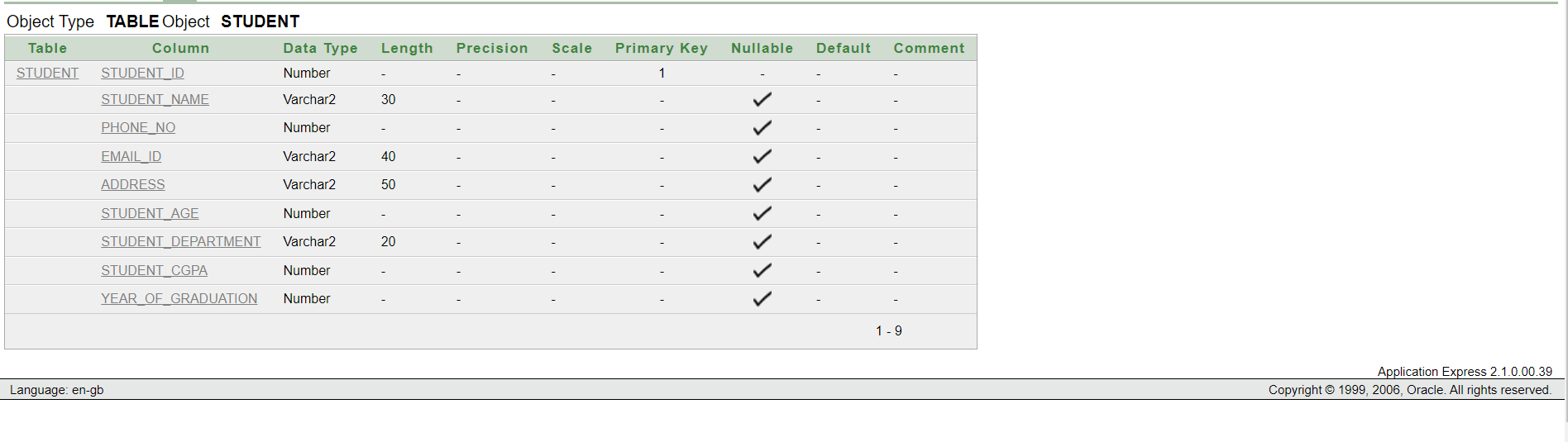


1. **MANAGEMENT TABLE:**

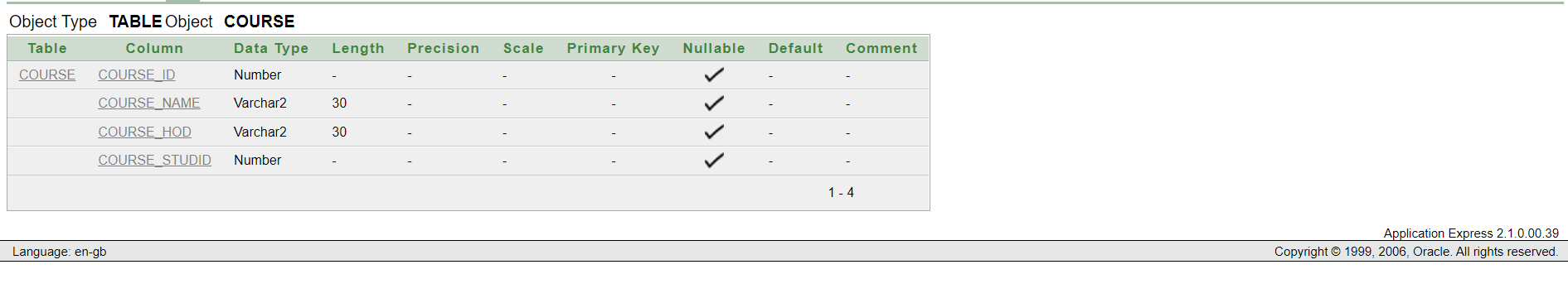
This table is a bit different from the others, as the Login username is set as primary key, student login IDs and faculty login IDs are set as foreign keys.



1. **DESCRIPTION OF STUDENT TABLE:**



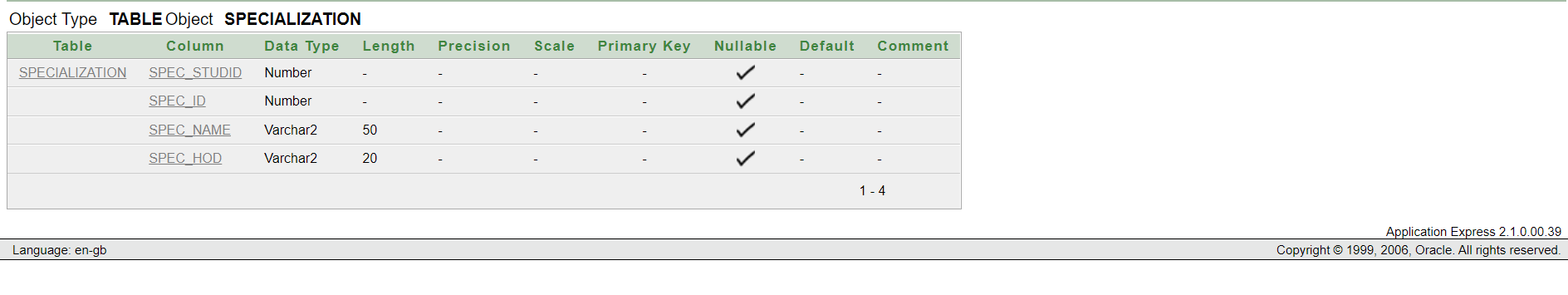
1. **DESCRIPTION OF COURSE TABLE:**



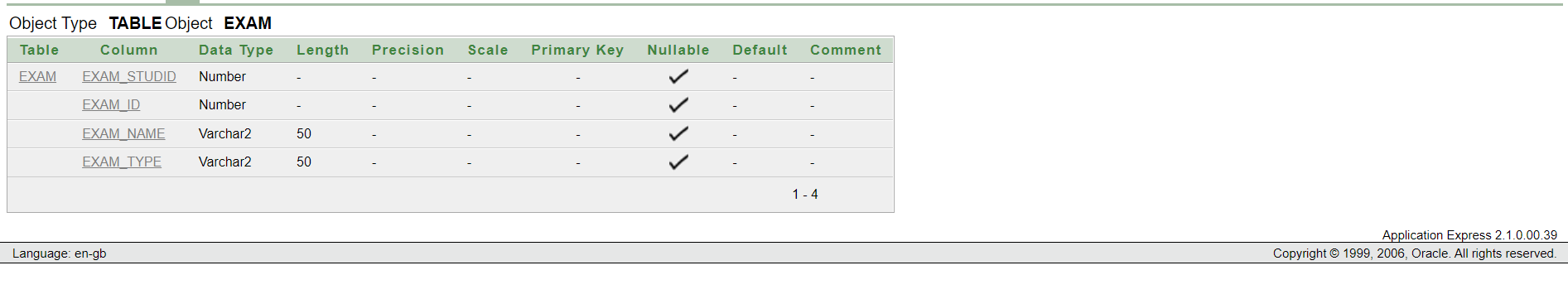
1. **DESCRIPTION OF FEES TABLE:**



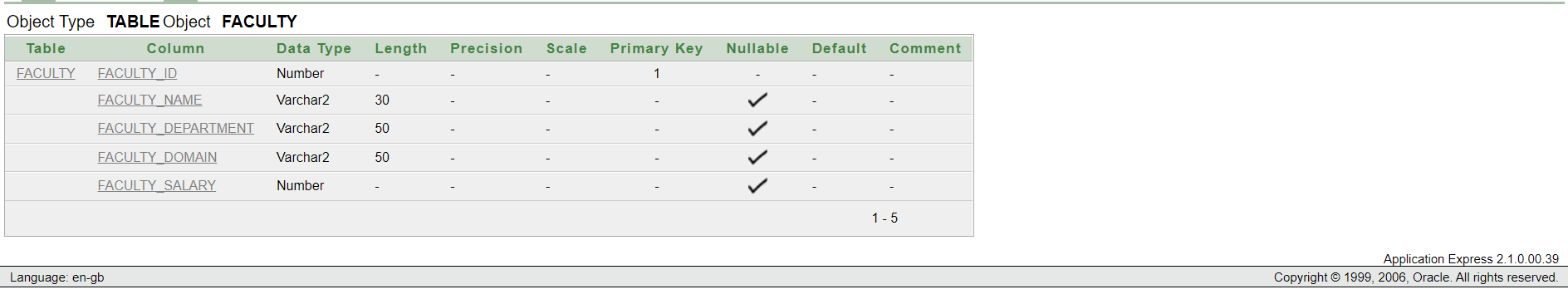
1. **DESCRIPTION OF SPECIALIZATION TABLE:**



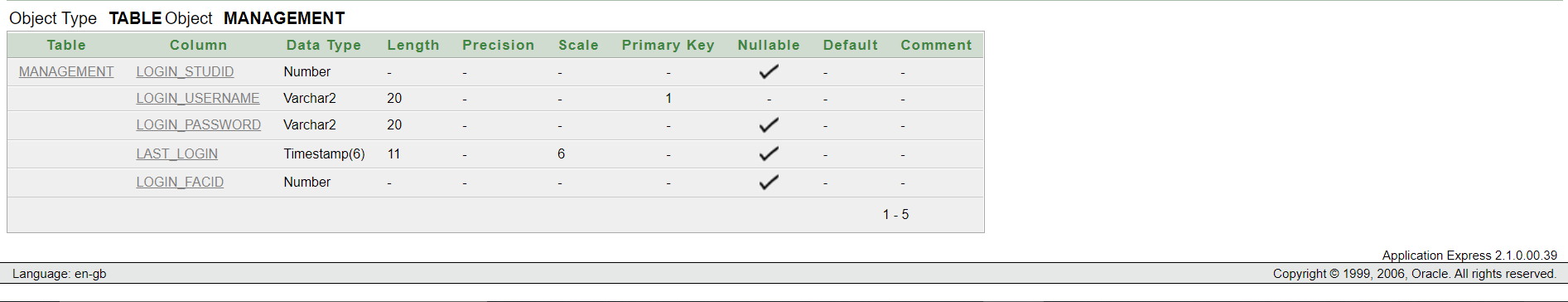
1. **DESCRIPTION OF EXAM TABLE:**



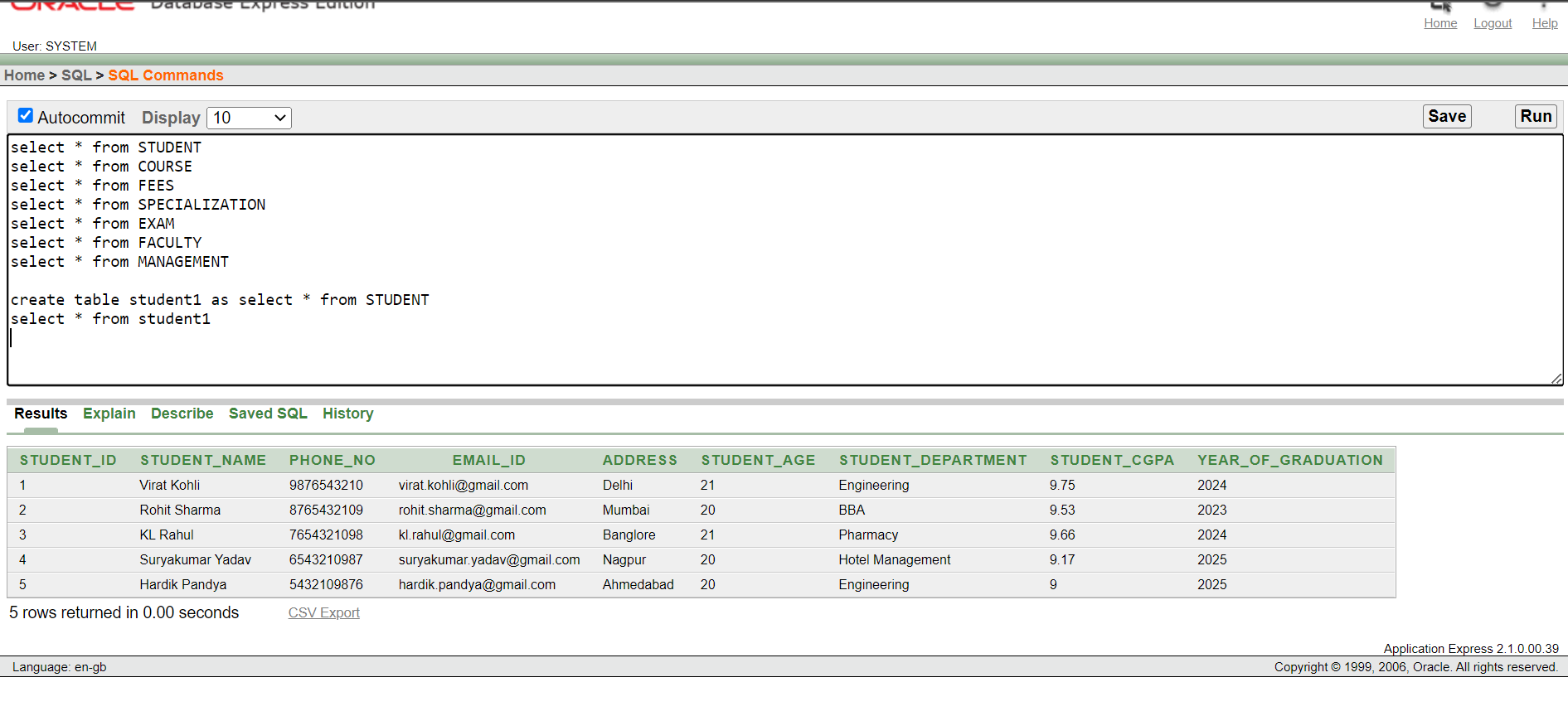
1. **DESCRIPTION OF FACULTY TABLE:**



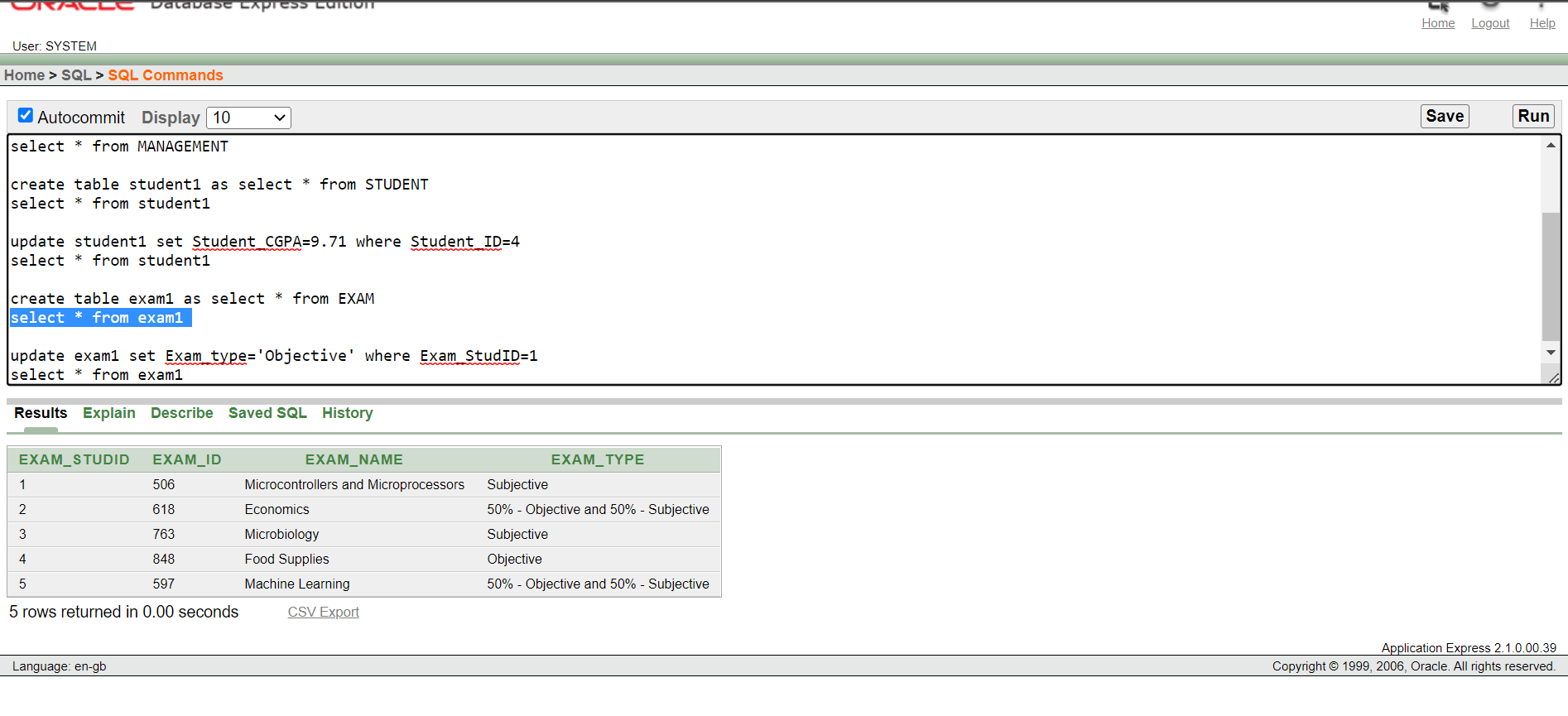
1. **DESCRIPTION OF MANAGEMENT TABLE:**



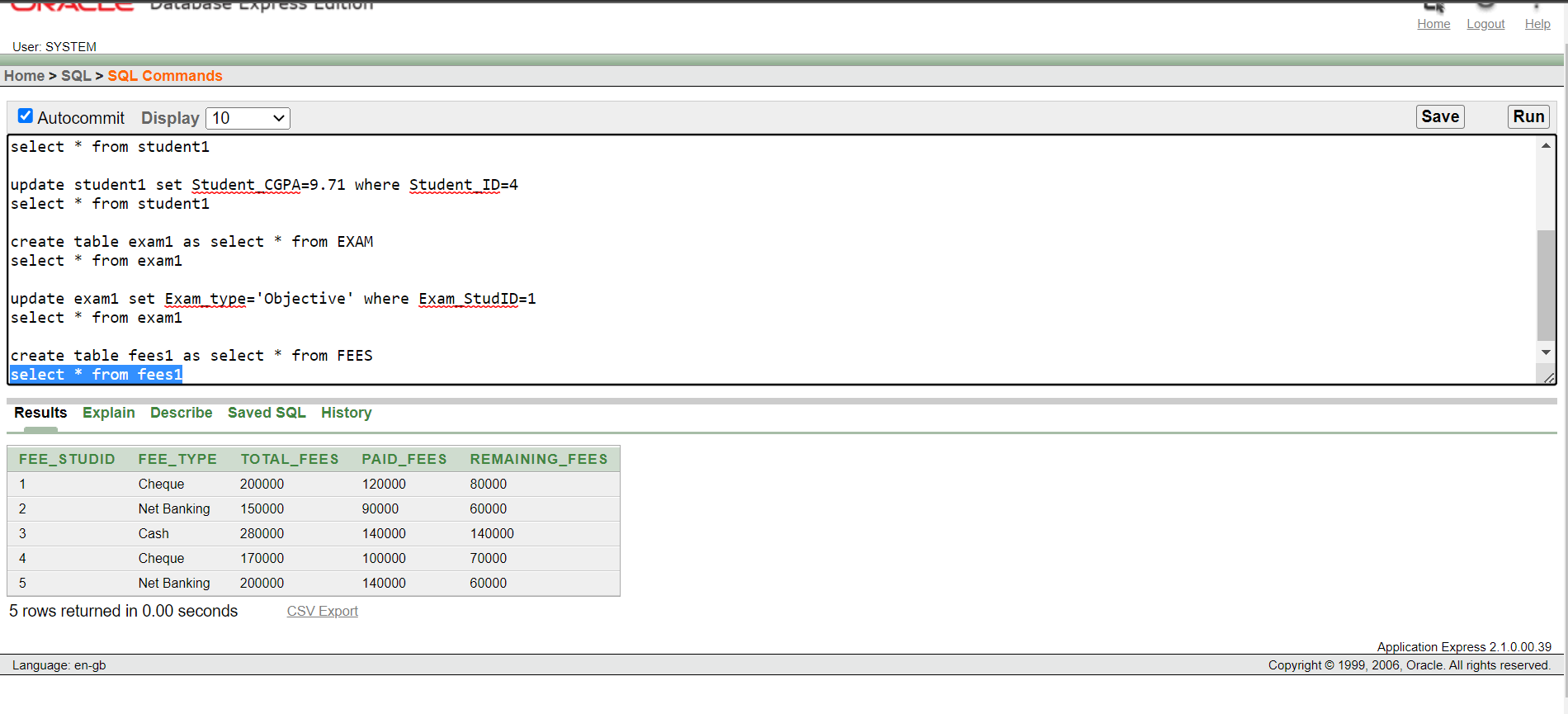
1. **CREATING TABLE STUDENT1 FROM STUDENT TO PERFORM DDL AND DML:**



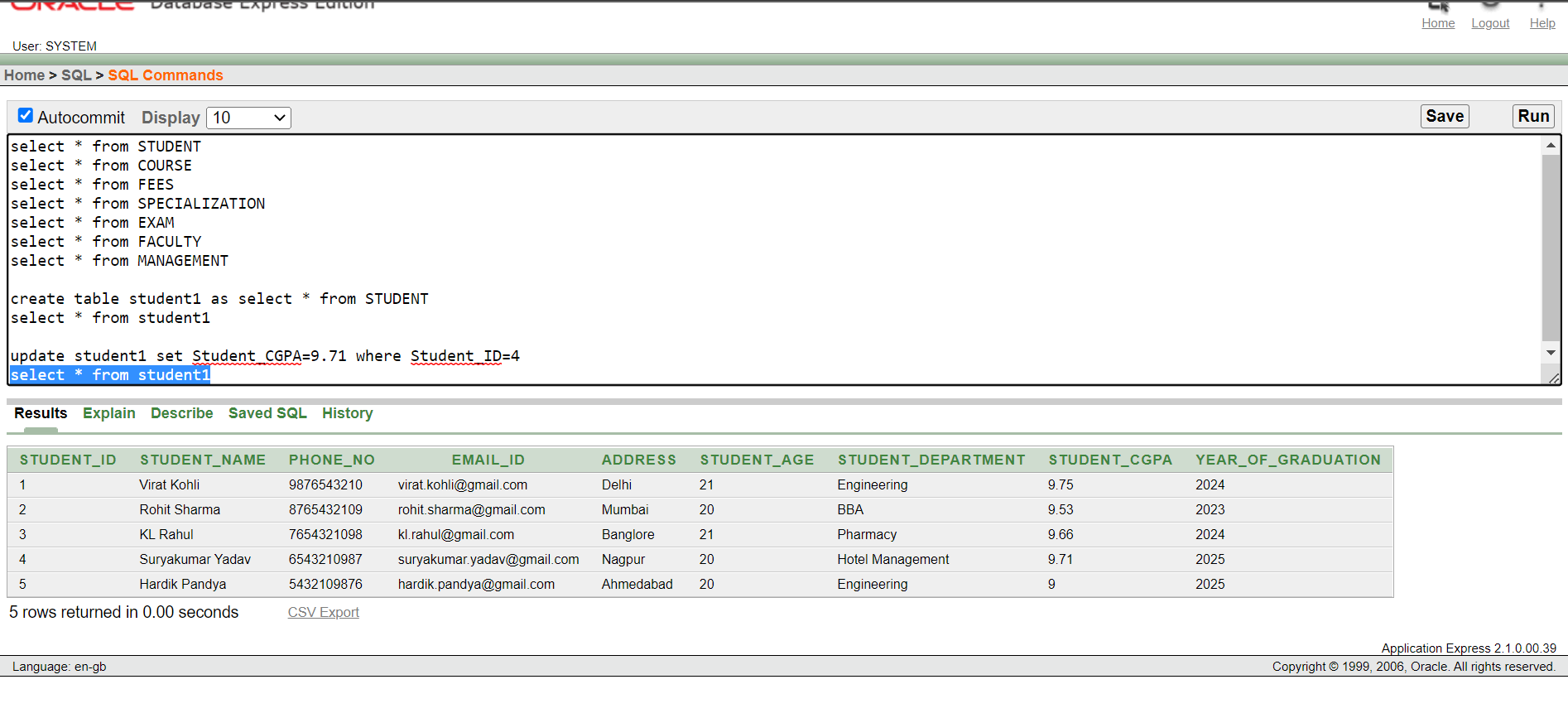
1. **CREATING TABLE EXAM1 FROM EXAM:**



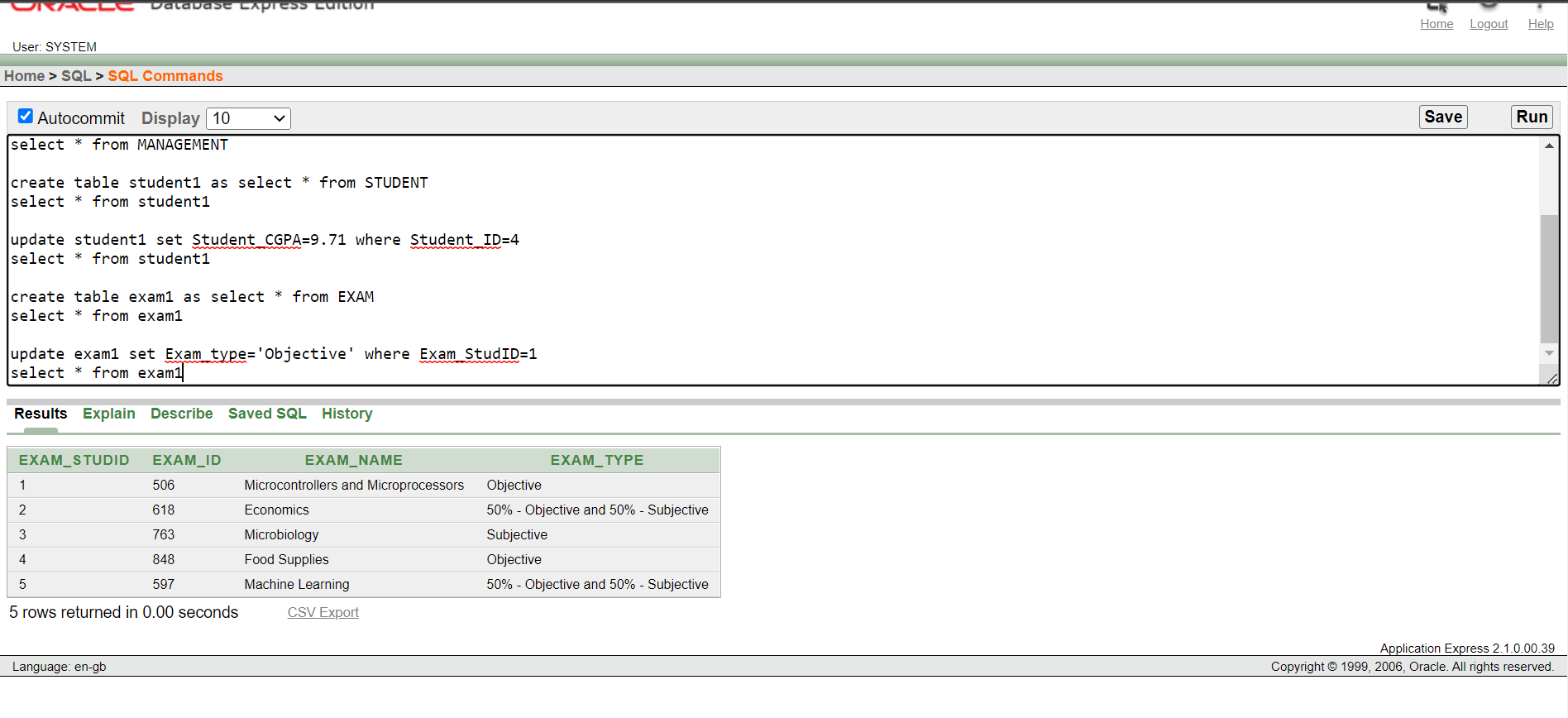
1. **CREATING TABLE FEES1 FROM FEES:**



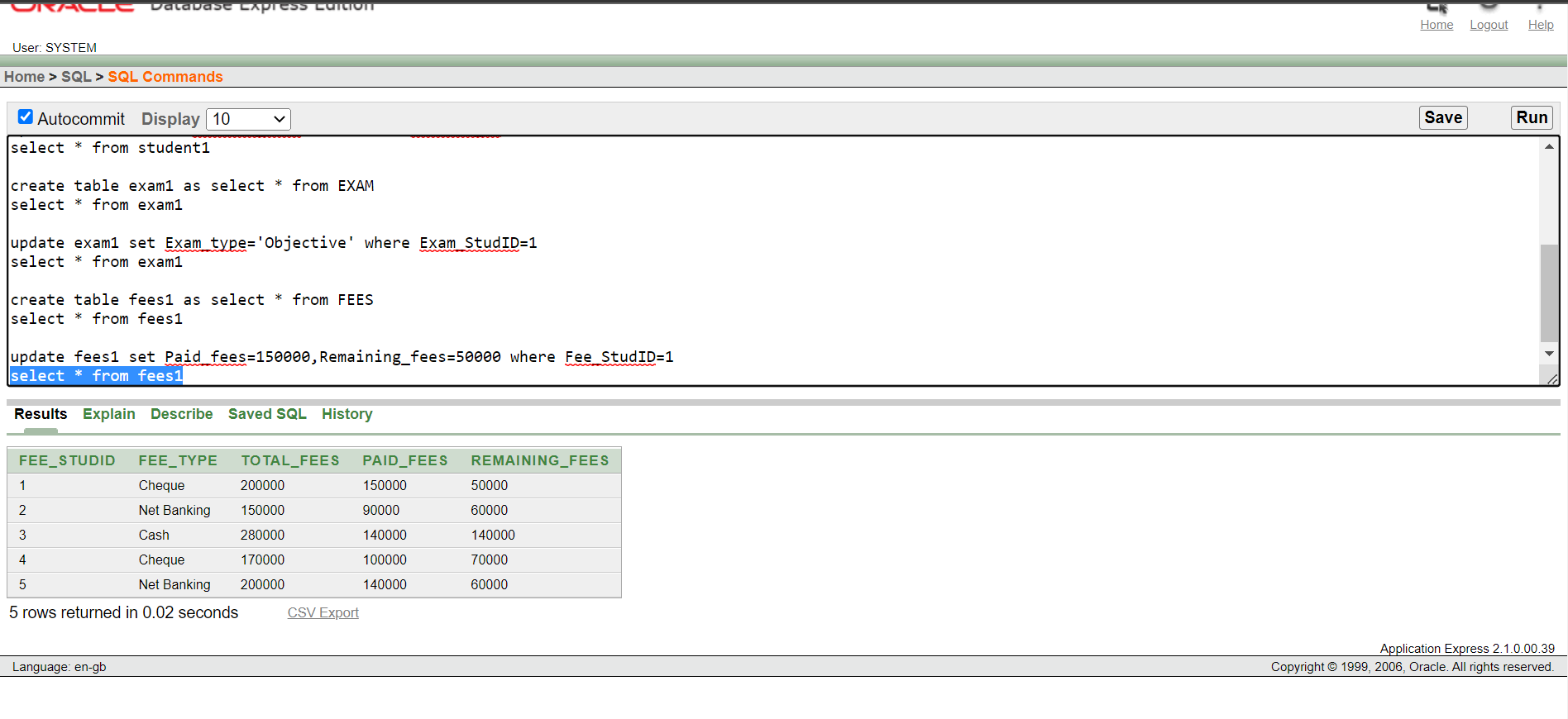
1. **UPDATING STUDENT1:**



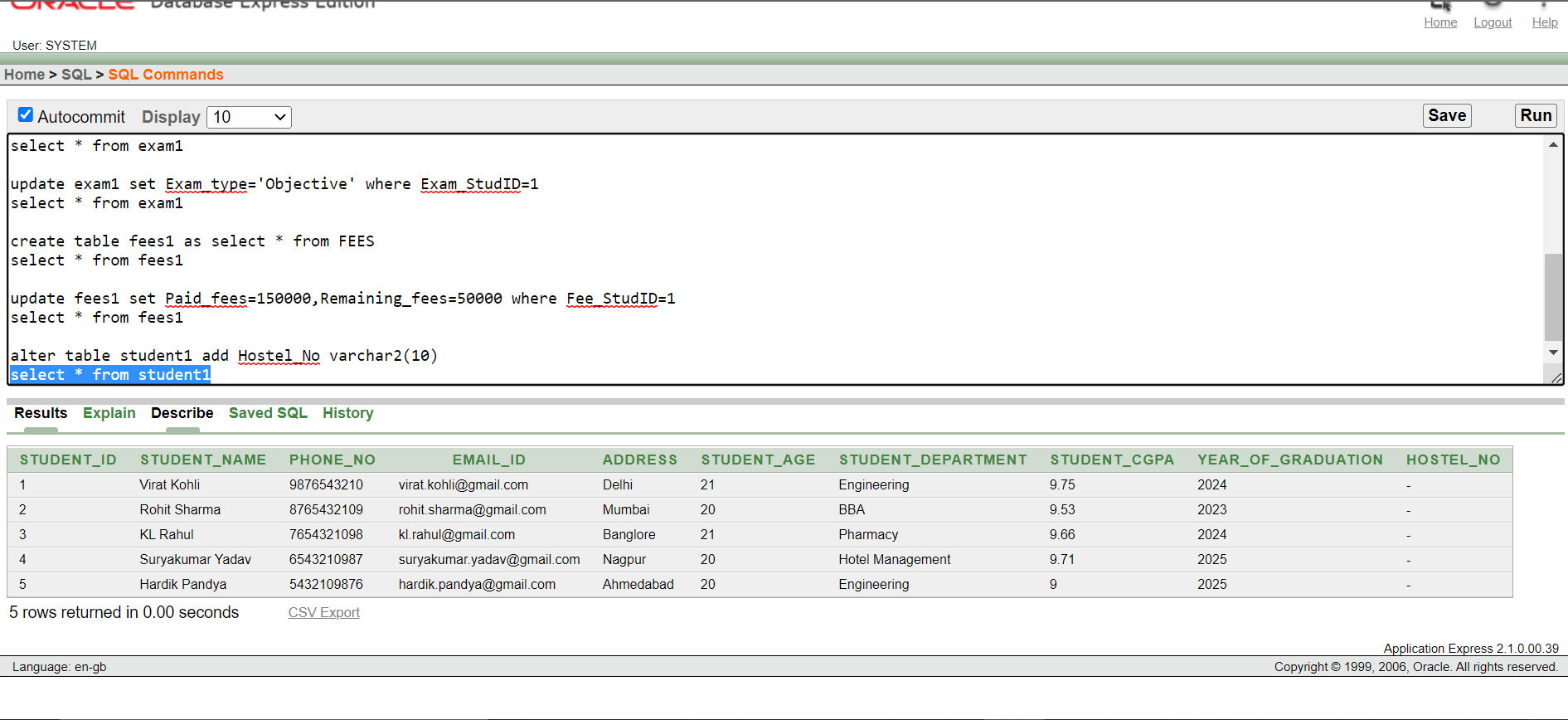
1. **UPDATING EXAM1:**



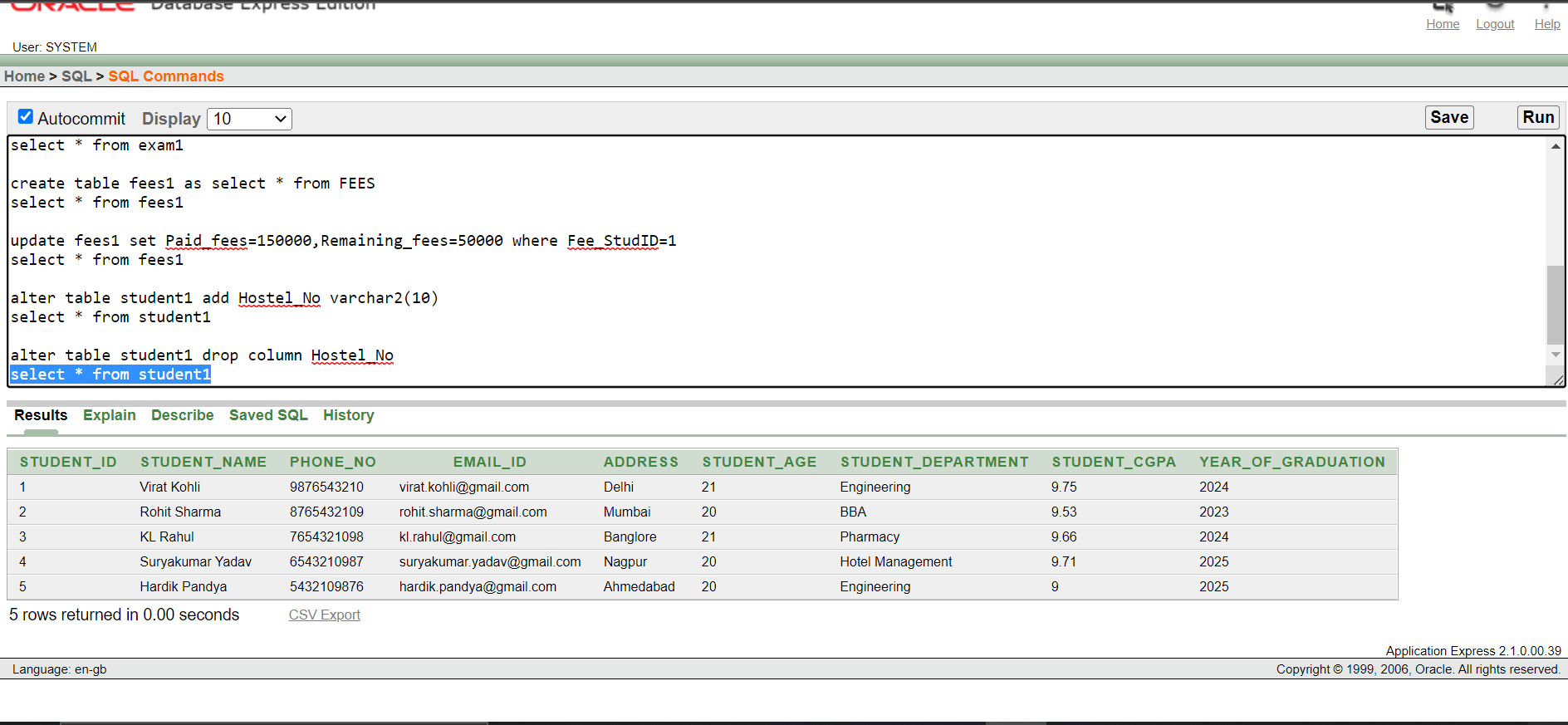
1. **UPDATING FEES1:**



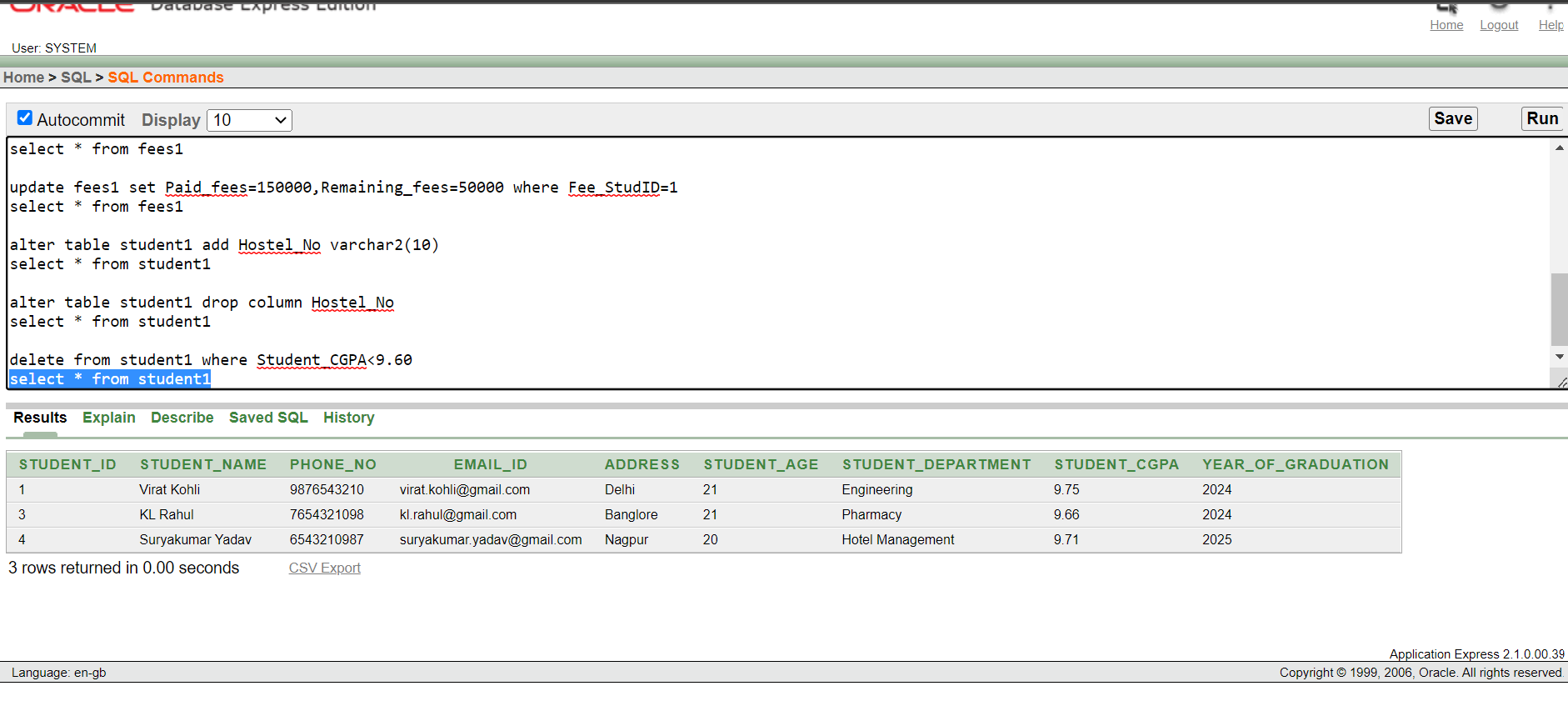
1. **ALTERING STUDENT1 BY ADDING A COLUMN HOSTEL NO:**



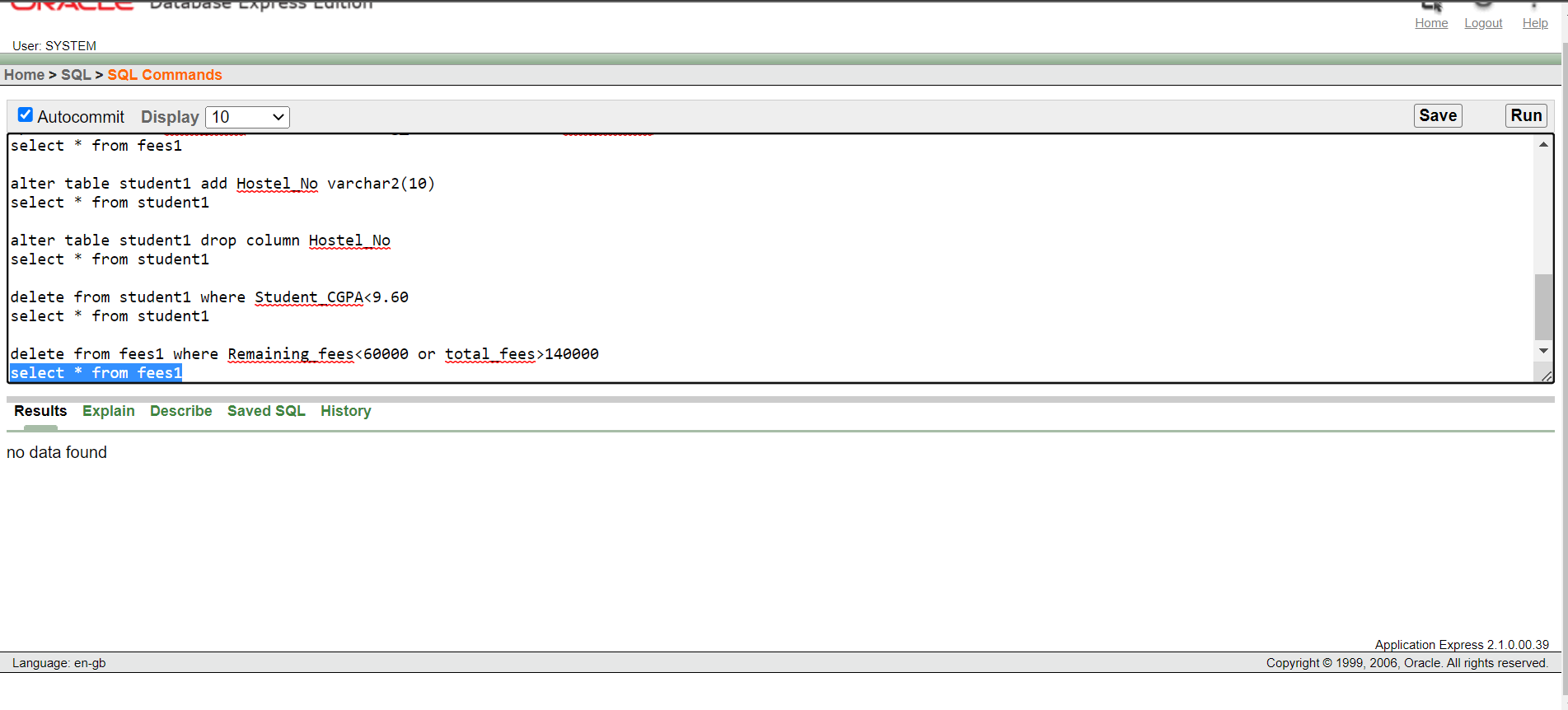
1. **ALTERING STUDENT1 BY DROPPING THE COLUMN HOSTEL NO:**



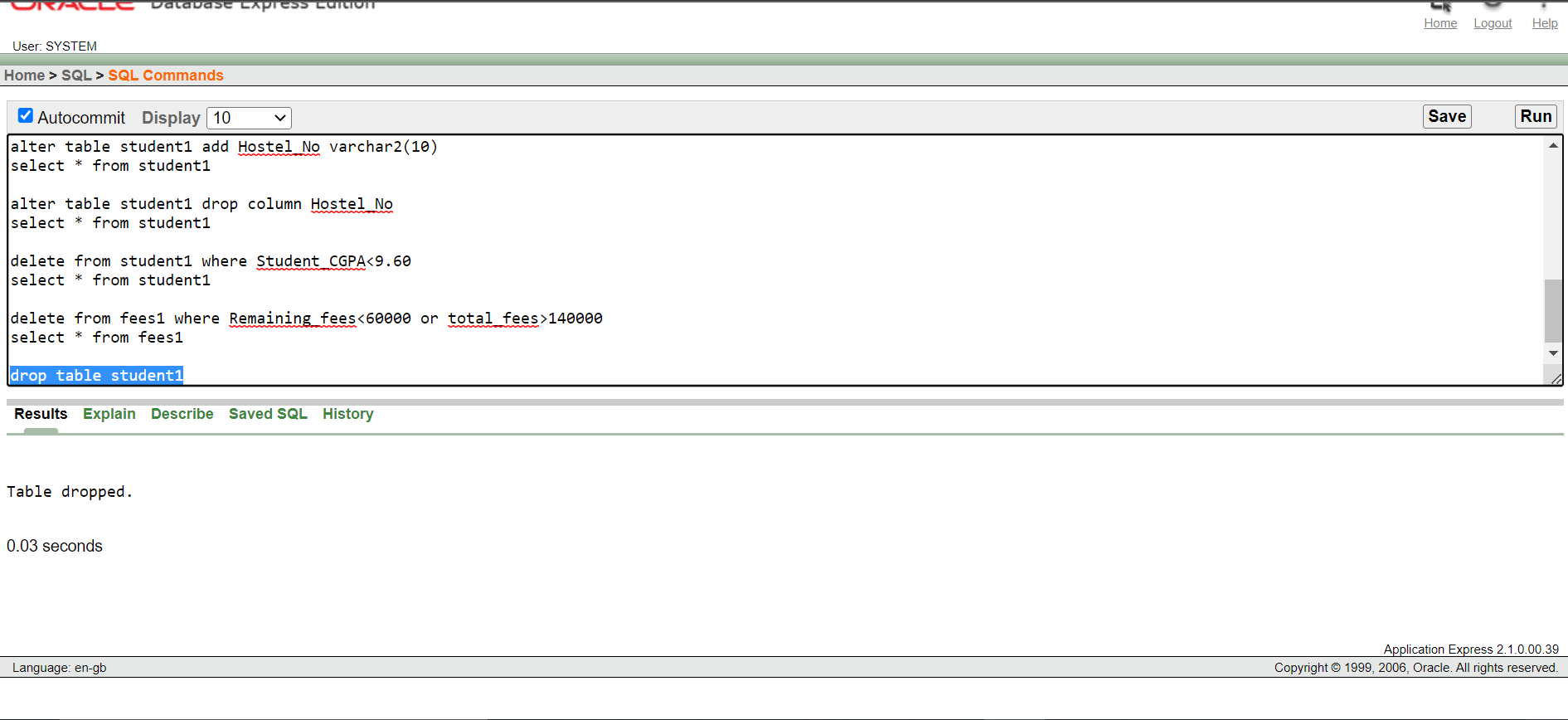
1. **DELETE COMMAND ON STUDENT1:**



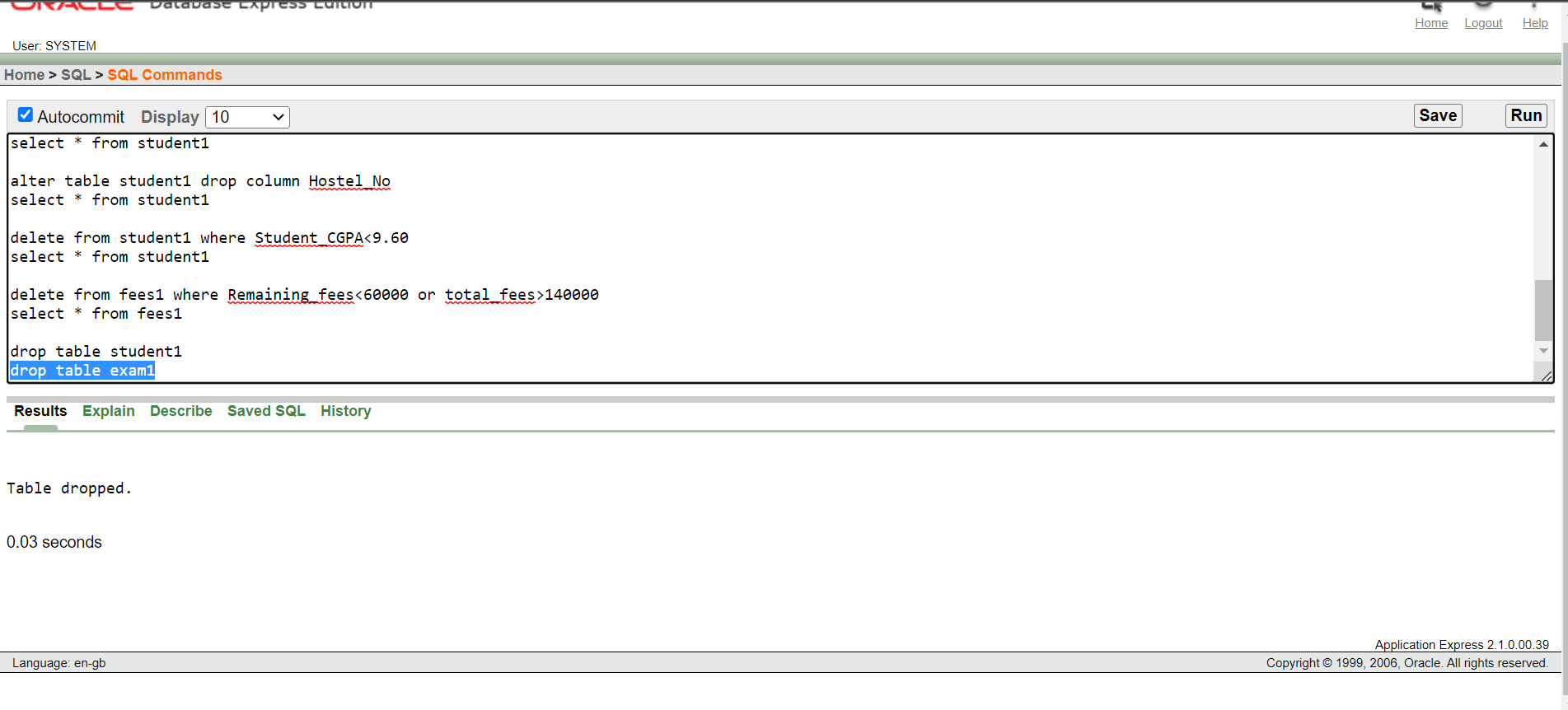
1. **DELETE COMMAND ON FEES1:**



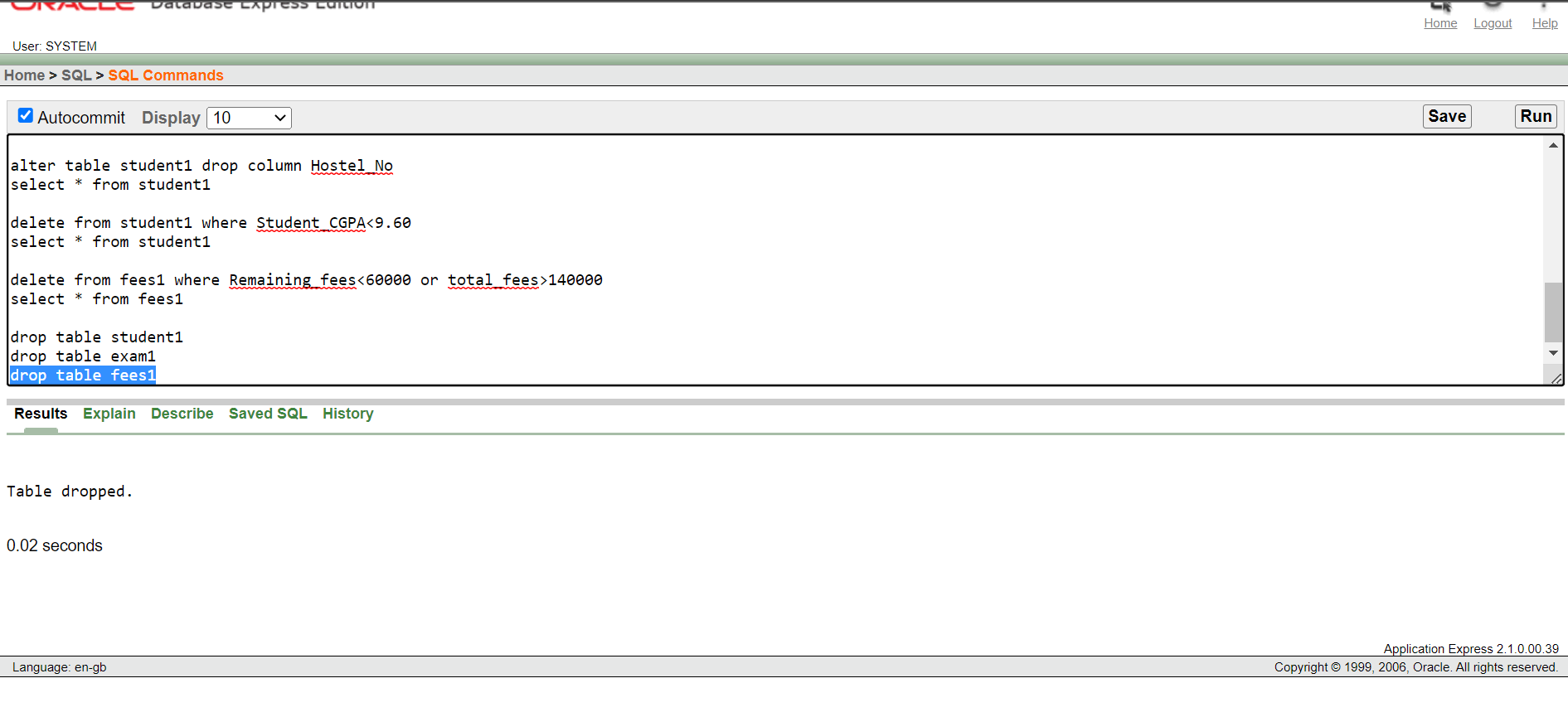
1. **DROPPING TABLE STUDENT1:**



1. **DROPPING TABLE EXAM1:**



1. **DROPPING TABLE FEES1:**



**CODES**

**CODE FOR CREATING THE TABLE (DDL), INSERTING THE VALUES (DML COMMANDS) AND TO GET THE DESCRIPTION OF THE TABLES.**

create table STUDENT(

Student\_ID number primary key,

Student\_Name varchar2(30),

Phone\_No number,

Email\_ID varchar2(40),

Address varchar2(50),

Student\_Age number,

Student\_Department varchar2(20),

Student\_CGPA number,

Year\_of\_Graduation number

)

Insert into STUDENT values(1,'Virat Kohli',9876543210,'virat.kohli@gmail.com','Delhi',21,'Engineering',9.75,2024)

insert into STUDENT values(2,'Rohit Sharma',8765432109,'rohit.sharma@gmail.com','Mumbai',20,'BBA',9.53,2023)

insert into STUDENT values(3,'KL Rahul',7654321098,'kl.rahul@gmail.com','Banglore',21,'Pharmacy',9.66,2024)

insert into STUDENT values(4,'Suryakumar Yadav',6543210987,'suryakumar.yadav@gmail.com','Nagpur',20,'Hotel Management',9.17,2025)

insert into STUDENT values(5,'Hardik Pandya',5432109876,'hardik.pandya@gmail.com','Ahmedabad',20,'Engineering',9.00,2025)

select \* from STUDENT

create table COURSE(

Course\_ID number,

Course\_Name varchar2(30),

Course\_HOD varchar2(30),

Course\_StudID number,

foreign key(Course\_StudID) references STUDENT(Student\_ID)

)

insert into COURSE values(101,'Engineering','Mr.Rahul Dravid',1)

insert into COURSE values(111,'BBA','Mr.Ravi Shastri',2)

insert into COURSE values(121,'Pharmacy','Mrs.Mithali Raj',3)

insert into COURSE values(131,'Hotel Management','Mr.Sachin Tendulkar',4)

insert into COURSE values(101,'Engineering','Mr.Rahul Dravid',5)

select \* from COURSE

create table FEES(

Fee\_StudID number,

Fee\_Type varchar2(30),

Total\_Fees number,

Paid\_Fees number,

Remaining\_Fees number,

foreign key(Fee\_StudID) references STUDENT(Student\_ID)

)

insert into FEES values(1,'Cheque',200000,120000,80000)

insert into FEES values(2,'Net Banking',150000,90000,60000)

insert into FEES values(3,'Cash',280000,140000,140000)

insert into FEES values(4,'Cheque',170000,100000,70000)

insert into FEES values(5,'Net Banking',200000,140000,60000)

select \* from FEES

create table SPECIALIZATION(

Spec\_StudID number,

Spec\_ID number,

Spec\_Name varchar2(50),

Spec\_HOD varchar2(20),

foreign key(Spec\_StudID) references STUDENT(Student\_ID)

)

insert into SPECIALIZATION values(1,102,'Electronics and Communication Engineering','Mr.Yuvraj Singh')

insert into SPECIALIZATION values(2,112,'Business Economics','Mr.Mohammad Kaif')

insert into SPECIALIZATION values(3,126,'Pharmaceutical Microbiology','Mr.RP Singh')

insert into SPECIALIZATION values(4,133,'Food and Beverage Management','Mr.Jaspreet Bumrah')

insert into SPECIALIZATION values(5,107,'Computer Science Engineering','Mr.Ravindra Jadeja')

select \* from SPECIALIZATION

create table EXAM(

Exam\_StudID number,

Exam\_ID number,

Exam\_Name varchar2(50),

Exam\_Type varchar2(50),

foreign key(Exam\_StudID) references STUDENT(Student\_ID)

)

insert into EXAM values(1,506,'Microcontrollers and Microprocessors','Subjective')

insert into EXAM values(2,618,'Economics','50% - Objective and 50% - Subjective')

insert into EXAM values(3,763,'Microbiology','Subjective')

insert into EXAM values(4,848,'Food Supplies','Objective')

insert into EXAM values(5,597,'Machine Learning','50% - Objective and 50% - Subjective')

select \* from EXAM

create table FACULTY(

Faculty\_ID number primary key,

Faculty\_Name varchar2(30),

Faculty\_Department varchar2(50),

Faculty\_Domain varchar2(50),

Faculty\_Salary number

)

insert into FACULTY values(1001,'Mr.Arshdeep Singh','Engineering','Mechanical Engineering',1785000)

insert into FACULTY values(1002,'Mr.Sunil Gavaskar','MBBS','Anatomy',2644000)

insert into FACULTY values(1003,'Mr.Kapil Dev','Pharmacy','Microbiology',2386000)

insert into FACULTY values(1004,'Mr.MS Dhoni','Law','Constitutional Law',4350000)

insert into FACULTY values(1005,'Mrs.Smriti Mandhana','B.Com','Accounts',4000000)

select \* from FACULTY

create table MANAGEMENT(

Login\_StudID number,

Login\_Username varchar2(20) primary key,

Login\_Password varchar2(20),

Last\_Login timestamp,

Login\_FacID number,

foreign key(Login\_StudID) references STUDENT(Student\_ID),

foreign key(Login\_FacID) references FACULTY(Faculty\_ID)

)

insert into MANAGEMENT values(1,'Virat.Kohli','Kohli@123','8-NOV-2022 2:00:00',1001)

insert into MANAGEMENT values(2,'Rohit.Sharma','Rohit@123','9-NOV-2022 11:35:00',1002)

insert into MANAGEMENT values(3,'KL.Rahul','Rahul@456','9-NOV-2022 9:17:38 PM',1003)

insert into MANAGEMENT values(4,'Suryakumar.Yadav','Surya@789','8-NOV-2022 6:48:55 PM',1004)

insert into MANAGEMENT values(5,'Hardik.Pandya','Pandya@345','7-NOV-2022 4:29:05',1005)

select \* from MANAGEMENT

description STUDENT

description COURSE

description FEES

description SPECIALIZATION

description EXAM

description FACULTY

description MANAGEMENT

**QUERIES**

**DDL: Already created 6 tables and hence 6 commands done for DDL**

**DDL: More 3 tables created and hence 9 commands done for DDL**

**DDL: 2 table altered and hence 11 commands done for DDL**

**DDL: 3 tables dropped and hence 14 commands done for DDL**

**DML: 5 rows inserted and hence 5 commands done for DML**

**DML: 3 rows updated and hence 8 commands done for DML**

**DML: 2 rows deleted and hence 10 commands done for DML**

create table student1 as select \* from STUDENT (DDL)

select \* from student1

update student1 set Student\_CGPA=9.71 where Student\_ID=4 (DML)

select \* from student1

create table exam1 as select \* from EXAM (DDL)

select \* from exam1

update exam1 set Exam\_type='Objective' where Exam\_StudID=1 (DML)

select \* from exam1

create table fees1 as select \* from FEES (DDL)

select \* from fees1

update fees1 set Paid\_fees=150000,Remaining\_fees=50000 where Fee\_StudID=1

select \* from fees1 (DML)

alter table student1 add Hostel\_No varchar2(10) (DDL)

select \* from student1

alter table student1 drop column Hostel\_No (DDL)

select \* from student1

delete from student1 where Student\_CGPA<9.60 (DML)

select \* from student1

delete from fees1 where Remaining\_fees<60000 or total\_fees>140000 (DML)

select \* from fees1

drop table student1 (DDL)

drop table exam1 (DDL)

drop table fees1 (DDL)

**PLSQL**

1. **SELECTING VALUES:**

declare

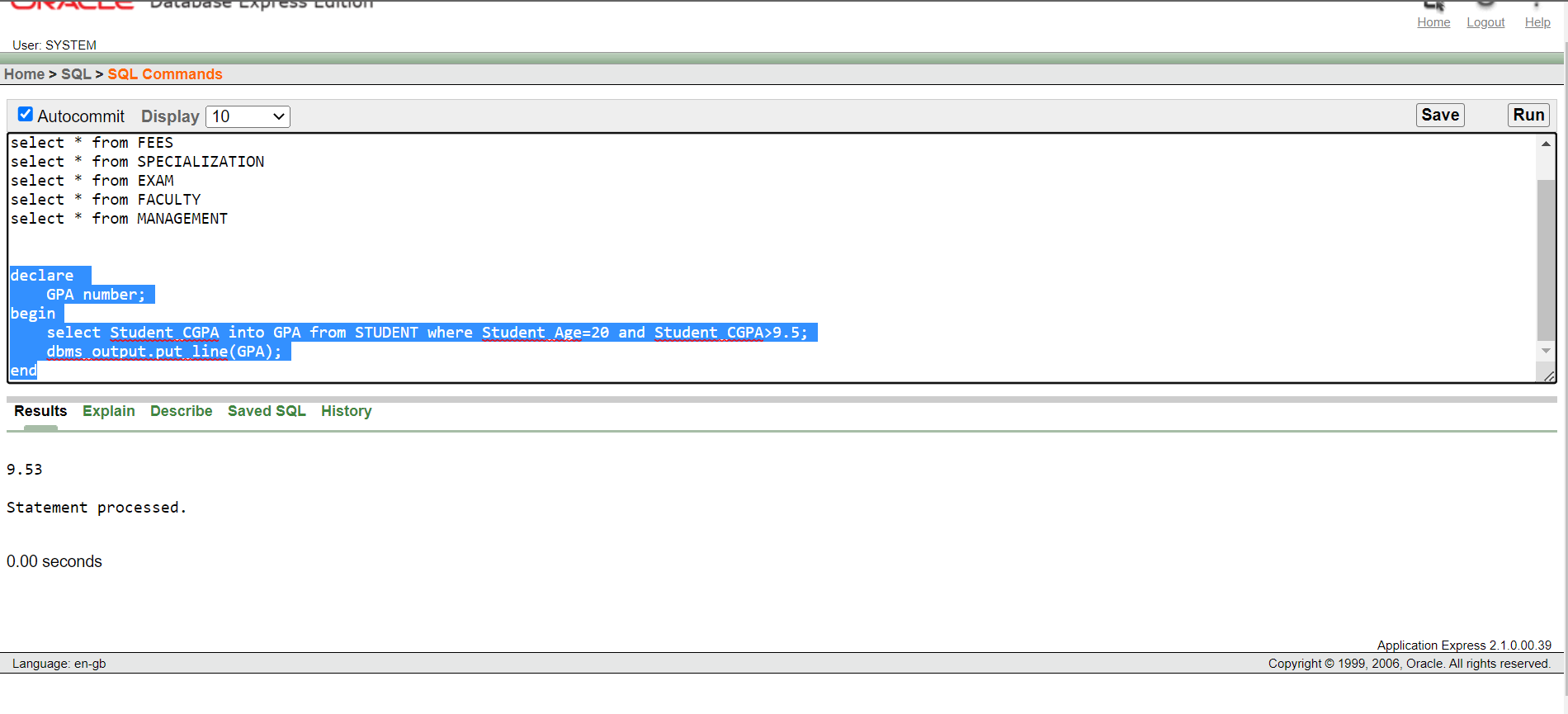
GPA number;

begin

select Student\_CGPA into GPA from STUDENT where Student\_Age=20 and Student\_CGPA>9.5;

dbms\_output.put\_line(GPA);

end



1. **APPLYING TRIGGER AND UPDATING A VALUE:**

create or replace trigger t1

after insert or delete or update on STUDENT

begin

if inserting then

dbms\_output.put\_line('Value is Inserted');

elsif deleting then

dbms\_output.put\_line('Value is Deleted');

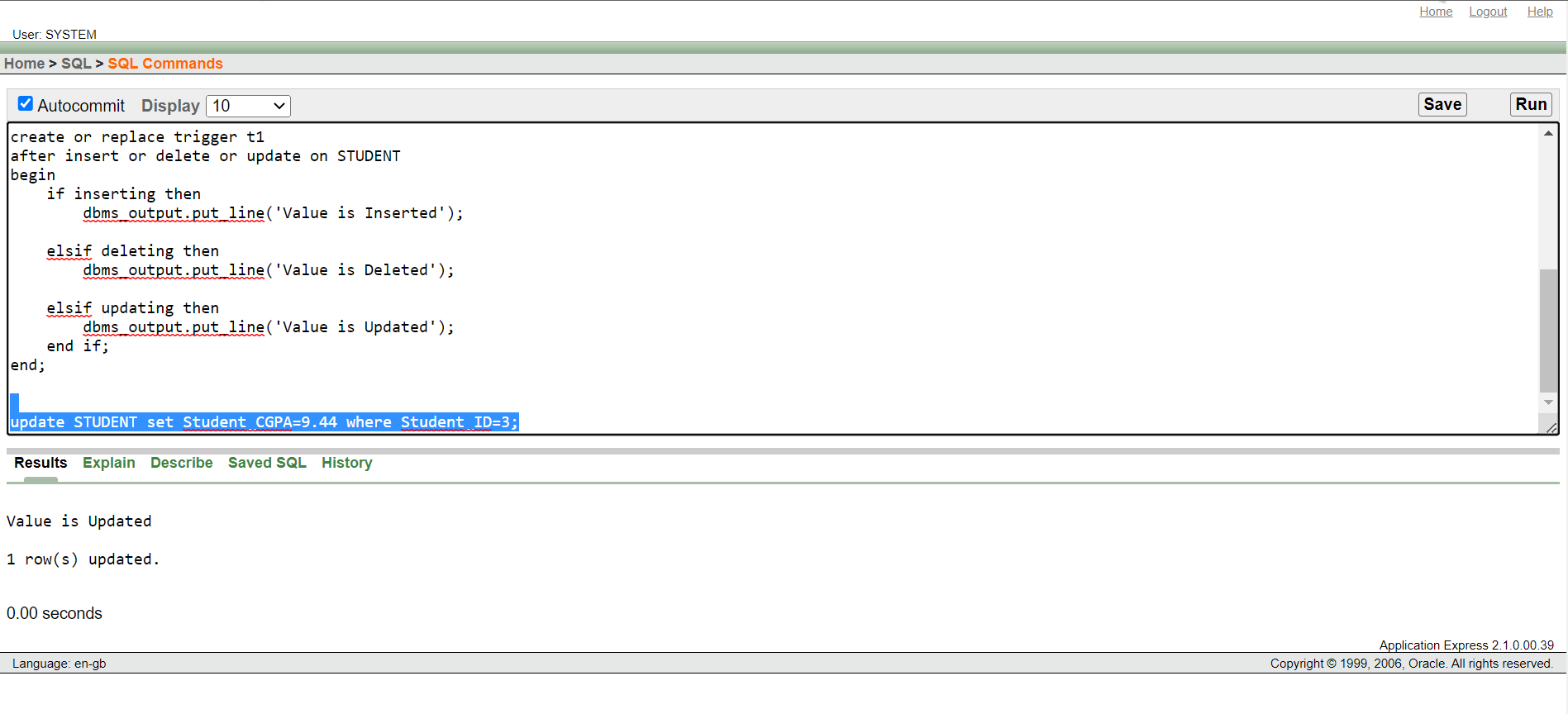
elsif updating then

dbms\_output.put\_line('Value is Updated');

end if;

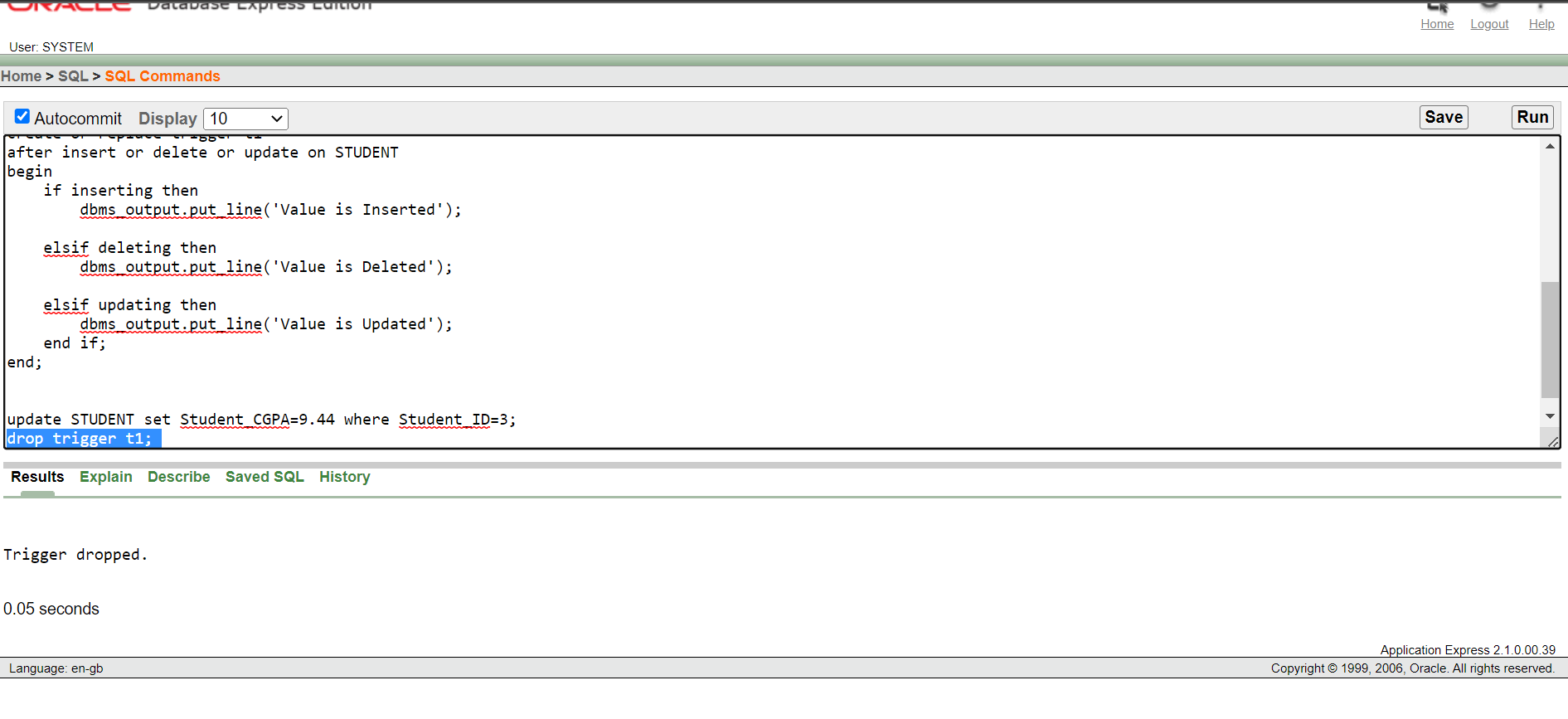
end;

update STUDENT set Student\_CGPA=9.44 where Student\_ID=3;

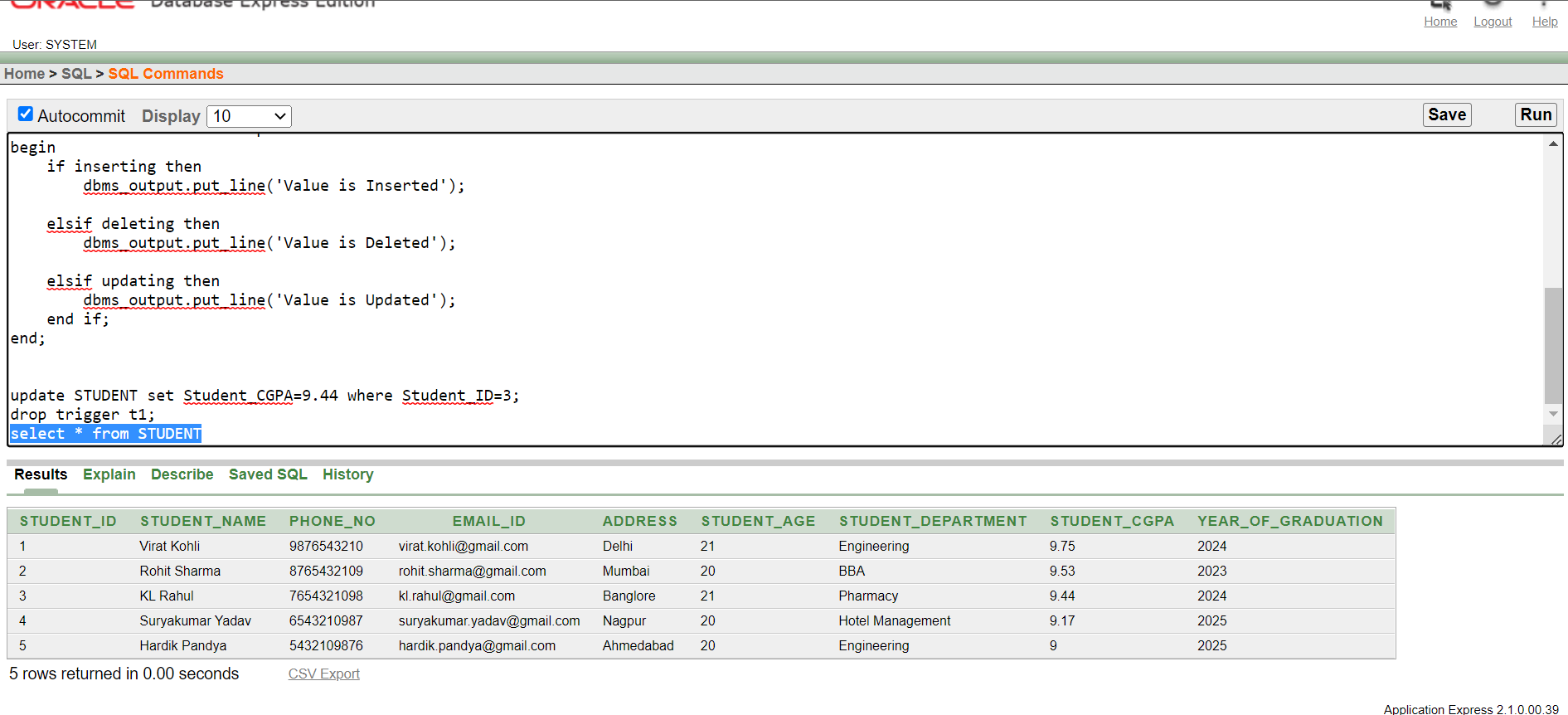


1. **DROPPING A TRIGGER:**

drop trigger t1;



1. **TABLE AFTER UPDATING:**



1. **PLSQL CODE FOR UPDATING SALARY IF FOUND:**

begin

update FACULTY set Faculty\_Salary=2899000 where Faculty\_Name='Mr.Kapil Dev';

if sql%found then

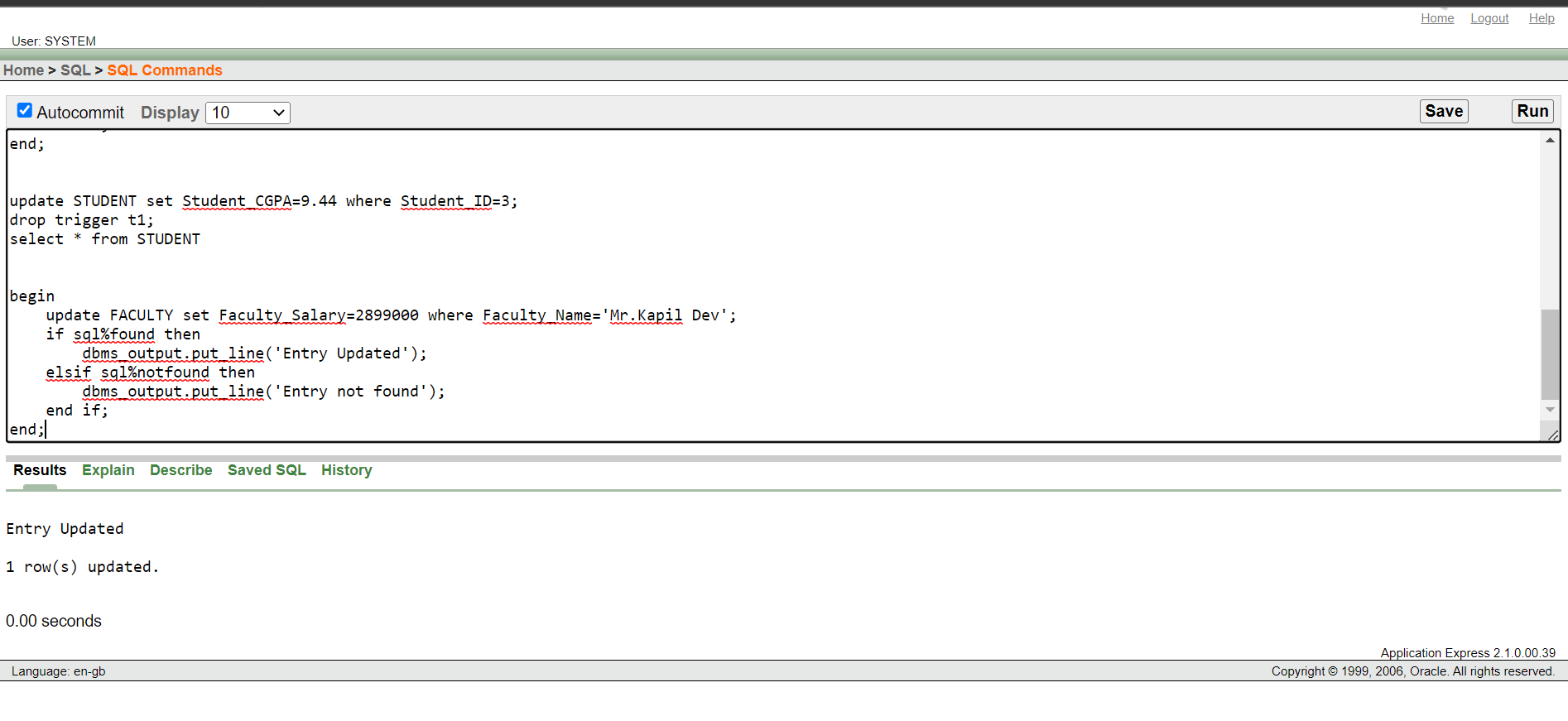
dbms\_output.put\_line('Entry Updated');

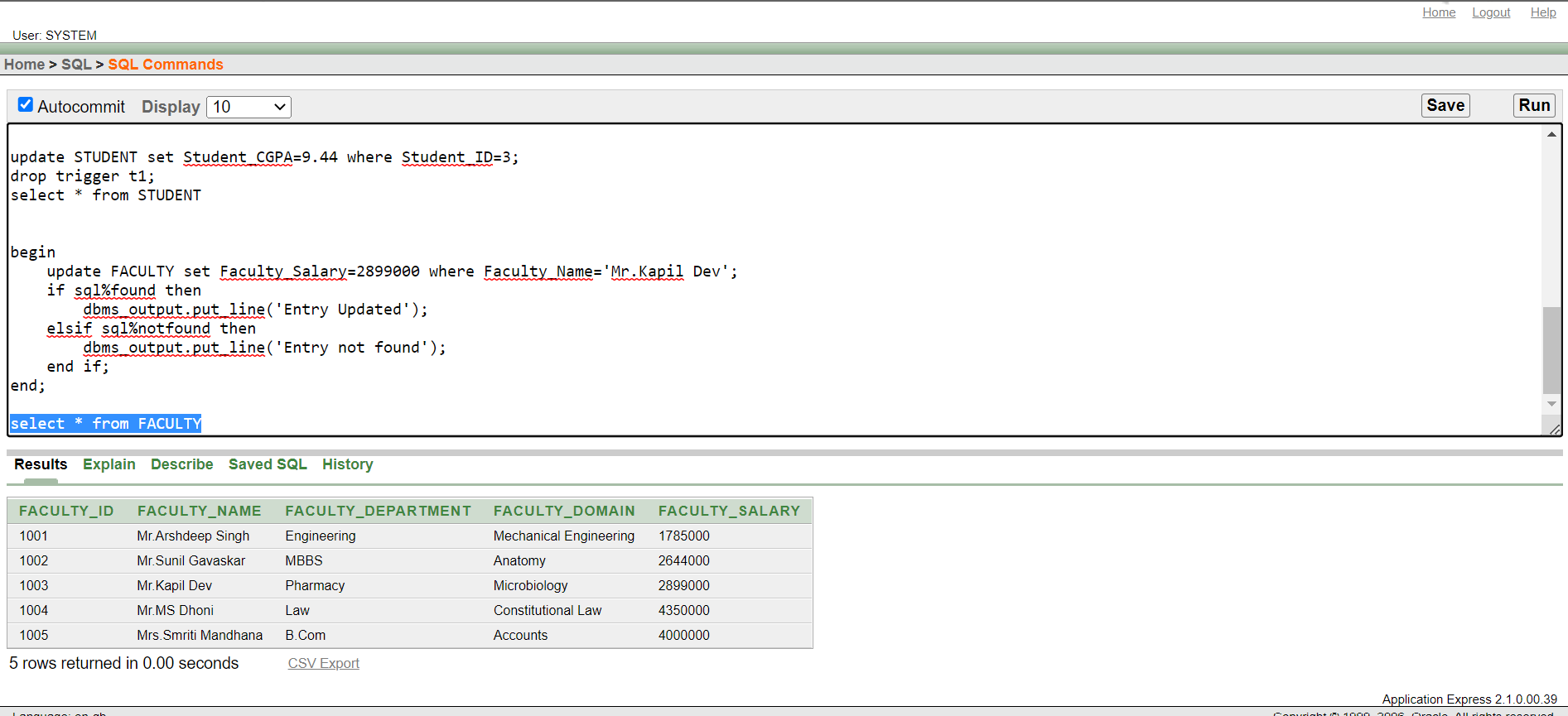
elsif sql%notfound then

dbms\_output.put\_line('Entry not found');

end if;

end;





1. PLSQL CODE IF ENTRY NOT FOUND:

begin

update FACULTY set Faculty\_Salary=2899000 where Faculty\_Name='Mr.Virat Kohli';

if sql%found then

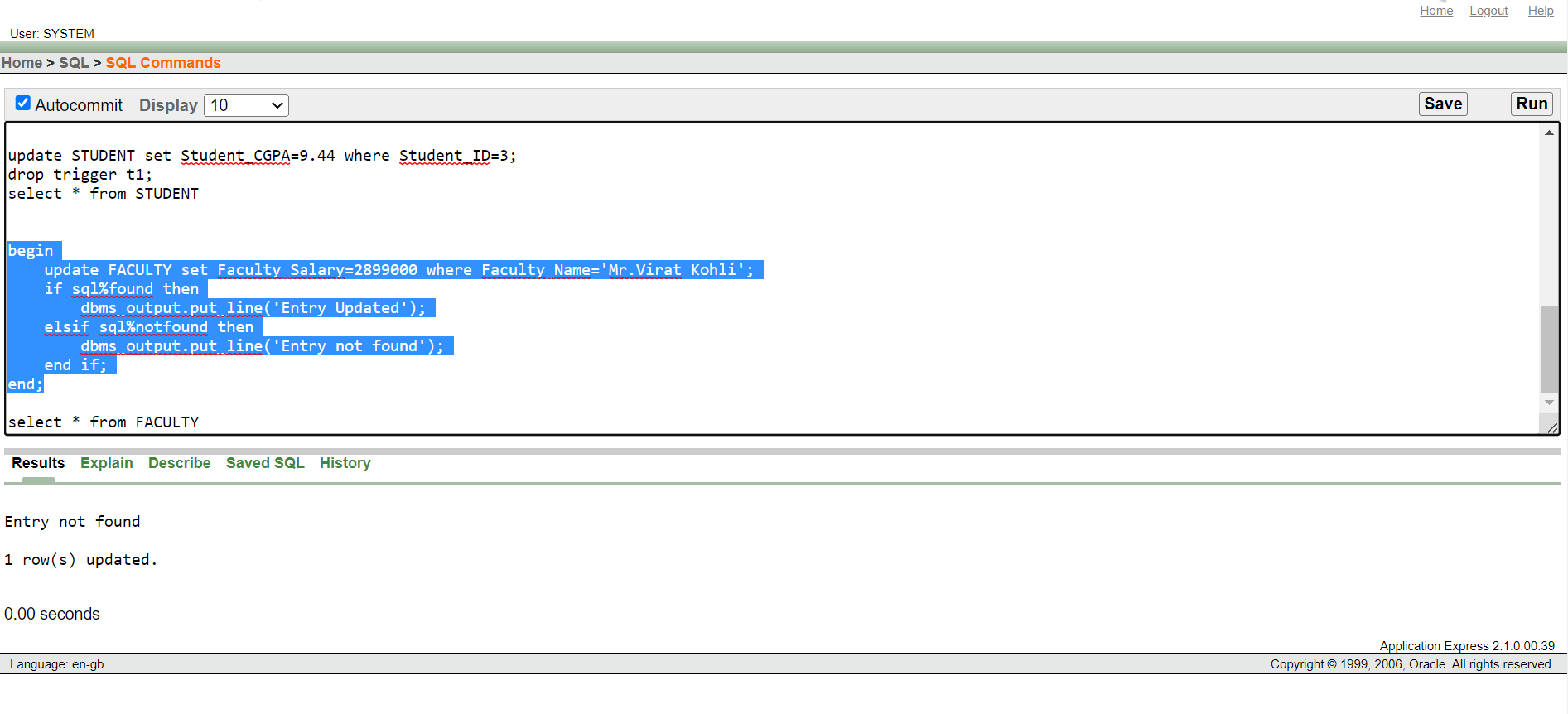
dbms\_output.put\_line('Entry Updated');

elsif sql%notfound then

dbms\_output.put\_line('Entry not found');

end if;

end;



THANK YOU