AMERICAN INTERNATIONAL UNIVERSITY–BANGLADESH (AIUB)

FACULTY OF SCIENCE & TECHNOLOGY

Introduction to Database

SPRING 2023-2024

Section: F, Group: 3

PROJECT ON

COACHING MANAGEMENT SYSTEM

Supervised By

SIFAT RAHMAN AHONA

Submitted By

Name	ID
MD MERAZ UDDIN	23-50322-1
SAKIB HOSSAIN ABIR	22-49309-3
SAMIA JANNAT LIZA	23-51085-1

Date of Submission: MAY 21, 2024

TABLE OF CONTENTS

Introduction

The Coaching Management System aims to enhance communication and information handling within coaching centers. This system simplifies administrative tasks and ensures prompt updates for students, teachers, and parents through an easy-to-use online platform. In today's dynamic educational environment, effective communication among students, teachers, and parents is vital. However, traditional methods often lead to missed information on classes, exams, and other significant events, hindering students' progress and creating unnecessary hurdles. To overcome this challenge, we developed the Coaching Management System to keep all stakeholders well-informed. This system offers a user-friendly solution for efficient information management and sharing. It delivers real-time updates on classes, exams, and events, enabling students to stay updated on their academic commitments. Teachers can utilize the system to update schedules, share results, and communicate essential information directly with students and parents. Moreover, a dedicated section for parents enables them to track their child's academic performance, encouraging greater parental involvement in their child's education. The primary goal of the Coaching Management System is to provide a seamless and accessible platform for sharing crucial information among students, teachers, and parents. Through the integration of technology, we aim to foster a more interconnected educational community, promoting academic success and collaboration of all stakeholders involved.

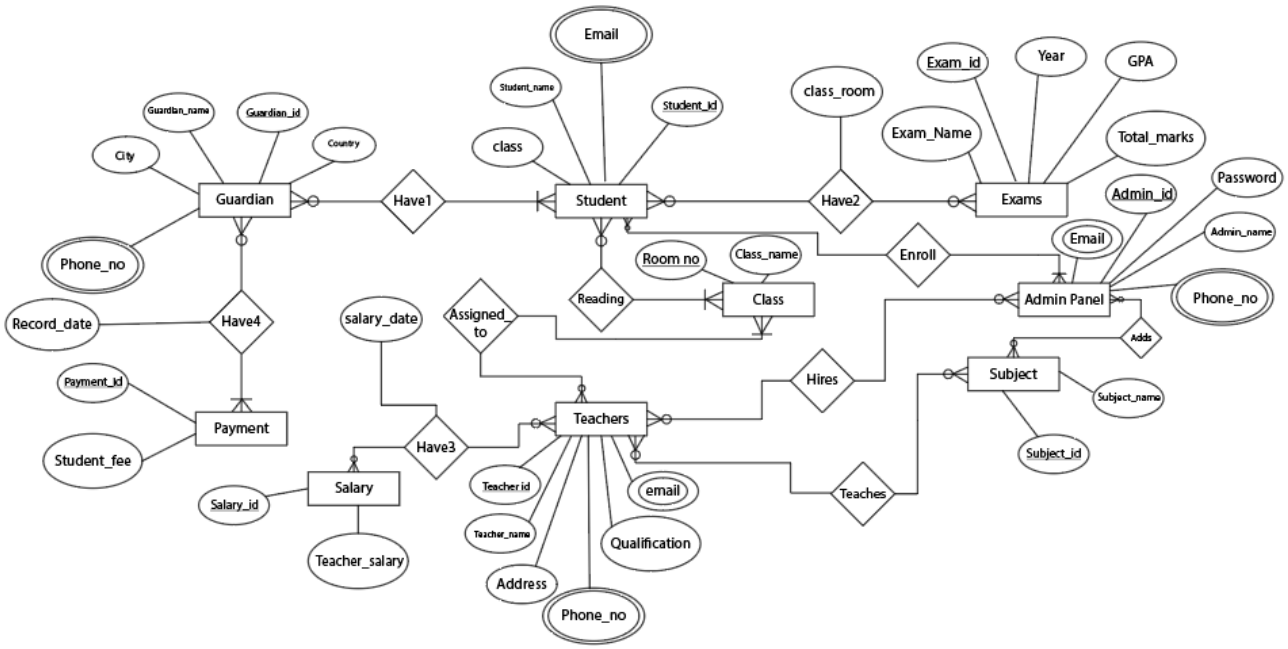
StudentID1: 23-50322-1 Name: MD MERAZ UDDIN	StudentID3: 23-51085-1 Name: SAMIA JANNAT LIZA
StudentID2: 22-49309-3 Name: SAKIB HOSSAIN ABIR	
CO2: Understand the fundamental concepts underlying database systems and gain hands-on experience with ER diagram Case study	
PO-c2: Develop process for complex computer science and engineering problems considering cultural and societal factors.	Marks

Scenario Description

In our Student Coaching Management System, each student is uniquely identified by the primary key, the Student ID, serving as the cornerstone for all student-related activities and information. Students also have some attributes like name, course, exam, phone number. This identifier plays a pivotal role in maintaining data integrity and facilitating efficient retrieval of student-specific details, forming the linchpin for various interactions within the system. Students, positioned as key entities, are intricately connected to diverse aspects crucial to their educational journey. The student-guardian relationship remains fundamental, with guardians uniquely identified by a Guardian ID. This connection ensures that students have dedicated support systems for their overall well-being, with guardian attributes, including Name, Phone Number, and Email (used as a password), intricately linked to corresponding student profiles. Moreover, students actively participate in academic assessments through exams, each uniquely distinguished by an Exam ID. This relationship empowers the systematic tracking of student performance and achievements across various examinations. The attributes within the student entity, including Name, Classroom, and Phone Number, collectively contribute to a comprehensive understanding of each student's academic profile. As we delve into the financial aspects, the Teachers Salary entity takes center stage, offering a nuanced perspective on the financial intricacies of the teaching faculty. Each teacher Salary record, identified by a unique ID, is seamlessly linked to specific teachers through the Teacher ID, teacher also has phone number, name, address, qualification, email establishing a robust foreign key relationship. The inclusion of the Salary Date ensures meticulous record-keeping, contributing to the transparency and accountability of our salary management system. The Amount attribute further encapsulates the financial dimension, detailing the specific payment amount associated with each salary record, providing a granular approach to salary transactions. Complementing the financial framework, the Admin Panel entity introduces administrative sophistication, defining user roles within the system. While it may not necessitate its own table in the database, the Admin Panel functions as a logical construct, influencing relationships such as "Hired by." This
relationship signifies that teachers are recruited into the system by the school administration, represented by the Admin Panel, enforcing organizational structure through user roles. Despite these innovations, core relationships like "Enrolled in," "Has Guardian," "Pays fees," "Teaches," and the recent addition, "Receives Salary," remain pillars defining the interconnected dynamics

of our educational system. As we navigate through these advancements, unresolved aspects like the attributes Reading and Room No. beckon for further clarification to enhance the overall comprehensibility and coherence of our Student Coaching Management System. Nevertheless, our system remains at the forefront, providing a sophisticated and dynamic platform for educators, administrators, and students alike.

ER DIAGRAM



Normalization

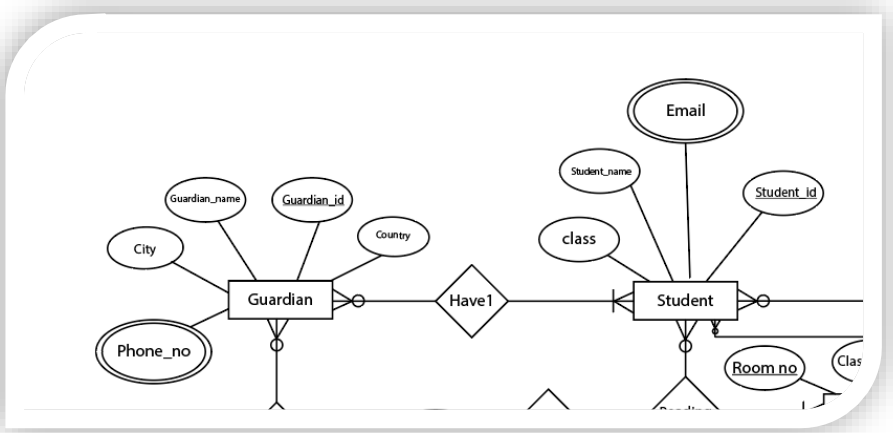


Fig-1: Normalization between Guardian and Student

Have1(Guardian_id, Guardian_name, City, Country, Phone_ no, Student_id, Student_name, Class, Email)

1NF: Phone_no, Email multivalued attribute

2NF: Guardian_id, Guardian_name, City, Country, Phone_ no
Student_id, Student_name, Class, Email
GS_id, Guardian_id, Student_id

3NF: Guardian_id, Guardian_name, ACid, Phone_ no
Student_id, Student_name, Class, Email
GS_id, Guardian_id, Student_id
ACid, City, Country

Table:

Guardian_id, Guardian_name, ACid, Phone_ no
Student_id, Student_name, Class, Email
GS_id, Guardian_id, Student_id
ACid, City, Country

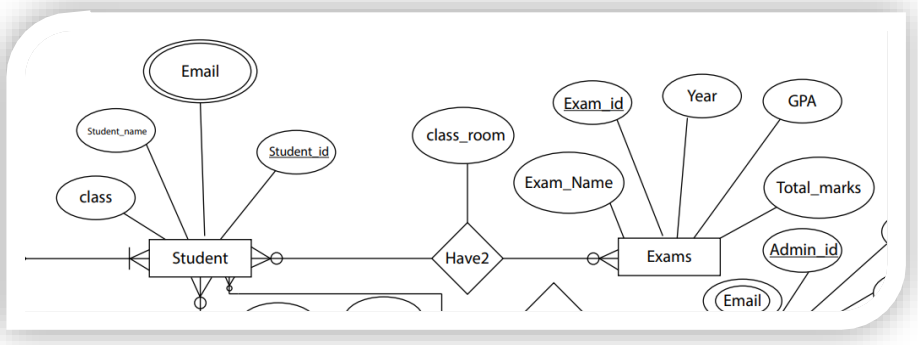


Fig-2: Normalization between Student and Exams

Have2 (Student_id, Student_name, Class, Email, Exam_id, Exam_name, Year, GPA, Total_marks)

1NF: Email multivalued attribute

2NF: Student_id, Student_name, Class, Email
Exam_id, Exam_name, Year, GPA, Total_marks
Exam_id, Student_id, Exam_name

3NF: Student_id, Student_name, Class, Email
Exam_id, Exam_name, Year, GPA, Total_marks
Exam_id, Student_id, Exam_name
No transitive dependency

Table:

Student_id, Student_name, Class, Email
Exam_id, Exam_name, Year, GPA, Total_marks
Exam_id, Student_id, Exam_name

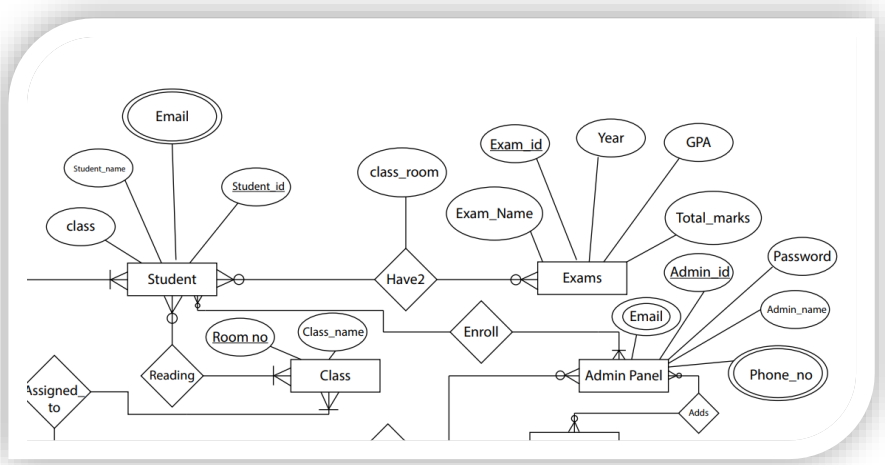


Fig-3: Normalization between Student and Admin Panel

Enroll (Student_id, Student_name, Class, Email, Admin_id, Admin_name, Password, Phone_no, Email)

1NF: Phone_no, Email multivalued attribute

2NF: Student_id, Student_name, Class, Email
Admin_id, Admin_name, Password, Phone_no, Email
SA_id, Student_id, Admin_id

3NF: Student_id, Student_name, Class, Email
Admin_id, Admin_name, Password, Phone_no, Email
SA_id, Student_id, Admin_id
No transitive dependency

Table:

Student_id, Student_name, Class, Email
Admin_id, Admin_name, Password, Phone_no, Email
SA_id, Student_id, Admin_id

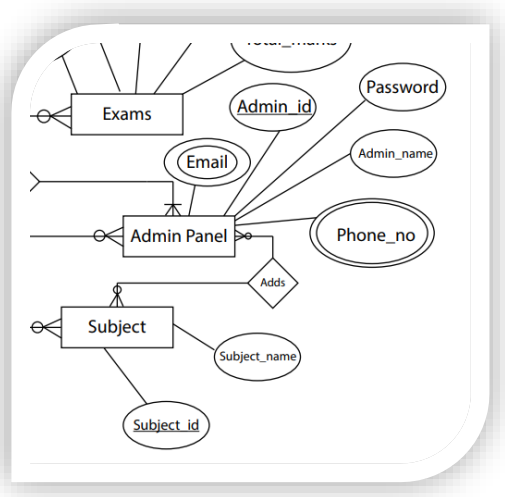


Fig-4: Normalization between Admin pamel and Subject

Adds (Admin_id, Admin_name, Password, Phone_no, Email, Subject_id, Subject_name)

1NF: Email, Phone_no multivalued attribute

2NF: Admin_id, Admin_name, Password, Phone_no, Email
Subject_id, Subject_name
AS_id, Admin_id, Subject_id

3NF: Admin_id, Admin_name, Password, Phone_no, Email
Subject_id, Subject_name
AS_id, Admin_id, Subject_id
No transitive dependency

Table:

Admin_id, Admin_name, Password, Phone_no, Email
Subject_id, Subject_name
AS_id, Admin_id, Subject_id

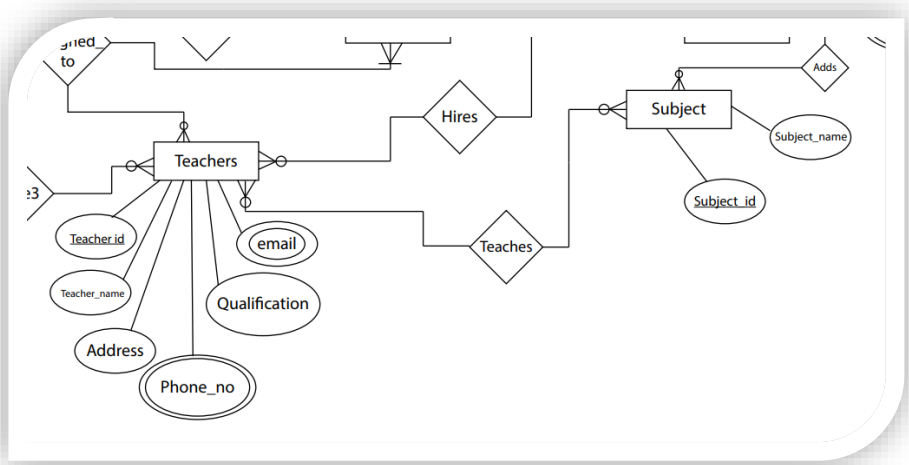


Fig-5: Normalization between Teachers and Subject

Teaches (Teacher_id, Teacher_name, Address, Qualification, Phone_no, Email, Subject_id, Subject_name)

1NF: Email,Phone_no multivalued attribute

2NF: Teacher_id, Teacher_name, Address, Qualification, Phone_no, Email
Subject_id, Subject_name
TS_id, Teacher_id, Subject_id

3NF: Teacher_id, Teacher_name, Address, Qualification, Phone_no, Email
Subject_id, Subject_name
TS_id, Teacher_id, Subject_id
No transitive dependency

Table:

Teacher_id, Teacher_name, Address, Qualification, Phone_no, Email
Subject_id, Subject_name
TS_id, Teacher_id, Subject_id

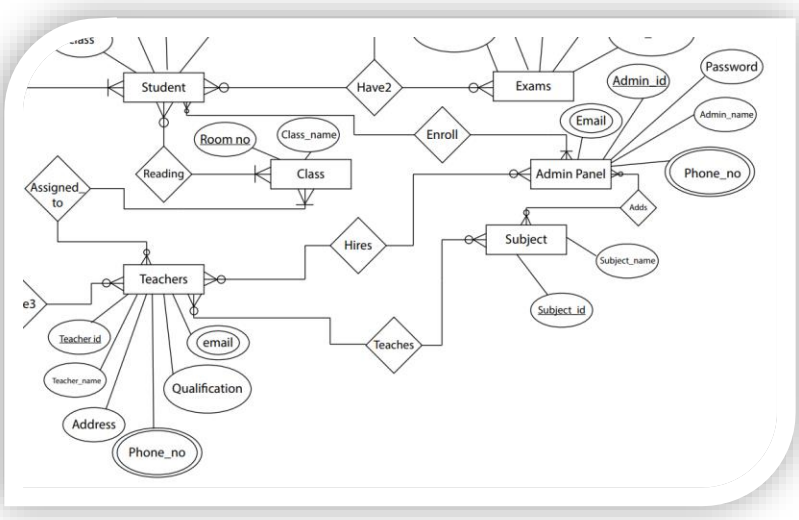


Fig-6: Normalization between Teachers and Admin panel

Hires (Teacher_id, Teacher_name, Address, Qualification, Phone_no, Email, Admin_id, Admin_name, Password, Phone_no, Email)

1NF: Email, Phone_no multivalued attribute

2NF: Teacher_id, Teacher_name, Address, Qualification, Phone_no, Email
Admin_id, Admin_name, Password, Phone_no, Email
TA_id, Teacher_id, Admin_id

3NF: Teacher_id, Teacher_name, Address, Qualification, Phone_no, Email
Admin_id, Admin_name, Password, Phone_no, Email
TA_id,Teacher_id,Admin_id
No transitive dependency

Table:

Teacher_id, Teacher_name, Address, Qualification, Phone_no, Email
Admin_id, Admin_name, Password, Phone_no, Email
TA_id,Teacher_id,Admin_id

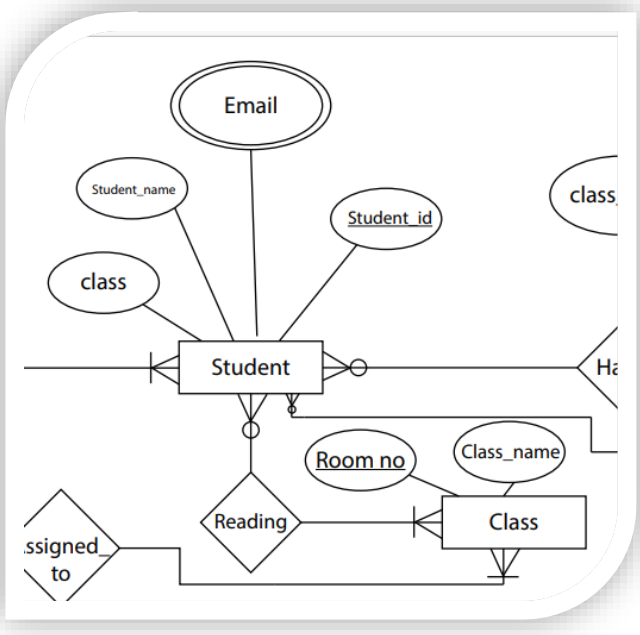


Fig-7: Normalization between Student and Class

Reading (Student_id, Student_name, Class, Email, Room_no, Class_name)

1NF: Email multivalued attribute

2NF: Student_id, Student_name, Class, Email
Room_no, Class_name
SR_id, Student_id,Room_no

3NF: Student_id, Student_name, Class, Email
Room_no, Class_name
SR_id, Student_id, Room_no
No transitive dependency

Table:

Student_id, Student_name, Class, Email
Room_no, Class_name
SR_id, Student_id, Room_no

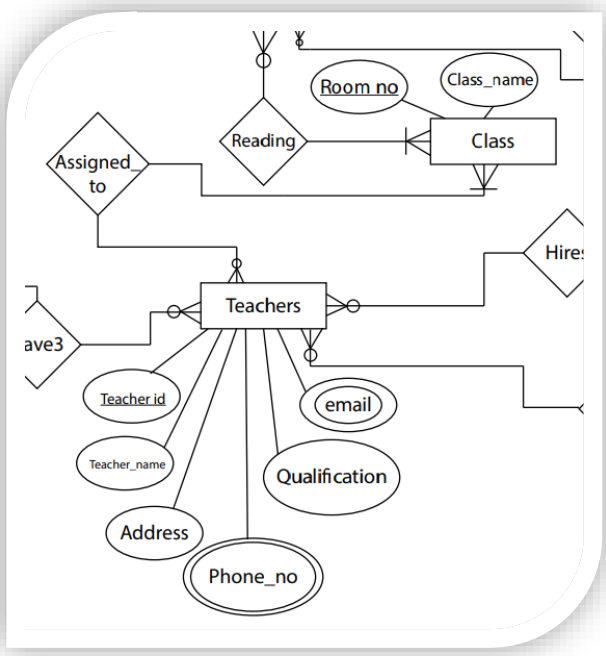


Fig-8: Normalization between Teachers and Class

Assigned_to (Teacher_id, Teacher_name, Address, Qualification, Phone_no, Email, Room_no, Class_name)

1NF: Email multivalued attribute

2NF: Teacher_id, Teacher_name, Address, Qualification, Phone_no, Email
Room_no, Class_name
TR_id, Teacher_id, Room _no

3NF: Teacher_id, Teacher_name, Address, Qualification, Phone_no, Email
Room_no, Class_name
TR_id, Teacher_id, Room _no
No transitive dependency

Table:

Teacher_id, Teacher_name, Address, Qualification, Phone_no, Email
Room_no, Class_name
TR_id, Teacher_id, Room

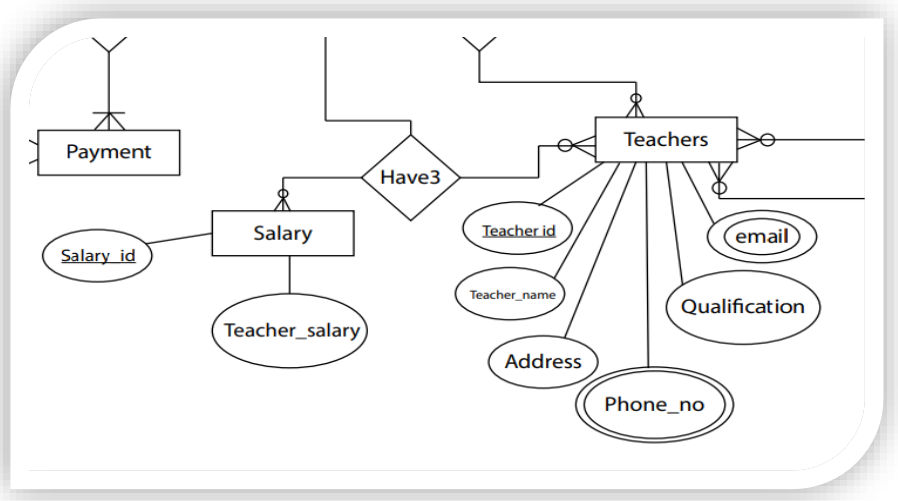


Fig-9: Normalization between Teachers and Salary

Have3 (Teacher_id, Teacher_name, Address, Qualification, Phone_no, Email, Salary_id, Teacher_salary)

1NF: Email, Phone_no multivalued attribute

2NF: Teacher_id, Teacher_name, Address, Qualification, Phone_no, Email
Salary_id, Teacher_salary
TS_id,Teacher_id, Salary_id

3NF: Teacher_id, Teacher_name, Address, Qualification, Phone_no, Email
Salary_id, Teacher_salary
TS_id,Teacher_id, Salary_id
No transitive dependency

Table:

Teacher_id, Teacher_name, Address, Qualification, Phone_no, Email
Salary_id, Teacher_salary
TS_id,Teacher_id, Salary_id

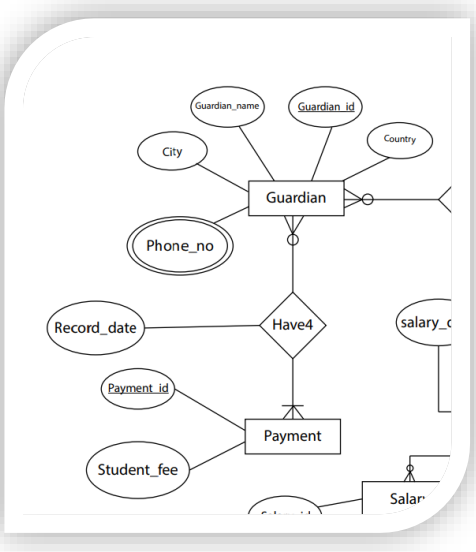


Fig-10: Normalization between Guardian and Payment

Have4 (Guardian_id, Guardian_name, Address, City, Country, Phone_ no, Payment_id, Student_fee)

1NF: Phone_no multivalued attribute

2NF: Guardian_id, Guardian_name, Address, City, Country, Phone_ no
Payment_id, Student_fee
GP_id, Guardian_id, Payment_id

3NF: Guardian_id, Guardian_name, ACid, Phone_ no
Payment_id, Student_fee
GP_id, Guardian_id, Payment_id
ACid, Address, City, Country

Table:

Guardian_id, Guardian_name, ACid, Phone_ no
Payment_id, Student_fee
GP_id, Guardian_id, Payment_id
ACid, Address, City, Country

Finalization

Total Table:

✓1. Guardian_id, Guardian_name, ACid, Phone_ no
✓2.Student_id, Student_name, Class, Email
✓3.GS_id, Guardian_id, Student_id
✓4.ACid, City, Country
✗5.Student_id, Student_name, Class, Email
✓6.Exam_id, Exam_name, Year, GPA, Total_marks
✓7.SE_id,Student_id,Exam_id
✗8.Student_id, Student_name, Class, Email
✓9.Admin_id, Admin_name, Password, Phone_no, Email
✓10.SA_id, Student_id, Admin_id
✗11.Admin_id, Admin_name, Password, Phone_no, Email
✓12.Subject_id, Subject_name
✓13.AS_id, Admin_id, Subject_id
✓14.Teacher_id, Teacher_name, Address, Qualification, Phone_no, Email
✗15.Subject_id, Subject_name
✓16.TS_id, Teacher_id, Subject_id
✗17.Teacher_id, Teacher_name, Address, Qualification, Phone_no, Email
✗18.Admin_id, Admin_name, Password, Phone_no, Email
✓19.TA_id,Teacher_id,Admin_id
✗20.Student_id, Student_name, Class, Email
✗21.Room_no, Class_name
✓22.SR_id, Student_id,Room_no
✗23.Teacher_id, Teacher_name, Address, Qualification, Phone_no, Email
✓24.Room_no, Class_name
✓25.TR_id, Teacher_id, Room _no
✗26.Teacher_id, Teacher_name, Address, Qualification, Phone_no, Email
✓27.Salary_id, Teacher_salary
✓28.TS_id,Teacher_id, Salary_id
✗29.Guardian_id, Guardian_name, ACid, Phone_ no
✓30.Payment_id, Student_fee
✓31.GP_id, Guardian_id, Payment_id
✗32.ACid, Address, City, Country

Final Table:

1.Guardian_id, Guardian_name, ACid, Phone_ no

2.Student_id, Student_name, Class, Email

3.GS_id, Guardian_id, Student_id

4.ACid, City, Country

5.Exam_id, Exam_name, Year, GPA, Total_marks

6.SE_id, Student_id, Exam_id

7.Admin_id, Admin_name, Password, Phone_no, Email

8.SA_id, Student_id, Admin_id

9.Subject_id, Subject_name

10.AS_id, Admin_id, Subject_id

11.Teacher_id, Teacher_name, Address, Qualification, Phone_no, Email

12.TS_id, Teacher_id, Subject_id

13.TA_id, Teacher_id, Admin_id

14.Room_no, Class_name

15.TR_id, Teacher_id, Room _no

16.SR_id, Student_id, Room_no

17.Salary_id, Teacher_salary

18.TS_id, Teacher_id, Salary_id

19.Payment_id, Student_fee

20 .GP_id, Guardian_id, Payment_id

Table Creation (DDL Operations) and Data Insertion

StudentID1: 22-49309-3 Name: SAKIB HOSSAIN ABIR	StudentID3: 23-51085-1 Name: SAMIA JANNAT LIZA
StudentID2: 23-50322-1 Name: MD MERAZ UDDIN	
CO4: Creating DML, DDL using Oracle and connection with ODBC/JDBC for existing JAVA application	
PO-e-2: Use modern engineering and IT tools for prediction and modeling of complex computer science and engineering problem	Marks

1. Guardian Table:

```
create table Guardian
 (
  Guardian_id varchar2(20) constraint Guardian_Gid_pk primary key,
  Guardian_name varchar2(30) not null,
 AC_id varchar2(20) constraint Guardian_acid_fk  references AC(AC_id),
 Phone_no varchar2 (11) unique
)
```

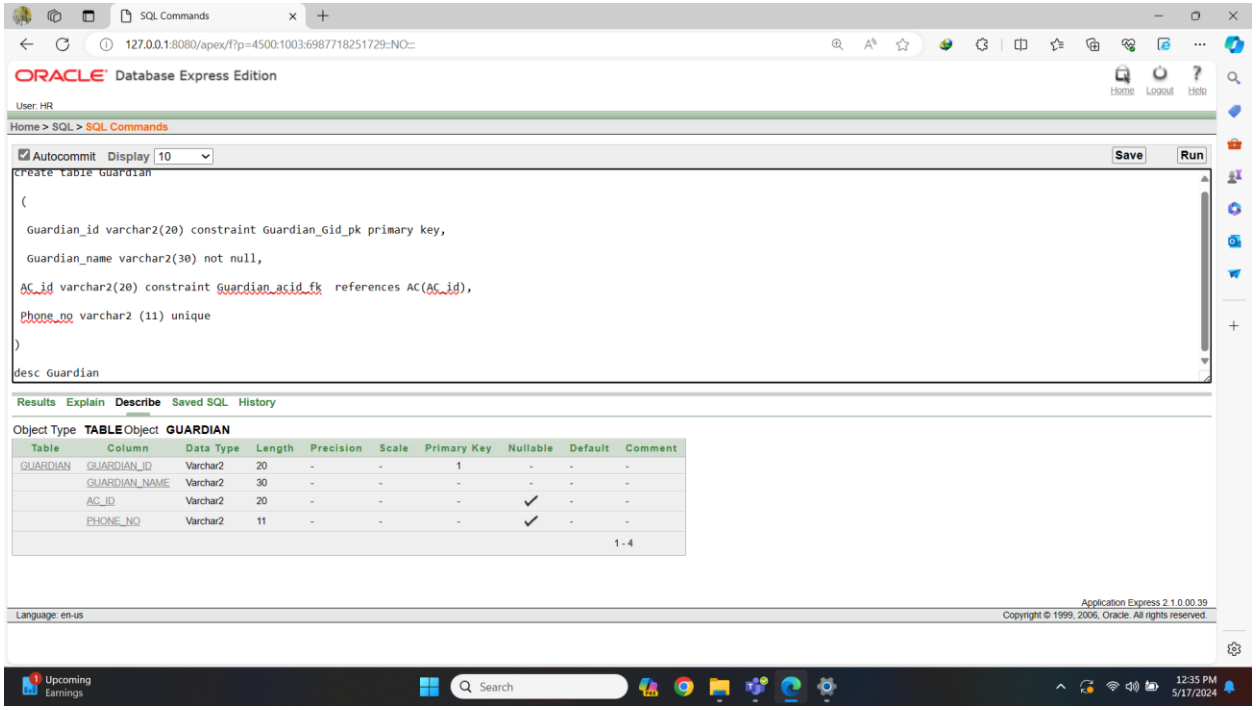


Fig-1: Guardian Table Create

insert into Guardian (Guardian_id, Guardian_name, AC_id, Phone_no) values ('G001', 'Mohammad Rahman', 'AC001', '01712345678')
insert into Guardian (Guardian_id, Guardian_name, AC_id, Phone_no) values ('G002', 'Fatima Akhtar', 'AC002', '01809876543')
insert into Guardian (Guardian_id, Guardian_name, AC_id, Phone_no) values ('G003', 'Abdul Haque', 'AC003', '01955512345')
insert into Guardian (Guardian_id, Guardian_name, AC_id, Phone_no) values ('G004', 'Aisha Begum', 'AC004', '01799988877')
insert into Guardian (Guardian_id, Guardian_name, AC_id, Phone_no) values ('G005', 'Sajjad Ali', 'AC005', '01811122233')
insert into Guardian (Guardian_id, Guardian_name, AC_id, Phone_no) values ('G006', 'Nazma Islam', 'AC006', '01744455566')
insert into Guardian (Guardian_id, Guardian_name, AC_id, Phone_no) values ('G007', 'Mizanur Rahman', 'AC007', '01977788899')
insert into Guardian (Guardian_id, Guardian_name, AC_id, Phone_no) values ('G008', 'Shahnaz Begum', 'AC008', '01866677788')
insert into Guardian (Guardian_id, Guardian_name, AC_id, Phone_no) values ('G009', 'Kamal Hossain', 'AC009', '01733344455')
insert into Guardian (Guardian_id, Guardian_name, AC_id, Phone_no) values ('G010', 'Farida Akhter', 'AC010', '01922233344')

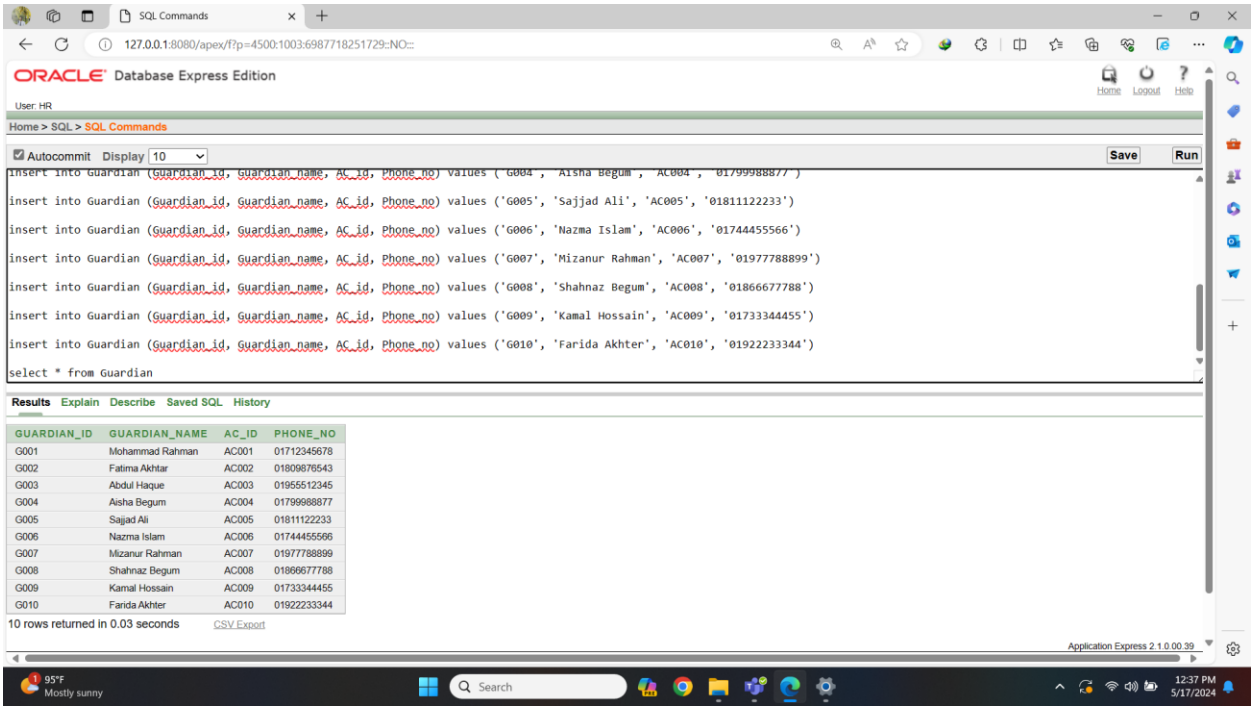


Fig-2: Guardian Table Data

2. Student Table:

```
create table Student
(
 Student_id varchar2(20) constraint Students_Sid_pk primary key,
 Student_name varchar2(30) not null,
 Class varchar2(10),
 Email varchar2(20) unique
)
```

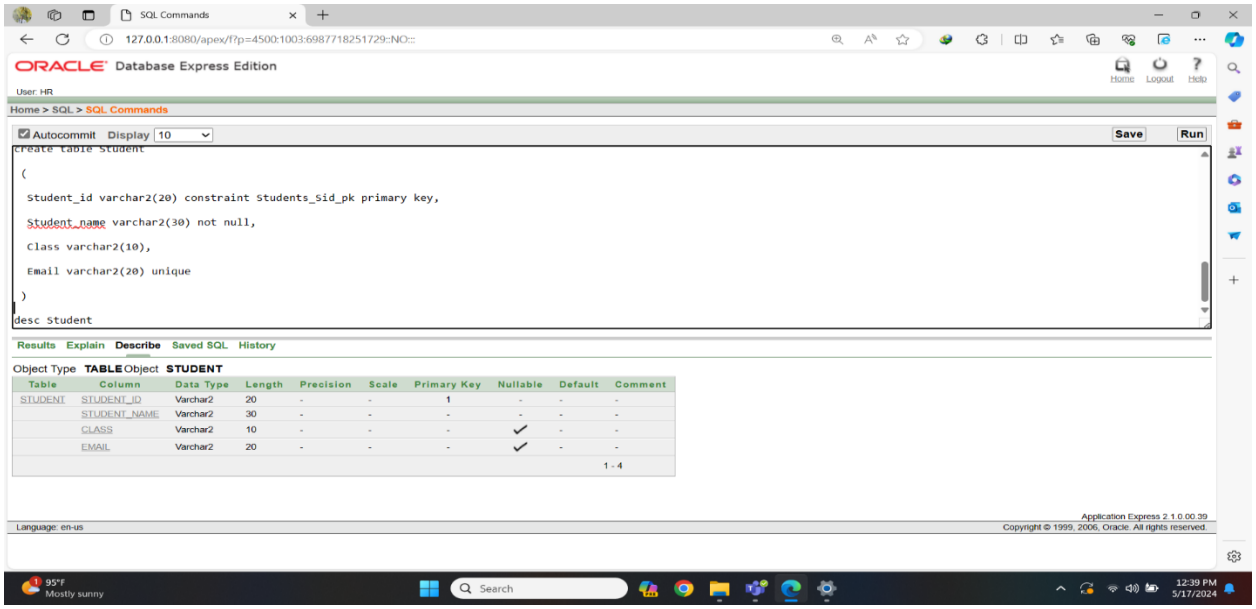


Fig-3: Student Table Create

insert into Student (Student_id, Student_name, Class, Email) values ('S001', 'Abdul Rahman', '10', 'abdul@gmail.com')
insert into Student (Student_id, Student_name, Class, Email) values ('S002', 'Fatima Begum', '9', 'fatima@outlook.com')
insert into Student (Student_id, Student_name, Class, Email) values ('S003', 'Ayesha Khan', '8', 'ayesha@yahoo.com')
insert into Student (Student_id, Student_name, Class, Email) values ('S004', 'Mohammad Ali', '10', 'mohammad@hotmail.com')
insert into Student (Student_id, Student_name, Class, Email) values ('S005', 'Nusrat Jahan', '9', 'nusrat@icloud.com')
insert into Student (Student_id, Student_name, Class, Email) values ('S006', 'Arif Hasan', '8', 'arif@protonmail.com')
insert into Student (Student_id, Student_name, Class, Email) values ('S007', 'Sadia Akter', '10', 'sadia@yandex.com')

insert into Student (Student_id, Student_name, Class, Email) values ('S008', 'Farhan Ahmed', '9', 'farhan@aol.com')
insert into Student (Student_id, Student_name, Class, Email) values ('S009', 'Nazia Islam', '8', 'nazia@mail.com')
insert into Student (Student_id, Student_name, Class, Email) values ('S010', 'Imran Khan', '10', 'imran@live.com')

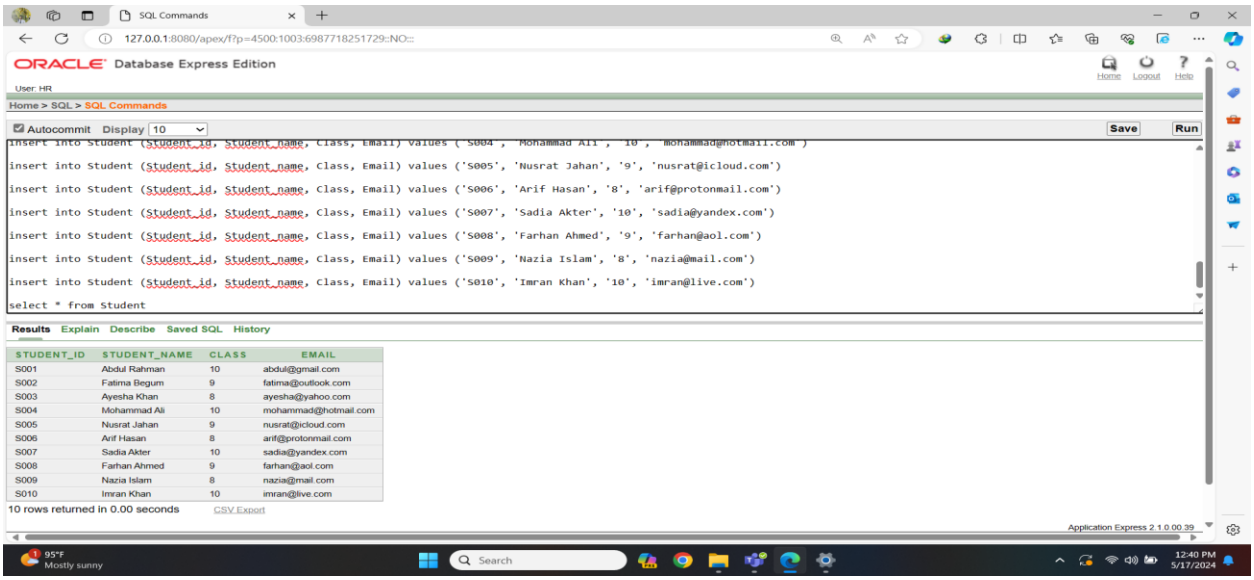


Fig-4: Student Table Data

3. Guardian Student table:

```
create table GuardianStudent
(
  GS_id varchar2(20) constraint GuardianStudent_GSid_pk primary key,
  Guardian_id varchar2(20) constraint GuardianStudent_Guardianid_fk references
Guardian(Guardian_id),
  Student_id varchar2(20) constraint GuardianStudent_Studentid_fk references
Student(Student_id)
)
```

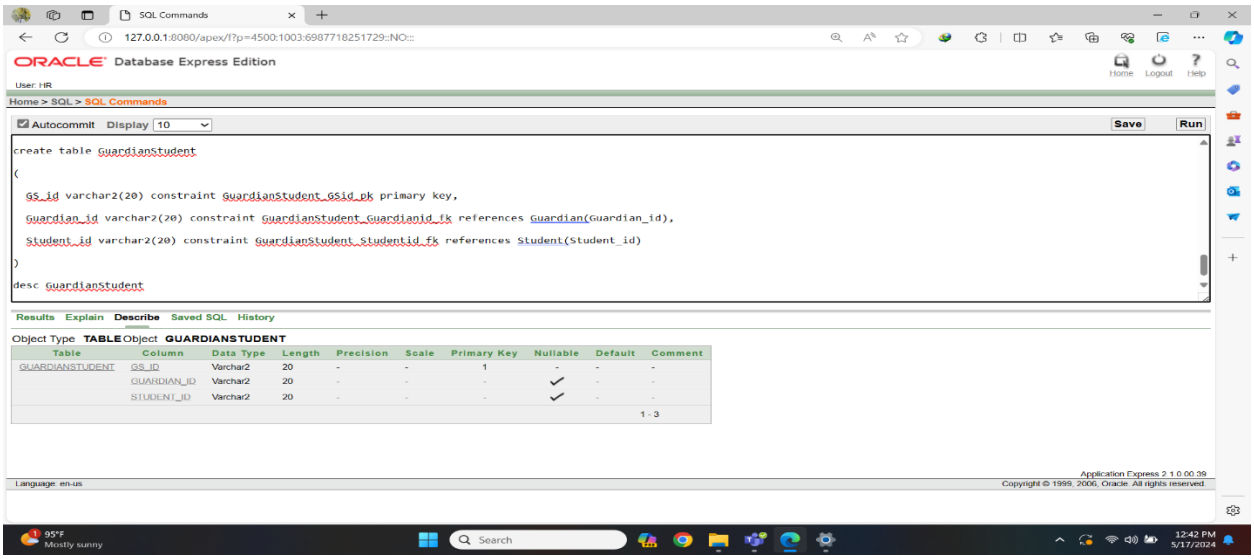


Fig-5: Guardian-Student Table Create

insert into GuardianStudent (GS_id, Guardian_id, Student_id) values ('GS001', 'G001', 'S001')
insert into GuardianStudent (GS_id, Guardian_id, Student_id) values ('GS002', 'G002', 'S002')
insert into GuardianStudent (GS_id, Guardian_id, Student_id) values ('GS003', 'G003', 'S003')
insert into GuardianStudent (GS_id, Guardian_id, Student_id) values ('GS004', 'G004', 'S004')
insert into GuardianStudent (GS_id, Guardian_id, Student_id) values ('GS005', 'G005', 'S005')
insert into GuardianStudent (GS_id, Guardian_id, Student_id) values ('GS006', 'G006', 'S006')
insert into GuardianStudent (GS_id, Guardian_id, Student_id) values ('GS007', 'G007', 'S007')
insert into GuardianStudent (GS_id, Guardian_id, Student_id) values ('GS008', 'G008', 'S008')
insert into GuardianStudent (GS_id, Guardian_id, Student_id) values ('GS009', 'G009', 'S009')
insert into GuardianStudent (GS_id, Guardian_id, Student_id) values ('GS010', 'G010', 'S010')

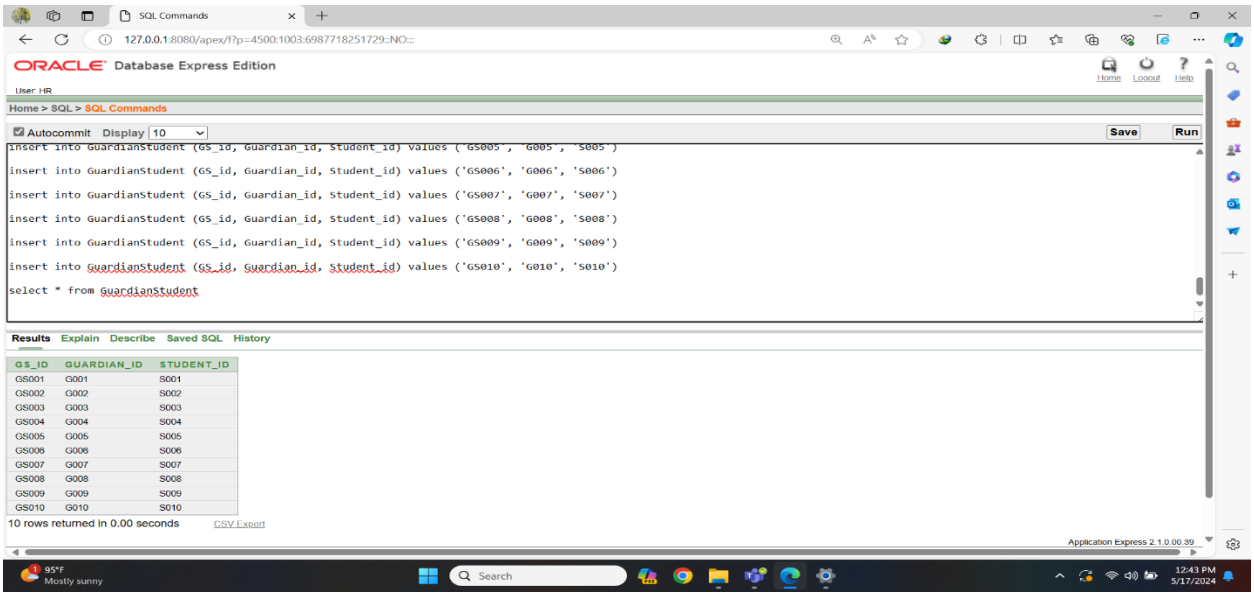


Fig-6: Guardian-Student Tabel Data

4. Address City Table:

```
create table AddressCity
(
  AC_id varchar(20) constraint AddressCity_acid_pk primary key,
  City varchar2(20) not null,
  Country varchar2(20) default 'Bangladesh'
)
```

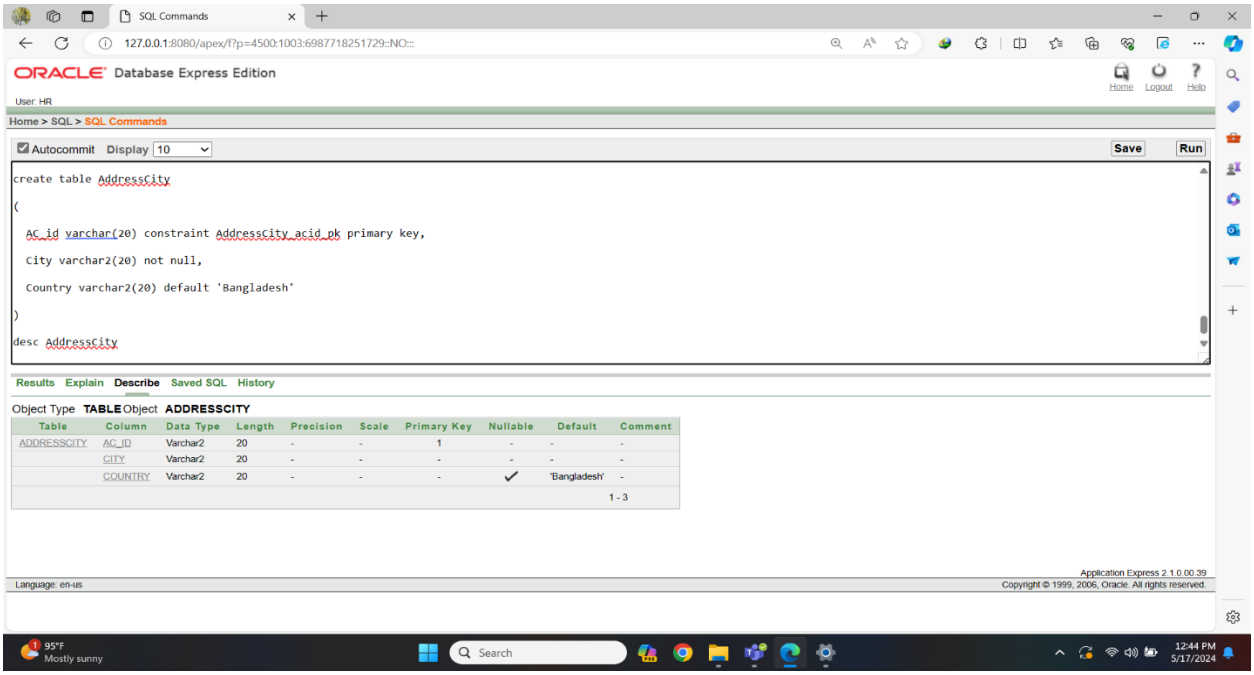


Fig-7:Address-City Table Create

```
insert into AddressCity (AC_id, City, Country) values ('AC001', 'Dhaka', 'Bangladesh')
insert into AddressCity (AC_id, City, Country) values ('AC002', 'Chittagong', 'Bangladesh')
insert into AddressCity (AC_id, City, Country) values ('AC003', 'Sylhet', 'Bangladesh')
insert into AddressCity (AC_id, City, Country) values ('AC004', 'Khulna', 'Bangladesh')
insert into AddressCity (AC_id, City, Country) values ('AC005', 'Rajshahi', 'Bangladesh')
insert into AddressCity (AC_id, City, Country) values ('AC006', 'Barisal', 'Bangladesh')
insert into AddressCity (AC_id, City, Country) values ('AC007', 'Comilla', 'Bangladesh')
insert into AddressCity (AC_id, City, Country) values ('AC008', 'Mymensingh', 'Bangladesh')
insert into AddressCity (AC_id, City, Country) values ('AC009', 'Rangpur', 'Bangladesh')
insert into AddressCity (AC_id, City, Country) values ('AC010', 'Dinajpur', 'Bangladesh')
```

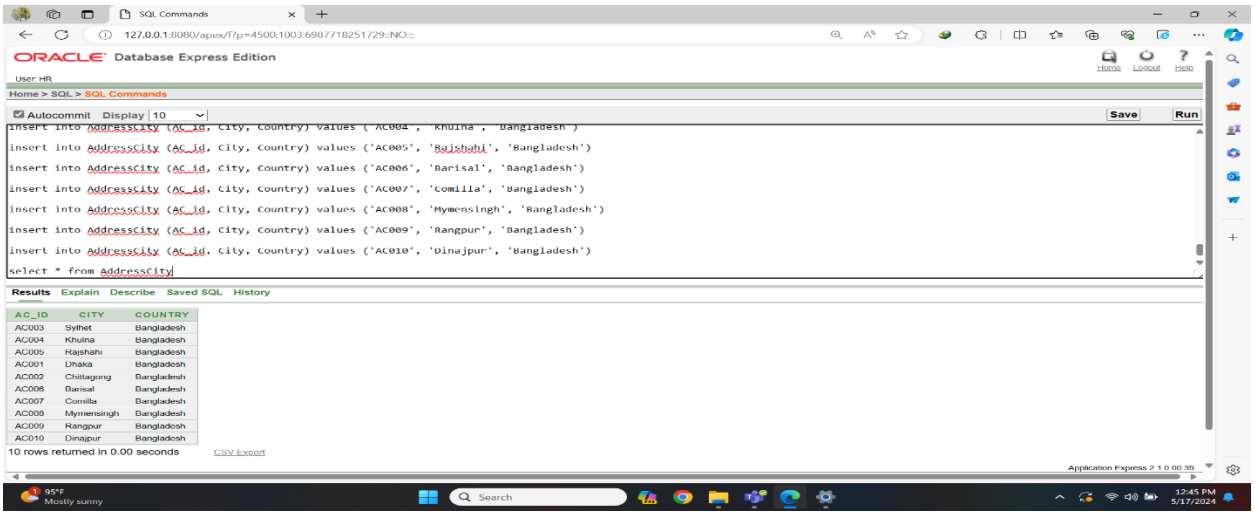


Fig-8: Address-City Table Data

5. Exam Table:

```
 create table Exam
(
   Exam_id varchar2(20) constraint Exam_Eid_pk primary key,
   Exam_name varchar2(30) unique,
   Year varchar2(10),
   GPA number(3,2),
   Total_marks number(10)
 )
```

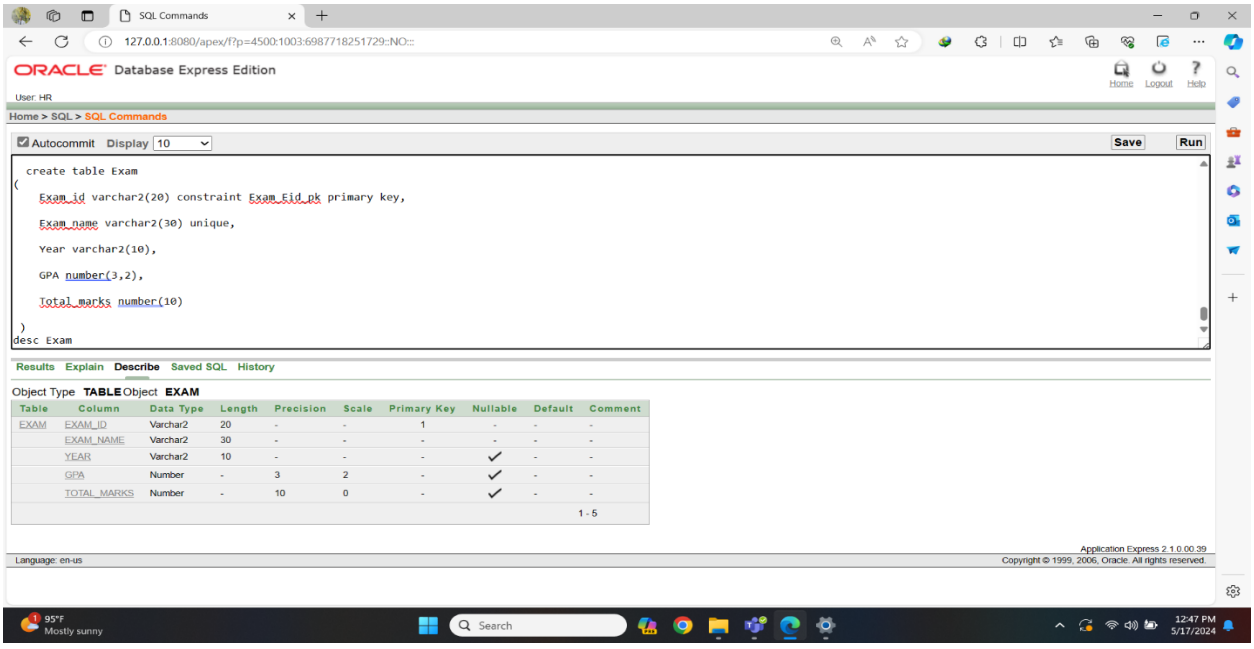


Fig-9: Exam Table Create

```
insert into Exam (Exam_id, Exam_name, Year, GPA, Total_marks) values ('E001', 'Mathematics', '2023', 5.00, 1000)
insert into Exam (Exam_id, Exam_name, Year, GPA, Total_marks) values ('E002', 'Science', '2023', 4.80, 1200)
insert into Exam (Exam_id, Exam_name, Year, GPA, Total_marks) values ('E003', 'English', '2023', 4.20, 800)
insert into Exam (Exam_id, Exam_name, Year, GPA, Total_marks) values ('E004', 'History', '2023', 3.00, 900)
insert into Exam (Exam_id, Exam_name, Year, GPA, Total_marks) values ('E005', 'Geography', '2023', 3.60, 1100)
insert into Exam (Exam_id, Exam_name, Year, GPA, Total_marks) values ('E006', 'Biology', '2023', 4.50, 1300)
insert into Exam (Exam_id, Exam_name, Year, GPA, Total_marks) values ('E007', 'Chemistry', '2023', 4.20, 1250)
insert into Exam (Exam_id, Exam_name, Year, GPA, Total_marks) values ('E008', 'Physics', '2023', 4.30, 1150)
insert into Exam (Exam_id, Exam_name, Year, GPA, Total_marks) values ('E009', 'Economics', '2023', 3.60, 950)
insert into Exam (Exam_id, Exam_name, Year, GPA, Total_marks) values ('E010', 'Database', '2023', 4.80, 1350)
```

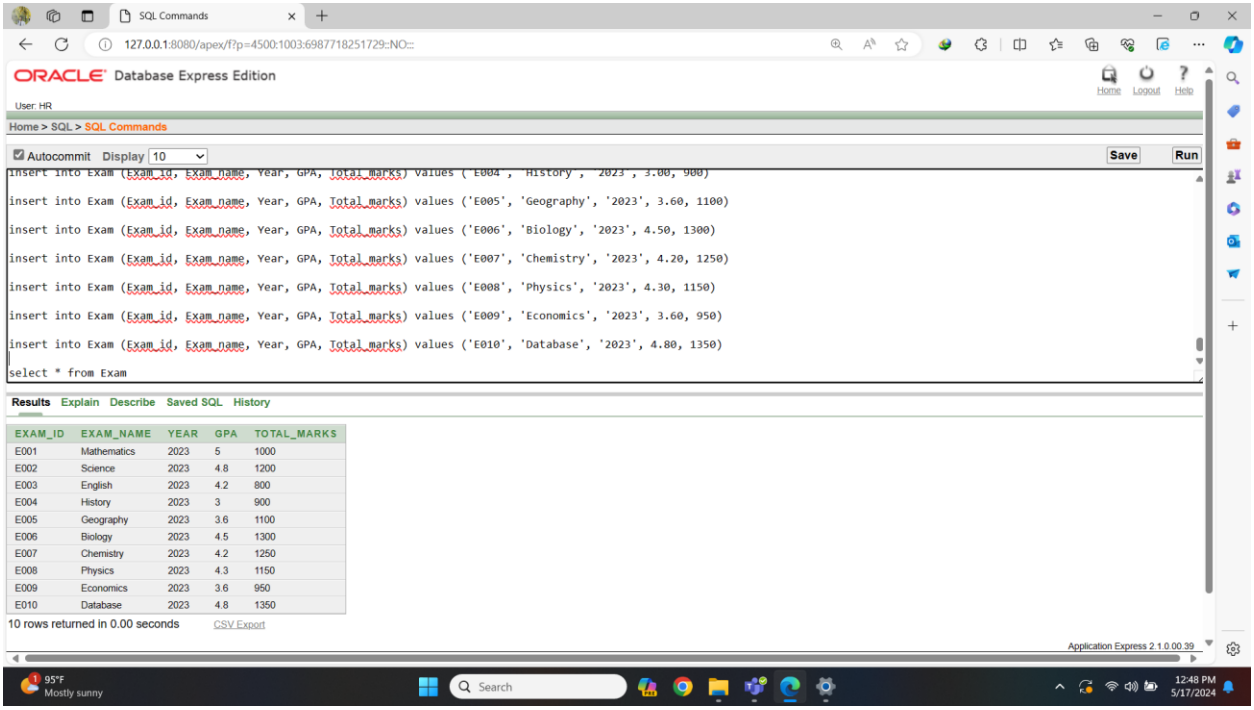


Fig-10: Exam Table Data

6. Student Exam Table:

```
create table StudentExam
(
SE_id varchar2(20) constraint StudentExam_SEid_pk primary key,
Student_id varchar2(20) constraint StudentExam_Studentid_fk references Student(Student_id),
Exam_id varchar2(20) constraint StudentExam_Examid_fk references Exam(Exam_id)
 )
```

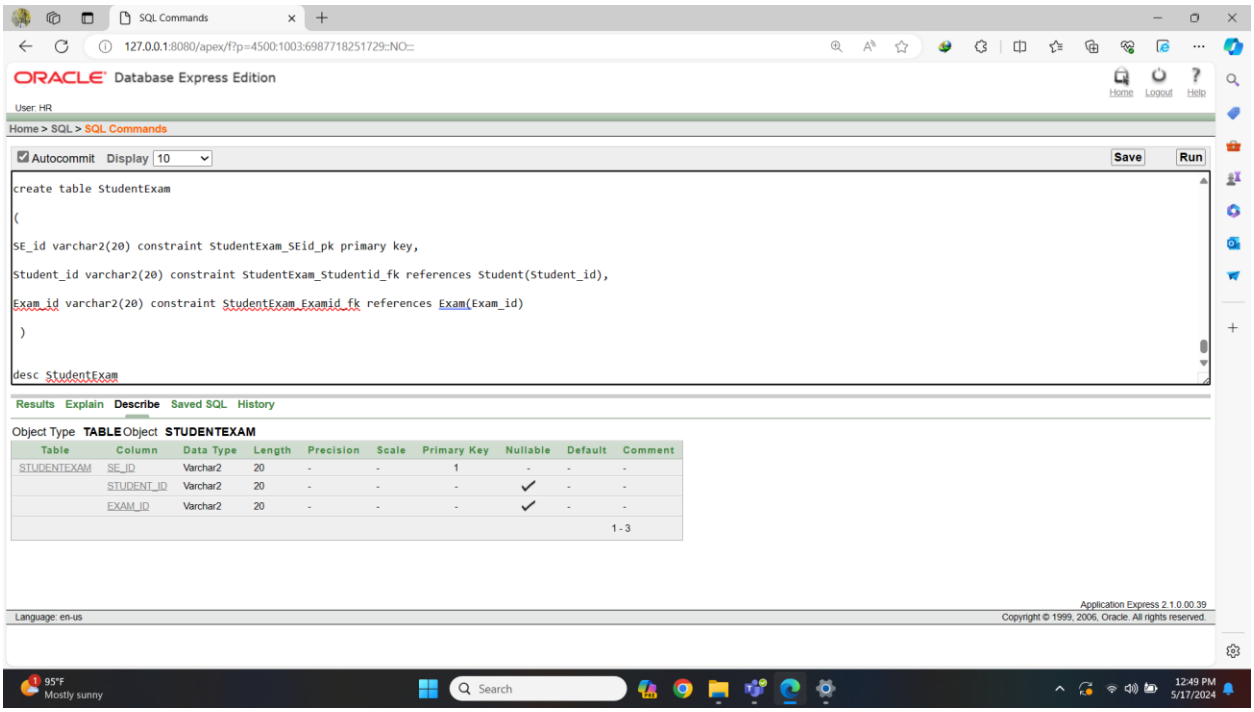


Fig-11: Student_Exam Table Create

```
insert into StudentExam (SE_id, Student_id, Exam_id) values ('se001', 's001', 'e001')
insert into StudentExam (SE_id, Student_id, Exam_id) values ('se002', 's002', 'e002')
insert into StudentExam (SE_id, Student_id, Exam_id) values ('se003', 's003', 'e003')
insert into StudentExam (SE_id, Student_id, Exam_id) values ('se004', 's004', 'e004')
insert into StudentExam (SE_id, Student_id, Exam_id) values ('se005', 's005', 'e005')
insert into StudentExam (SE_id, Student_id, Exam_id) values ('se006', 's006', 'e006')
insert into StudentExam (SE_id, Student_id, Exam_id) values ('se007', 's007', 'e007')
insert into StudentExam (SE_id, Student_id, Exam_id) values ('se008', 's008', 'e008')
insert into StudentExam (SE_id, Student_id, Exam_id) values ('se009', 's009', 'e009')
insert into StudentExam (SE_id, Student_id, Exam_id) values ('se010', 's010', 'e010')
```

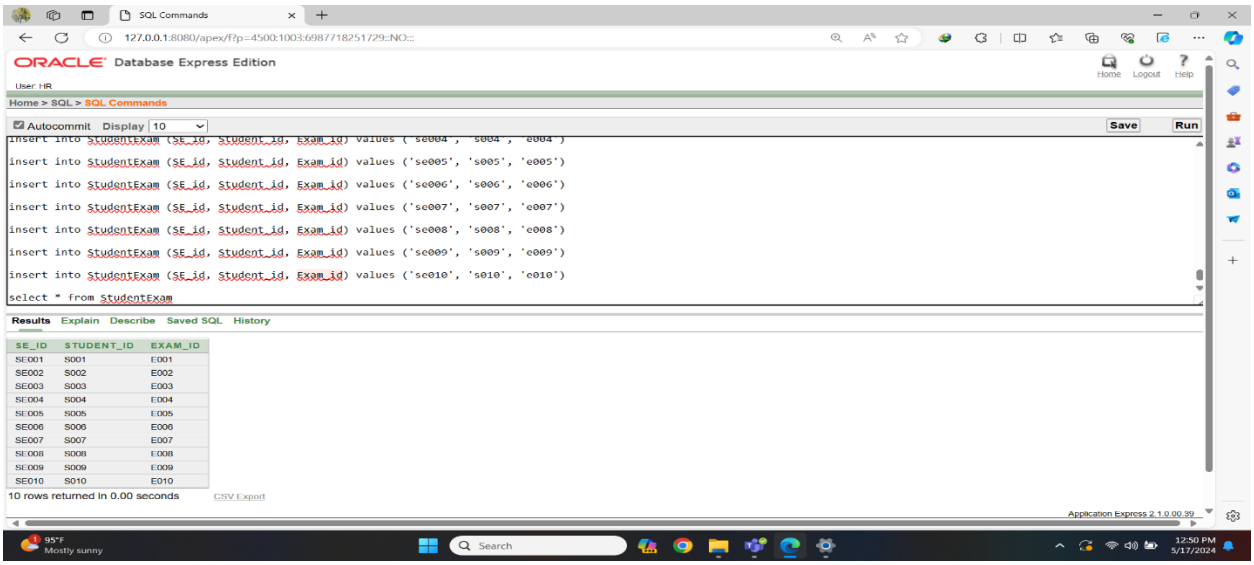


Fig-12: Student-Exam Table Data

7. Admin Table:

```
create table Admin
(
   Admin_id varchar2(20) constraint Admin_Adminid_pk primary key,
   Admin_name varchar2(30) not null,
   Password varchar2(20),
   Phone_no number(11),
   Email varchar2(30) unique
)
```

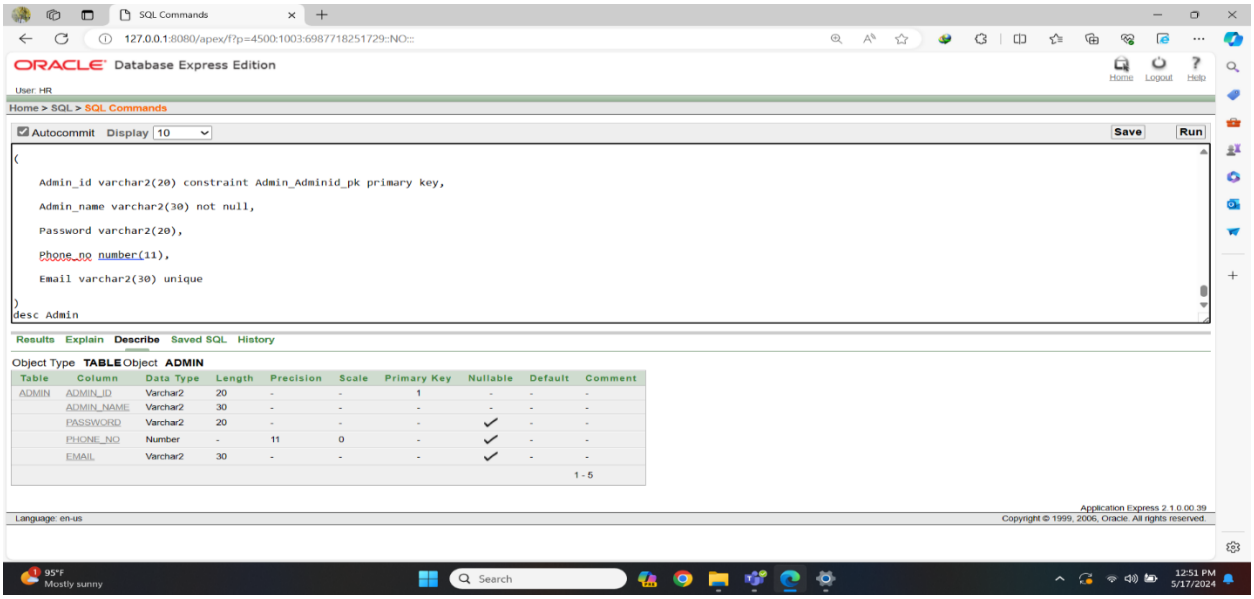


Fig-13: Admin Table Create

insert into Admin (Admin_id, Admin_name, Password, Phone_no, Email) values ('A001', 'Sohel Ahmed', 'password123', 01712345678, 'sohel@example.com')
insert into Admin (Admin_id, Admin_name, Password, Phone_no, Email) values ('A002', 'Nazneen Khan', 'admin123', 01809876543, 'nazneen@example.com')
insert into Admin (Admin_id, Admin_name, Password, Phone_no, Email) values ('A003', 'Imran Chowdhury', 'admin1234', 01955512345, 'imran@example.com')
insert into Admin (Admin_id, Admin_name, Password, Phone_no, Email) values ('A004', 'Sumaiya Islam', 'securepass', 01799988877, 'sumaiya@example.com')
insert into Admin (Admin_id, Admin_name, Password, Phone_no, Email) values ('A005', 'Rahat Rahman', 'password', 01811122233, 'rahat@example.com')
insert into Admin (Admin_id, Admin_name, Password, Phone_no, Email) values ('A006', 'Tasnim Ahmed', 'adminpass', 01744455566, 'tasnim@example.com')
insert into Admin (Admin_id, Admin_name, Password, Phone_no, Email) values ('A007', 'Nafisa Khan', 'passadmin', 01977788899, 'nafisa@example.com')
insert into Admin (Admin_id, Admin_name, Password, Phone_no, Email) values ('A008', 'Sarwar Islam', 'admin@123', 01866677788, 'sarwar@example.com')
insert into Admin (Admin_id, Admin_name, Password, Phone_no, Email) values ('A009', 'Anika Chowdhury', 'admin12345', 01733344455, 'anika@example.com')
insert into Admin (Admin_id, Admin_name, Password, Phone_no, Email) values ('A010', 'Tahmid Akhter', 'admin12', 01922233344, 'tahmid@example.com')

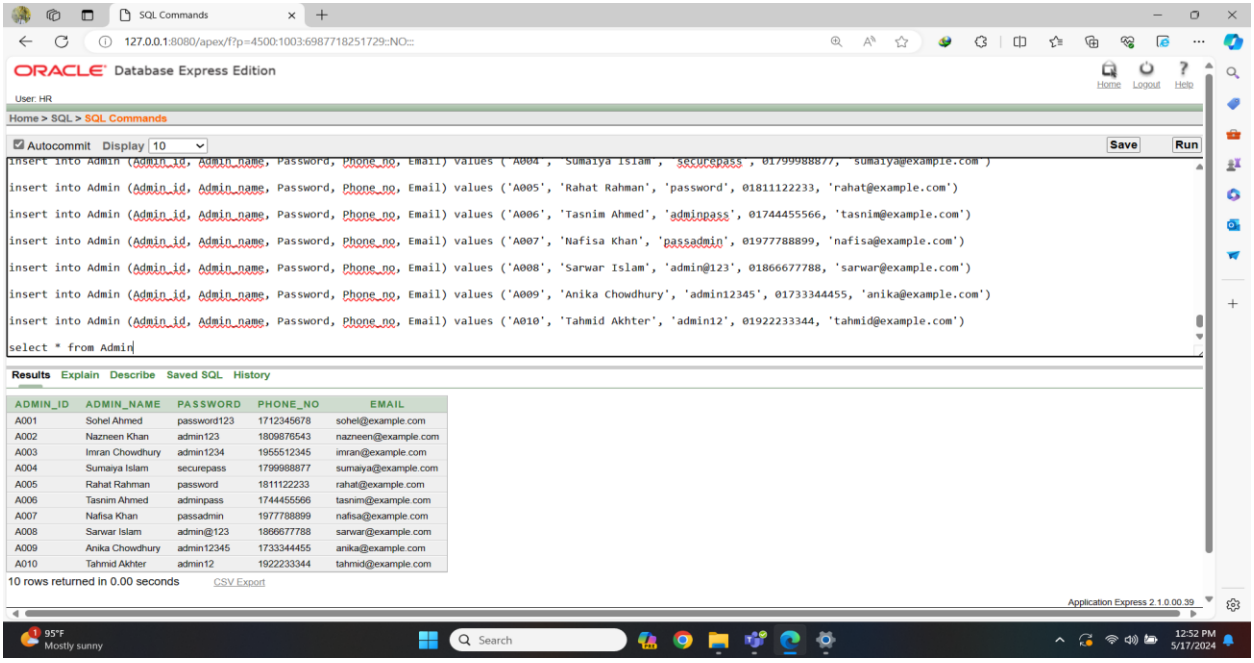


Fig-14: Admin Table Data

8. Student Admin Table:

```
create table StudentAdmin
(
   SA_id varchar2(20) constraint StudentAdmin _SAid_pk primary key,
   Student_id varchar2(20) constraint StudentAdmin _Studentid_fk references
Student(Student_id),
   Admin_id varchar2(20) constraint StudentAdmin _Adminid_fk references Admin(Admin_id)
)
```

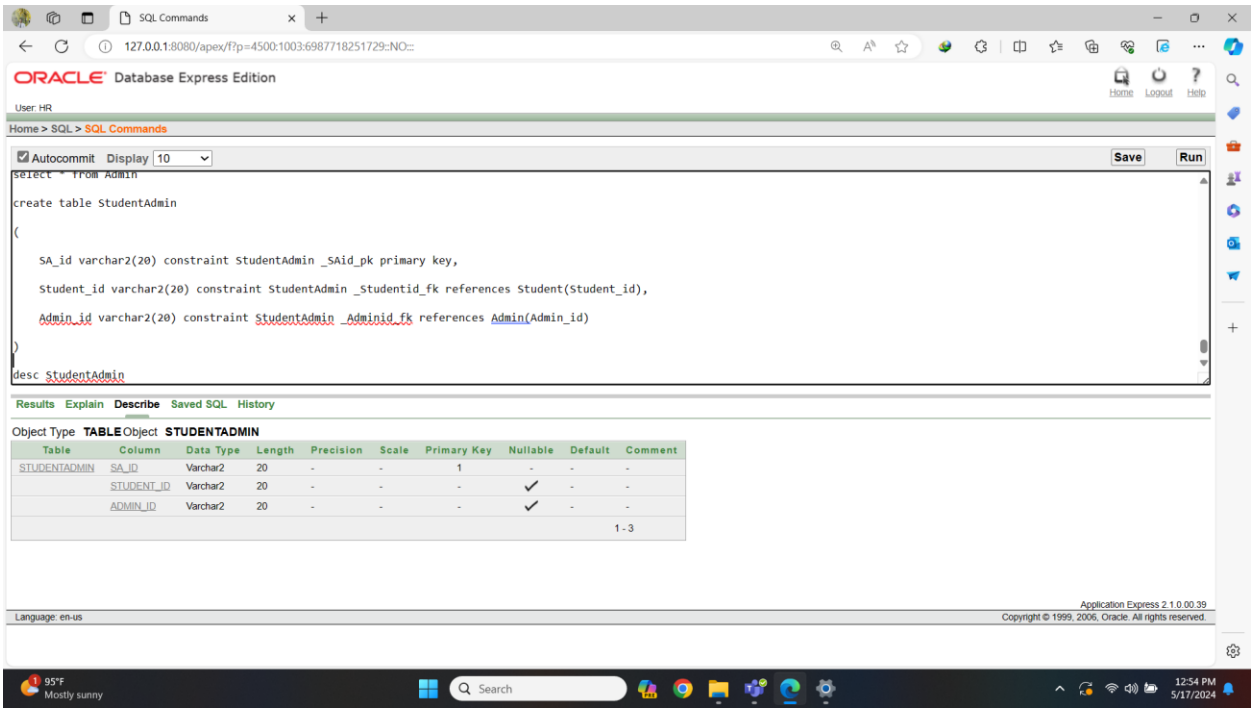


Fig-15: Student-Admin Table Create

```
insert into StudentAdmin (SA_id, Student_id, Admin_id) values ('SA001', 'S001', 'A001')
insert into StudentAdmin (SA_id, Student_id, Admin_id) values ('SA002', 'S002', 'A002')
insert into StudentAdmin (SA_id, Student_id, Admin_id) values ('SA003', 'S003', 'A003')
insert into StudentAdmin (SA_id, Student_id, Admin_id) values ('SA004', 'S004', 'A004')
insert into StudentAdmin (SA_id, Student_id, Admin_id) values ('SA005', 'S005', 'A005')
insert into StudentAdmin (SA_id, Student_id, Admin_id) values ('SA006', 'S006', 'A006')
insert into StudentAdmin (SA_id, Student_id, Admin_id) values ('SA007', 'S007', 'A007')
insert into StudentAdmin (SA_id, Student_id, Admin_id) values ('SA008', 'S008', 'A008')
insert into StudentAdmin (SA_id, Student_id, Admin_id) values ('SA009', 'S009', 'A009')
insert into StudentAdmin (SA_id, Student_id, Admin_id) values ('SA010', 'S010', 'A010')
```

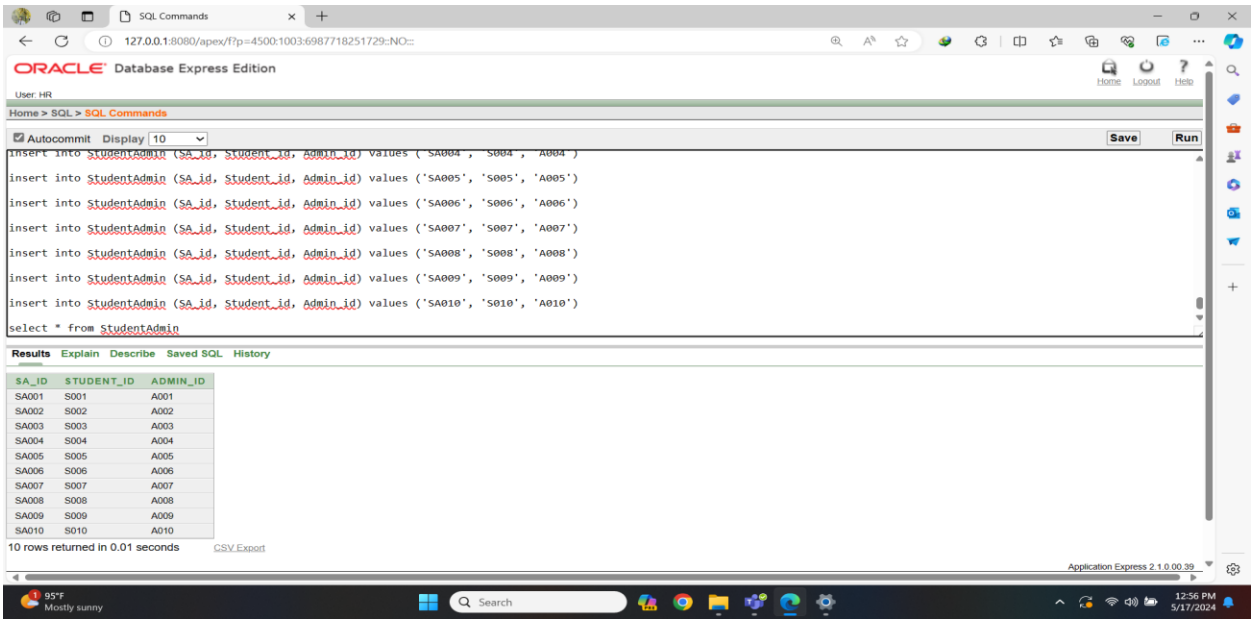


Fig-16: Student-Admin Table Data

9.Subject Table:

```
create table Subject
(
   Subject_id varchar2(20) constraint Subject_Subjectid_pk primary key,
   Subject_name varchar2(30) not null
)
```

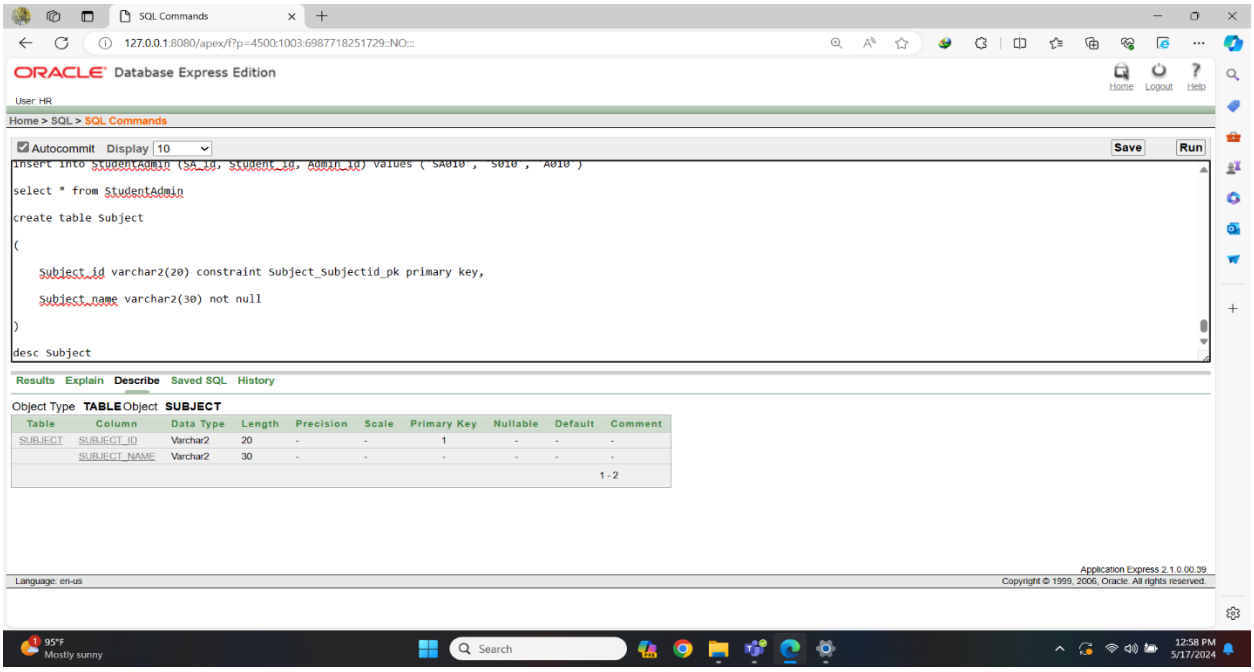


Fig-17: Subject Table Create

```
insert into Subject (Subject_id, Subject_name) values ('Sub001', 'Mathematics')
insert into Subject (Subject_id, Subject_name) values ('Sub002', 'Physics')
insert into Subject (Subject_id, Subject_name) values ('Sub003', 'Chemistry')
insert into Subject (Subject_id, Subject_name) values ('Sub004', 'Biology')
insert into Subject (Subject_id, Subject_name) values ('Sub005', 'English')
insert into Subject (Subject_id, Subject_name) values ('Sub006', 'History')
insert into Subject (Subject_id, Subject_name) values ('Sub007', 'Geography')
insert into Subject (Subject_id, Subject_name) values ('Sub008', 'Computer Science')
insert into Subject (Subject_id, Subject_name) values ('Sub009', 'Economics')
insert into Subject (Subject_id, Subject_name) values ('Sub010', 'Accounting')
```

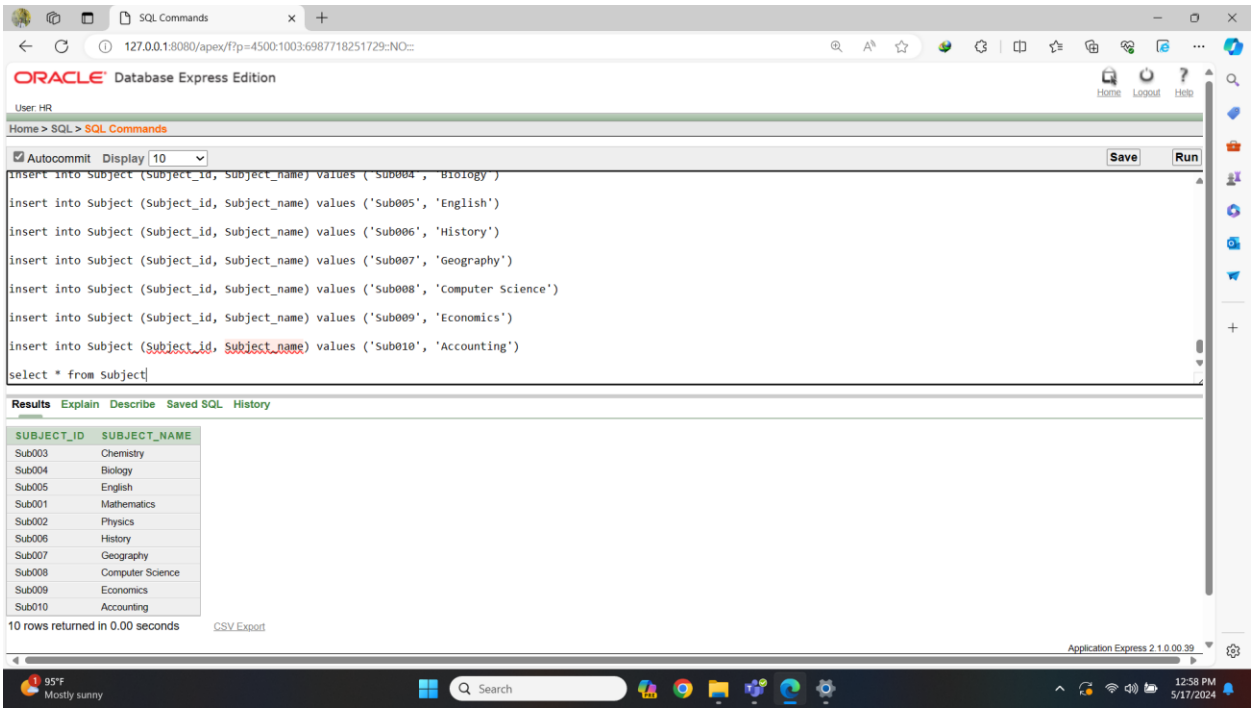


Fig-18: Subject Table Data

10. Admin Subject Table:

```
create table AdminSubject
(

AS_id varchar2(20) constraint AdminSubject_id_pk primary key,
Admin_id varchar2(20) constraint  AdminSubject_Adminid_fk references Admin(Admin_id),
Subject_id varchar2(20) constraint  AdminSubject_Subjectid_fk references
Subject(Subject_id)

)
```

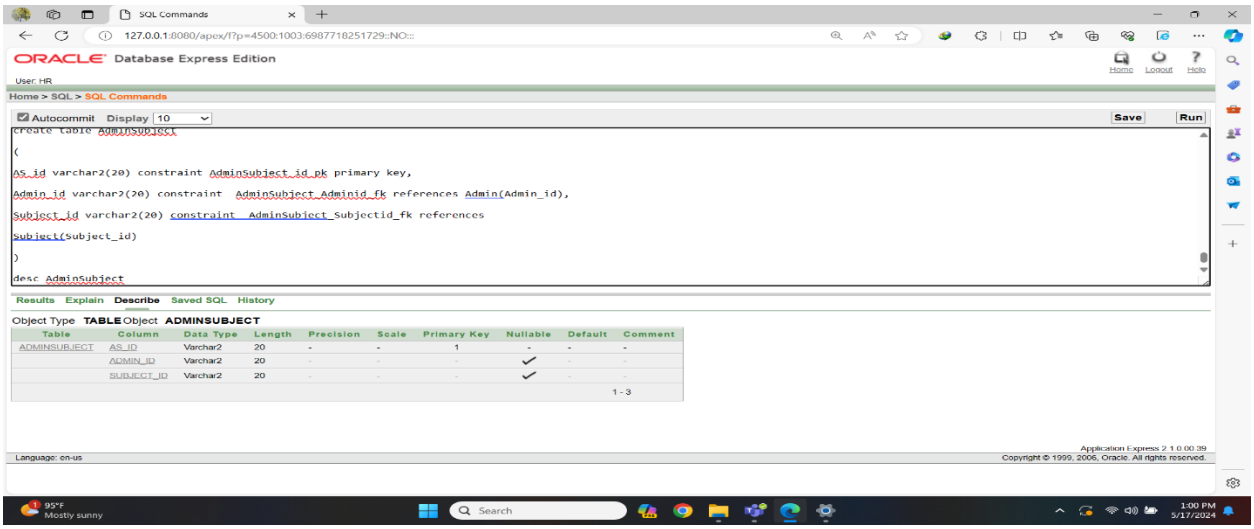


Fig-19: Admin-Subject Table Create

insert into AdminSubject (AS_id, Admin_id, Subject_id) values ('AS001', 'A001', 'Sub001')
insert into AdminSubject (AS_id, Admin_id, Subject_id) values ('AS002', 'A002', 'Sub002')
insert into AdminSubject (AS_id, Admin_id, Subject_id) values ('AS003', 'A003', 'Sub003')
insert into AdminSubject (AS_id, Admin_id, Subject_id) values ('AS004', 'A004', 'Sub004')
insert into AdminSubject (AS_id, Admin_id, Subject_id) values ('AS005', 'A005', 'Sub005')
insert into AdminSubject (AS_id, Admin_id, Subject_id) values ('AS006', 'A006', 'Sub006')
insert into AdminSubject (AS_id, Admin_id, Subject_id) values ('AS007', 'A007', 'Sub007')
insert into AdminSubject (AS_id, Admin_id, Subject_id) values ('AS008', 'A008', 'Sub008')
insert into AdminSubject (AS_id, Admin_id, Subject_id) values ('AS009', 'A009', 'Sub009')
insert into AdminSubject (AS_id, Admin_id, Subject_id) values ('AS010', 'A010', 'Sub010')

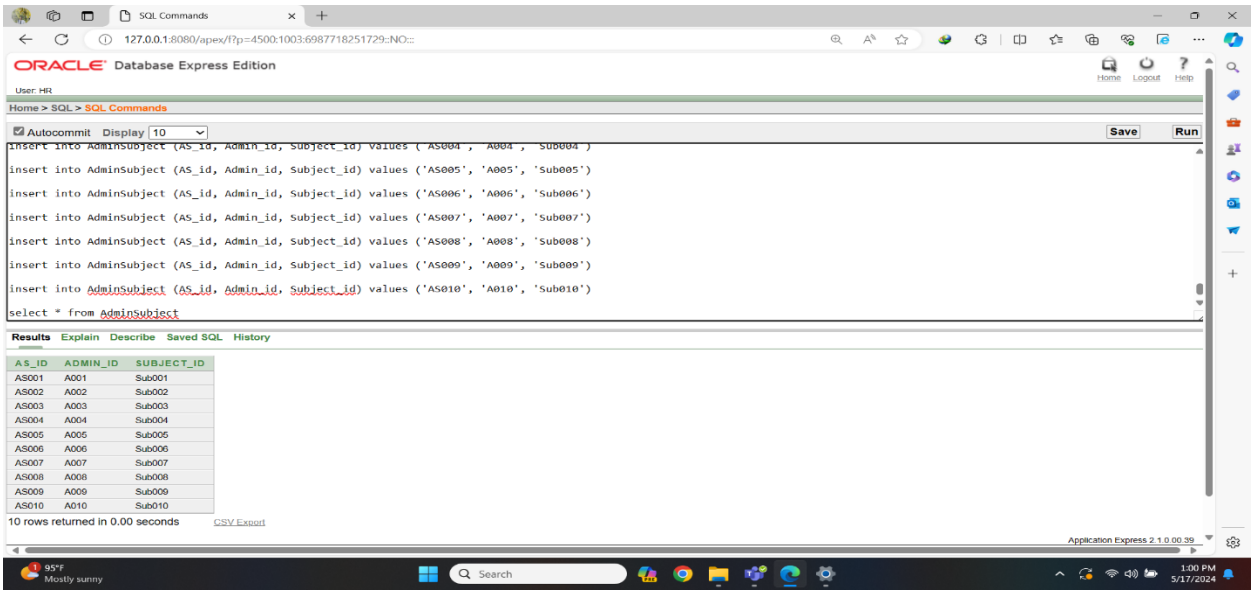


Fig-20: Admin-Subject Table Data

11. Teacher Table:

```
 create table Teacher
(
    Teacher_id varchar2(20) constraint Teacher_Teacherid_pk primary key,
    Teacher_name varchar2(30) not null,
    Address varchar2(30),
    Qualification varchar2(30),
    Phone_no number(11) ,
    Email varchar2(30)  unique
)
```

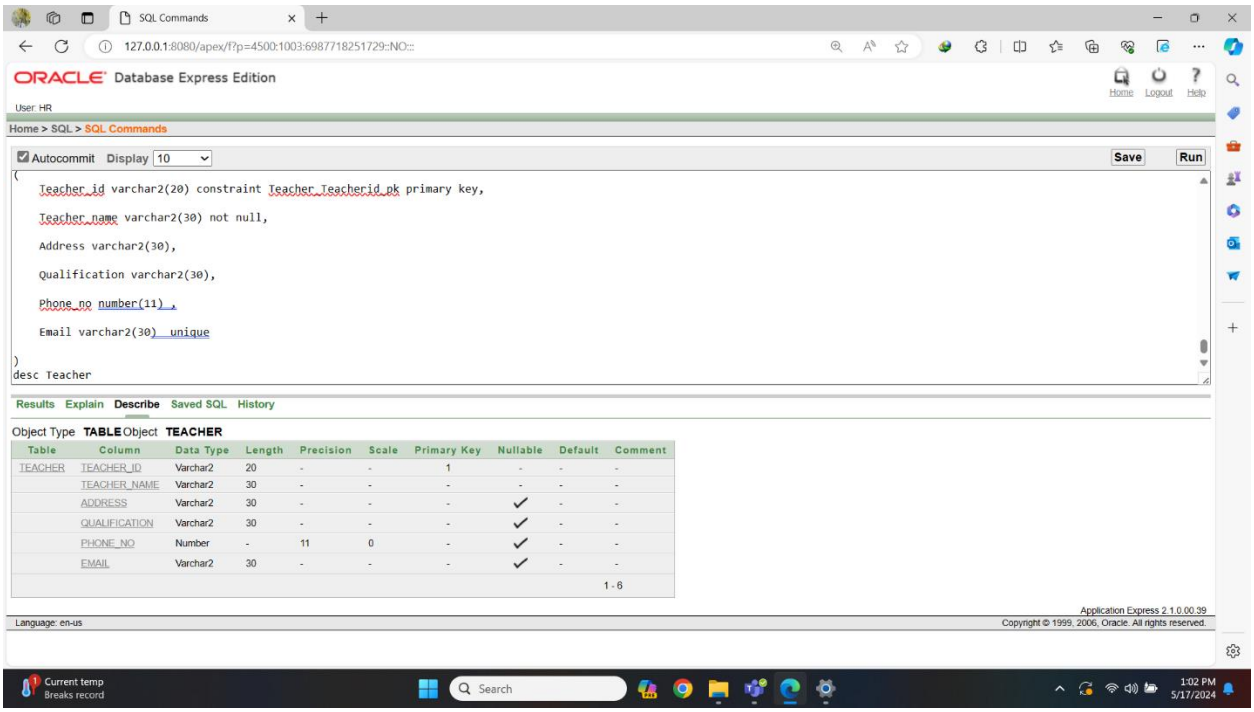


Fig-21: Teacher Table Create

```
insert into Teacher (Teacher_id, Teacher_name, Address, Qualification, Phone_no, Email)
values ('T1', 'Sadia Akhtar', 'Dhaka, Bangladesh', 'PhD in Biology', 01712345678,
'sadia.akhtar@gmail.com')
insert into Teacher (Teacher_id, Teacher_name, Address, Qualification, Phone_no, Email)
values ('T2', 'Imran Khan', 'Chittagong, Bangladesh', 'Masters in Chemistry', 01823456789,
'imran.khan@yahoo.com')
insert into Teacher (Teacher_id, Teacher_name, Address, Qualification, Phone_no, Email)
values ('T3', 'Aisha Rahman', 'Sylhet, Bangladesh', 'Bachelor in Physics', 01534567890,
'aisha.rahman@outlook.com')
```

```
insert into Teacher (Teacher_id, Teacher_name, Address, Qualification, Phone_no, Email)
values ('T4', 'Rahman Ali', 'Rajshahi, Bangladesh', 'PhD in Mathematics', 01745678901,
'rahman.ali@gmail.com')
insert into Teacher (Teacher_id, Teacher_name, Address, Qualification, Phone_no, Email)
values ('T5', 'Saima Islam', 'Khulna, Bangladesh', 'Masters in Computer Science', 01856789012,
'saima.islam@yahoo.com')
insert into Teacher (Teacher_id, Teacher_name, Address, Qualification, Phone_no, Email)
values ('T6', 'Fahim Rahman', 'Barisal, Bangladesh', 'Bachelor in English', 01567890123,
'fahim.rahman@outlook.com')
insert into Teacher (Teacher_id, Teacher_name, Address, Qualification, Phone_no, Email)
values ('T7', 'Tasnim Akhtar', 'Rangpur, Bangladesh', 'PhD in History', 01778901234,
'tasnim.akhtar@gmail.com')
insert into Teacher (Teacher_id, Teacher_name, Address, Qualification, Phone_no, Email)
values ('T8', 'Kamal Mia', 'Dinajpur, Bangladesh', 'Masters in Economics', 01889012345,
'kamal.mia@yahoo.com')
insert into Teacher (Teacher_id, Teacher_name, Address, Qualification, Phone_no, Email)
values ('T9', 'Sumaya Khan', 'Jessore, Bangladesh', 'Bachelor in Political Science', 01590123456,
'sumaya.khan@outlook.com')
insert into Teacher (Teacher_id, Teacher_name, Address, Qualification, Phone_no, Email)
values ('T10', 'Farhana Ahmed', 'Comilla, Bangladesh', 'PhD in Sociology', 01701234567,
'farhana.ahmed@gmail.com')
```

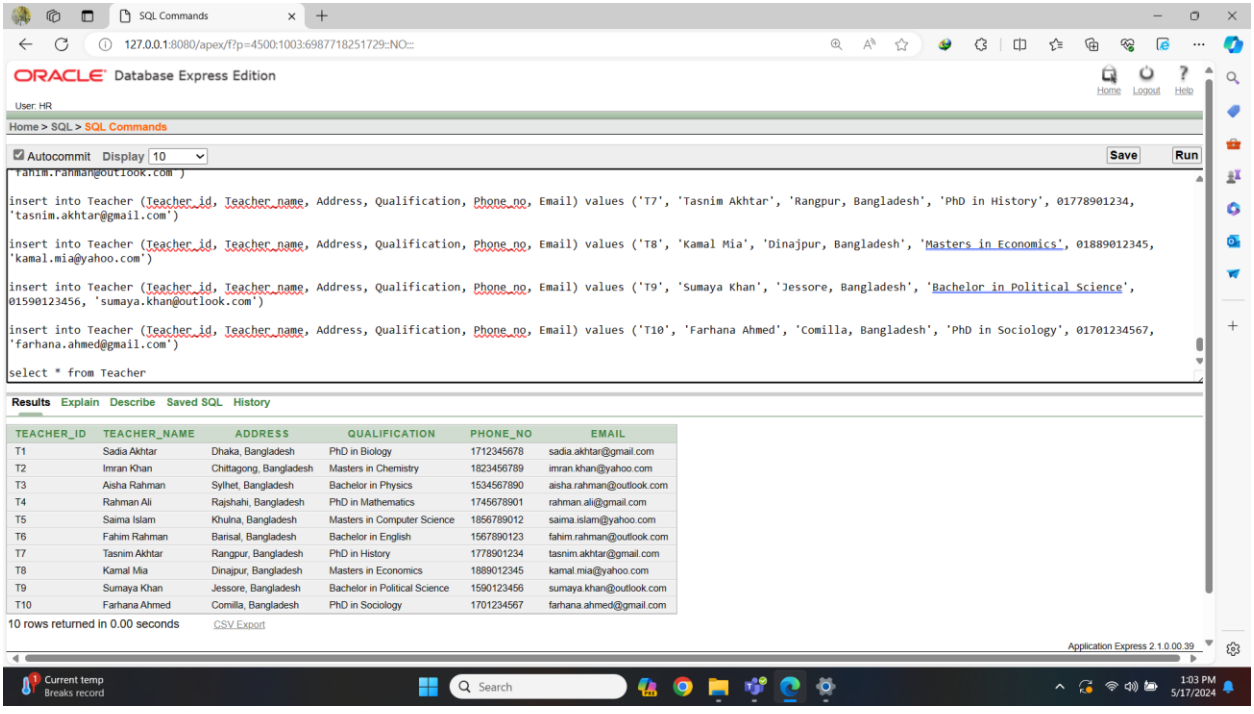


Fig-22: Teacher Table Data

12. Teacher Subject Table:

```
create table TeacherSubject
(
   TS_id varchar2(20) constraint TeacherSubject_TSid_pk primary key,
   Teacher_id varchar2(20) constraint TeacherSubject_Tid_fk references Teacher(Teacher_id),
   Subject_id varchar2(20) constraint TeacherSubject_Sid_fk references Subject(Subject_id)
)
```

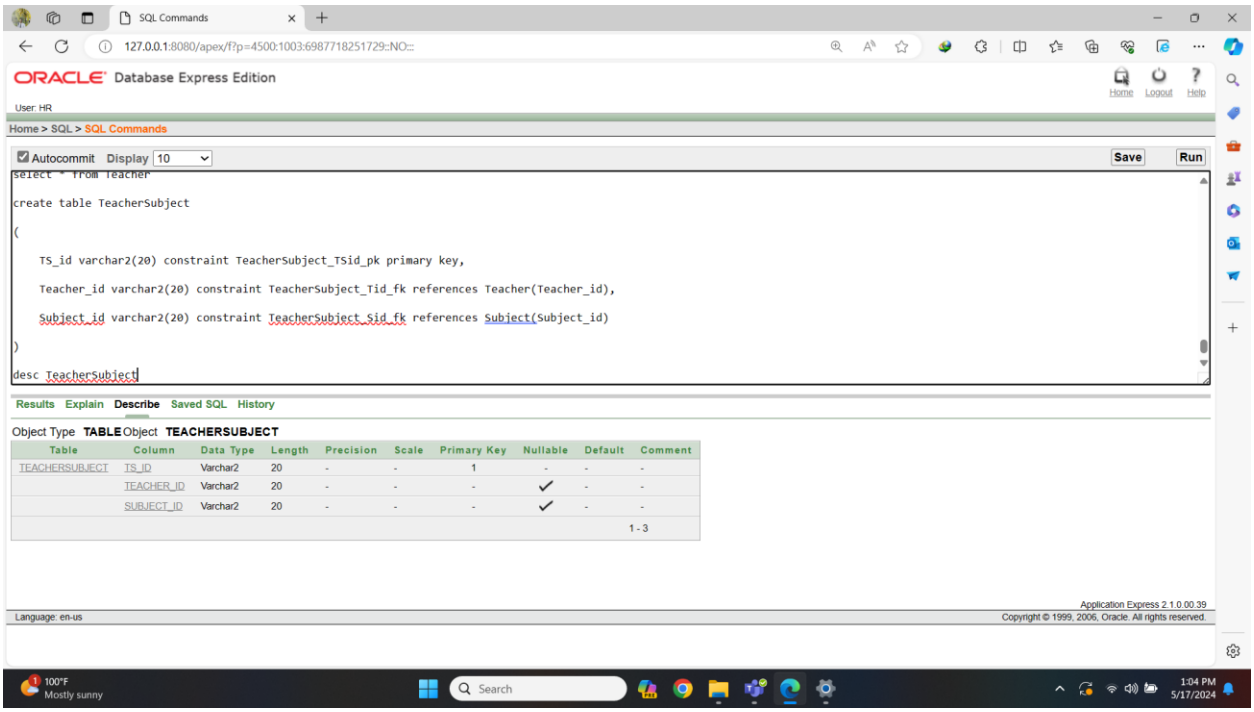


Fig-23: Teacher-Subject Table Create

```
insert into TeacherSubject (TS_id, Teacher_id, Subject_id) values ('TS1', 'T1', 'Sub001')
insert into TeacherSubject (TS_id, Teacher_id, Subject_id) values ('TS2', 'T2', 'Sub002')
insert into TeacherSubject (TS_id, Teacher_id, Subject_id) values ('TS3', 'T3', 'Sub003')
insert into TeacherSubject (TS_id, Teacher_id, Subject_id) values ('TS4', 'T4', 'Sub004')
insert into TeacherSubject (TS_id, Teacher_id, Subject_id) values ('TS5', 'T5', 'Sub005')
insert into TeacherSubject (TS_id, Teacher_id, Subject_id) values ('TS6', 'T6', 'Sub006')
insert into TeacherSubject (TS_id, Teacher_id, Subject_id) values ('TS7', 'T7', 'Sub007')
insert into TeacherSubject (TS_id, Teacher_id, Subject_id) values ('TS8', 'T8', 'Sub008')
insert into TeacherSubject (TS_id, Teacher_id, Subject_id) values ('TS9', 'T9', 'Sub009')
insert into TeacherSubject (TS_id, Teacher_id, Subject_id) values ('TS10', 'T10', 'Sub010')
```

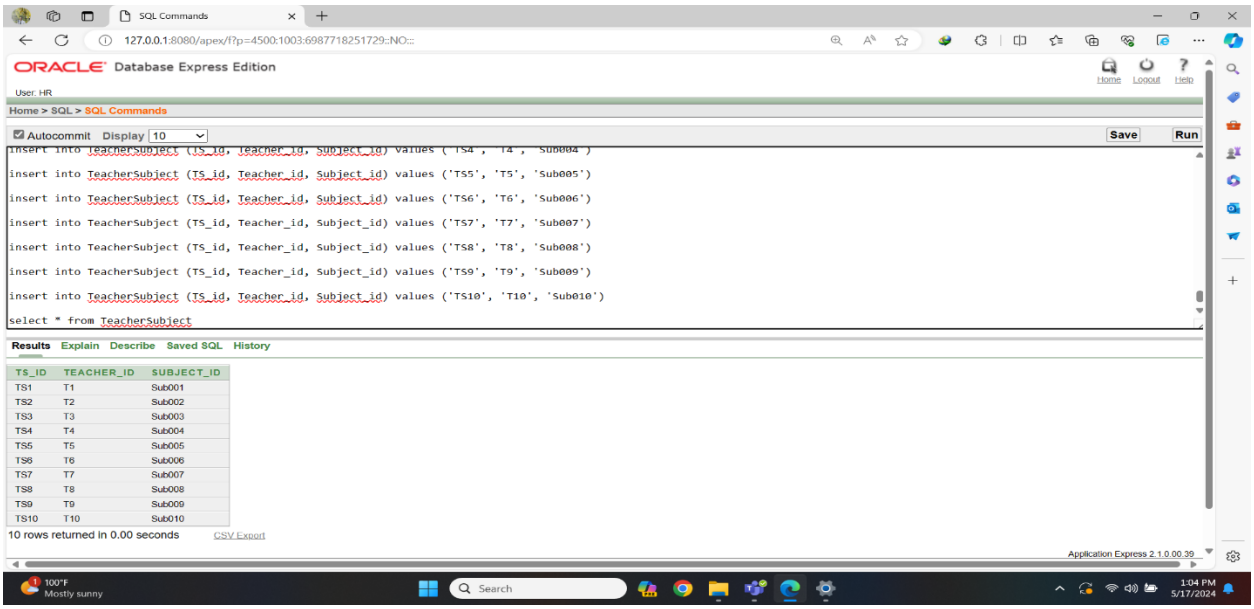


Fig-24: Teacher-Subject Table Data

13. Teacher Admin Table:

```
create table TeacherAdmin
(
    TA_id varchar2(20) constraint TeacherAdmin_Tid_pk primary key,
    Teacher_id varchar2(20) constraint TeacherAdmin_Tid_fk references Teacher(Teacher_id),
    Admin_id varchar2(20) constraint TeacherAdmin_Aid_fk references Admin(Admin_id)
)
```

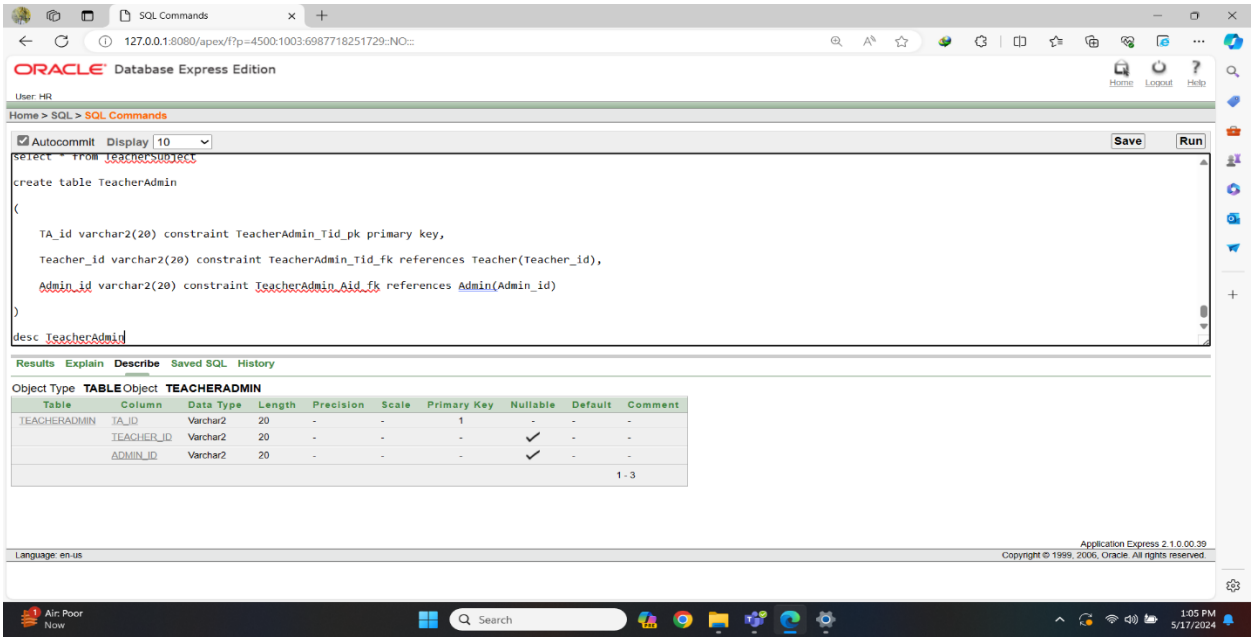


Fig-25: Teacher-Admin Table Create

```
insert into TeacherAdmin (TA_id, Teacher_id, Admin_id) values ('TA1', 'T1', 'A001')
insert into TeacherAdmin (TA_id, Teacher_id, Admin_id) values ('TA2', 'T2', 'A002')
insert into TeacherAdmin (TA_id, Teacher_id, Admin_id) values ('TA3', 'T3', 'A003')
insert into TeacherAdmin (TA_id, Teacher_id, Admin_id) values ('TA4', 'T4', 'A004')
insert into TeacherAdmin (TA_id, Teacher_id, Admin_id) values ('TA5', 'T5', 'A005')
insert into TeacherAdmin (TA_id, Teacher_id, Admin_id) values ('TA6', 'T6', 'A006')
insert into TeacherAdmin (TA_id, Teacher_id, Admin_id) values ('TA7', 'T7', 'A007')
insert into TeacherAdmin (TA_id, Teacher_id, Admin_id) values ('TA8', 'T8', 'A008')
insert into TeacherAdmin (TA_id, Teacher_id, Admin_id) values ('TA9', 'T9', 'A009')
insert into TeacherAdmin (TA_id, Teacher_id, Admin_id) values ('TA10', 'T10', 'A010')
```

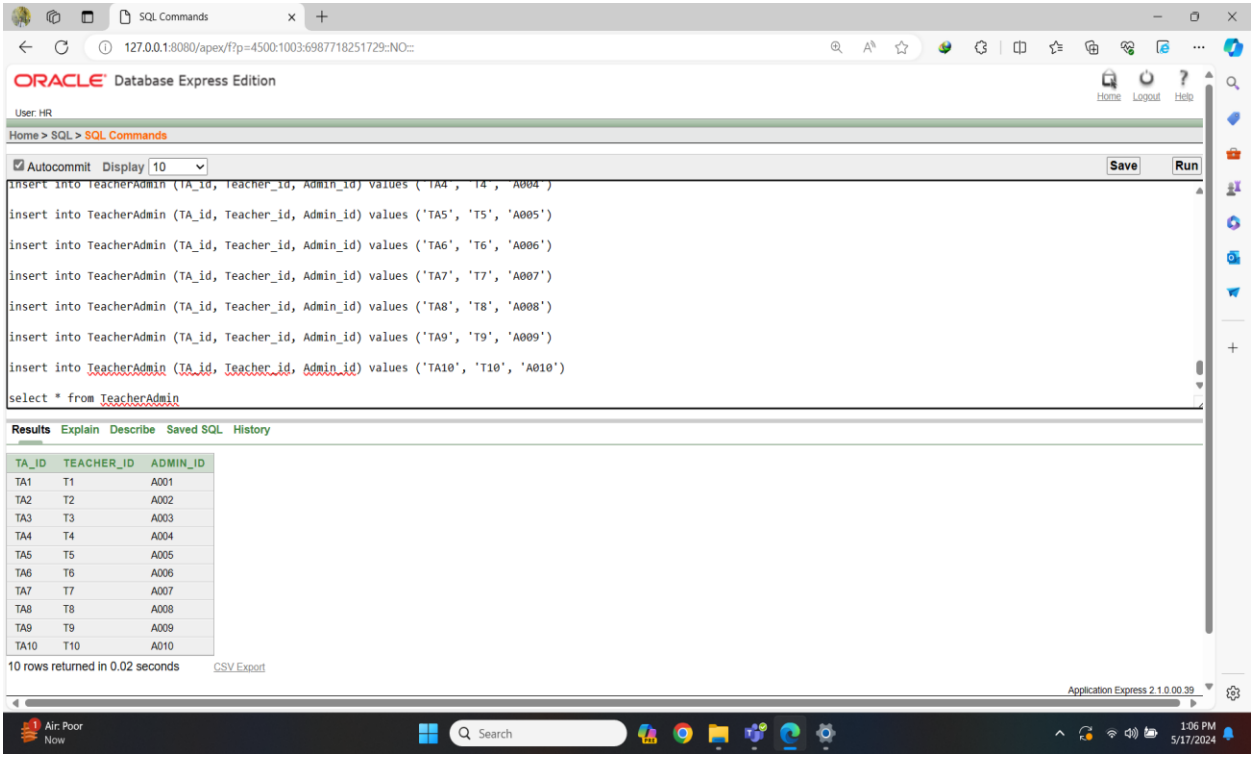


Fig-26: Teacher-Admin Table Data

14. Room Table:

```
create table Room
(
   Room_no varchar2(20) constraint Room_Roomno_pk primary key,
   Class_name varchar2(30) not null
)
```

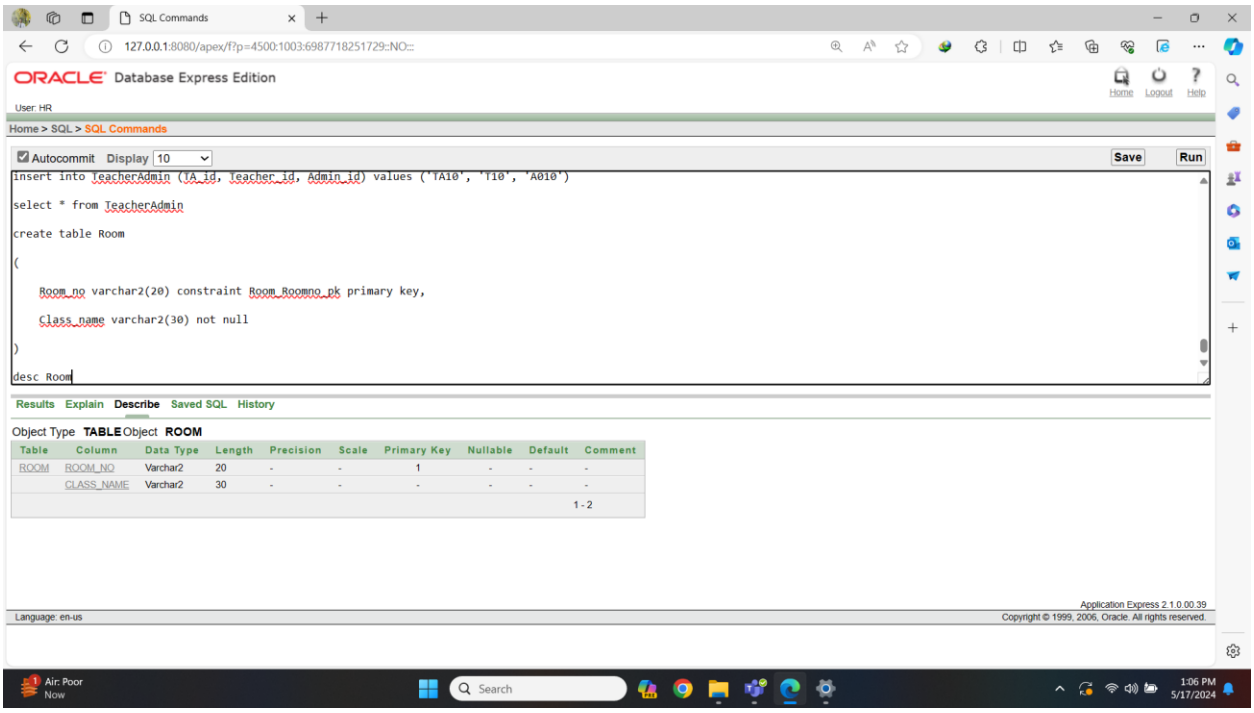


Fig-27: Room Table Create

```
insert into Room (Room_no, Class_name) values ('R1', 'class 1A')
insert into Room (Room_no, Class_name) values ('R2', 'class 1B')
insert into Room (Room_no, Class_name) values ('R3', 'class 2A')
insert into Room (Room_no, Class_name) values ('R4', 'class 2B')
insert into Room (Room_no, Class_name) values ('R5', 'class 3A')
insert into Room (Room_no, Class_name) values ('R6', 'class 3B')
insert into Room (Room_no, Class_name) values ('R7', 'class 4A')
insert into Room (Room_no, Class_name) values ('R8', 'class 4B')
insert into Room (Room_no, Class_name) values ('R9', 'class 5A')
insert into Room (Room_no, Class_name) values ('R10', 'class 5B')
```

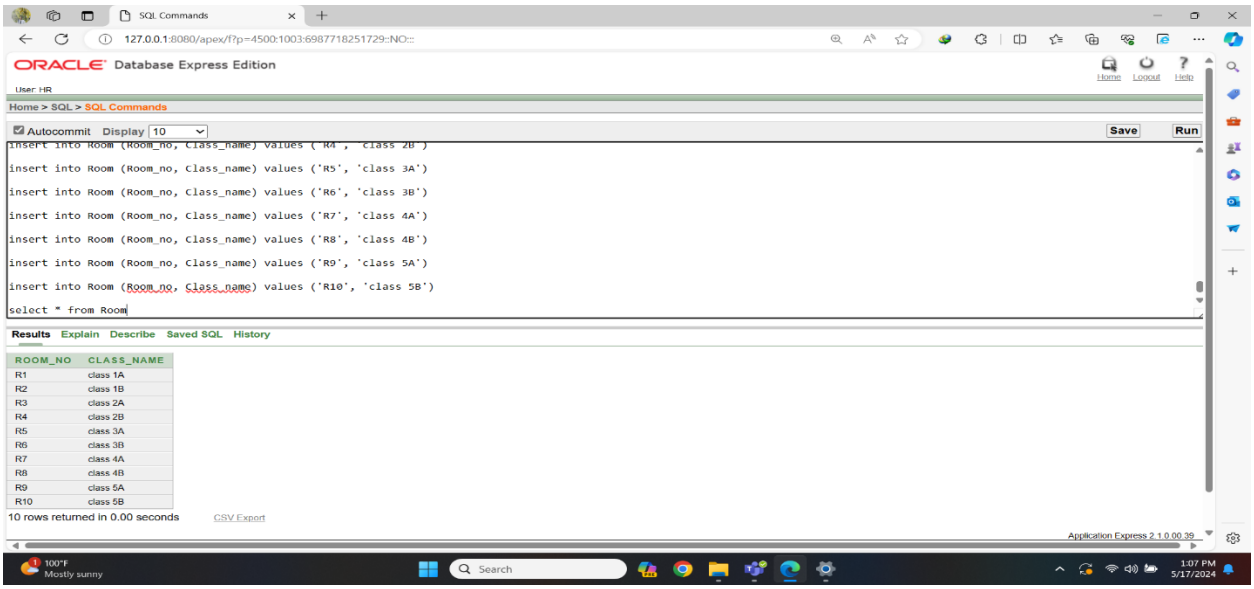


Fig-28: Room Table Data

15. Teacher Room Table:

```
create table TeacherRoom
(
    TR_id varchar2(20) constraint TeacherRoom_TRid_pk primary key,
    Teacher_id varchar2(20) constraint TeacherRoom_Tid_fk references Teacher(Teacher_id),
    Room_no varchar2(20) constraint TeacherRoom_Roomno_fk references Room(Room_no)
)
```

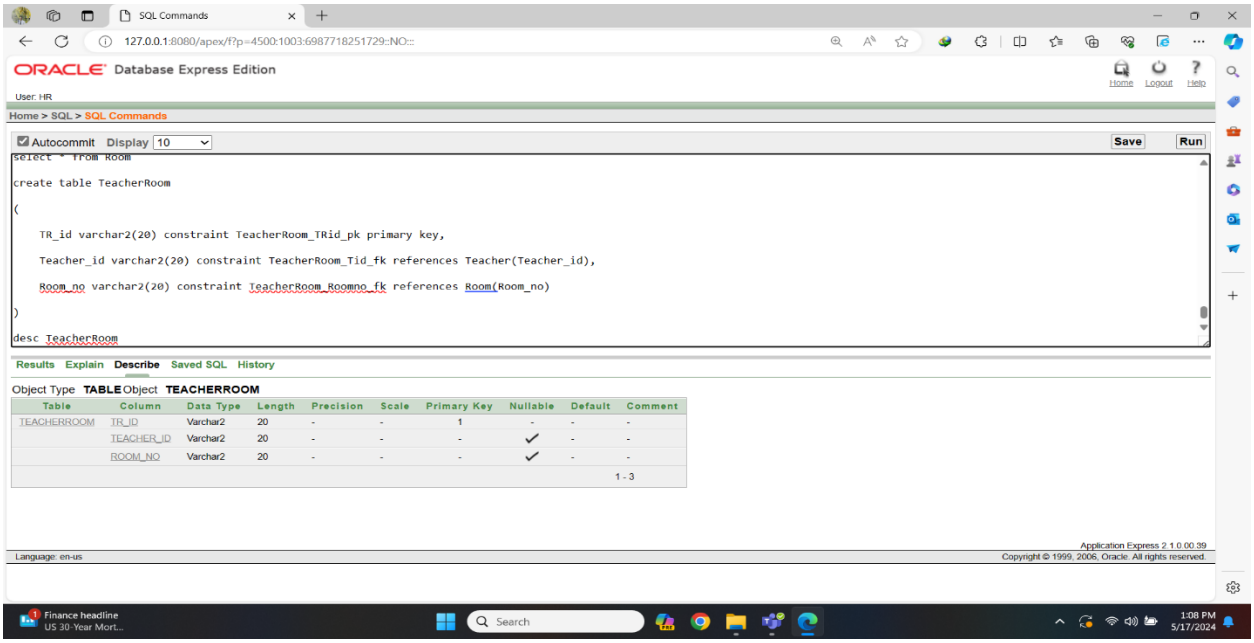


Fig-29: Teacher-Room Table Create

```
insert into TeacherRoom (TR_id, Teacher_id, Room_no) values ('TR1', 'T1', 'R1')
insert into TeacherRoom (TR_id, Teacher_id, Room_no) values (‘TR2', 'T2', 'R2')
insert into TeacherRoom (TR_id, Teacher_id, Room_no) values ('TR3', 'T3', 'R3')
insert into TeacherRoom (TR_id, Teacher_id, Room_no) values ('TR4', 'T4', 'R4')
insert into TeacherRoom (TR_id, Teacher_id, Room_no) values ('TR5', 'T5', 'R5')
insert into TeacherRoom (TR_id, Teacher_id, Room_no) values ('TR6', 'T6', 'R6')
insert into TeacherRoom (TR_id, Teacher_id, Room_no) values ('TR7', 'T7', 'R7')
insert into TeacherRoom (TR_id, Teacher_id, Room_no) values ('TR8', 'T8', 'R8')
insert into TeacherRoom (TR_id, Teacher_id, Room_no) values ('TR9', 'T9', 'R9')
insert into TeacherRoom (TR_id, Teacher_id, Room_no) values ('TR10', 'T10', 'R10')
```

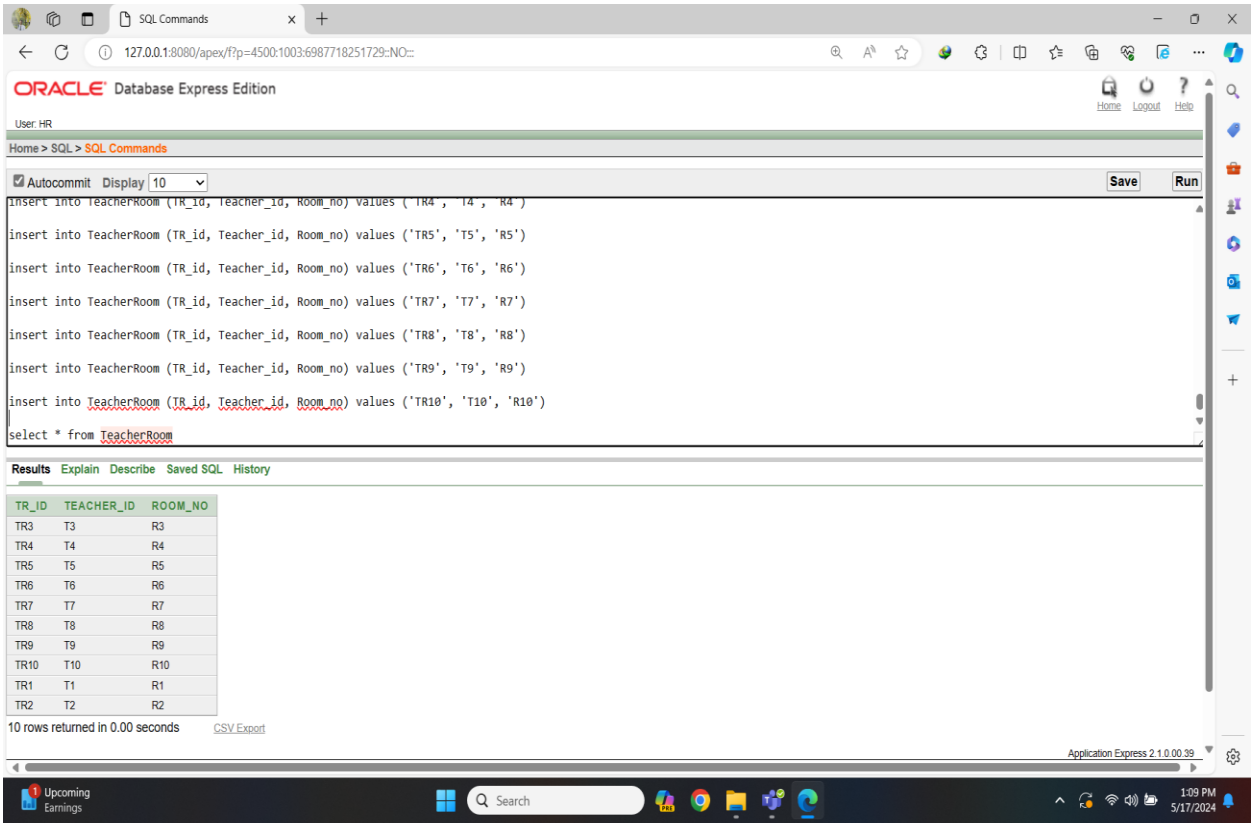


Fig-30: Teacher-Room Table Data

16. Student Room Table:

```
create table StudentRoom
(
    SR_id varchar2(20) constraint StudentRoom_SRid_pk primary key,
    Student_id varchar2(20) constraint StudentRoom_Sid_fk references Student(Student_id),
    Room_no varchar2(20) constraint StudentRoom_Roomno_fk references Room(Room_no)
)
```

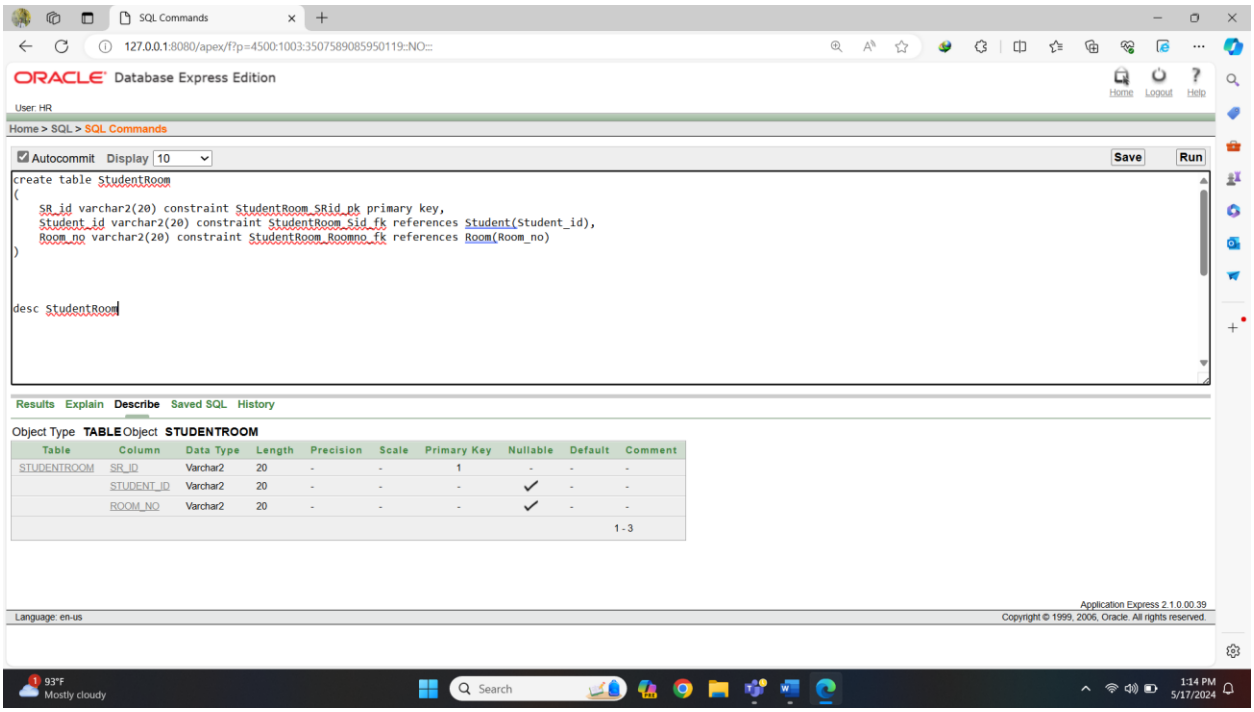


Fig-31: Student-Room Table Create

```
insert into StudentRoom (SR_id, Student_id, Room_no) values ('SR01', 'S001', 'R1' )
insert into StudentRoom (SR_id, Student_id, Room_no) values ('SR02', 'S002', 'R2' )
insert into StudentRoom (SR_id, Student_id, Room_no) values ('SR03', 'S003', 'R3' )
insert into StudentRoom (SR_id, Student_id, Room_no) values ('SR04', 'S004', 'R4' )
insert into StudentRoom (SR_id, Student_id, Room_no) values ('SR05', 'S005', 'R5' )
insert into StudentRoom (SR_id, Student_id, Room_no) values ('SR06', 'S006', 'R6' )
insert into StudentRoom (SR_id, Student_id, Room_no) values ('SR07', 'S007', 'R7' )
insert into StudentRoom (SR_id, Student_id, Room_no) values ('SR08', 'S008', 'R8' )
```

insert into StudentRoom (SR_id, Student_id, Room_no) values ('SR09', 'S009', 'R9')

insert into StudentRoom (SR_id, Student_id, Room_no) values ('SR10', 'S010', 'R10')

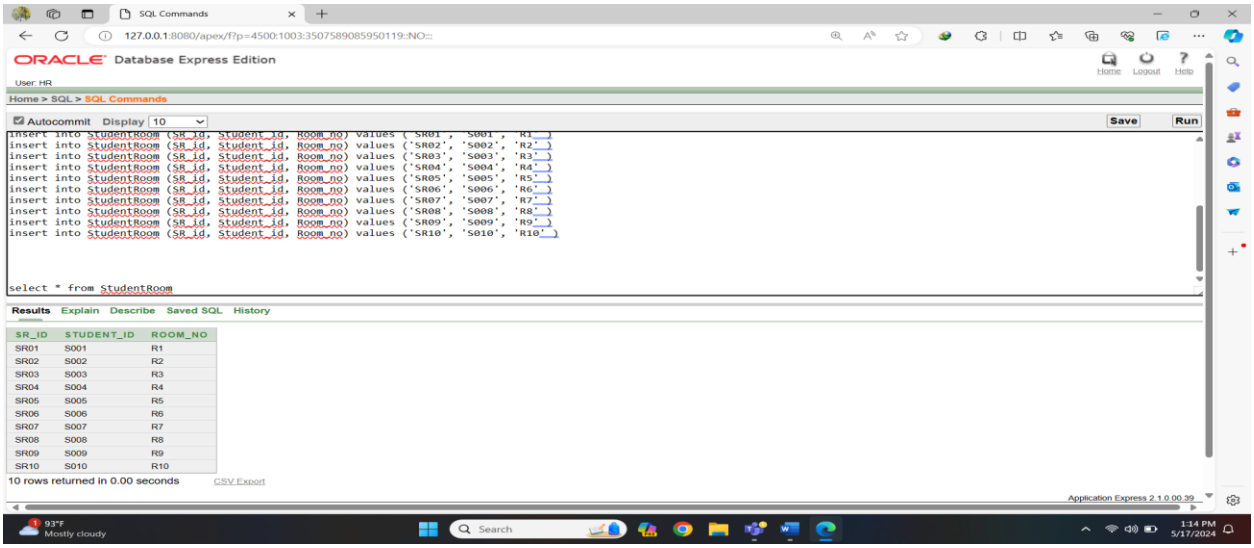


Fig-32: Student-Room Table Data

17. Salary Table:

```
create table Salary
(
   Salary_id varchar2(20) constraint Salary_Sid_pk primary key,
   Teacher_Salary number(10) not null
)
```

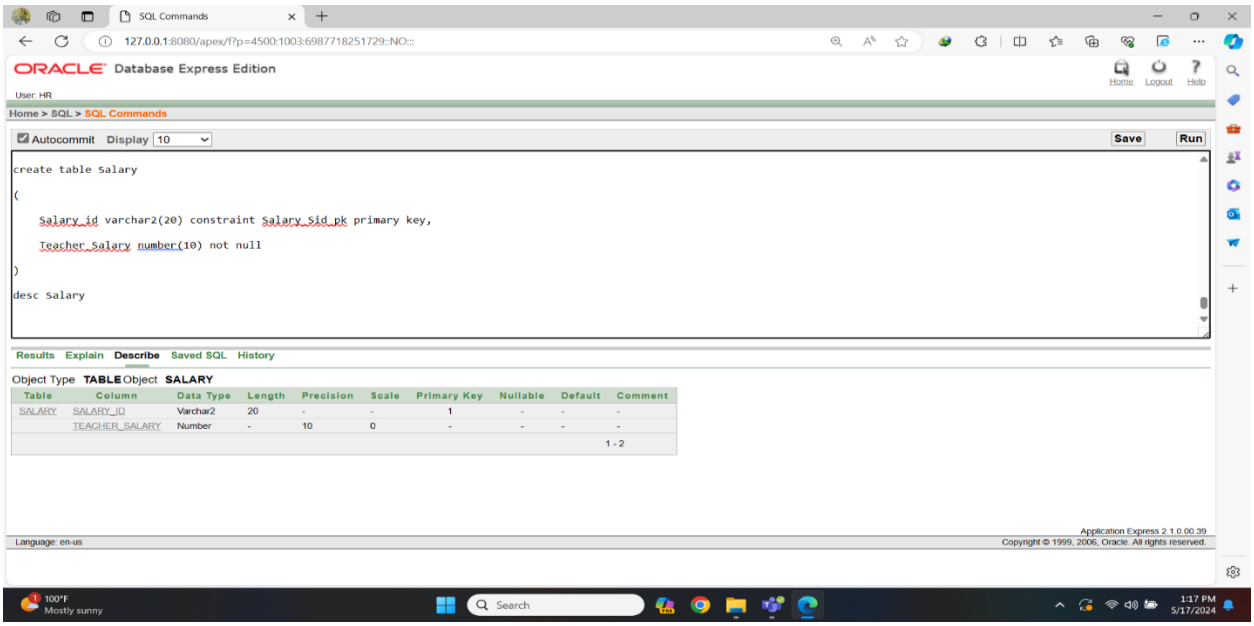


Fig-33: Salary Table Create

insert into Salary (Salary_id, Teacher_salary) values ('S1', 50000)
insert into Salary (Salary_id, Teacher_salary) values ('S2', 55000)
insert into Salary (Salary_id, Teacher_salary) values ('S3', 60000)
insert into Salary (Salary_id, Teacher_salary) values ('S4', 65000)
insert into Salary (Salary_id, Teacher_salary) values ('S5', 70000)
insert into Salary (Salary_id, Teacher_salary) values ('S6', 75000)
insert into Salary (Salary_id, Teacher_salary) values ('S7', 80000)
insert into Salary (Salary_id, Teacher_salary) values ('S8', 85000)
insert into Salary (Salary_id, Teacher_salary) values ('S9', 90000)
insert into Salary (Salary_id, Teacher_salary) values ('S10', 95000)

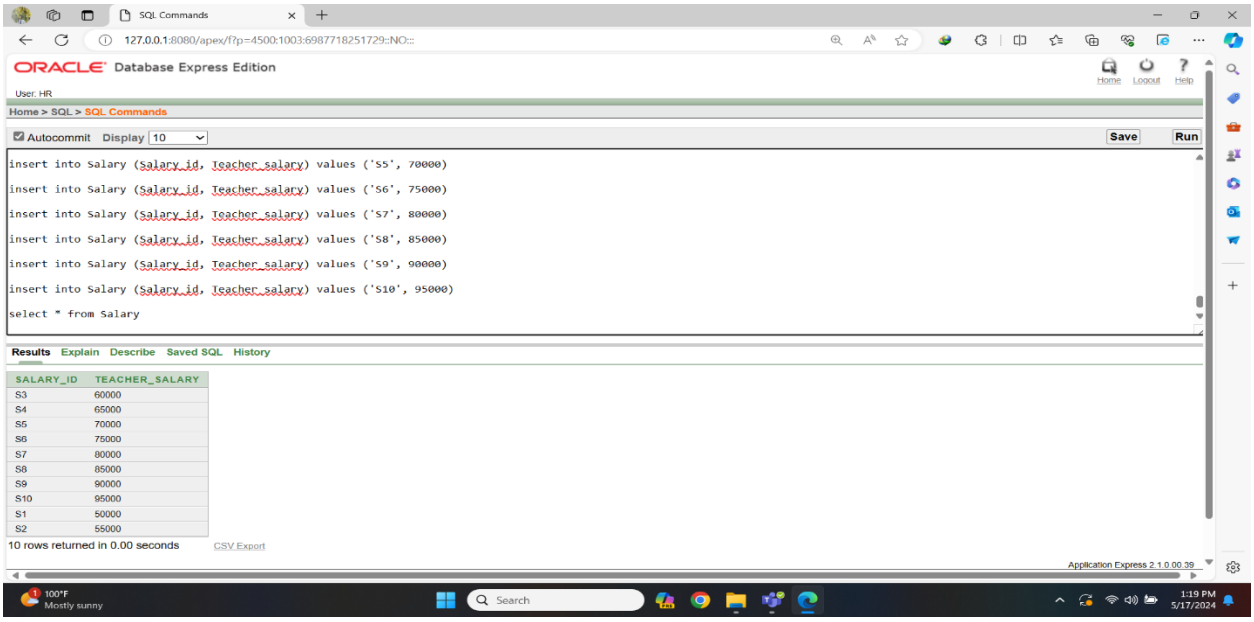


Fig-34: Salary Table Data

18. Teacher Salary Table:

```
create table TeacherSalary
(
  TS_id varchar2(20) constraint TeacherSalary_TSid_pk primary key,
  Teacher_id varchar2(20) constraint TeacherSalary_Tid_fk references Teacher(Teacher_id),
  Salary_id varchar2(20) constraint TeacherSalary_Sid_fk references Salary(Salary_id)
)
```

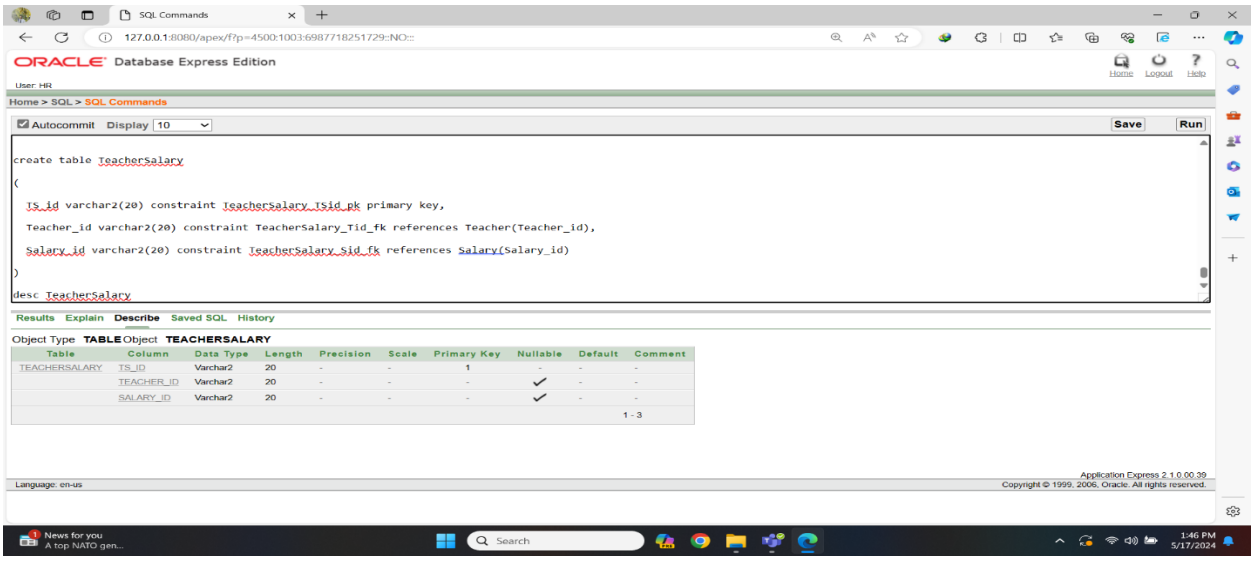


Fig-35: Teacher-Salary Table Create

```
insert into TeacherSalary (TS_id, Teacher_id, Salary_id) values ('TS1', 'T1', 'S1')
insert into TeacherSalary (TS_id, Teacher_id, Salary_id) values ('TS2', 'T2', 'S2')
insert into TeacherSalary (TS_id, Teacher_id, Salary_id) values ('TS3', 'T3', 'S3')
insert into TeacherSalary (TS_id, Teacher_id, Salary_id) values ('TS4', 'T4', 'S4')
insert into TeacherSalary (TS_id, Teacher_id, Salary_id) values ('TS5', 'T5', 'S5')
insert into TeacherSalary (TS_id, Teacher_id, Salary_id) values ('TS6', 'T6', 'S6')
insert into TeacherSalary (TS_id, Teacher_id, Salary_id) values ('TS7', 'T7', 'S7')
insert into TeacherSalary (TS_id, Teacher_id, Salary_id) values ('TS8', 'T8', 'S8')
insert into TeacherSalary (TS_id, Teacher_id, Salary_id) values ('TS9', 'T9', 'S9')
insert into TeacherSalary (TS_id, Teacher_id, Salary_id) values ('TS10', 'T10', 'S10')
```

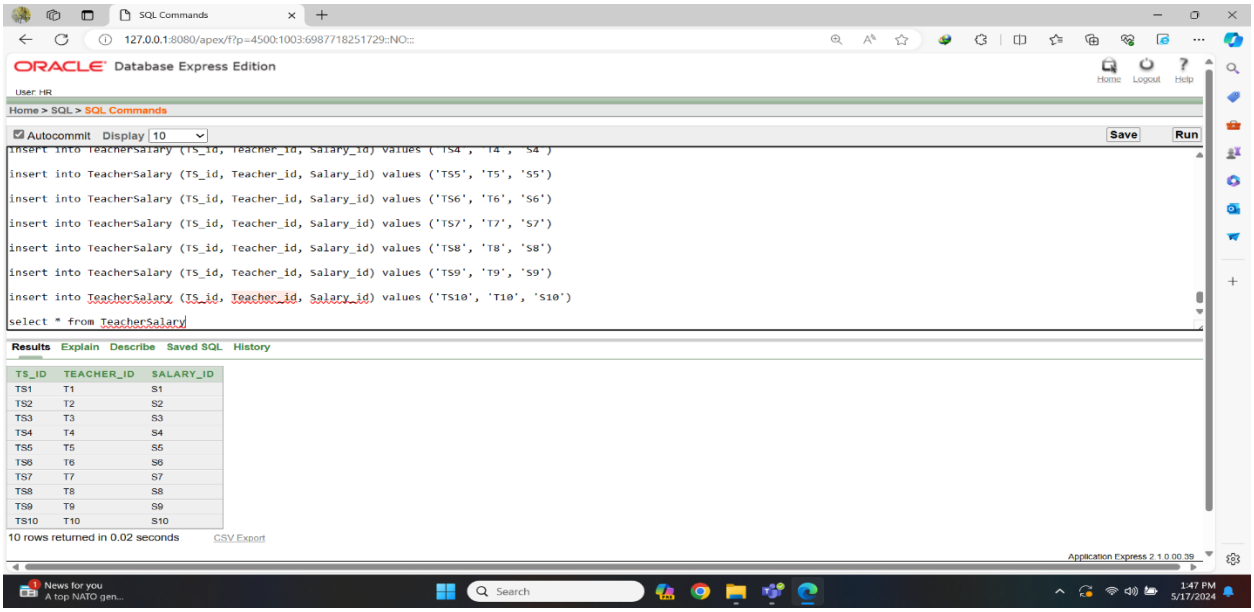


Fig-36: Teacher-Salary Table Data

19. Payment Table:

```
create table Payment
(
   Payment_id varchar2(20) constraint Payment_Paymentid_pk primary key,
   Student_fee number(10) not null
)
```

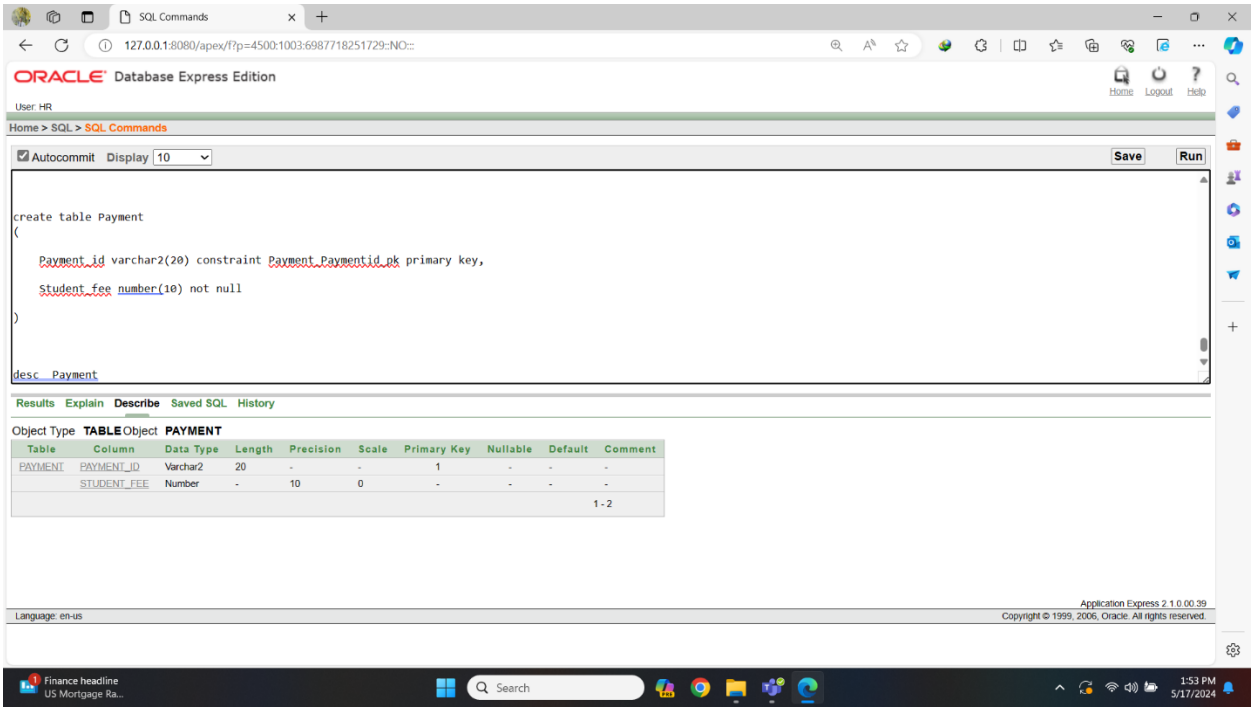


Fig-37: Payment Table Create

```
insert into Payment (Payment_id, Student_fee) values ('P1', 5000)
insert into Payment (Payment_id, Student_fee)values ('P2', 5500)
insert into Payment (Payment_id, Student_fee) values ('P3', 6000)
insert into Payment (Payment_id, Student_fee) values ('P4', 6500)
insert into Payment (Payment_id, Student_fee) values ('P5', 7000)
insert into Payment (Payment_id, Student_fee) values ('P6', 7500)
insert into Payment (Payment_id, Student_fee) values ('P7', 8000)
insert into Payment (Payment_id, Student_fee) values ('P8', 8500)
insert into Payment (Payment_id,Student_fee) values ('P9', 9000)
insert into Payment (Payment_id,Student_fee) values ('P10', 9500)
```

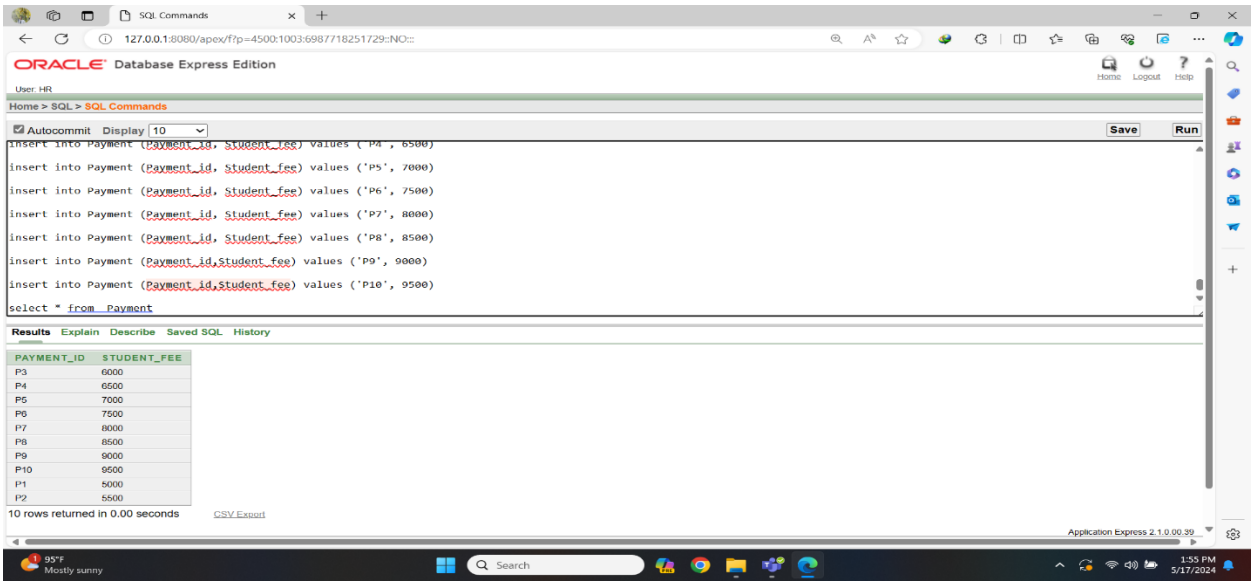


Fig-38: Payment Table Data

20. Guardian Payment Table:

```
create table GuardianPayment
(
    GP_id varchar2(20) constraint GuardianPayment_GPid_pk primary key,
    Guardian_id varchar2(20) constraint GuardianPayment_Gid_fk references
Guardian(Guardian_id),
    Payment_id varchar2(20)  constraint GuardianPayment_Pid_fk references
Payment(Payment_id)
)
```

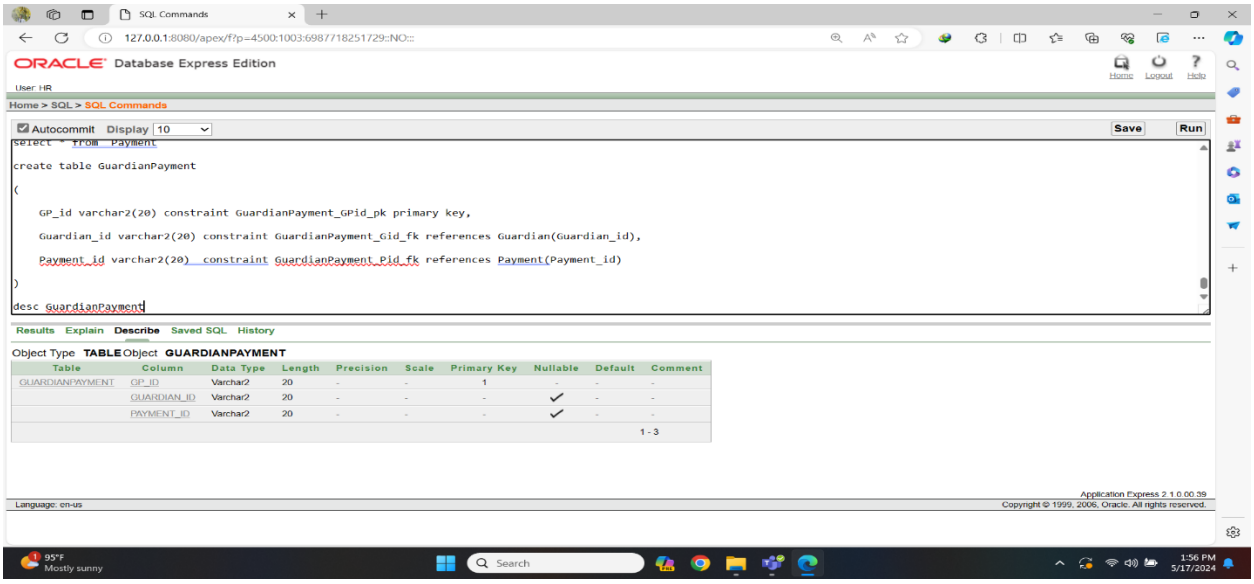


Fig-39: Guardian-Payment Table Create

```
insert into GuardianPayment (GP_id, Guardian_id, Payment_id) values ('GP1', 'G001', 'P1')
insert into GuardianPayment (GP_id, Guardian_id, Payment_id) values ('GP2', 'G002', 'P2')
insert into GuardianPayment (GP_id, Guardian_id, Payment_id) values ('GP3', 'G003', 'P3')
insert into GuardianPayment (GP_id, Guardian_id, Payment_id) values ('GP4', 'G004', 'P4')
insert into GuardianPayment (GP_id, Guardian_id, Payment_id) values ('GP5', 'G005', 'P5')
insert into GuardianPayment (GP_id, Guardian_id, Payment_id) values ('GP6', 'G006', 'P6')
insert into GuardianPayment (GP_id, Guardian_id, Payment_id) values ('GP7', 'G007', 'P7')
insert into GuardianPayment (GP_id, Guardian_id, Payment_id) values ('GP8', 'G008', 'P8')
insert into GuardianPayment (GP_id, Guardian_id, Payment_id) values ('GP9', 'G009', 'P9')
insert into GuardianPayment (GP_id, Guardian_id, Payment_id) values ('GP10', G010', 'P10')
```

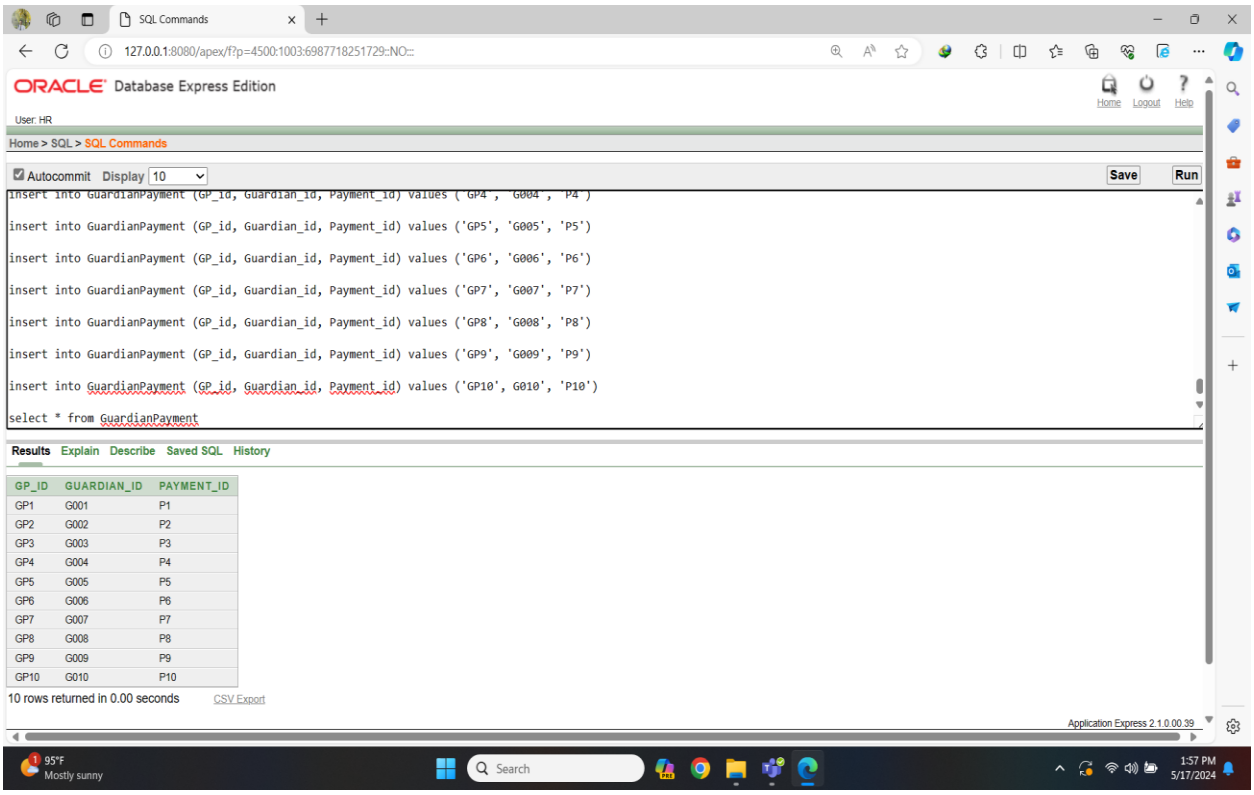


Fig-40: Guardian-Payment Table Data

Query Test

A) Conditional Statement:

1.Display the names of all students whose names have 'a' as the second character.
⇒ select Student_name from Student where Student_name like '_a%'

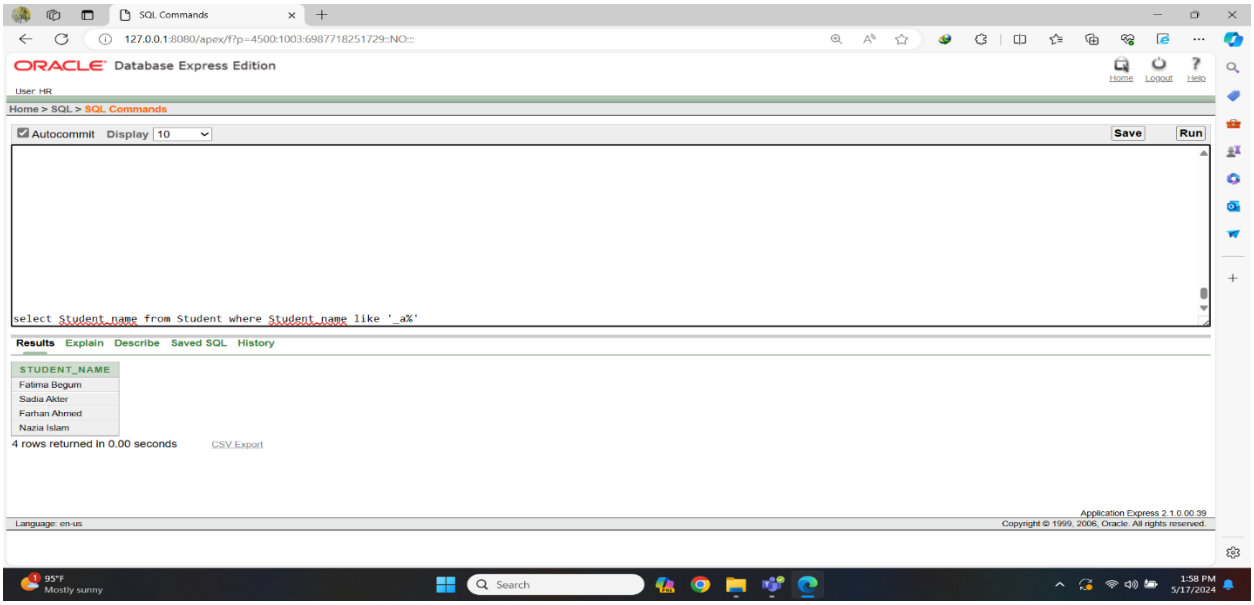


Fig-1: Conditional Statement Query

2. Display all information for teachers who earn more than 50,000.
⇒ select * from Salary where Teacher_salary > 50000

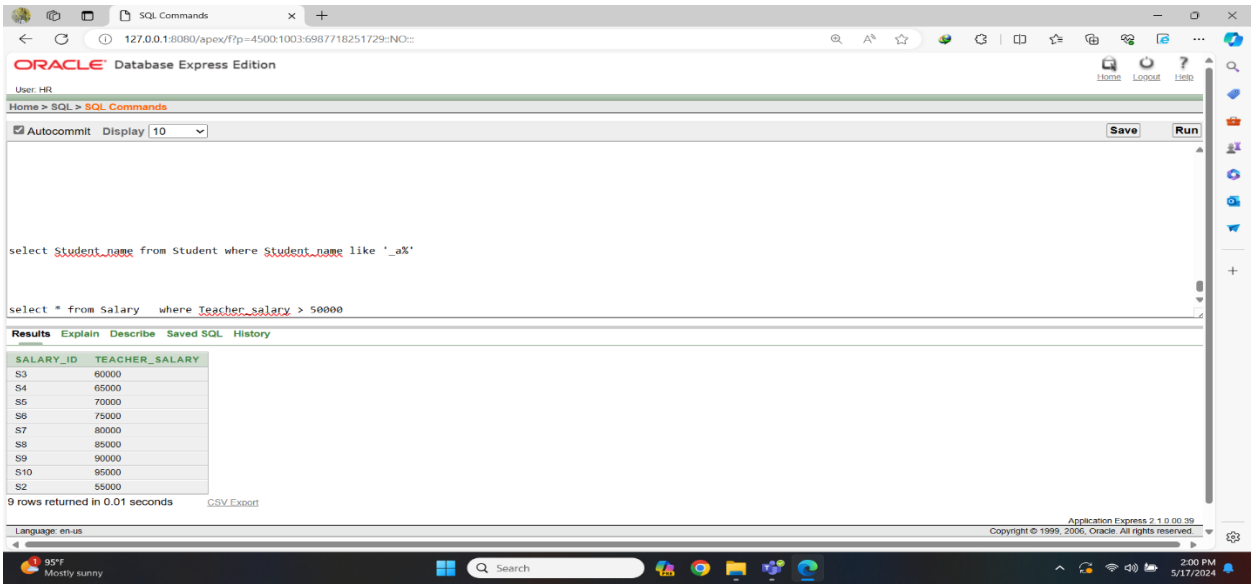


Fig-2: Conditional Statement Query

3. Display all information for students who are in class '10'.
⇒ select * from Student where Class = '10'

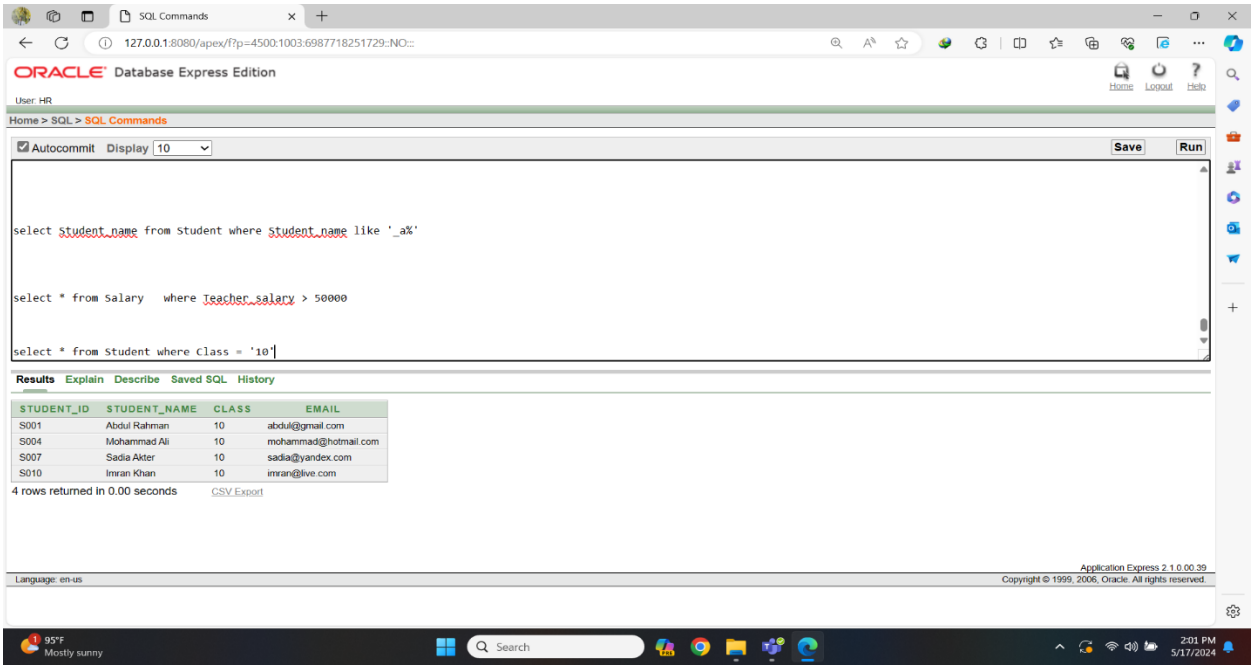


Fig-3: Conditional Statement Query

4.Display all information for the administrator with the Admin_id 'A003'.
⇒ select * from Admin where Admin_id = 'A003'

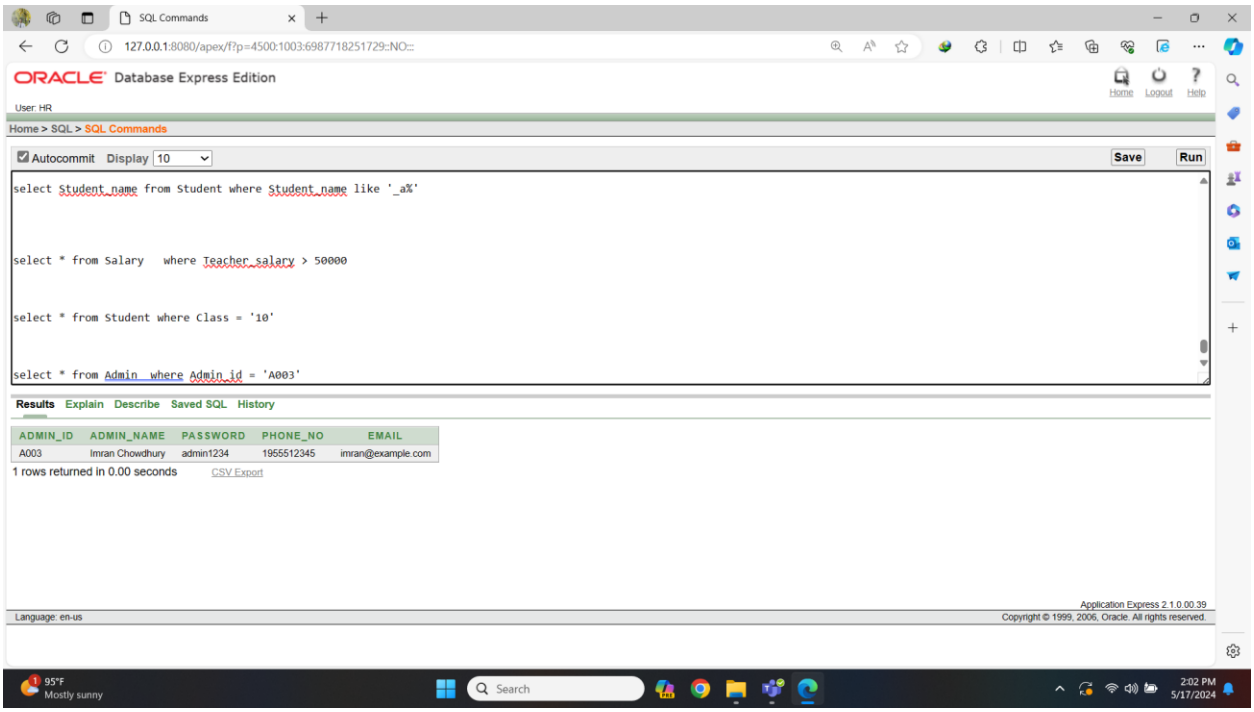


Fig-4: Conditional Statement Query

5. Display the Teacher_id and Address of teachers who have a qualification of 'Masters in Computer Science'.
⇒ select Teacher_id,Address from Teacher where Qualification = 'Masters in Computer Science'

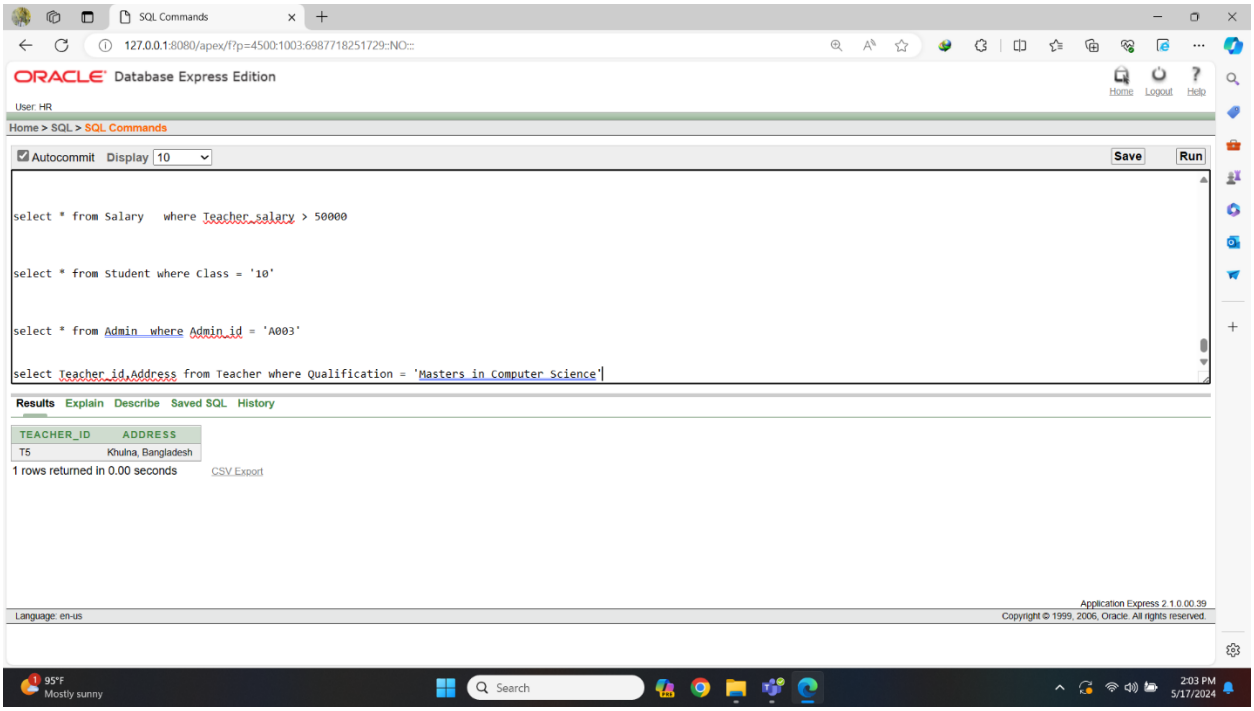


Fig-5: Conditional Statement Query

B) Single row function:

1.Display the Guardian_id, Guardian_name and a concatenated string of Guardian_id and Guardian_name for all guardians.
⇒ select Guardian_id,Guardian_name, CONCAT(Guardian_id,Guardian_name) from Guardian

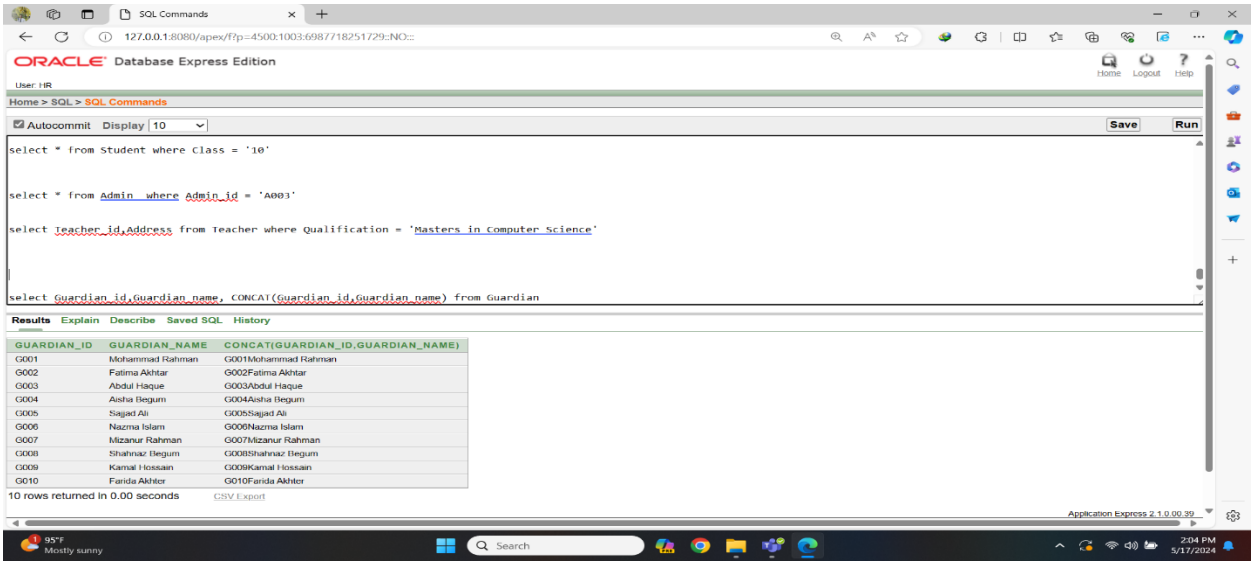


Fig-6: Single Row Function Query

2. Display the Guardian_id and Guardian_name in uppercase for all guardians.
⇒ select Guardian_id,UPPER(Guardian_name) from Guardian

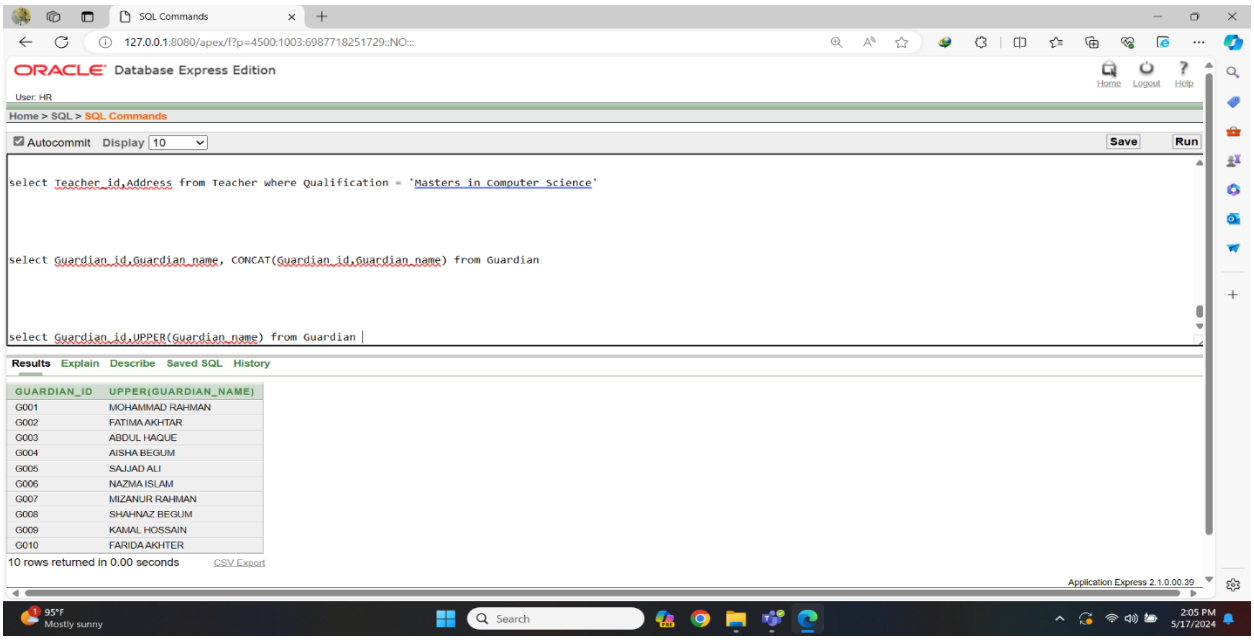


Fig-7: Single Row Function Query

3. Display the Guardian_id, Guardian_name, and the first three characters of Guardian_name for all guardians. Alias this substring as first_three_characters_name.
⇒ select Guardian_id, Guardian_name ,SUBSTR(Guardian_name, 1, 3) as first_three_characters_name from Guardian

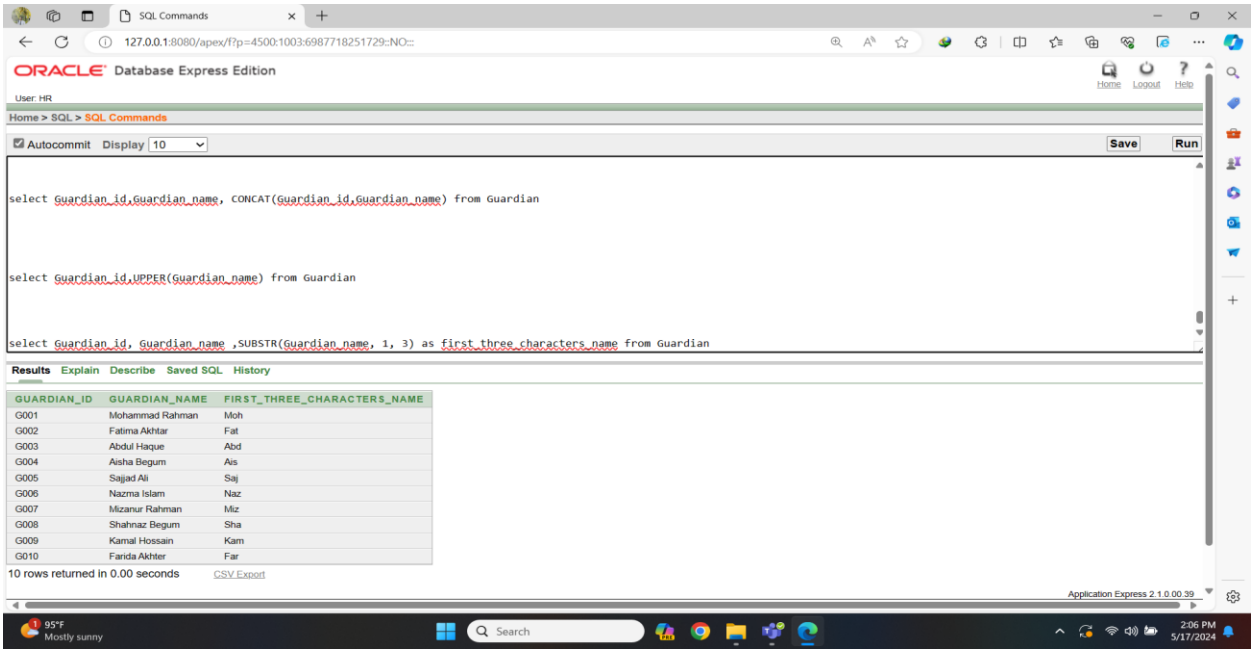


Fig-8: Single Row Function Query

4. Display all columns for teachers whose Teacher_id is between 'T2' and 'T4'.
⇒ select * from Teacher where Teacher_id between 'T2' and 'T4'

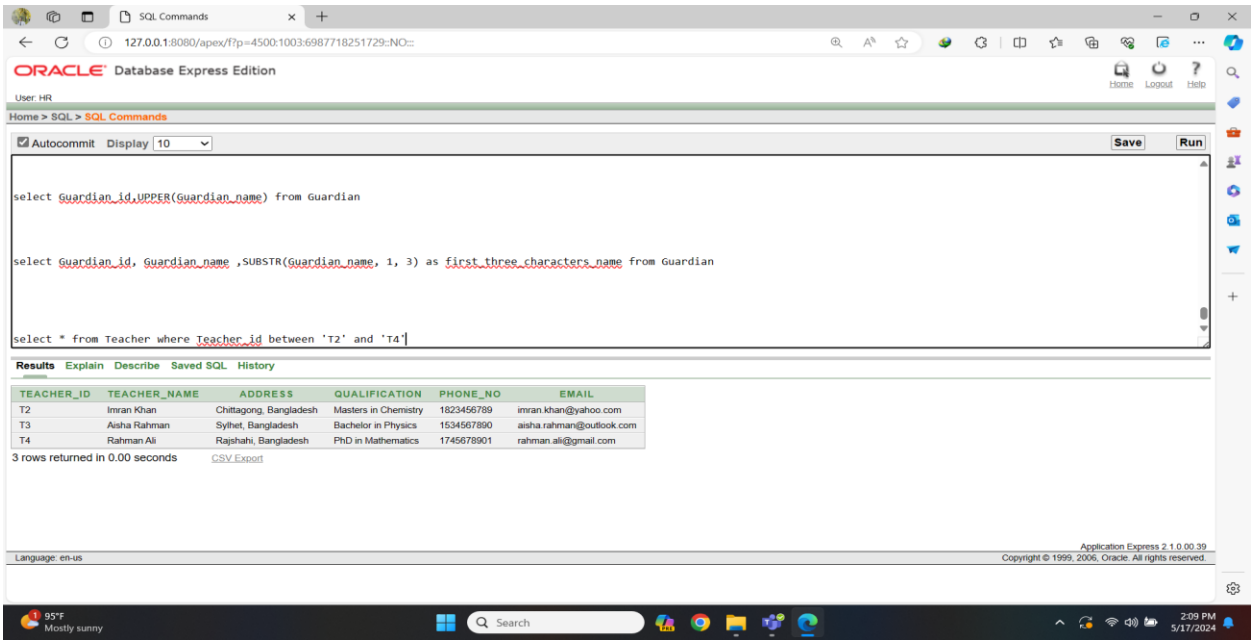


Fig-9: Single Row Function Query

10.Display the Admin_name and the length of Admin_name for all administrators.
⇒ select Admin_name, LENGTH(Admin_name) as Admin_name_length from Admin

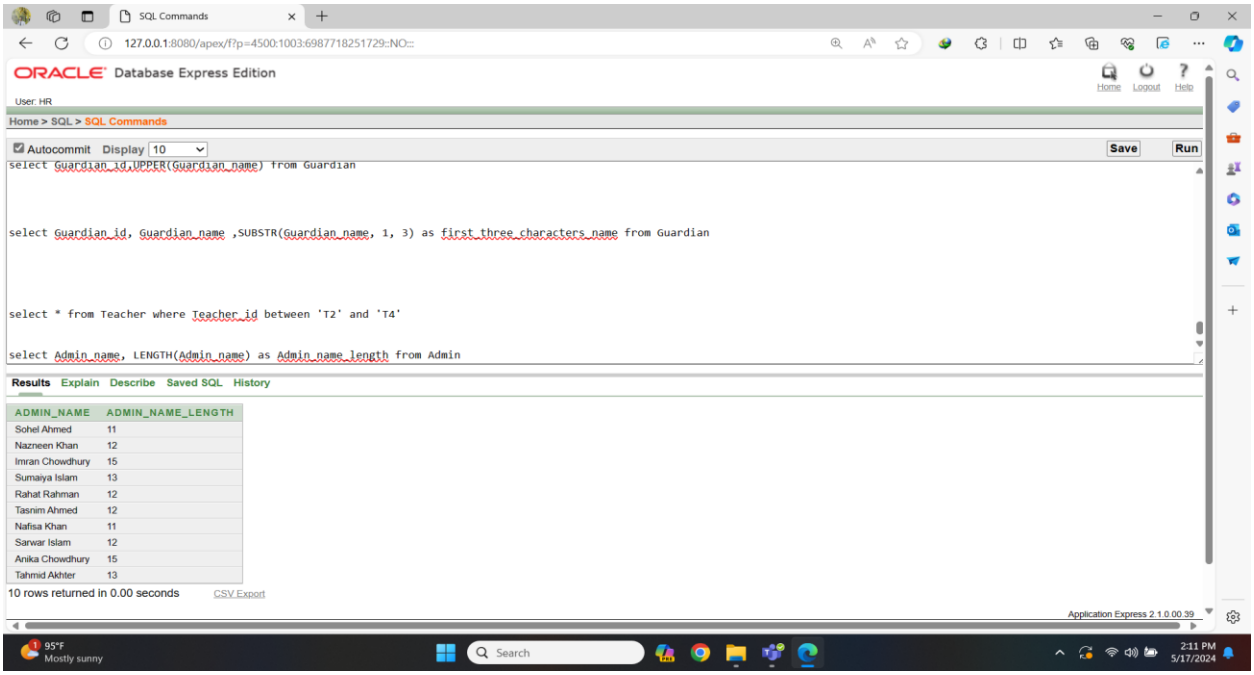


Fig-10: Single Row Function Query

C) Multiple row function:

1. Display the minimum amount paid by any guardian from payment table.
⇒ select min(Student_fee) from Payment

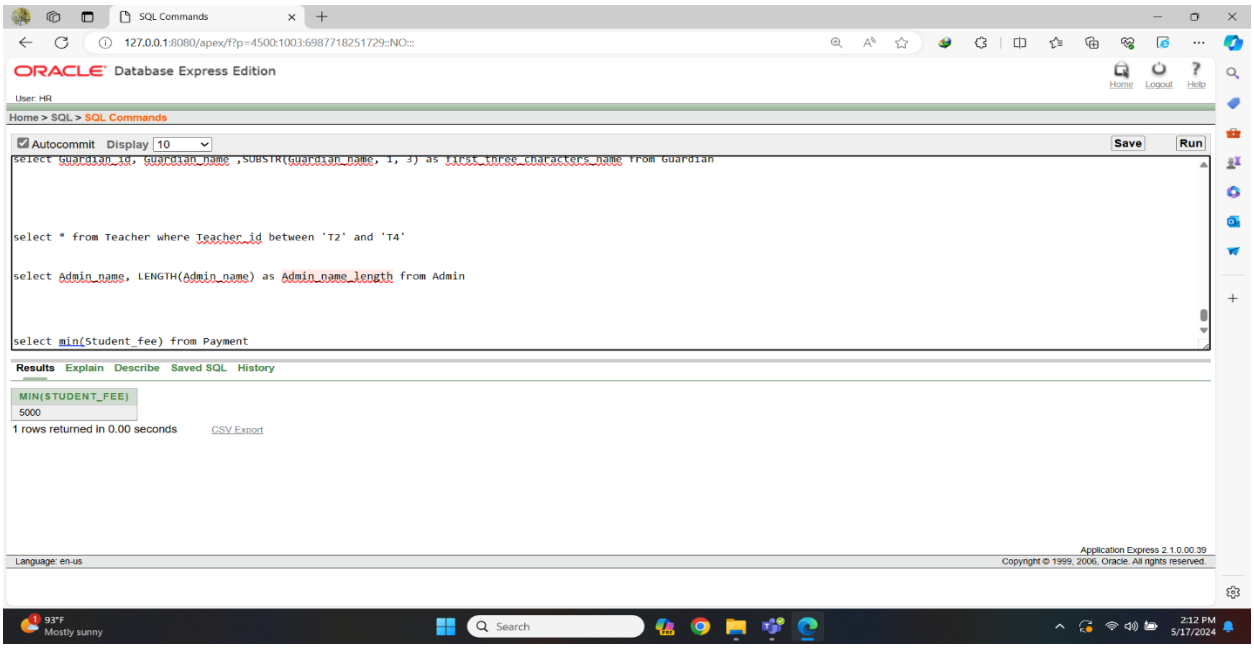


Fig-11: Multiple Row Function Query

2. Write an SQL query to count the total number of rooms in the room table.
⇒ select count(Room_no) from Room

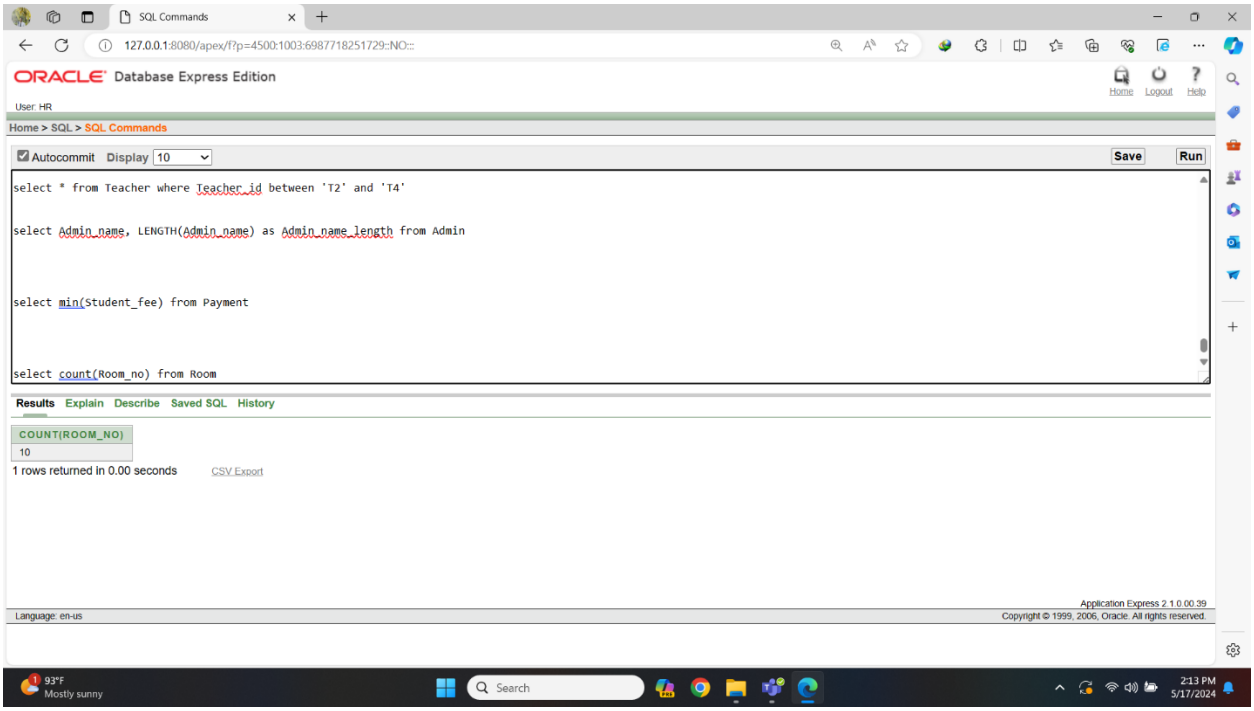


Fig-12: Multiple Row Function Query

3. Display all information for teachers whose Teacher_id is not T3, T5, T8 and T10.
⇒ select * from Teacher where Teacher_id not in('T3', 'T5', 'T8', 'T10')

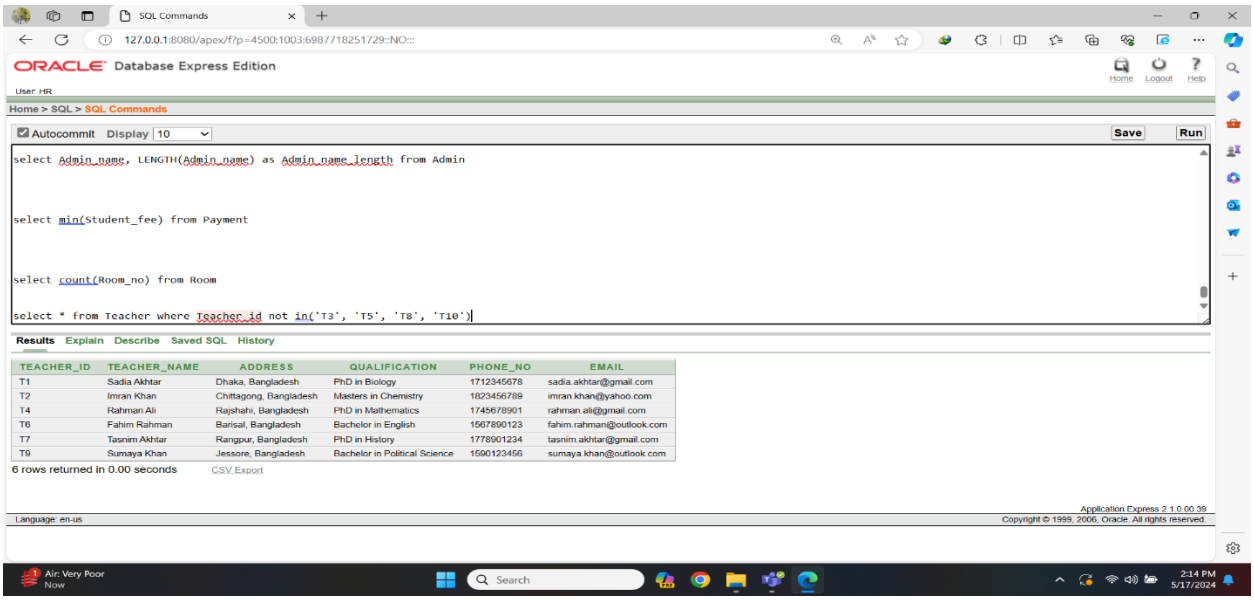


Fig-13: Multiple Row Function Query

D) Subquery:

1.Display the Student_name and Student_id of students who belong to the highest class in the school.
⇒ select Student_name, Student_id from Student where Class = (select max(Class) from Student)

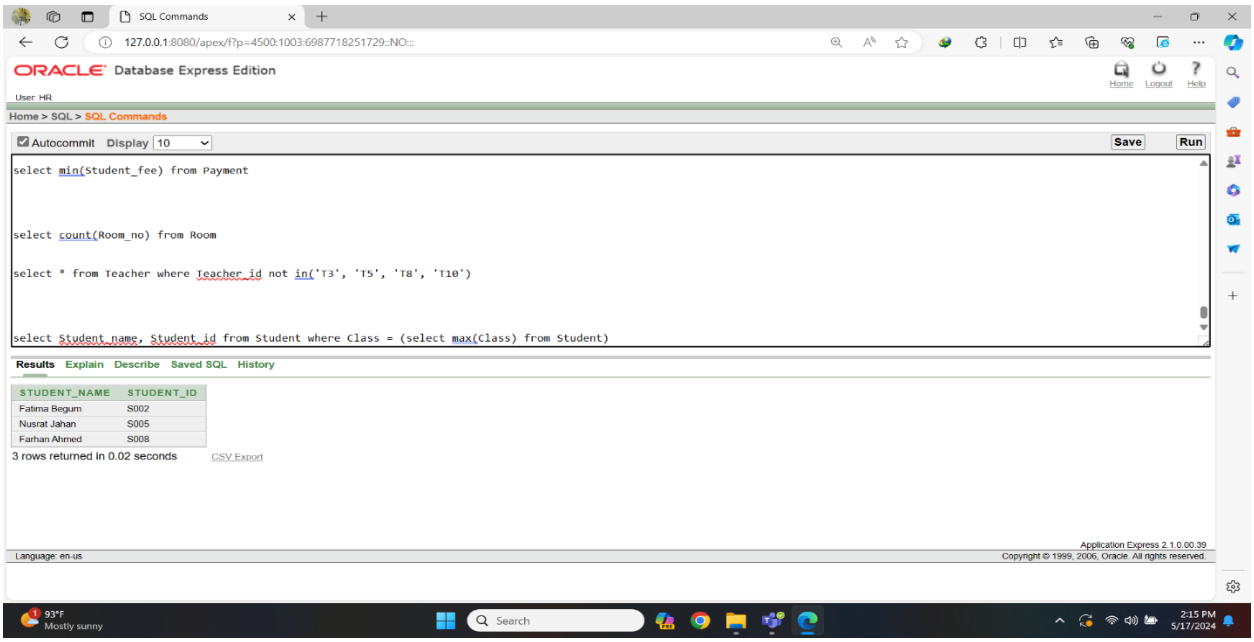


Fig-14: Subquery

2. Display the password of administrators whose Admin_id are greater than 'A004'.
⇒select Password from Admin where Admin_id in(select Admin_id from Admin where Admin_id > 'A004')

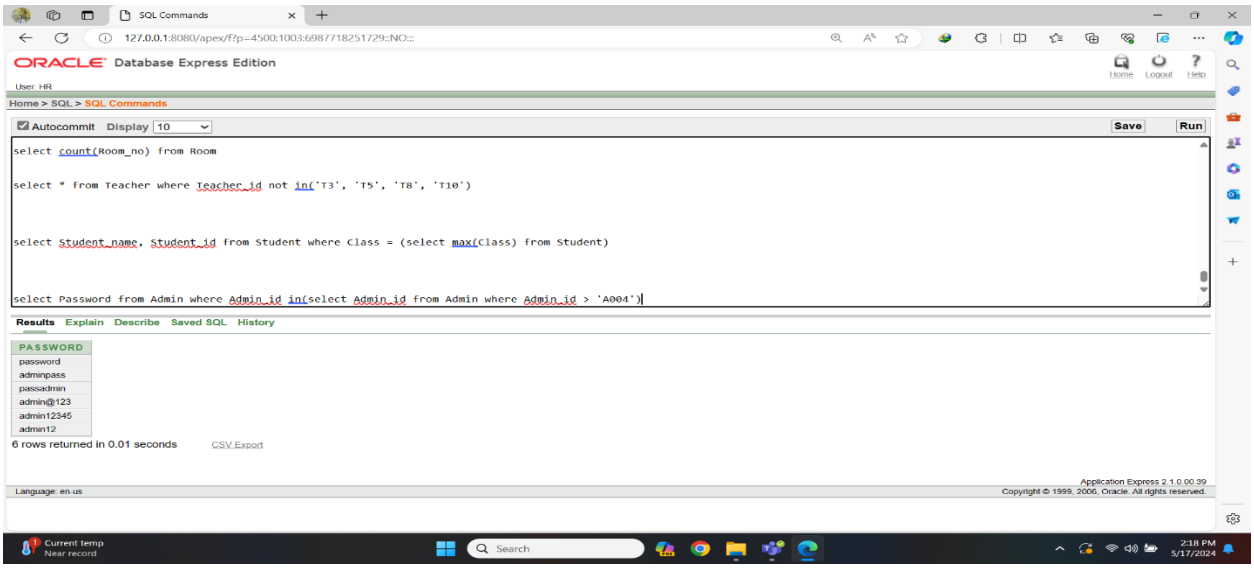


Fig-15: Subquery

E) Join Query:

- Equi Join:

Display the Teacher_id, Teacher_name, Qualification, and Subject_id from the Teacher table and TeacherSubject table, where the records in both tables are related by the Teacher_id column.
⇒select t.Teacher_id, t.Teacher_name , t.Qualification , ts.Subject_id from Teacher t, TeacherSubject ts where t.Teacher_id = ts.Teacher_id

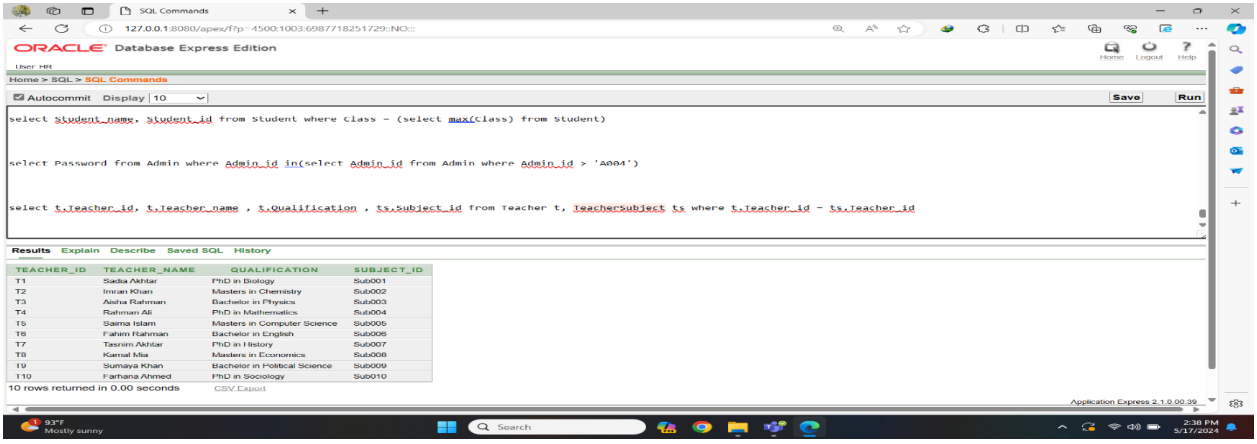


Fig-16: Equi Join Query

- Non-equi Join:

Display the Teacher_id, Teacher_name, Qualification, and Subject_id for each teacher, excluding cases where a teacher is assigned a subject that doesn't match their own ID.
⇒select t.Teacher_id, t.Teacher_name , t.Qualification , ts.Subject_id from Teacher t, TeacherSubject ts where ts.Teacher_id != t.Teacher_id

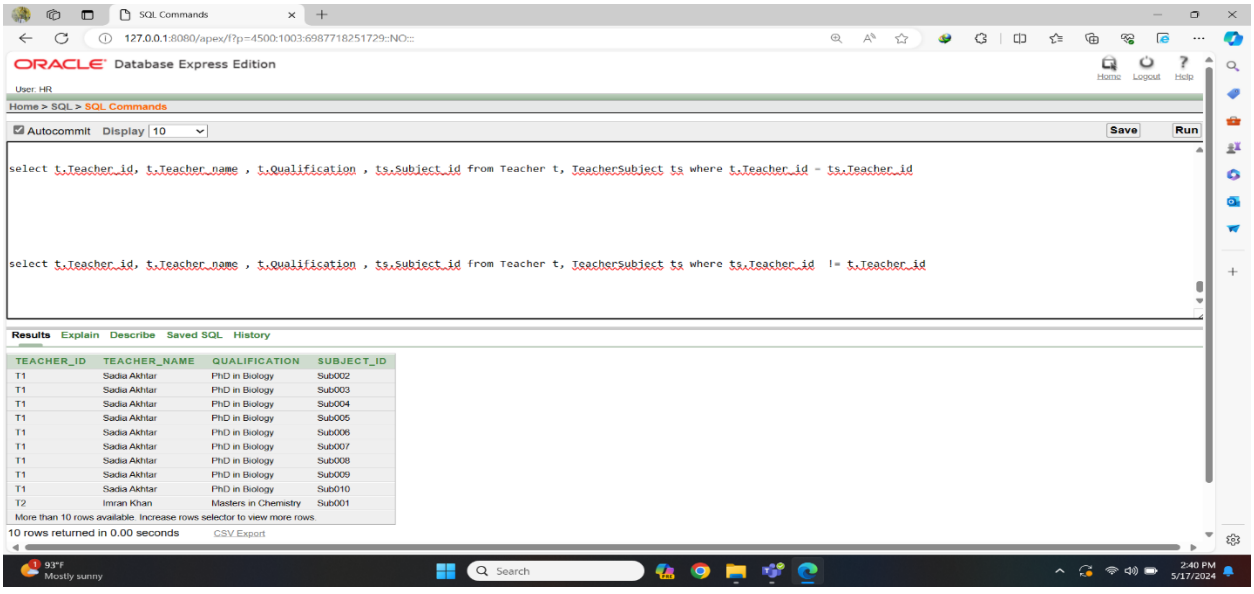


Fig-17: Non-equi Join Query

- Self join:

Display the information of exam id and exam name from the self join of both in Exam table whose gpa is greater than 4.50 and total marks is less than 1000

⇒ select 'Exam id is ' || e1.Exam_id || ' and exam name is ' || e2.Exam_name from Exam e1, Exam e2 where e1.GPA > 4.50 and e2.Total_marks < 1000

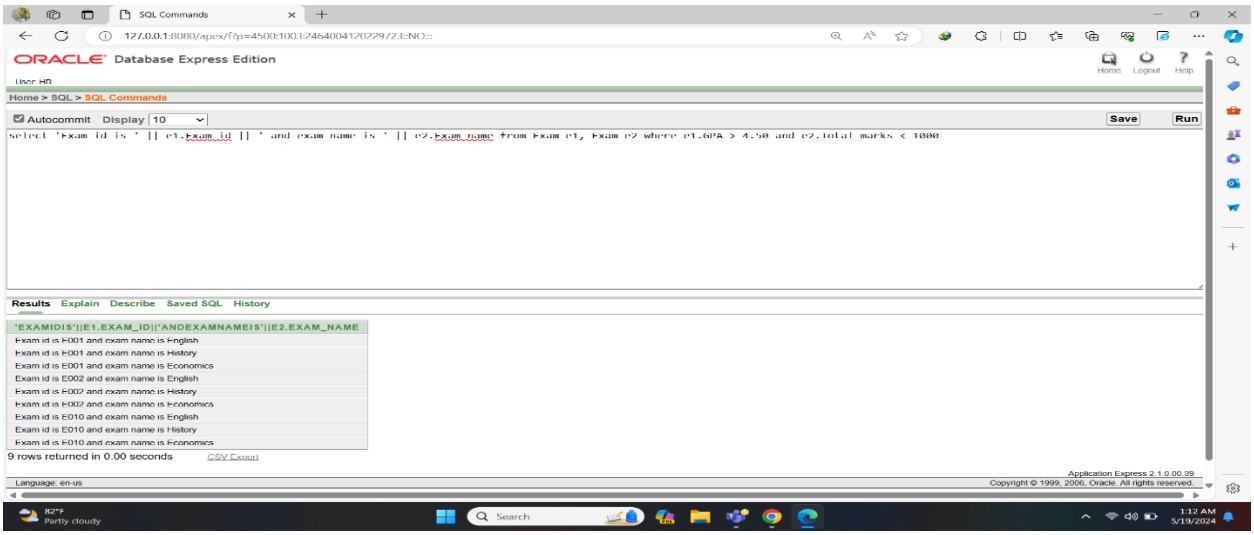


Fig-18: Self join Query

- Outer join:

Display the guardian id, name,address, phone number along with their corresponding payment information payment id from the Guardian and GuardianPayment table

⇒ select g.Guardian_id, g.Guardian_name, g.AC_id, g.Phone_no, gp.GP_id, gp.Payment_id from Guardian g, GuardianPayment gp where g.Guardian_id = gp.GP_id(+) order by g.Guardian_id

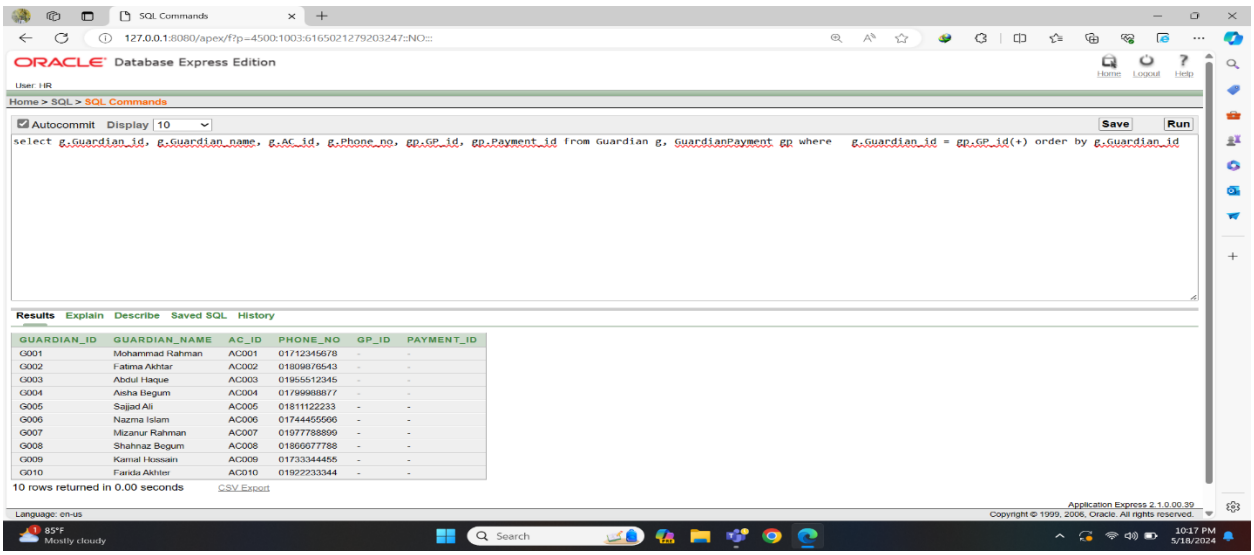


Fig-19: Outer join Query

F) View:

Create a view for the Guardian_id, Guardian_name, AC_id, Phone_no, Student_id, Student_name, Class , Email of the guardians and students included in the Guardian_Student_View and Guardian_id in ‘G001’, ‘G002’ and Student_id in ‘S001’, ‘S002’

⇒ create view Guardian_Student_View as select g.Guardian_id, g.Guardian_name, g.AC_id, g.Phone_no, s.Student_id, s.Student_name, s.Class, s.Email from Guardian g, Student s where g.Guardian_id in('G001','G010') and s.Student_id in('S001','S010')

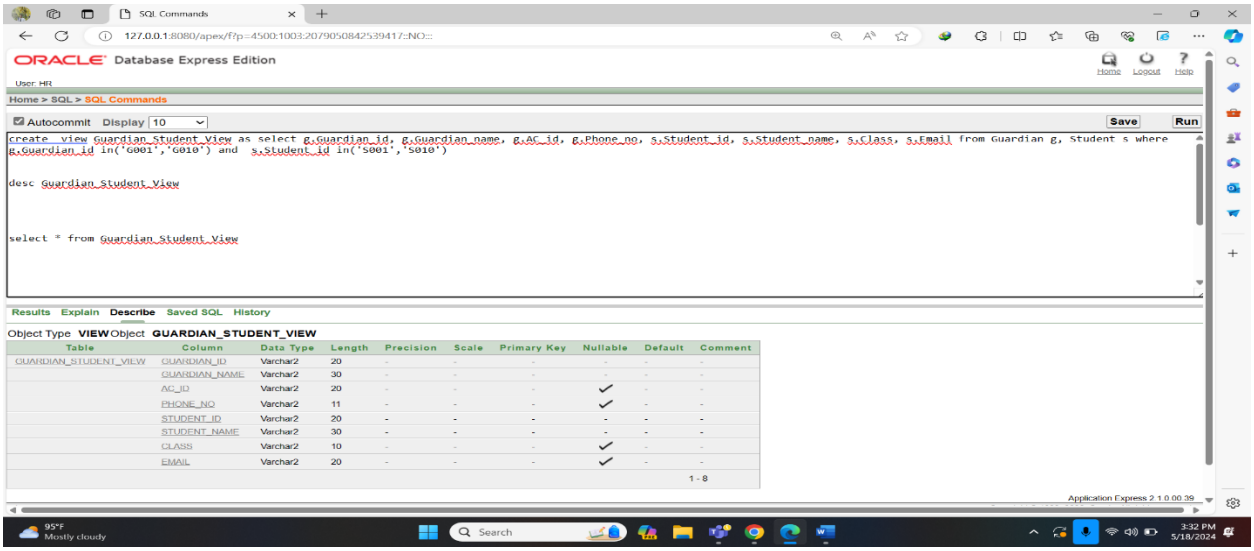


Fig-20: View Creation

select * from Guardian_Student_View

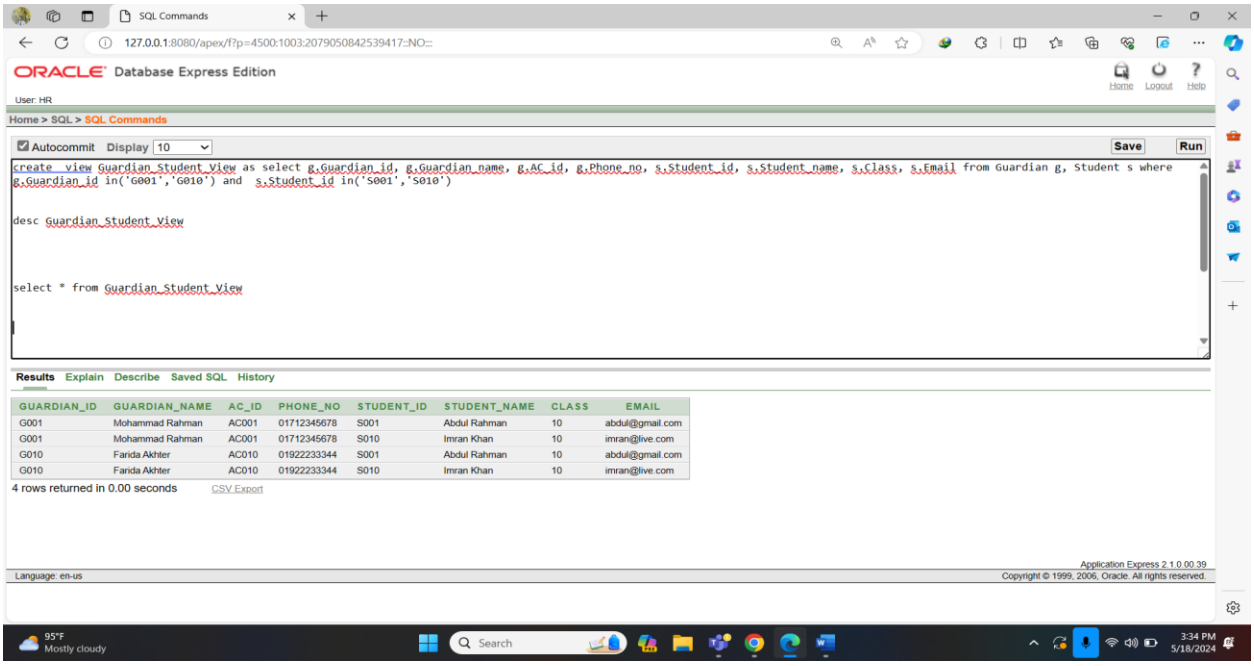


Fig-21: View Data

G) Sequence:

```
create sequence Receipt_id

minvalue 1
maxvalue 100
start with 6
increment by 5
nocycle
nocache


create table Receipt
(
   Receipt_id number(38),
   Student_id varchar2(20) constraint Receipt_Sid_fk references Student(Student_id),
   Student_name varchar2(30),
   Total_Ammount number(10),
   Receipt_date date
)
```

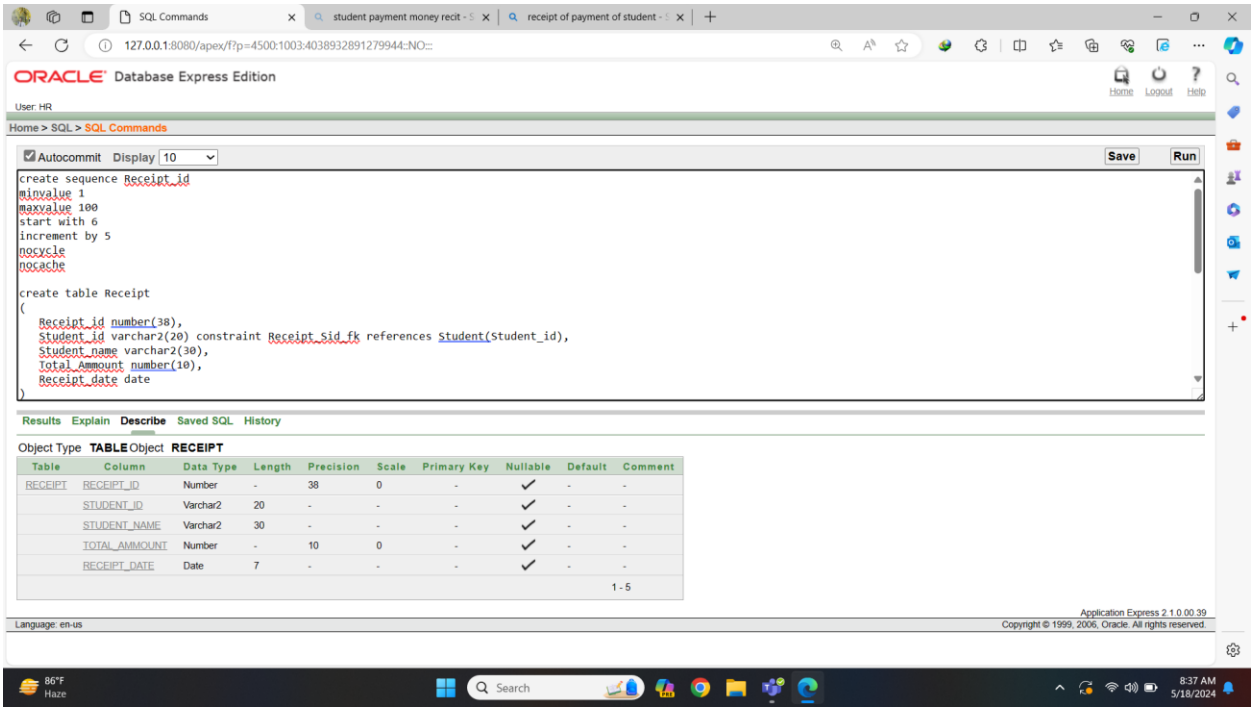


Fig-22: Sequence Create

```
insert into Receipt (Receipt_id, Student_id, Student_name, Total_Ammount, Receipt_date)
values (Receipt_id.nextval,'S001', 'Abdul Rahman', 5000, sysdate)
insert into Receipt (Receipt_id, Student_id, Student_name, Total_Ammount, Receipt_date)
values (Receipt_id.nextval,'S002', 'Fatima Begum', 5500, sysdate)
insert into Receipt (Receipt_id, Student_id, Student_name, Total_Ammount, Receipt_date)
values (Receipt_id.nextval,'S003', 'Ayesha Khan', 6000, sysdate)
insert into Receipt (Receipt_id, Student_id, Student_name, Total_Ammount, Receipt_date)
values (Receipt_id.nextval,'S004', 'Mohammad Ali', 6500, sysdate)
insert into Receipt (Receipt_id, Student_id, Student_name, Total_Ammount, Receipt_date)
values (Receipt_id.nextval,'S005', 'Nusrat Jahan', 7000, sysdate)
insert into Receipt (Receipt_id, Student_id, Student_name, Total_Ammount, Receipt_date)
values (Receipt_id.nextval,'S006', 'Arif Hasan', 7500, sysdate)
insert into Receipt (Receipt_id, Student_id, Student_name, Total_Ammount, Receipt_date)
values (Receipt_id.nextval,'S007', 'Sadia Akter', 8000, sysdate)
insert into Receipt (Receipt_id, Student_id, Student_name, Total_Ammount, Receipt_date)
values (Receipt_id.nextval,'S008', 'Farhan Ahmed', 8500, sysdate)
insert into Receipt (Receipt_id, Student_id, Student_name, Total_Ammount, Receipt_date)
values (Receipt_id.nextval,'S009', 'Nazia Islam',  9000, sysdate)
insert into Receipt (Receipt_id, Student_id, Student_name, Total_Ammount, Receipt_date)
values (Receipt_id.nextval,'S010', 'Imran Khan',  9500, sysdate)
```

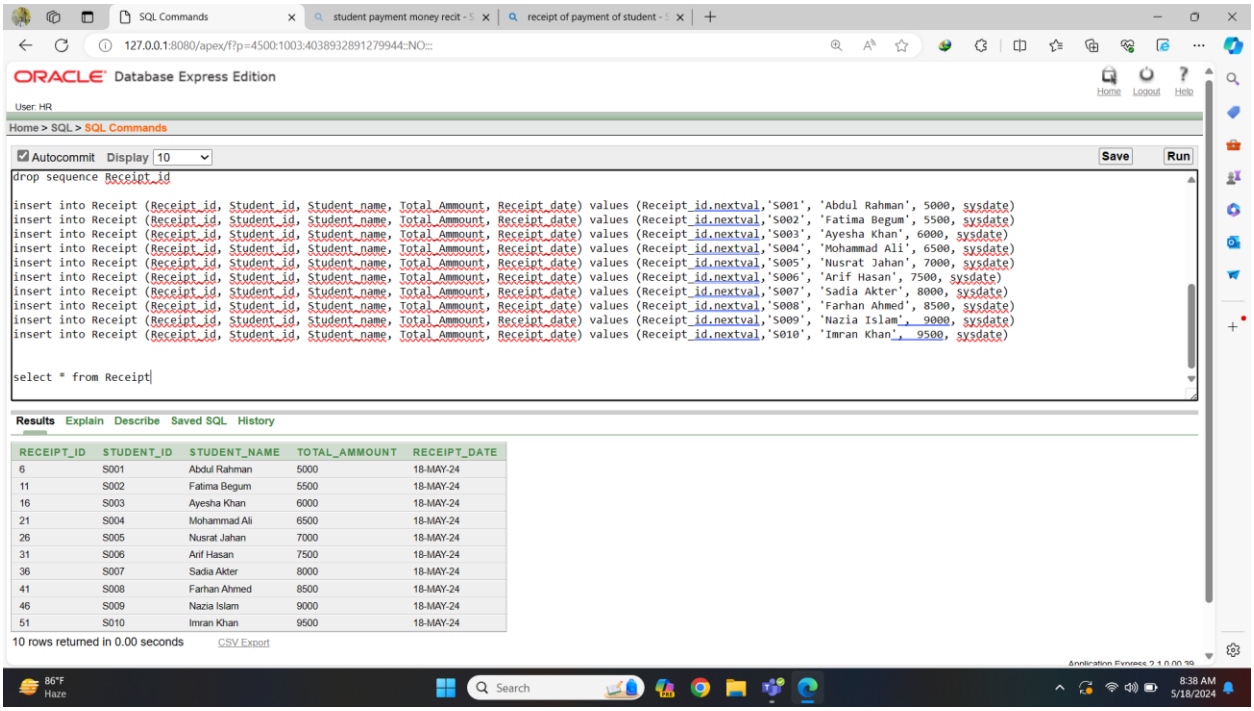


Fig-23: Sequence Data

Relational Algebra

1. Display all information from Admin where Admin id is A005.

$$\text{Admin}\ \sigma\ \text{Admin\_id} = \text{'A005'}$$

2. Display Student id, email those class is 9

$$\left[\ \text{Student}\ \sigma\ \text{Class} = \text{'9'}\ \right]$$

$$\pi$$

Student_id,Email

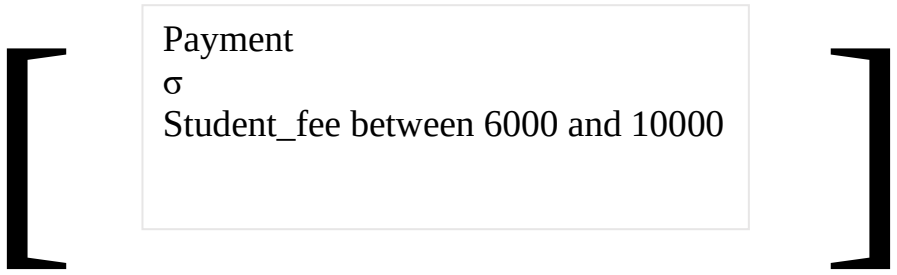
3. Display the Guardian ID, name whose id under G007.

$$\left[\ \text{Guardian}\ \sigma\ \text{Guardian\_id} < \text{'G007'}\ \right]$$

$$\pi$$

Guardian_id, Guardian_name

4. Display the Payment id and student fee whose fee between 6000 and 10000

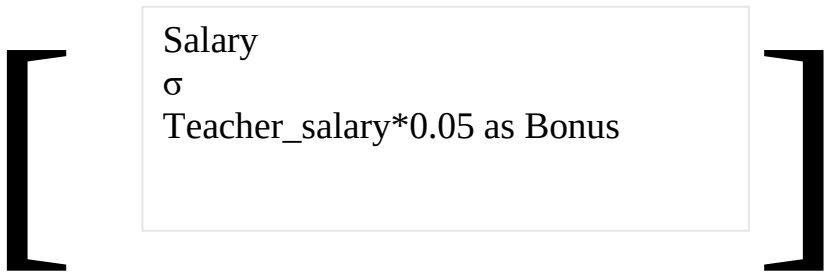


π

Payment_id, Student_fee

5. Display the Teacher maximum and minimum salary

6. Write a query to display the salary id, teacher salary where bonus is 5% of their salary.



π

Salary_id, Teacher_salary

7. Display Admin id and name where the third letter of their name is h.

$$\left[\begin{array}{l}\text{Admin}\\ \sigma\\ \text{Admin\_name like '\_\_h\%'}\end{array}\right]$$

$$\pi$$

Admin_id, Admin_name

8. Display teacher id , name and salary for all teacher.

$$\left[\begin{array}{l}\text{Teacher x TeacherSalary x Salary}\\ \sigma\\ \text{t.Teacher\_id = ts.Teacher\_id and ts.Salary\_id = s.Salary\_id}\end{array}\right]$$

$$\pi$$

t.Teacher_id, t.Teacher_name , s.Teacher_salary

9. Display all information of guardian whose id is G002

Guardian
σ
Guardian_id = ‘G002’

10. Display the student maximum and minimum total marks in exam

Exam
σ
max(Total_marks), min(Total_marks)

Conclusion

In conclusion, adding the student-guardian-teacher examination module to the coaching management system is a major improvement in educational technology. This addition promotes better collaboration, transparency, and accountability among students, parents, and teachers. Using database technology, the system simplifies communication, allows real-time progress tracking, and provides useful insights for making informed decisions. These changes not only enhance the educational experience but also create a solid foundation for ongoing improvements and innovations. By giving everyone involved timely information and a clear understanding of each student's progress, this system fosters a supportive and engaged learning environment. Parents can offer more effective support, teachers can adjust their instruction to meet individual needs, and students can benefit from a more unified and responsive educational framework. The integration of this module helps in identifying trends and areas for improvement, ensuring that the coaching management system adapts to the changing needs of education. Overall, this advancement shows the powerful impact of technology in education, driving continuous improvement and enabling all participants to actively contribute to the learning journey. As a result, the coaching management system is better equipped to prepare students for future challenges, making it a vital tool in the pursuit of academic excellence and overall development.