Database Management System(UE19CS301)

Assignment - 3

Team Details

	NAME	SRN
1	**Shatakshi Mohan**	**PES2UG19CS379**
2	**Siddhi Patil**	**PES2UG19CS389**
3	**Sonit Pradhan**	**PES2UG19CS399**

Problem Statement

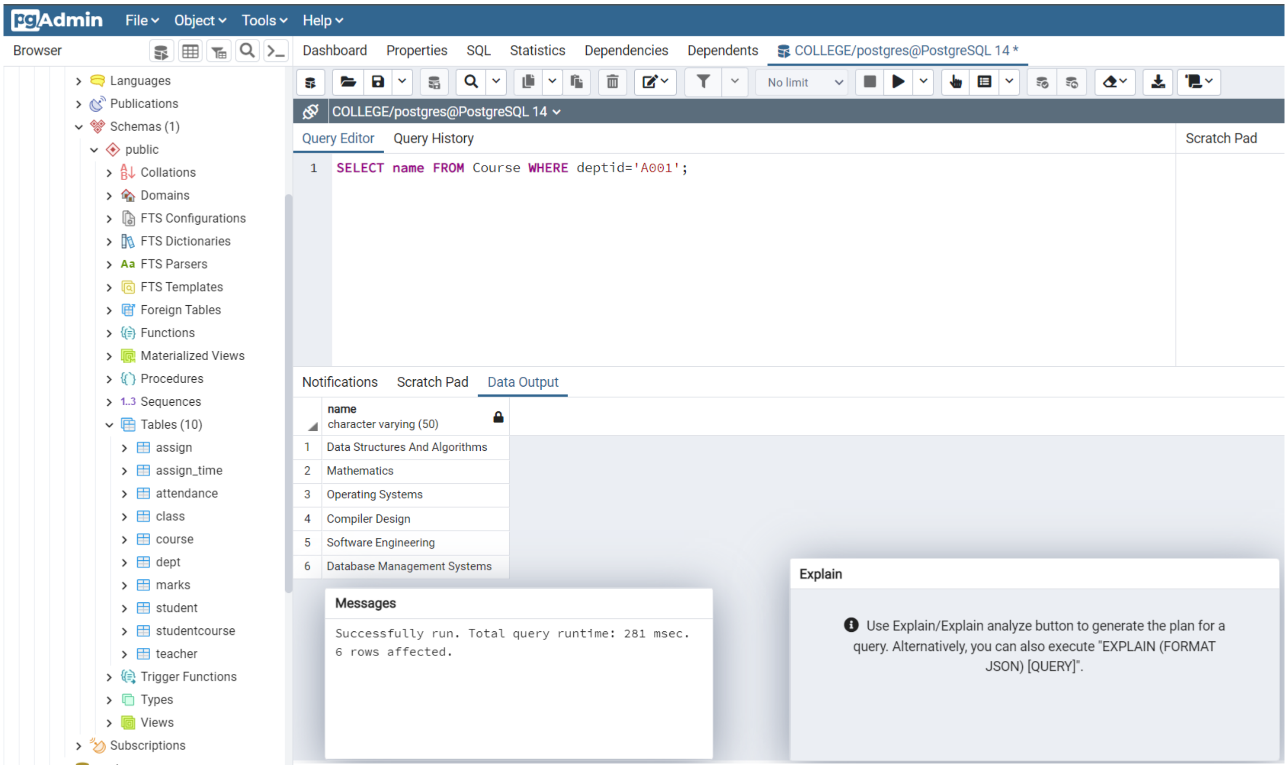
Our project aims to integrate the modules and functionalities of a college system in a single application with our **College Enterprise Resource Planning (ERP)** system. Using the Django framework we intend to design it for interactions between students and teachers. A database management system based on sqlite is what we have used to store and manage our data.

Queries

Simple, Complex, Nested Queries

Simple Queries:

1. Select all the course names offered by the CSE department.(‘A001’)



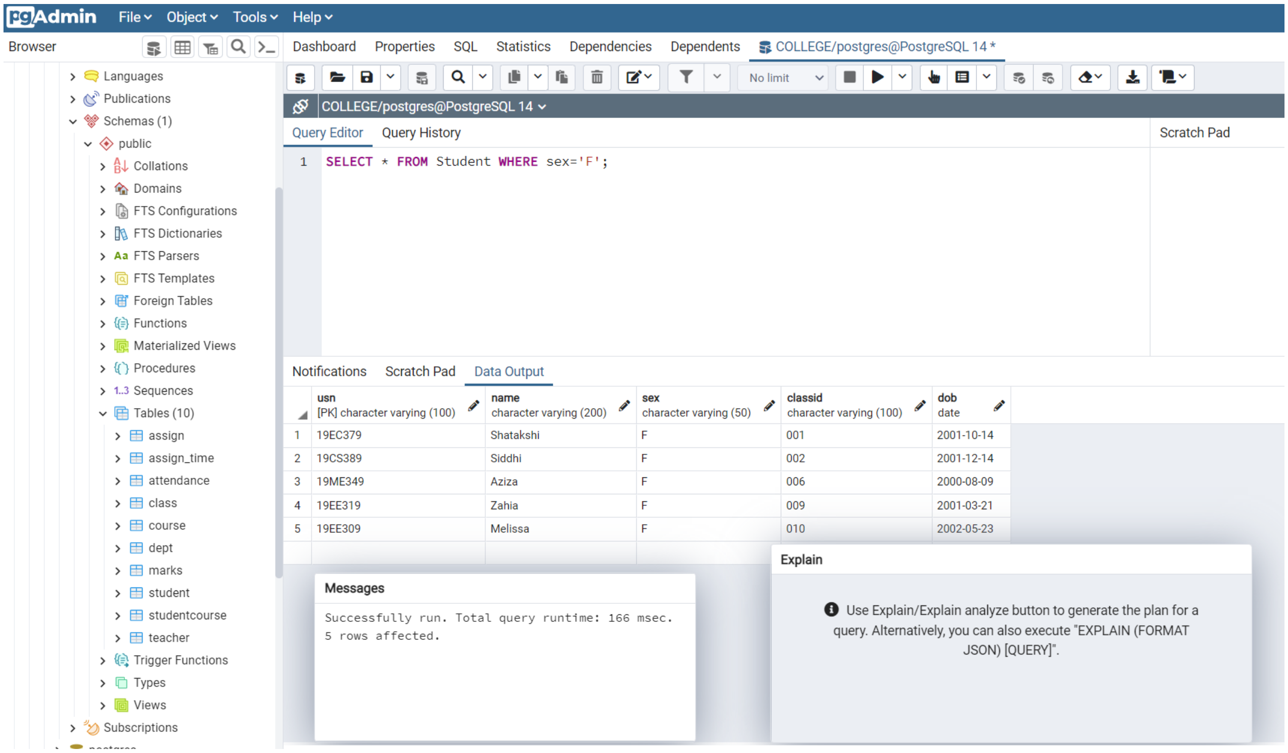
```sql
SELECT name FROM Course WHERE deptid='A001';
```

	name character varying (50)
1	Data Structures And Algorithms
2	Mathematics
3	Operating Systems
4	Compiler Design
5	Software Engineering
6	Database Management Systems

```
Successfully run. Total query runtime: 281 msec.
6 rows affected.
```

Use Explain/Explain analyze button to generate the plan for a query. Alternatively, you can also execute "EXPLAIN (FORMAT JSON) [QUERY]".

2. Display the details of all female students.



```
SELECT * FROM Student WHERE sex='F';
```

	usn [PK] character varying (100)	name character varying (200)	sex character varying (50)	classid character varying (100)	dob date
1	19EC379	Shatakshi	F	001	2001-10-14
2	19CS389	Siddhi	F	002	2001-12-14
3	19ME349	Aziza	F	006	2000-08-09
4	19EE319	Zahia	F	009	2001-03-21
5	19EE309	Melissa	F	010	2002-05-23

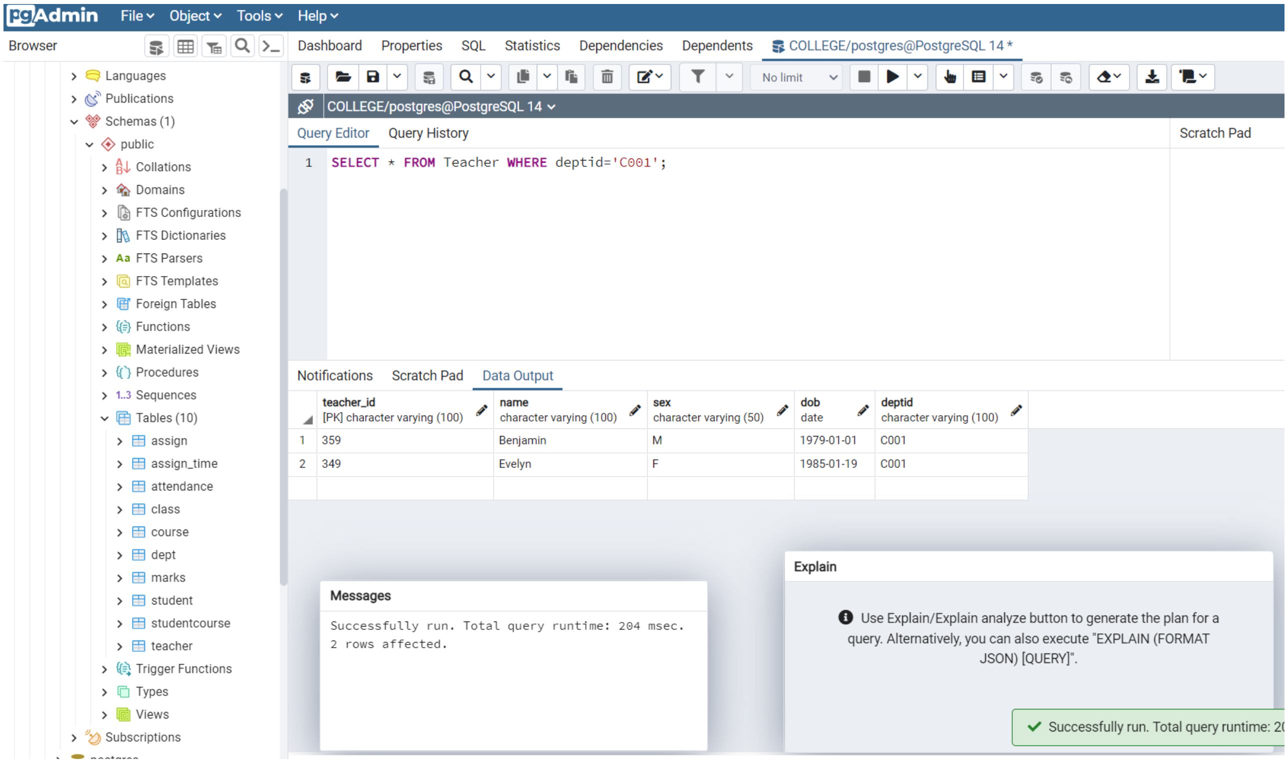
Messages

```
Successfully run. Total query runtime: 166 msec.
5 rows affected.
```

Explain

Use Explain/Explain analyze button to generate the plan for a query. Alternatively, you can also execute "EXPLAIN (FORMAT JSON) [QUERY]".

3. Display details of teachers belonging to the Civil department.



```
SELECT * FROM Teacher WHERE deptid='C001';
```

	teacher_id [PK] character varying (100)	name character varying (100)	sex character varying (50)	dob date	deptid character varying (100)
1	359	Benjamin	M	1979-01-01	C001
2	349	Evelyn	F	1985-01-19	C001

Messages

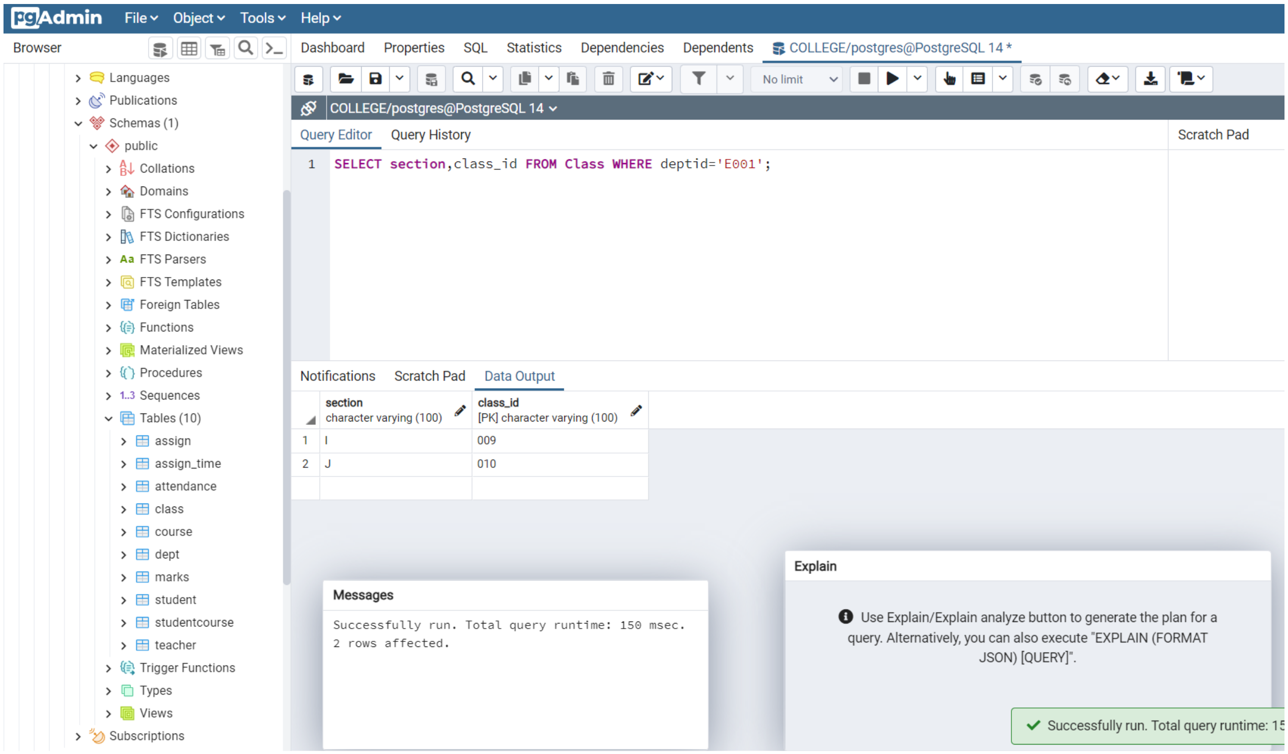
```
Successfully run. Total query runtime: 204 msec.
2 rows affected.
```

Explain

Use Explain/Explain analyze button to generate the plan for a query. Alternatively, you can also execute "EXPLAIN (FORMAT JSON) [QUERY]".

✓ Successfully run. Total query runtime: 2

4. Display the section and class_id of classes belonging to the Electrical department.



```
SELECT section,class_id FROM Class WHERE deptid='E001';
```

	section character varying (100)	class_id [PK] character varying (100)
1	I	009
2	J	010

Messages

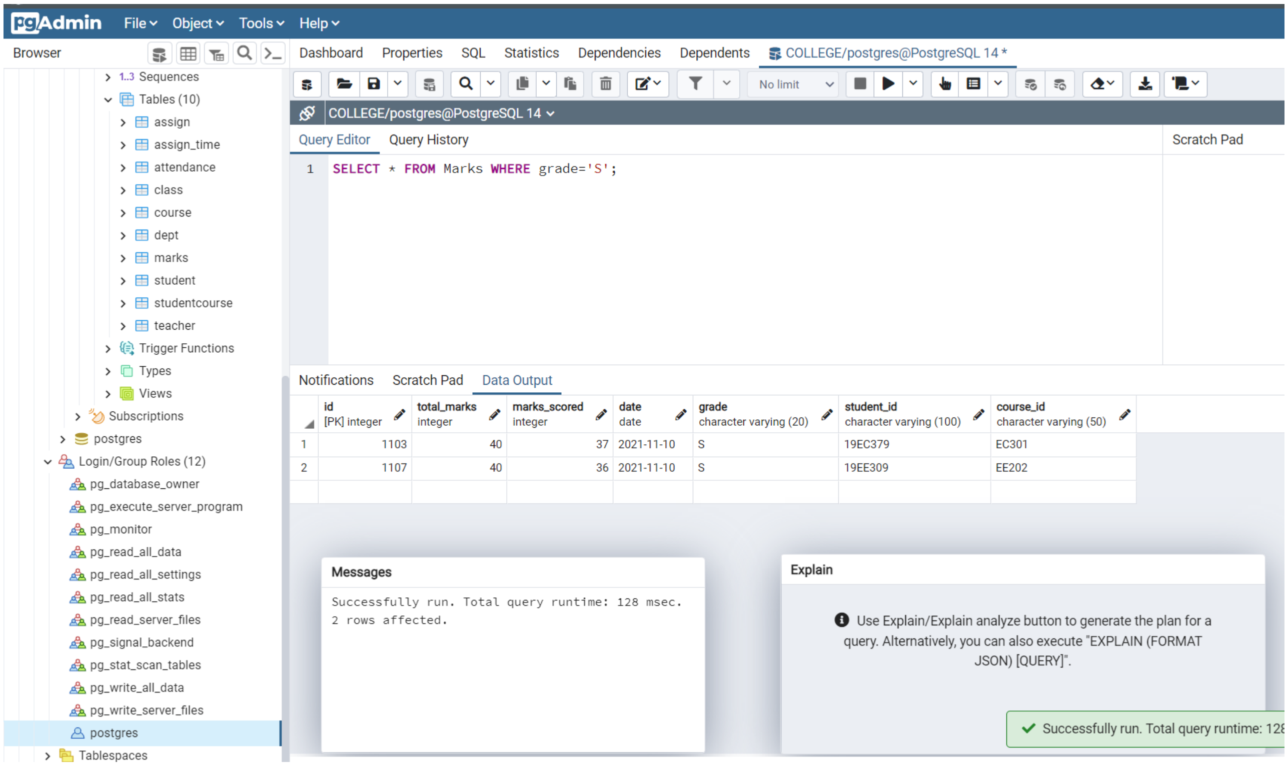
```
Successfully run. Total query runtime: 150 msec.
2 rows affected.
```

Explain

Use Explain/Explain analyze button to generate the plan for a query. Alternatively, you can also execute "EXPLAIN (FORMAT JSON) [QUERY]".

✓ Successfully run. Total query runtime: 15

5. Display all marks related details of students who obtained an ‘S’ grade.



```sql
SELECT * FROM Marks WHERE grade='S';
```

	id [PK] integer	total_marks integer	marks_scored integer	date date	grade character varying (20)	student_id character varying (100)	course_id character varying (50)
1	1103	40	37	2021-11-10	S	19EC379	EC301
2	1107	40	36	2021-11-10	S	19EE309	EE202

Messages

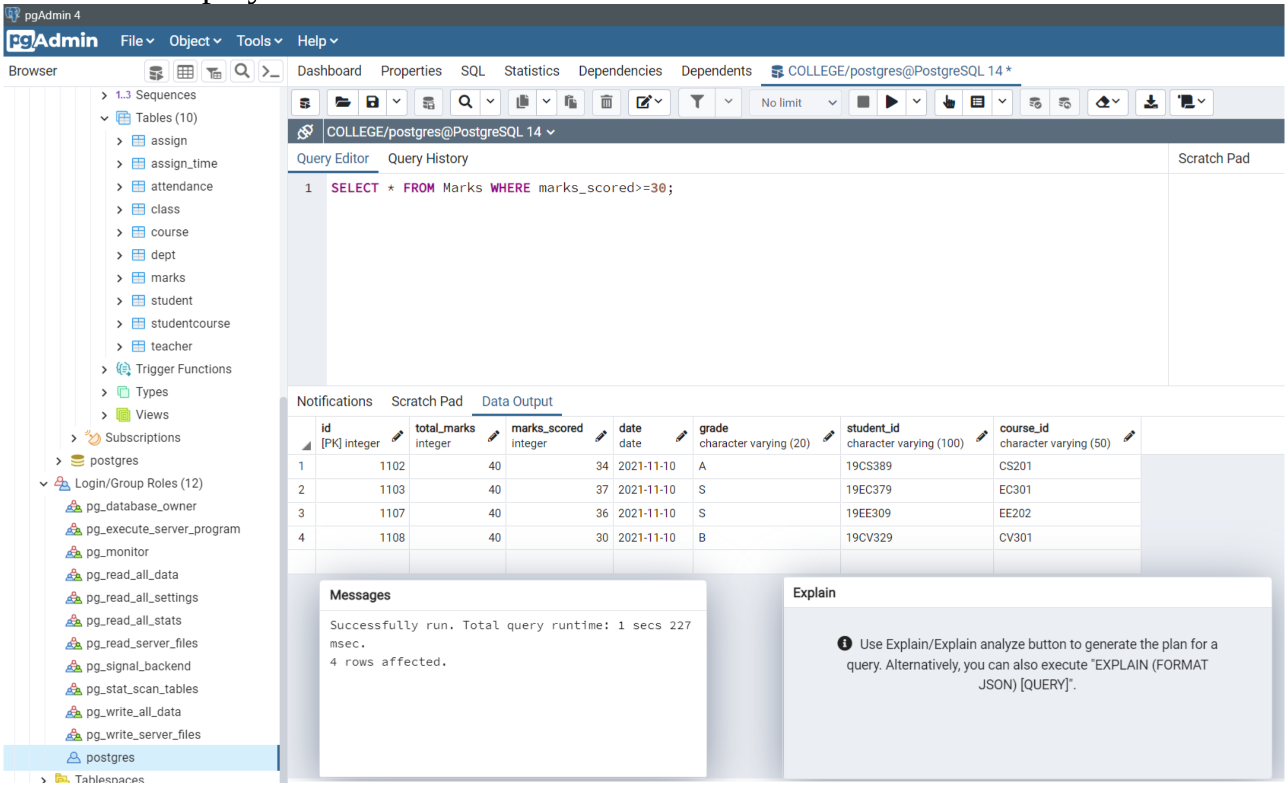
```
Successfully run. Total query runtime: 128 msec.
2 rows affected.
```

Explain

Use Explain/Explain analyze button to generate the plan for a query. Alternatively, you can also execute "EXPLAIN (FORMAT JSON) [QUERY]".

✓ Successfully run. Total query runtime: 128

6. Display all marks related details of students who scored more than 30/40.



```sql
SELECT * FROM Marks WHERE marks_scored>=30;
```

	id [PK] integer	total_marks integer	marks_scored integer	date date	grade character varying (20)	student_id character varying (100)	course_id character varying (50)
1	1102	40	34	2021-11-10	A	19CS389	CS201
2	1103	40	37	2021-11-10	S	19EC379	EC301
3	1107	40	36	2021-11-10	S	19EE309	EE202
4	1108	40	30	2021-11-10	B	19CV329	CV301

Messages

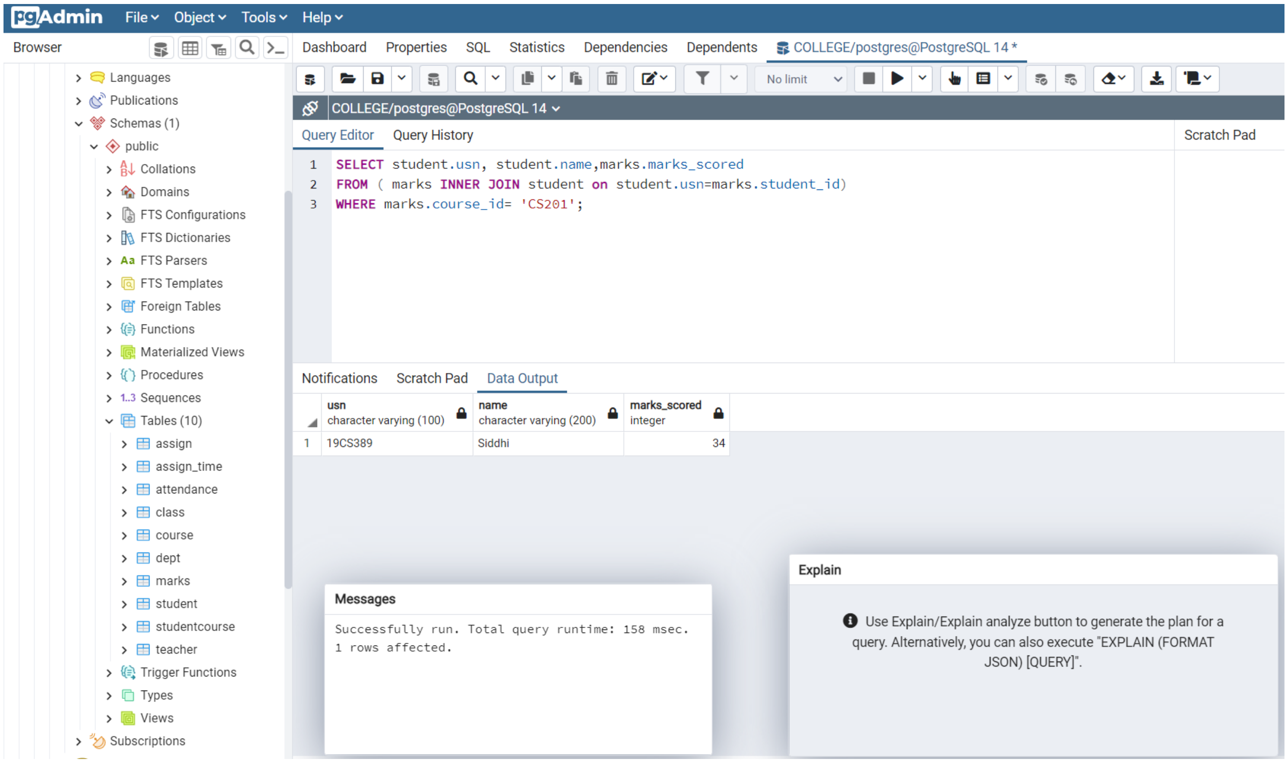
```
Successfully run. Total query runtime: 1 secs 227
msec.
4 rows affected.
```

Explain

Use Explain/Explain analyze button to generate the plan for a query. Alternatively, you can also execute "EXPLAIN (FORMAT JSON) [QUERY]".

Complex Queries:

7. Display USN, Name and marks of students enrolled in the course 'Data Structures And Algorithms'.



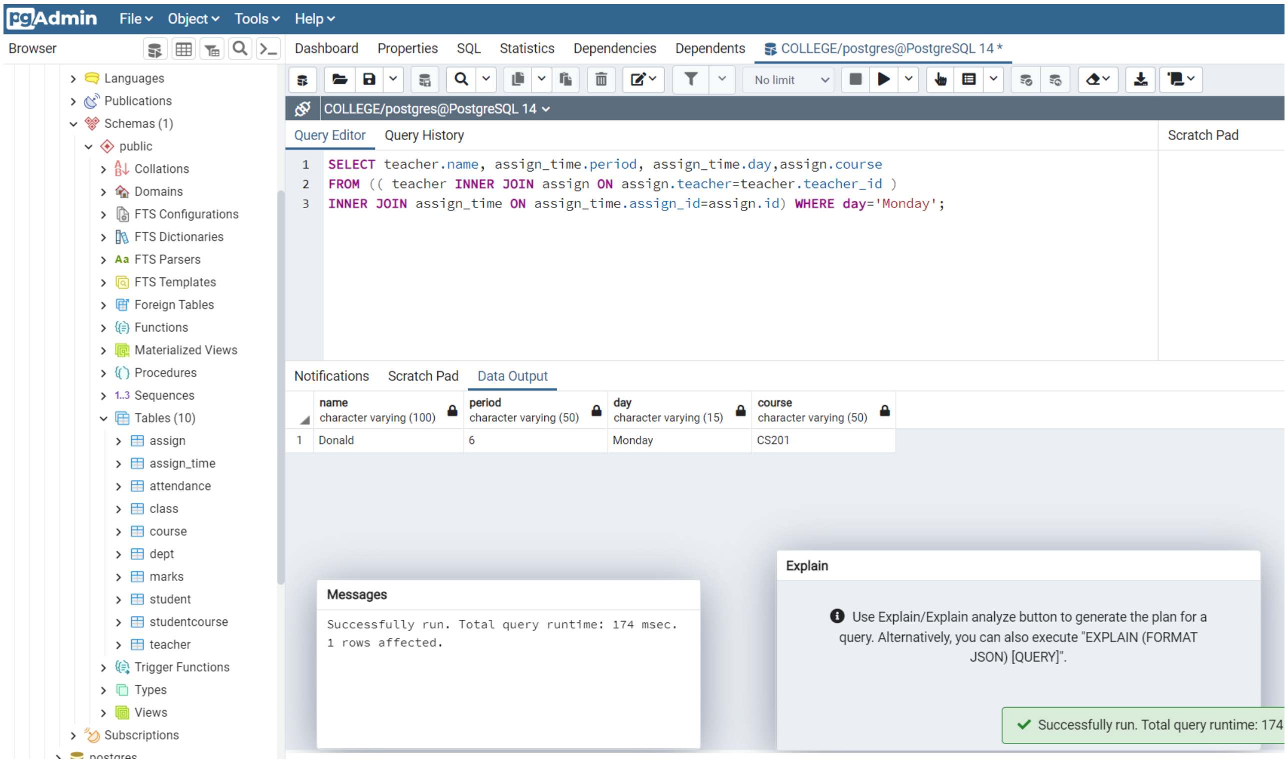
```sql
SELECT student.usn, student.name,marks.marks_scored
FROM ( marks INNER JOIN student on student.usn=marks.student_id)
WHERE marks.course_id= 'CS201';
```

	usn character varying (100)	name character varying (200)	marks_scored integer
1	19CS389	Siddhi	34

```
Successfully run. Total query runtime: 158 msec.
1 rows affected.
```

Use Explain/Explain analyze button to generate the plan for a query. Alternatively, you can also execute "EXPLAIN (FORMAT JSON) [QUERY]".

8. Display the name of the teachers taking classes on Monday's along with the period and course details.



```
SELECT teacher.name, assign_time.period, assign_time.day,assign.course
FROM (( teacher INNER JOIN assign ON assign.teacher=teacher.teacher_id )
INNER JOIN assign_time ON assign_time.assign_id=assign.id) WHERE day='Monday';
```

	name character varying (100)	period character varying (50)	day character varying (15)	course character varying (50)
1	Donald	6	Monday	CS201

Messages

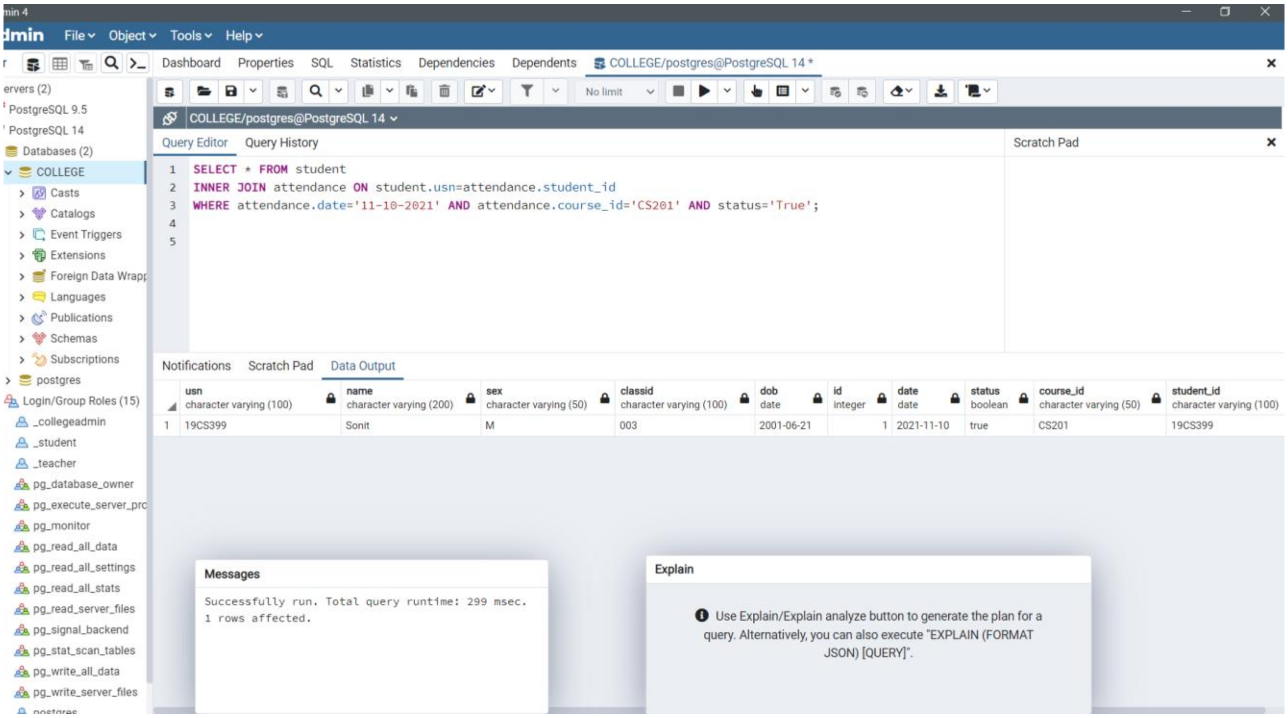
```
Successfully run. Total query runtime: 174 msec.
1 rows affected.
```

Explain

Use Explain/Explain analyze button to generate the plan for a query. Alternatively, you can also execute "EXPLAIN (FORMAT JSON) [QUERY]".

✓ Successfully run. Total query runtime: 174

9. Display names of students who attended CS201 on '11-10-2021'.



```sql
SELECT * FROM student
INNER JOIN attendance ON student.usn=attendance.student_id
WHERE attendance.date='11-10-2021' AND attendance.course_id='CS201' AND status='True';
```

	usn character varying (100)	name character varying (200)	sex character varying (50)	classid character varying (100)	dob date	id integer	date date	status boolean	course_id character varying (50)	student_id character varying (100)
1	19CS399	Sonit	M	003	2001-06-21	1	2021-11-10	true	CS201	19CS399

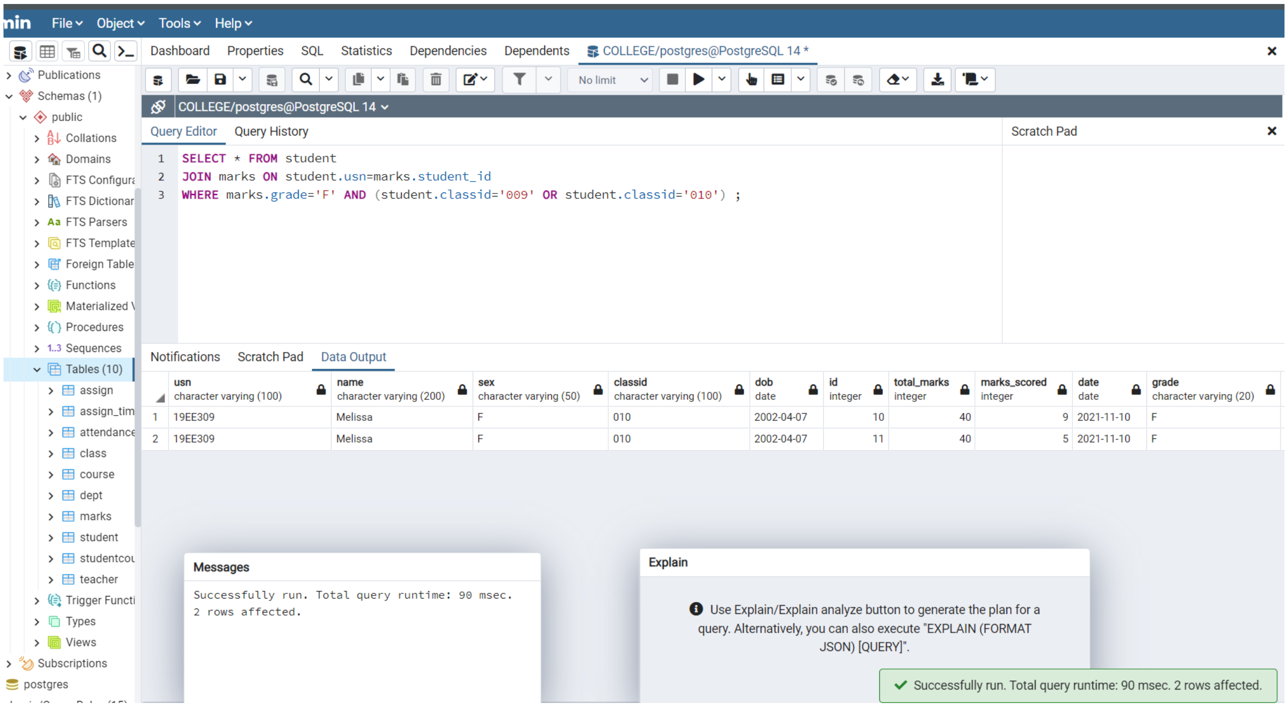
Messages

```
Successfully run. Total query runtime: 299 msec.
1 rows affected.
```

Explain

Use Explain/Explain analyze button to generate the plan for a query. Alternatively, you can also execute "EXPLAIN (FORMAT JSON) [QUERY]".

10. Display all details of students with failing grade in course Electrical department.



```
SELECT * FROM student
JOIN marks ON student.usn=marks.student_id
WHERE marks.grade='F' AND (student.classid='009' OR student.classid='010') ;
```

	usn character varying (100)	name character varying (200)	sex character varying (50)	classid character varying (100)	dob date	id integer	total_marks integer	marks_scored integer	date date	grade character varying (20)
1	19EE309	Melissa	F	010	2002-04-07	10	40	9	2021-11-10	F
2	19EE309	Melissa	F	010	2002-04-07	11	40	5	2021-11-10	F

Messages

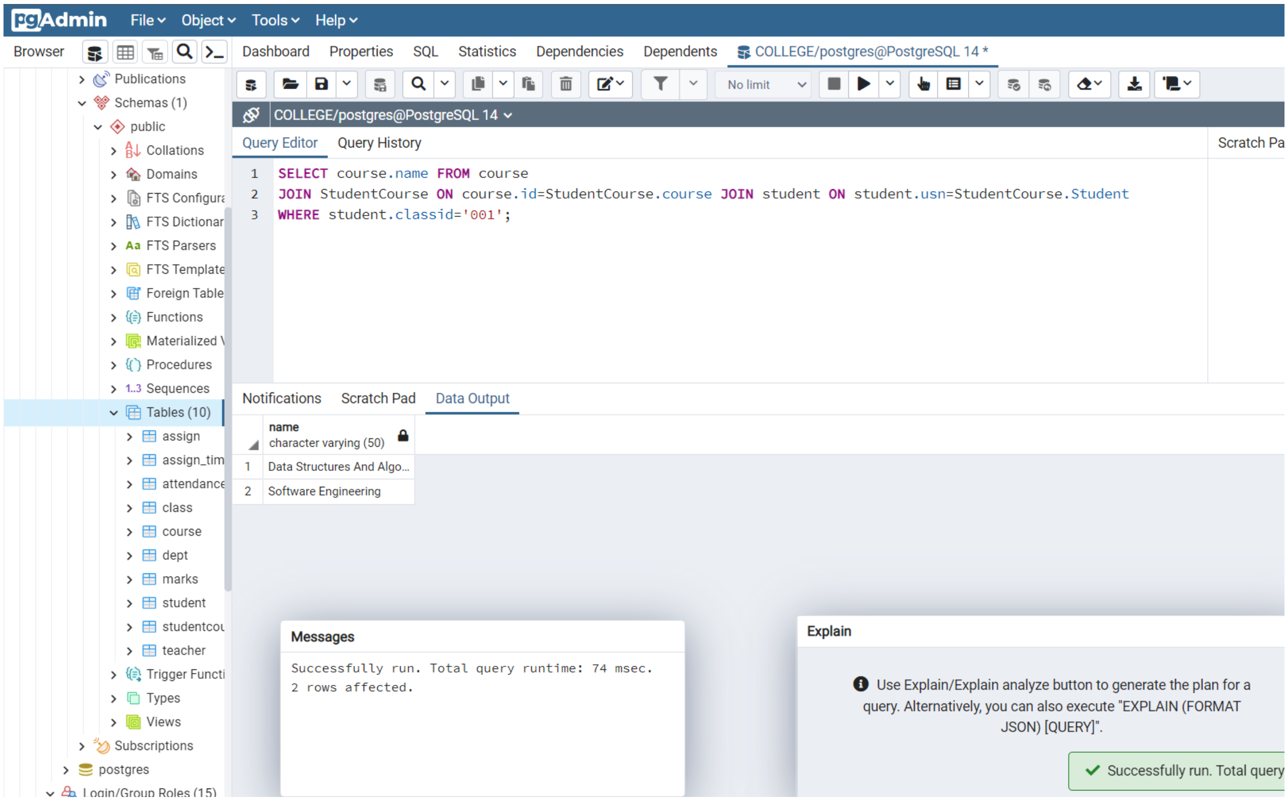
```
Successfully run. Total query runtime: 90 msec.
2 rows affected.
```

Explain

Use Explain/Explain analyze button to generate the plan for a query. Alternatively, you can also execute "EXPLAIN (FORMAT JSON) [QUERY]".

✓ Successfully run. Total query runtime: 90 msec. 2 rows affected.

11. Find all courses taken by students in section A.



```
SELECT course.name FROM course
JOIN StudentCourse ON course.id=StudentCourse.course JOIN student ON student.usn=StudentCourse.Student
WHERE student.classid='001';
```

	name character varying (50)
1	Data Structures And Algo...
2	Software Engineering

Messages

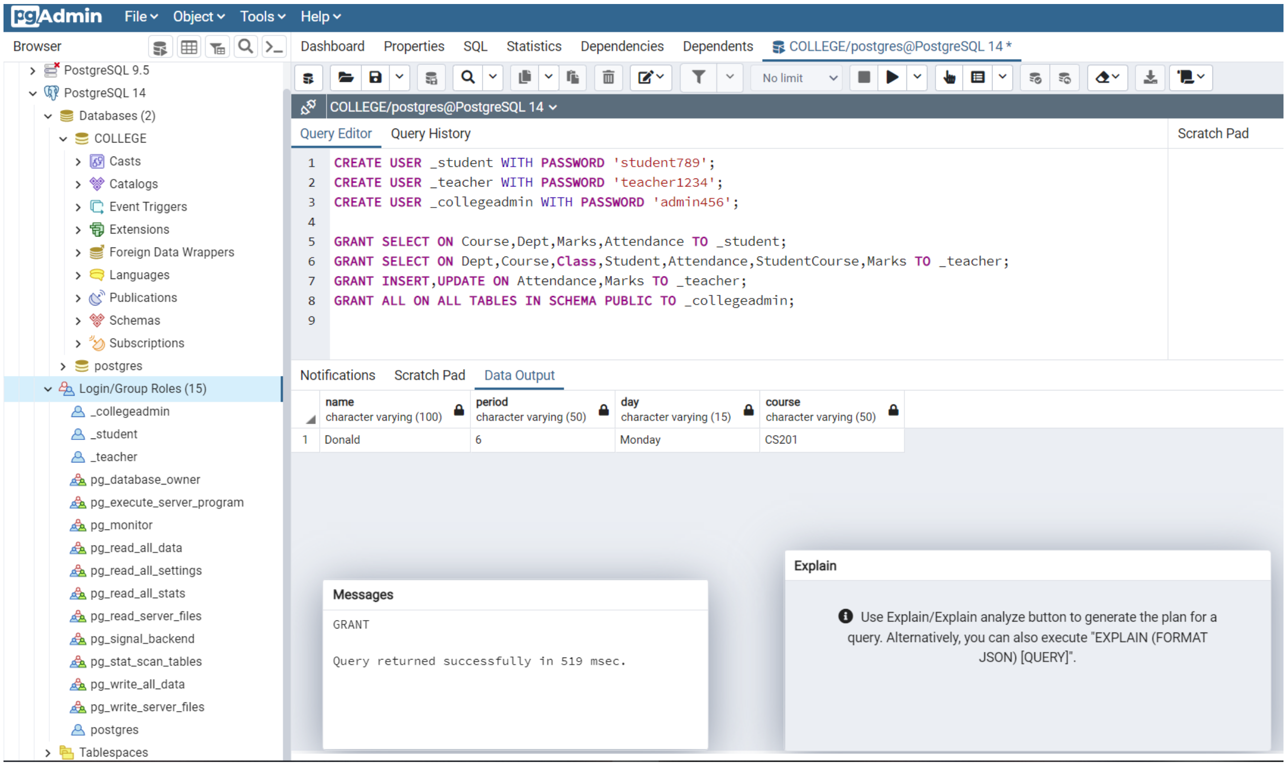
```
Successfully run. Total query runtime: 74 msec.
2 rows affected.
```

Explain

Use Explain/Explain analyze button to generate the plan for a query. Alternatively, you can also execute "EXPLAIN (FORMAT JSON) [QUERY]".

✓ Successfully run. Total query

Creating multiple users with different access privilege levels for different parts of the database.



```
CREATE USER _student WITH PASSWORD 'student789';
CREATE USER _teacher WITH PASSWORD 'teacher1234';
CREATE USER _collegeadmin WITH PASSWORD 'admin456';

GRANT SELECT ON Course,Dept,Marks,Attendance TO _student;
GRANT SELECT ON Dept,Course,Class,Student,Attendance,StudentCourse,Marks TO _teacher;
GRANT INSERT,UPDATE ON Attendance,Marks TO _teacher;
GRANT ALL ON ALL TABLES IN SCHEMA PUBLIC TO _collegeadmin;
```

	name character varying (100)	period character varying (50)	day character varying (15)	course character varying (50)
1	Donald	6	Monday	CS201

```
GRANT

Query returned successfully in 519 msec.
```

Use Explain/Explain analyze button to generate the plan for a query. Alternatively, you can also execute "EXPLAIN (FORMAT JSON) [QUERY]".

Contributions

Shatakshi Mohan- Simple queries and further population of the database.
Siddhi Patil- Complex and nested queries.
Sonit Pradhan- Complex queries and access privilege creation.

Time Spent- 6 hrs