

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’)

The screenshot displays the pgAdmin 4 web interface. On the left, the 'Browser' pane shows the database structure, with 'Tables (10)' expanded under the 'public' schema. The main 'Query Editor' pane contains the following SQL query:

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

Below the query editor, the 'Data Output' tab is active, showing a table with 6 rows of results:

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

At the bottom, a 'Messages' pane shows the execution status: 'Successfully run. Total query runtime: 281 msec. 6 rows affected.' To the right of the messages, an 'Explain' pane provides a tip: '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.

The screenshot shows the pgAdmin 4 web interface. On the left is a tree view of the database structure, including Languages, Publications, Schemas (1), and public. The main area is divided into several tabs: Query Editor, Query History, and Scratch Pad. The Query Editor tab is active, displaying a SQL query: `1 SELECT * FROM Student WHERE sex='F';`. Below the query editor, there are tabs for Notifications, Scratch Pad, and Data Output. The Data Output tab is active, showing a table with 5 rows of data. The table has columns: usn, name, sex, classid, and dob. The data is as follows:

usn	name	sex	classid	dob
19EC379	Shatakshi	F	001	2001-10-14
19CS389	Siddhi	F	002	2001-12-14
19ME349	Aziza	F	006	2000-08-09
19EE319	Zahia	F	009	2001-03-21
19EE309	Melissa	F	010	2002-05-23

Below the Data Output table, there are two pop-up windows. The first is titled "Messages" and contains the text: "Successfully run. Total query runtime: 166 msec. 5 rows affected." The second is titled "Explain" and contains the text: "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.

The screenshot shows the pgAdmin 4 web interface. The left sidebar displays the database structure, including the 'public' schema and a list of tables. The main pane is divided into three sections: 'Query Editor', 'Data Output', and 'Messages'.

Query Editor: The query being executed is:

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

Data Output: The results of the query are displayed in a table with the following columns: teacher_id, name, sex, dob, and deptid.

	teacher_id	name	sex	dob	deptid
1	359	Benjamin	M	1979-01-01	C001
2	349	Evelyn	F	1985-01-19	C001

Messages: A message box indicates that the query was successfully run, with a total query runtime of 204 msec and 2 rows affected.

Explain: A message box provides instructions on how to use the 'Explain/Explain analyze' button to generate the plan for a query. It also mentions that the query can be executed using 'EXPLAIN (FORMAT JSON) [QUERY]'.

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

The screenshot shows the pgAdmin 4 interface with the following components:

- Browser:** A tree view on the left showing the database structure. The 'public' schema is expanded, showing various objects like Collations, Domains, FTS Configurations, FTS Dictionaries, FTS Parsers, FTS Templates, Foreign Tables, Functions, Materialized Views, Procedures, Sequences, and Tables (10). The 'Tables' folder is expanded, showing tables like assign, assign_time, attendance, class, course, dept, marks, student, studentcourse, teacher, Trigger Functions, Types, Views, and Subscriptions.
- Query Editor:** The main window for writing SQL queries. The query is: `SELECT section, class_id FROM Class WHERE deptid='E001';`
- Data Output:** A table showing the results of the query. It has two columns: 'section' (character varying (100)) and 'class_id' ([FK] character varying (100)). The results are:

	section	class_id
1	I	009
2	J	010
- Messages:** A box at the bottom left showing the message: "Successfully run. Total query runtime: 150 msec. 2 rows affected."
- Explain:** A box at the bottom right showing the message: "Use Explain/Explain analyze button to generate the plan for a query. Alternatively, you can also execute 'EXPLAIN (FORMAT JSON) [QUERY]'."

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

The screenshot displays the pgAdmin 4 web interface. On the left, the 'Browser' pane shows the database structure, with 'postgres' selected. The main area is divided into several sections:

- Query Editor:** Contains the SQL query: `1 SELECT * FROM Marks WHERE grade='S';`
- Data Output:** A table showing the results of the query. It has 9 columns: `id` (integer), `total_marks` (integer), `marks_scored` (integer), `date` (date), `grade` (character varying (20)), `student_id` (character varying (100)), and `course_id` (character varying (50)). The table contains 2 rows of data.
- Messages:** A box showing the execution status: "Successfully run. Total query runtime: 128 msec. 2 rows affected."
- Explain:** A box with a message: "Use Explain/Explain analyze button to generate the plan for a query. Alternatively, you can also execute 'EXPLAIN (FORMAT JSON) [QUERY]'."

id	total_marks	marks_scored	date	grade	student_id	course_id
1	1103	40	2021-11-10	S	19EC379	EC301
2	1107	40	2021-11-10	S	19EE309	EE202

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

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 msec. 2 rows affected.

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

pgAdmin 4

File Object Tools Help

COLLEGE/postgres@PostgreSQL 14 *

COLLEGE/postgres@PostgreSQL 14

Query Editor Query History Scratch Pad

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

Notifications Scratch Pad Data Output

	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	40	2021-11-10	A	19CS389	CS201
2	1103	40	40	2021-11-10	S	19EC379	EC301
3	1107	40	40	2021-11-10	S	19EE309	EE202
4	1108	40	40	2021-11-10	B	19CV329	CV301

Messages

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

Explain

1 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'.

The screenshot displays the pgAdmin 4 web interface. On the left, the 'Browser' pane shows a tree view of the database structure, including 'Schemas (1)' and 'Tables (10)'. The main area is divided into three panes: 'Query Editor', 'Query History', and 'Scratch Pad'. The 'Query Editor' contains the following SQL query:

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

Below the query editor, the 'Data Output' pane shows the results of the query in a table format:

usn	name	marks_scored
19CS389	Siddhi	34

At the bottom, there are two panes: 'Messages' and 'Explain'. The 'Messages' pane shows a success message: 'Successfully run. Total query runtime: 158 msec. 1 rows affected.' The 'Explain' pane contains a message: '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.

The screenshot shows the PgAdmin interface with a SQL query executed in the Query Editor. The query is:

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

The Data Output tab shows the following results:

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

The Messages pane shows:

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

The Explain pane shows:

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

A green status bar at the bottom right indicates: **✓ Successfully run. Total query runtime: 174**

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

The screenshot shows the pgAdmin 4 web interface. The left sidebar displays the database structure, including the 'COLLEGE' database and its tables. The main pane shows a SQL query in the Query Editor:

```
1 SELECT * FROM student
2 INNER JOIN attendance ON student.usn=attendance.student_id
3 WHERE attendance.date='11-10-2021' AND attendance.course_id='CS201' AND status='True';
4
5
```

The Data Output tab shows the results of the query as a table with 10 columns: usn, name, sex, classid, dob, id, date, status, course_id, and student_id. The results show one row of data for student 19CS399, Sonit, who attended CS201 on 2021-11-10.

usn	name	sex	classid	dob	id	date	status	course_id	student_id
19CS399	Sonit	M	003	2001-06-21	1	2021-11-10	true	CS201	19CS399

At the bottom, the Messages pane shows a success message: "Successfully run. Total query runtime: 299 msec. 1 rows affected." The Explain pane is also visible, showing a message about using the Explain/Explain analyze button to generate the plan for a query.

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

The screenshot shows a database management tool interface. The top menu bar includes File, Object, Tools, and Help. The main toolbar contains various icons for database operations. The left sidebar shows a tree view of the database schema, with 'Tables (10)' expanded. The central pane displays a SQL query in the 'Query Editor' tab:

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

Below the query editor, the 'Data Output' tab shows the results of the query. The results are displayed in a table with the following columns: usn, name, sex, classid, dob, id, total_marks, marks_scored, date, and grade. The table contains two rows of data:

usn	name	sex	classid	dob	id	total_marks	marks_scored	date	grade
19EE309	Melissa	F	010	2002-04-07	10	40	40	2021-11-10	F
19EE309	Melissa	F	010	2002-04-07	11	40	40	2021-11-10	F

At the bottom of the interface, there are two panels: 'Messages' and 'Explain'. The 'Messages' panel shows a success message: 'Successfully run. Total query runtime: 90 msec. 2 rows affected.' The 'Explain' panel shows a message: 'Use Explain/Explain analyze button to generate the plan for a query. Alternatively, you can also execute "EXPLAIN (FORMAT JSON) [QUERY]".' A green status bar at the bottom right indicates: 'Successfully run. Total query runtime: 90 msec. 2 rows affected.'

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

The screenshot shows the pgAdmin 4 web interface. The left sidebar displays the database structure, with 'Tables (10)' expanded under the 'public' schema. The main pane shows the 'Query Editor' with the following SQL query:

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

The 'Data Output' tab shows the results of the query:

name
Data Structures And Algo...
Software Engineering

Below the query editor, there are two panels: 'Messages' and 'Explain'.

Messages:

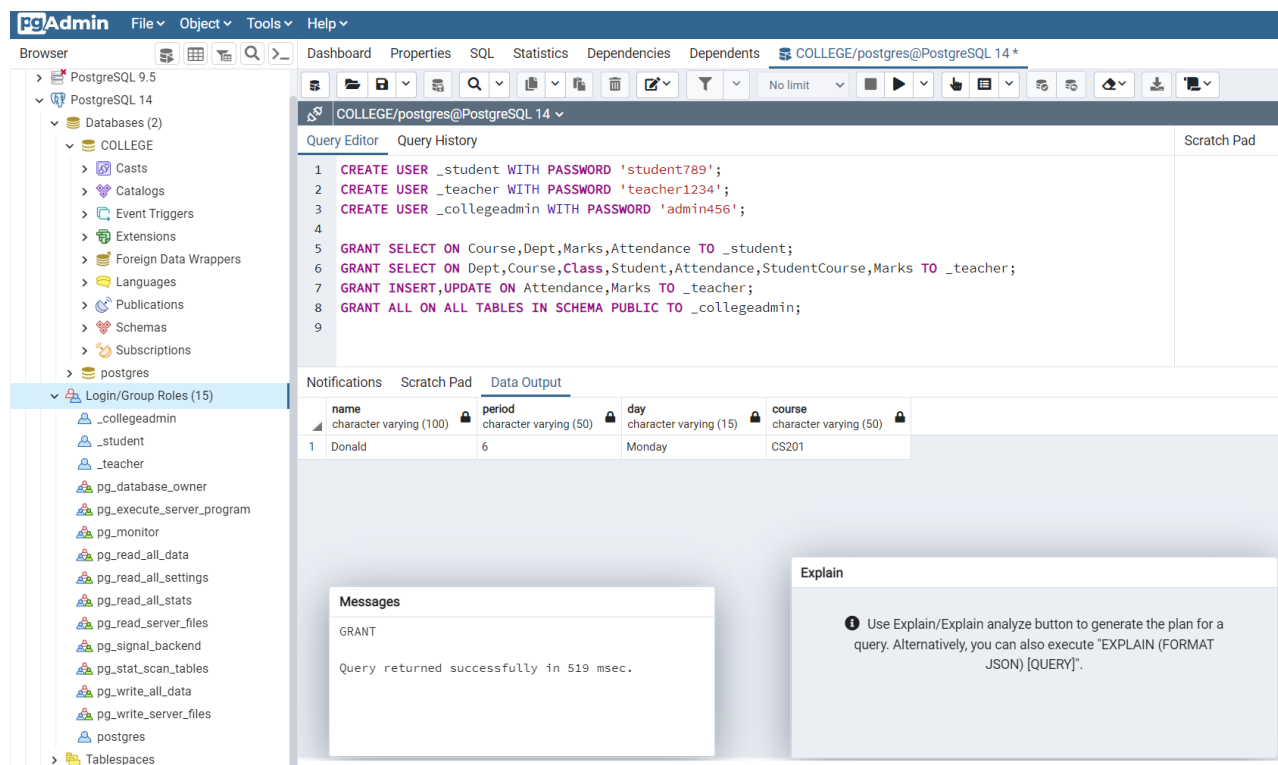
```
Successfully run. Total query runtime: 74 msec.
2 rows affected.
```

Explain:

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

At the bottom right, a green status bar indicates: "Successfully run. Total query runtime: 74 msec. 2 rows affected."

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



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

