

## College Fest Database

### Group 15:

Aditya Kumar (202212046)

Saheb Singh (202212071)

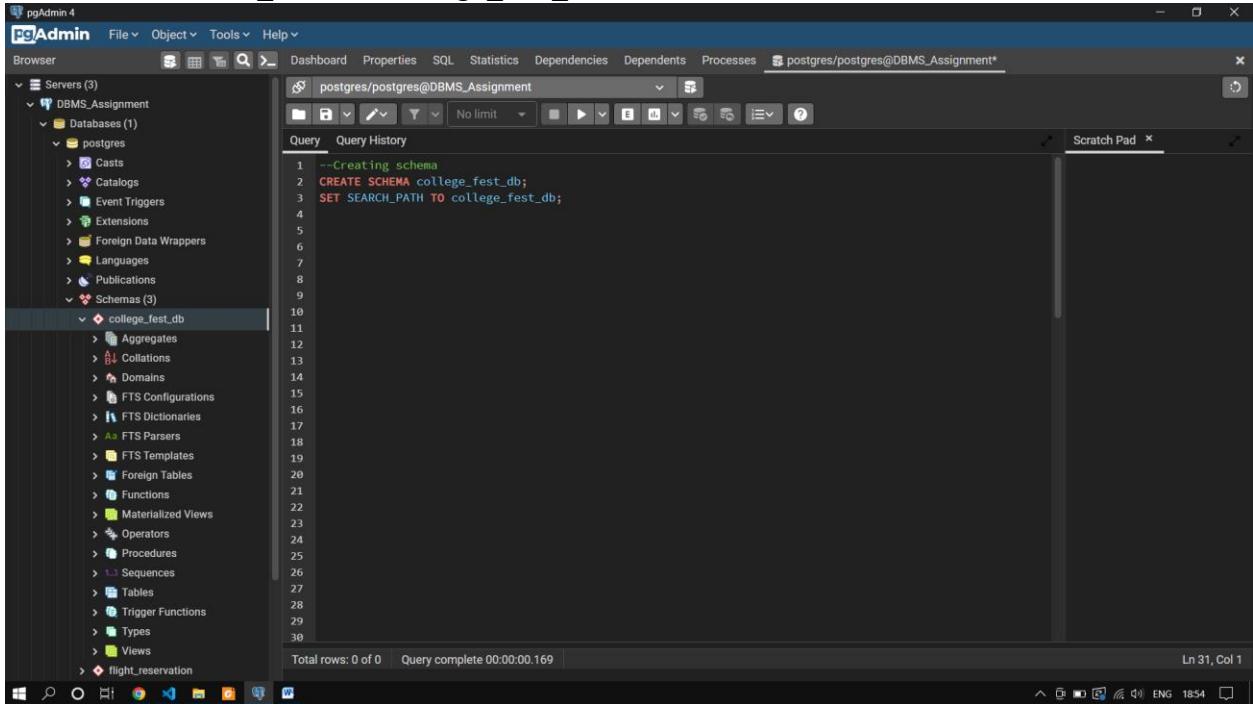
---

### Creating Schema and Tables

#### 1. --Creating Schema:

Query:

```
CREATE SCHEMA college_fest_db;  
SET SEARCH_PATH TO college_fest_db;
```



The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure under 'DBMS\_Assignment'. The 'Schemas' section contains a single entry: 'college\_fest\_db'. The main pane shows a query editor with the following SQL code:

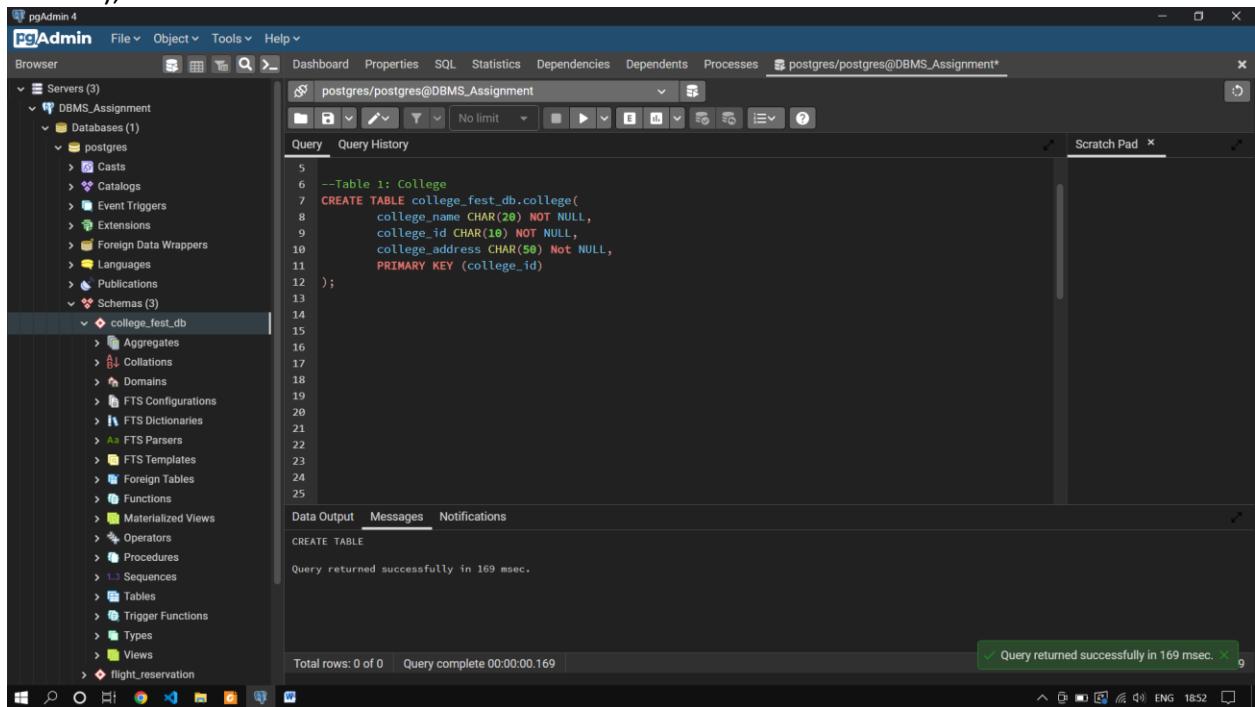
```
1 --Creating schema  
2 CREATE SCHEMA college_fest_db;  
3 SET SEARCH_PATH TO college_fest_db;  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30
```

The status bar at the bottom indicates 'Query complete 00:00:00.169' and 'Ln 31, Col 1'.

## 2. --Table 1: College

Query:

```
CREATE TABLE college_fest_db.college(
    college_name CHAR(20) NOT NULL,
    college_id CHAR(10) NOT NULL,
    college_address CHAR(50) Not NULL,
    PRIMARY KEY (college_id)
);
```



The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure under 'DBMS\_Assignment'. The 'college\_fest\_db' schema is selected. The main window shows the SQL tab with the following code:

```
5
6 --Table 1: College
7 CREATE TABLE college_fest_db.college(
8     college_name CHAR(20) NOT NULL,
9     college_id CHAR(10) NOT NULL,
10    college_address CHAR(50) Not NULL,
11    PRIMARY KEY (college_id)
12 );
13
14
15
16
17
18
19
20
21
22
23
24
25
```

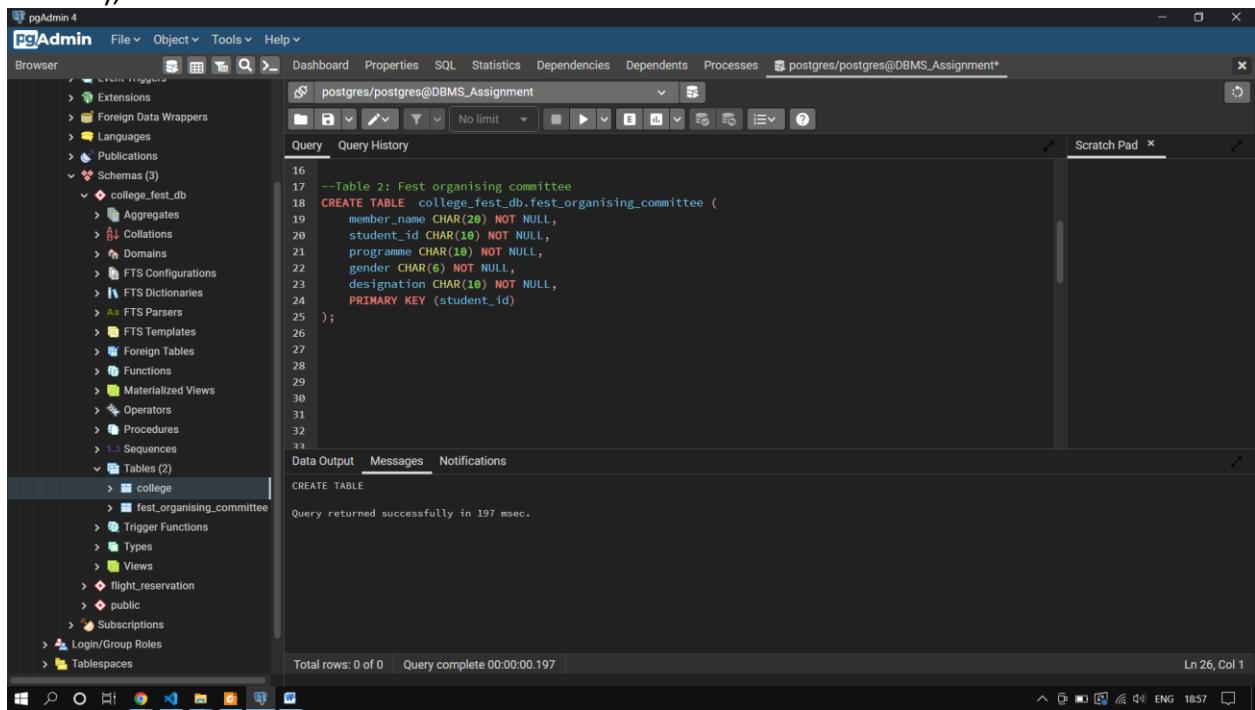
Below the code, the 'Messages' tab shows the message: 'Query returned successfully in 169 msec.' The status bar at the bottom right indicates 'Query returned successfully in 169 msec.' and shows system status like 'ENG 1852'.

### 3. Table 2: Fest organising committee

Query:

--Table 2: Fest organising committee

```
CREATE TABLE college_fest_db.fest_organising_committee (
    member_name CHAR(20) NOT NULL,
    student_id CHAR(10) NOT NULL,
    programme CHAR(10) NOT NULL,
    gender CHAR(6) NOT NULL,
    designation CHAR(10) NOT NULL,
    PRIMARY KEY (student_id)
);
```



The screenshot shows the pgAdmin 4 interface. The left sidebar (Browser) displays the database structure, including the 'fest\_organising\_committee' table under the 'Tables (2)' section of the 'college\_fest\_db' schema. The main area (Query) shows the SQL code for creating the table. The code is as follows:

```
16 --Table 2: Fest organising committee
17 CREATE TABLE college_fest_db.fest_organising_committee (
18     member_name CHAR(20) NOT NULL,
19     student_id CHAR(10) NOT NULL,
20     programme CHAR(10) NOT NULL,
21     gender CHAR(6) NOT NULL,
22     designation CHAR(10) NOT NULL,
23     PRIMARY KEY (student_id)
24 );
25 ;
26
27
28
29
30
31
32
33
```

The 'Data Output' tab shows the successful creation of the table:

```
CREATE TABLE
Query returned successfully in 197 msec.
```

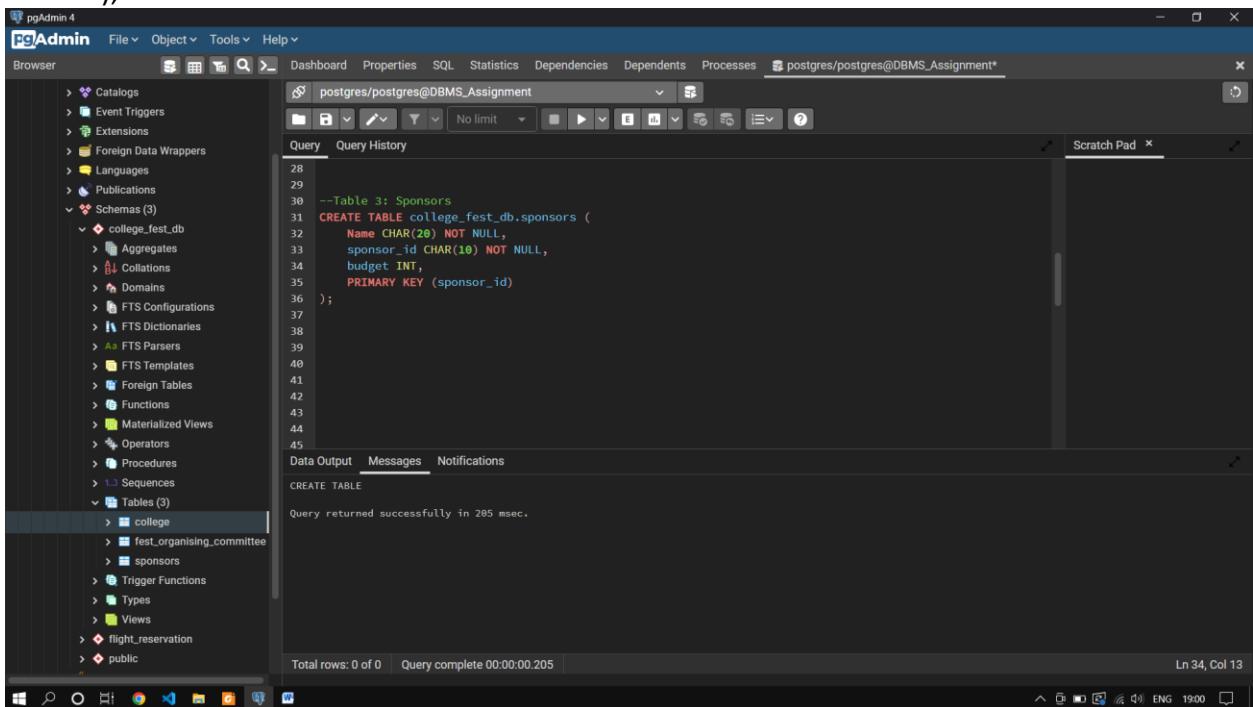
The status bar at the bottom right indicates 'Ln 26, Col 1'.

#### 4. Table 3: Sponsors

Query:

--Table 3: Sponsors

```
CREATE TABLE college_fest_db.sponsors (
    Name CHAR(20) NOT NULL,
    sponsor_id CHAR(10) NOT NULL,
    budget INT,
    PRIMARY KEY (sponsor_id)
);
```



The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure under the 'Browser' tab, including Catalogs, Event Triggers, Extensions, Foreign Data Wrappers, Languages, Publications, Schemas (3), college\_fest\_db (with Aggregates, Collations, Domains, FTS Configurations, FTS Dictionaries, FTS Parsers, FTS Templates, Foreign Tables, Functions, Materialized Views, Operators, Procedures, Sequences, Tables (3) containing college, fest\_organising\_committee, and sponsors), Trigger Functions, Types, Views, flight\_reservation, and public. The main window shows the SQL tab with the following code:

```
28
29
30 --Table 3: Sponsors
31 CREATE TABLE college_fest_db.sponsors (
32     Name CHAR(20) NOT NULL,
33     sponsor_id CHAR(10) NOT NULL,
34     budget INT,
35     PRIMARY KEY (sponsor_id)
36 );
37
38
39
40
41
42
43
44
45
```

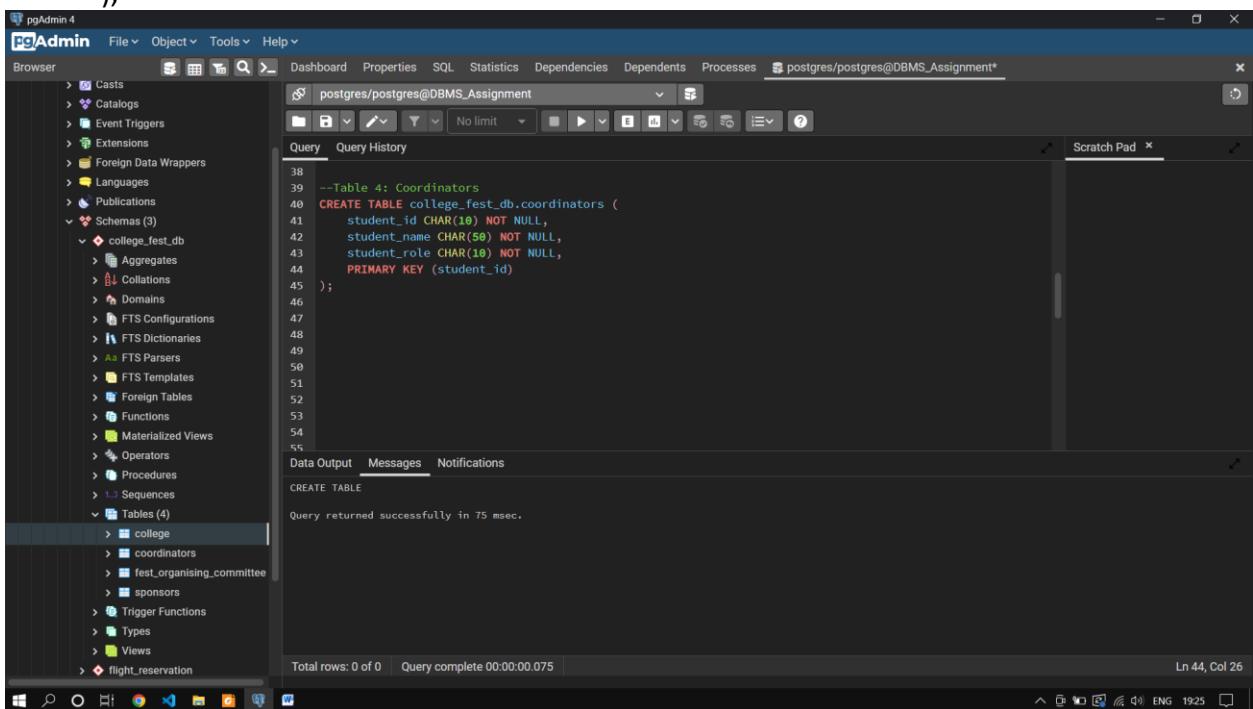
The 'Messages' tab below the SQL tab shows the message: 'Query returned successfully in 285 msec.' The status bar at the bottom indicates 'Total rows: 0 of 0' and 'Query complete 00:00:00.205'. The bottom right corner shows 'Ln 34, Col 13'.

## 5. Table: Coordinators

Query:

--Table 4: Coordinators

```
CREATE TABLE college_fest_db.coordinators (
    student_id CHAR(10) NOT NULL,
    student_name CHAR(50) NOT NULL,
    student_role CHAR(10) NOT NULL,
    PRIMARY KEY (student_id)
);
```

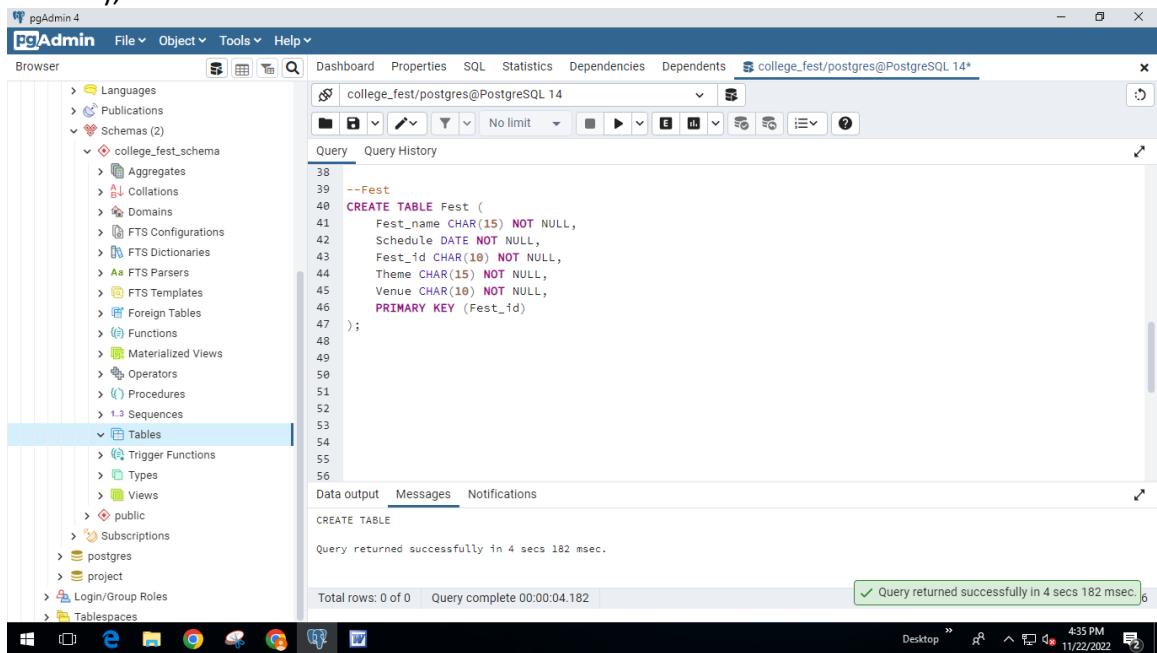


## 6. Table: Fest

Query:

--Table 5: Fest

```
CREATE TABLE college_fest_db.fest (
    fest_name CHAR(15) NOT NULL,
    schedule DATE NOT NULL,
    fest_id CHAR(10) NOT NULL,
    theme CHAR(15) NOT NULL,
    venue CHAR(10) NOT NULL,
    PRIMARY KEY (fest_id)
);
```

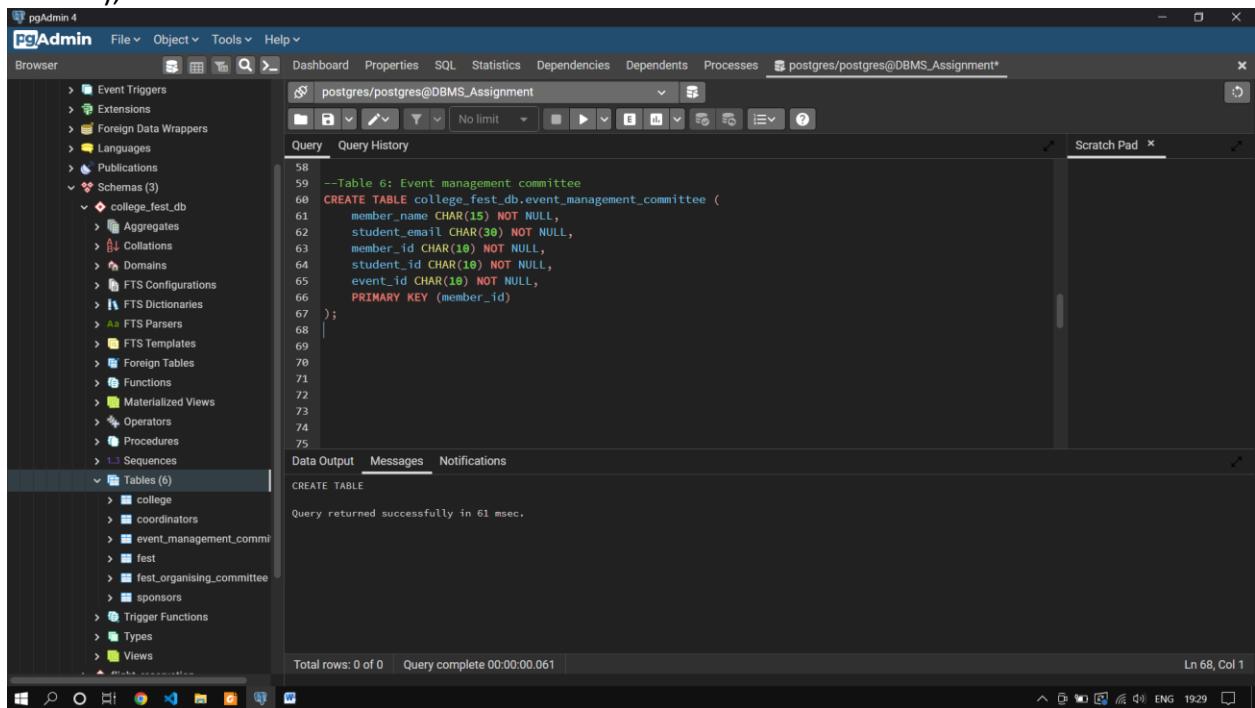


## 7. Table: Event management committee

Query:

--Table 6: Event management committee

```
CREATE TABLE college_fest_db.event_management_committee (
    member_name CHAR(15) NOT NULL,
    student_email CHAR(30) NOT NULL,
    member_id CHAR(10) NOT NULL,
    student_id CHAR(10) NOT NULL,
    event_id CHAR(10) NOT NULL,
    PRIMARY KEY (member_id)
);
```



The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure under the 'college\_fest\_db' schema, including tables like 'college', 'coordinators', 'event\_management\_committee', 'fest', 'fest\_organising\_committee', 'sponsors', and 'Trigger Functions'. The main window shows the SQL query being run in the 'Query' tab:

```
58 --Table 6: Event management committee
59 CREATE TABLE college_fest_db.event_management_committee (
60     member_name CHAR(15) NOT NULL,
61     student_email CHAR(30) NOT NULL,
62     member_id CHAR(10) NOT NULL,
63     student_id CHAR(10) NOT NULL,
64     event_id CHAR(10) NOT NULL,
65     PRIMARY KEY (member_id)
66 );
67 ;
68
69
70
71
72
73
74
75
```

The 'Data Output' tab shows the successful creation of the table:

```
CREATE TABLE
```

Query returned successfully in 61 msec.

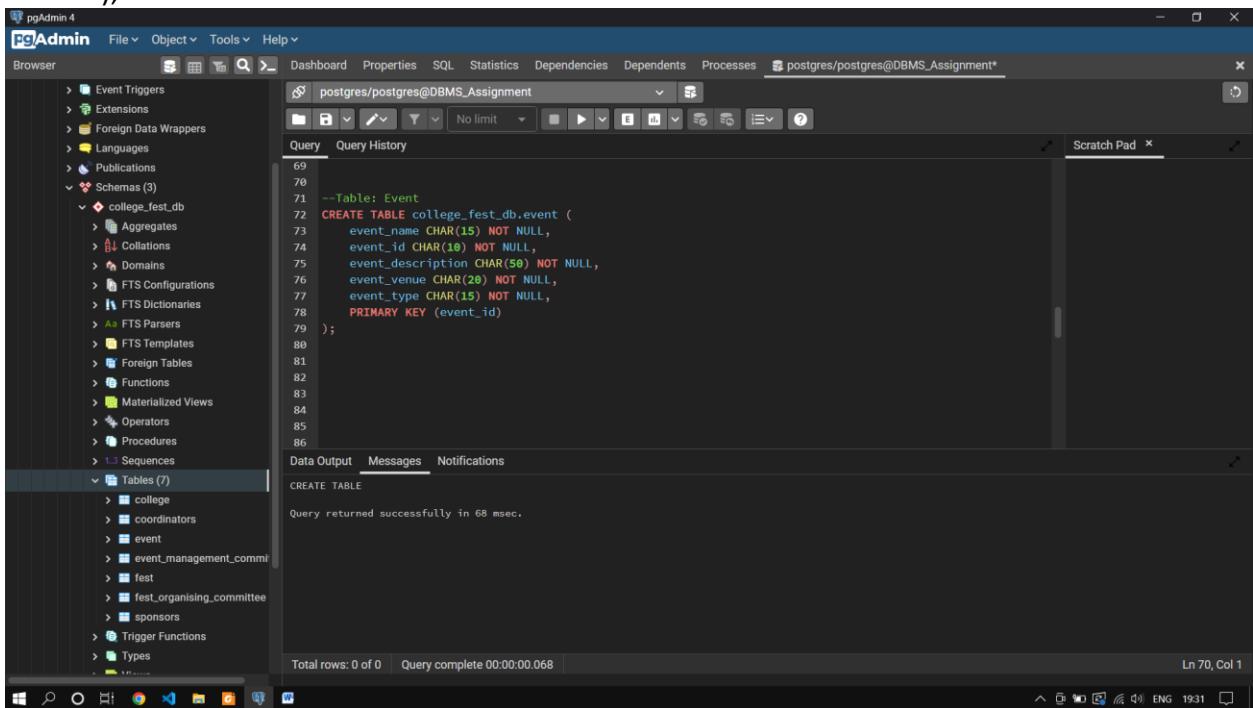
Total rows: 0 of 0 | Query complete 00:00:00.061 | Ln 68, Col 1

## 8. Table: Event

Query:

--Table: Event

```
CREATE TABLE college_fest_db.event (
    event_name CHAR(15) NOT NULL,
    event_id CHAR(10) NOT NULL,
    event_description CHAR(50) NOT NULL,
    event_venue CHAR(20) NOT NULL,
    event_type CHAR(15) NOT NULL,
    PRIMARY KEY (event_id)
);
```



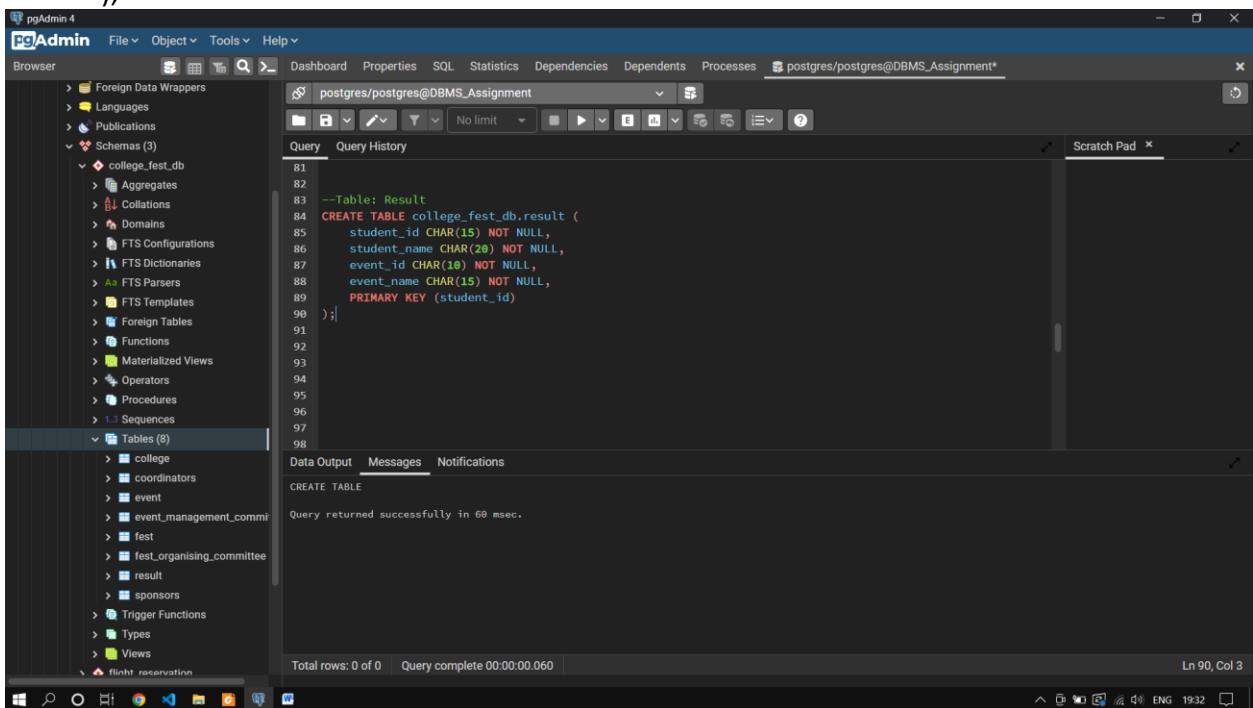
The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure with the 'Tables (7)' section selected, showing tables like 'college', 'coordinators', 'event', etc. The main window shows the SQL query being run in the 'Query' tab of the central pane. The query is the same as the one provided above. The 'Data Output' tab shows the result of the query: 'CREATE TABLE'. Below the output, a message states 'Query returned successfully in 68 msec.' The bottom status bar indicates 'Total rows: 0 of 0' and 'Query complete 00:00:00.068'. The bottom right corner shows the line and column numbers 'Ln 70, Col 1'.

## 9. Table: Result

Query:

--Table: Result

```
CREATE TABLE college_fest_db.result (
    student_id CHAR(15) NOT NULL,
    student_name CHAR(20) NOT NULL,
    event_id CHAR(10) NOT NULL,
    event_name CHAR(15) NOT NULL,
    PRIMARY KEY (student_id)
);
```



The screenshot shows the pgAdmin 4 interface. The left sidebar (Browser) displays the database structure, including the 'college\_fest\_db' schema which contains tables like 'college', 'coordinators', 'event', etc. The main window shows a query editor with the SQL code for creating the 'result' table. The code is as follows:

```
81
82
83 --Table: Result
84 CREATE TABLE college_fest_db.result (
85     student_id CHAR(15) NOT NULL,
86     student_name CHAR(20) NOT NULL,
87     event_id CHAR(10) NOT NULL,
88     event_name CHAR(15) NOT NULL,
89     PRIMARY KEY (student_id)
90 );
91
92
93
94
95
96
97
98
```

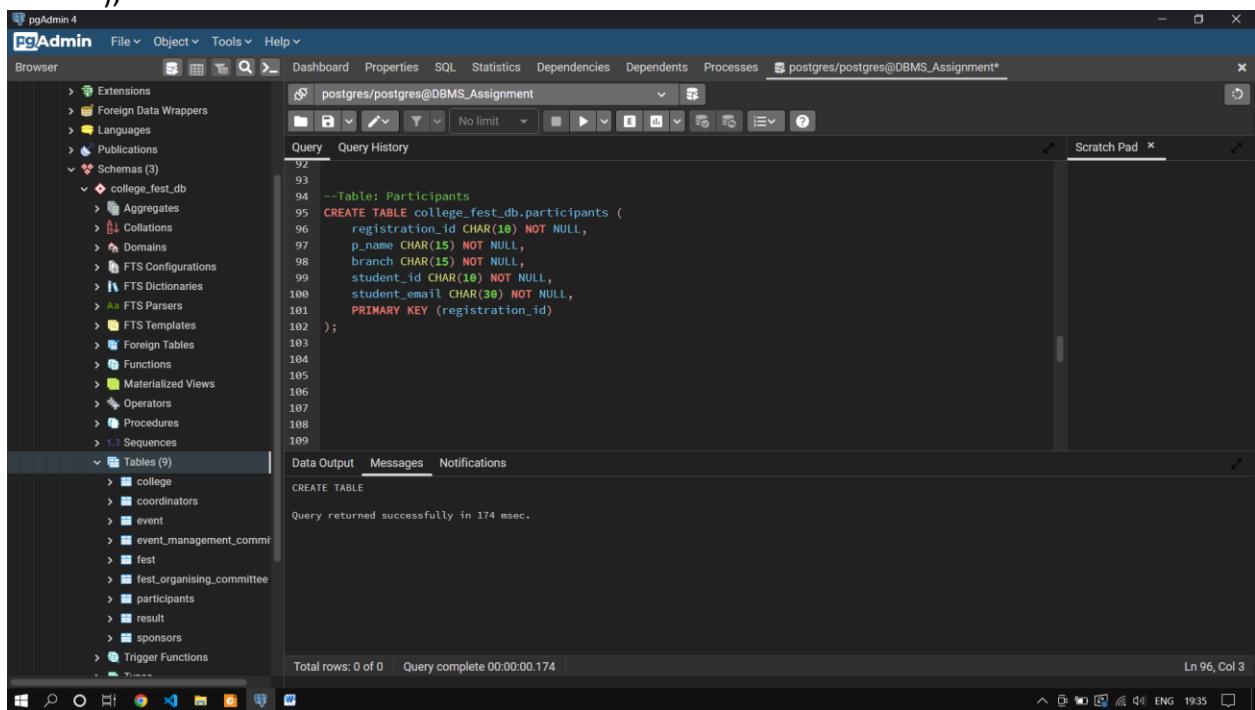
The 'Messages' tab below the query editor shows the message: "Query returned successfully in 60 msec." The status bar at the bottom right indicates "Total rows: 0 of 0" and "Query complete 00:00:00.060".

## 10. Table: Participants

Query:

--Table: Participants

```
CREATE TABLE college_fest_db.participants (
    registration_id CHAR(10) NOT NULL,
    p_name CHAR(15) NOT NULL,
    branch CHAR(15) NOT NULL,
    student_id CHAR(10) NOT NULL,
    student_email CHAR(30) NOT NULL,
    PRIMARY KEY (registration_id)
);
```



The screenshot shows the pgAdmin 4 interface. The left sidebar (Browser) displays the database structure, including the 'college\_fest\_db' schema which contains nine tables: college, coordinators, event, event\_management\_committee, fest, fest\_organising\_committee, participants, result, and sponsors. The 'Tables (9)' section is currently selected. The main window shows the SQL query being run in the 'Query' tab:

```
92
93
94 --Table: Participants
95 CREATE TABLE college_fest_db.participants (
96     registration_id CHAR(10) NOT NULL,
97     p_name CHAR(15) NOT NULL,
98     branch CHAR(15) NOT NULL,
99     student_id CHAR(10) NOT NULL,
100    student_email CHAR(30) NOT NULL,
101    PRIMARY KEY (registration_id)
102 );
103
104
105
106
107
108
109
```

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

```
CREATE TABLE
```

Query returned successfully in 174 msec.

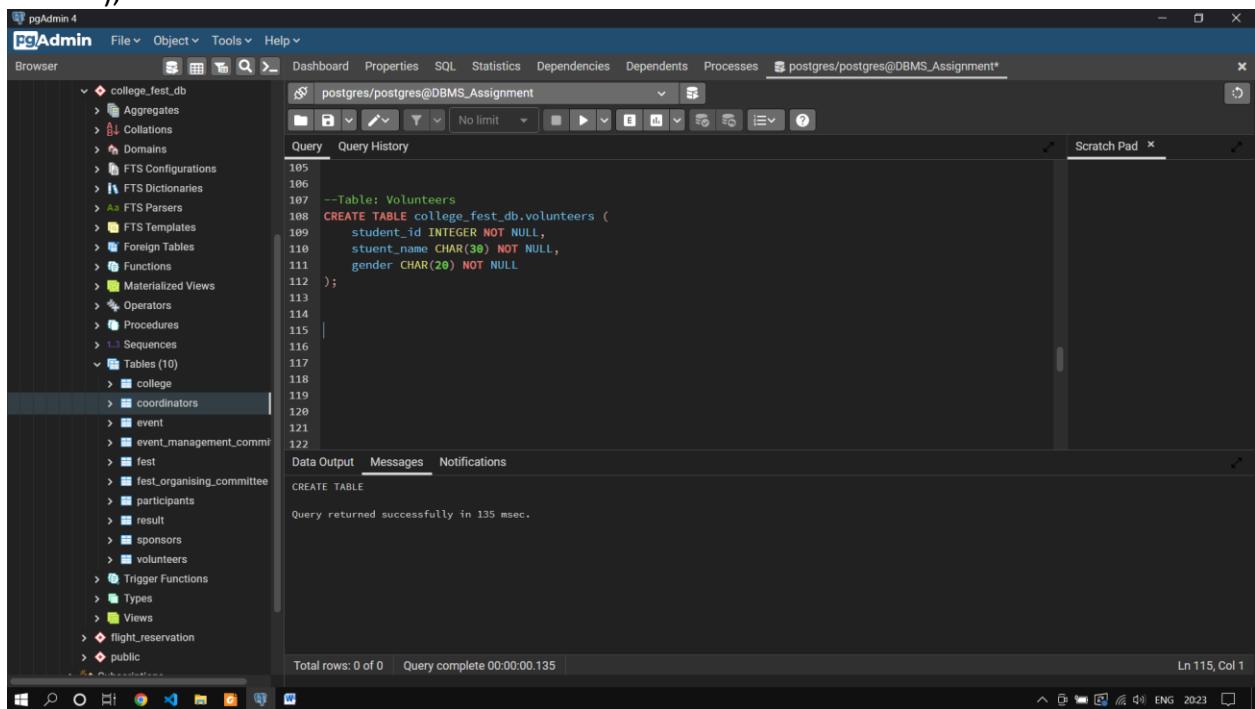
Total rows: 0 of 0 | Query complete 00:00:00.174 | Ln 96, Col 3

## 11. Table: Volunteers

Query:

--Table: Volunteers

```
CREATE TABLE college_fest_db.volunteers (
    student_id INTEGER NOT NULL,
    student_name CHAR(30) NOT NULL,
    gender CHAR(20) NOT NULL
);
```



The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure for 'college\_fest\_db', including tables like 'college', 'coordinators', 'event', 'event\_management\_committee', 'fest', 'fest\_organising\_committee', 'participants', 'result', 'sponsors', and 'volunteers'. The 'volunteers' table is currently selected. The main window shows the SQL query being run:

```
105
106
107 --Table: Volunteers
108 CREATE TABLE college_fest_db.volunteers (
109     student_id INTEGER NOT NULL,
110     student_name CHAR(30) NOT NULL,
111     gender CHAR(20) NOT NULL
112 );
113
114
115
116
117
118
119
120
121
122
```

The 'Messages' tab shows the result of the query:

```
CREATE TABLE
Query returned successfully in 135 msec.
```

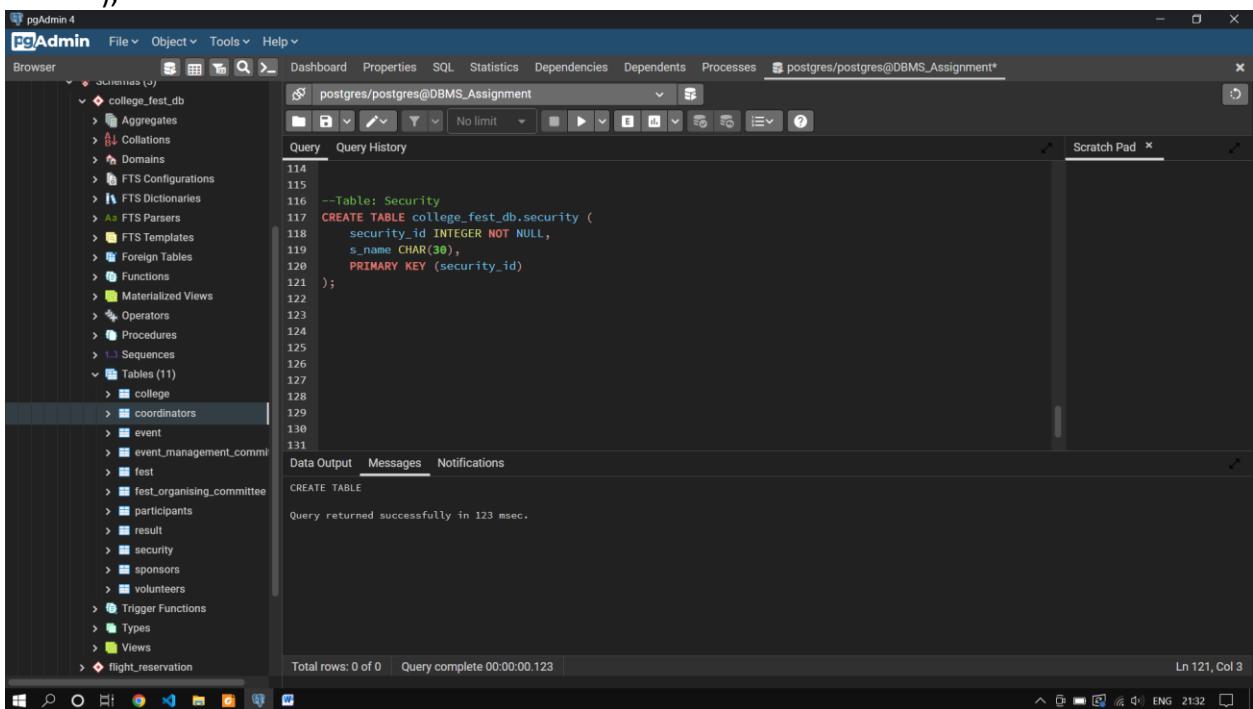
At the bottom, the status bar indicates 'Total rows: 0 of 0' and 'Query complete 00:00:00.135'. The bottom right corner shows the date and time as 'Ln 115, Col 1' and 'ENG 2023'.

## 12. Security

Query:

--Table: Security

```
CREATE TABLE college_fest_db.security (
    security_id INTEGER NOT NULL,
    s_name CHAR(30),
    PRIMARY KEY (security_id)
);
```



The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure under 'college\_fest\_db', including Schemas, Aggregates, Collations, Domains, FTS Configurations, FTS Dictionaries, FTS Parsers, FTS Templates, Foreign Tables, Functions, Materialized Views, Operators, Procedures, Sequences, and Tables (11). The 'Tables (11)' section is expanded, showing 'college', 'coordinators', 'event', 'event\_management\_committee', 'fest', 'fest\_organising\_committee', 'participants', 'result', 'security', 'sponsors', 'volunteers', 'Trigger Functions', 'Types', and 'Views'. The 'fest' table is currently selected. The main pane shows the SQL query being run:

```
114
115
116 --Table: Security
117 CREATE TABLE college_fest_db.security (
118     security_id INTEGER NOT NULL,
119     s_name CHAR(30),
120     PRIMARY KEY (security_id)
121 );
122
123
124
125
126
127
128
129
130
131
```

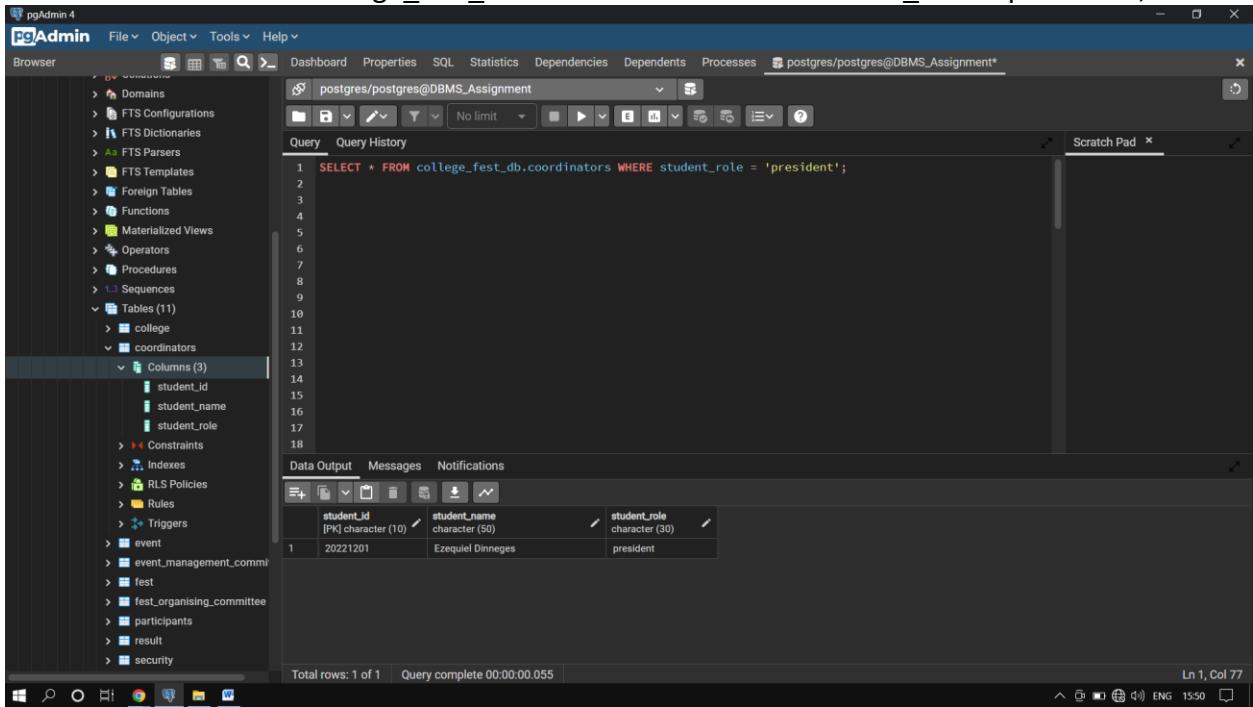
The 'Messages' tab shows the result of the query: "Query returned successfully in 123 msec." The status bar at the bottom indicates "Total rows: 0 of 0" and "Query complete 00:00:00.123". The bottom right corner shows "Ln 121, Col 3".

## Executing queries

1. English query: To display the President from the Co-ordinators.

Query:

```
SELECT * FROM college_fest_db.coordinators WHERE student_role = 'president';
```



The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure under 'Browser', including 'Domains', 'FTS Configurations', 'FTS Dictionaries', 'FTS Parsers', 'FTS Templates', 'Foreign Tables', 'Functions', 'Materialized Views', 'Operators', 'Procedures', 'Sequences', and 'Tables (11)'. The 'coordinators' table is selected, and its 'Columns (3)' are listed: 'student\_id', 'student\_name', and 'student\_role'. The main area shows the SQL query: 'SELECT \* FROM college\_fest\_db.coordinators WHERE student\_role = 'president';'. The 'Data Output' tab shows the result of the query:

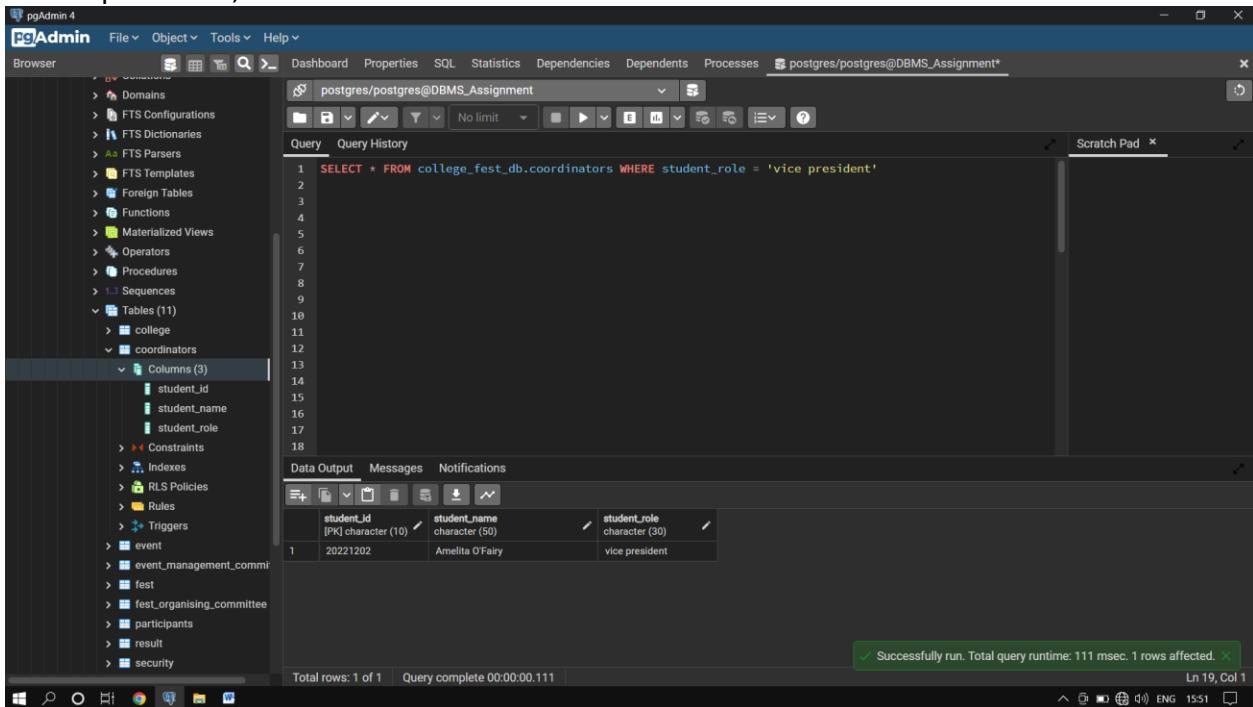
	student_id	student_name	student_role
1	20221201	Ezequiel Dinneges	president

Total rows: 1 of 1 | Query complete 00:00:00.055 | Ln 1, Col 77

2. English query: To display the Vice President from the Co-ordinators.

Query:

```
SELECT * FROM college_fest_db.coordinators WHERE student_role = 'vice president';
```



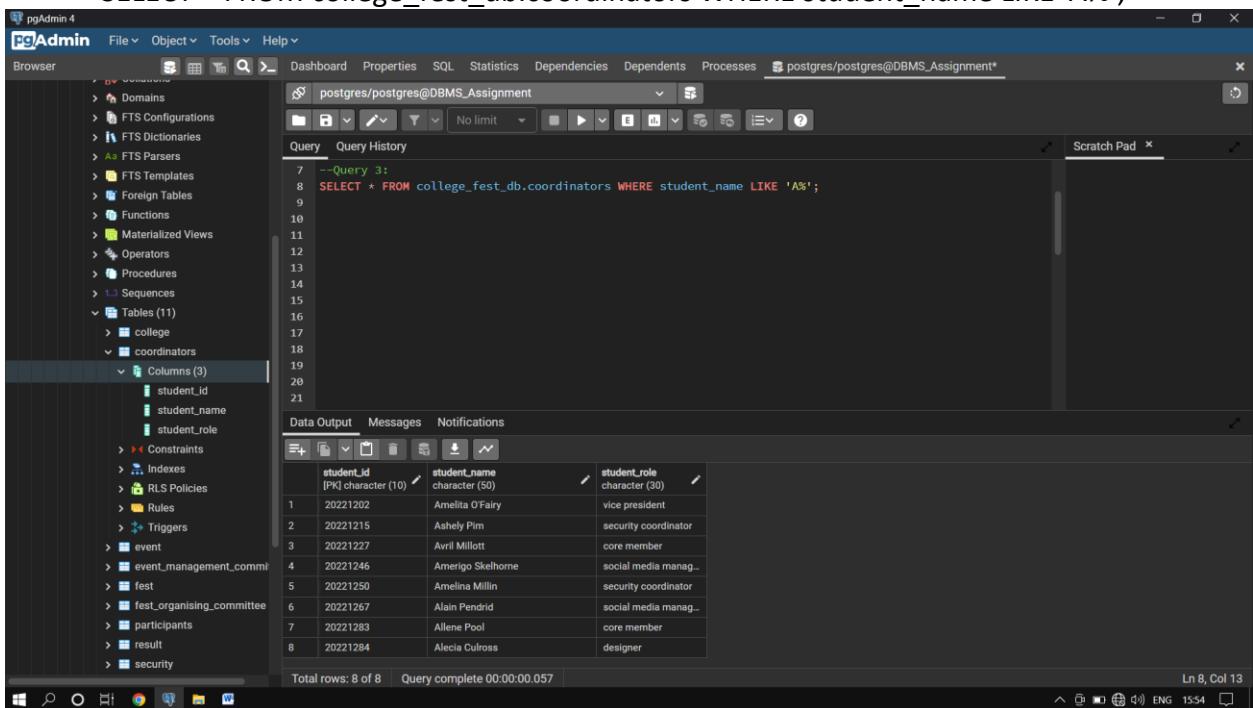
The screenshot shows the pgAdmin 4 interface. The left sidebar is the 'Browser' pane, showing the database structure with 'coordinators' selected. The main pane is the 'Query' pane, displaying the SQL query. The 'Data Output' pane shows the results of the query, which is a single row: student\_id 20221202, student\_name Amelita O'Fairy, and student\_role vice president. A status bar at the bottom right indicates the query was successfully run with a runtime of 111 msec and 1 row affected.

student_id	student_name	student_role
20221202	Amelita O'Fairy	vice president

3. English query: List all the o-ordinators whose names start with 'A'.

Query:

```
SELECT * FROM college_fest_db.coordinators WHERE student_name LIKE 'A%';
```

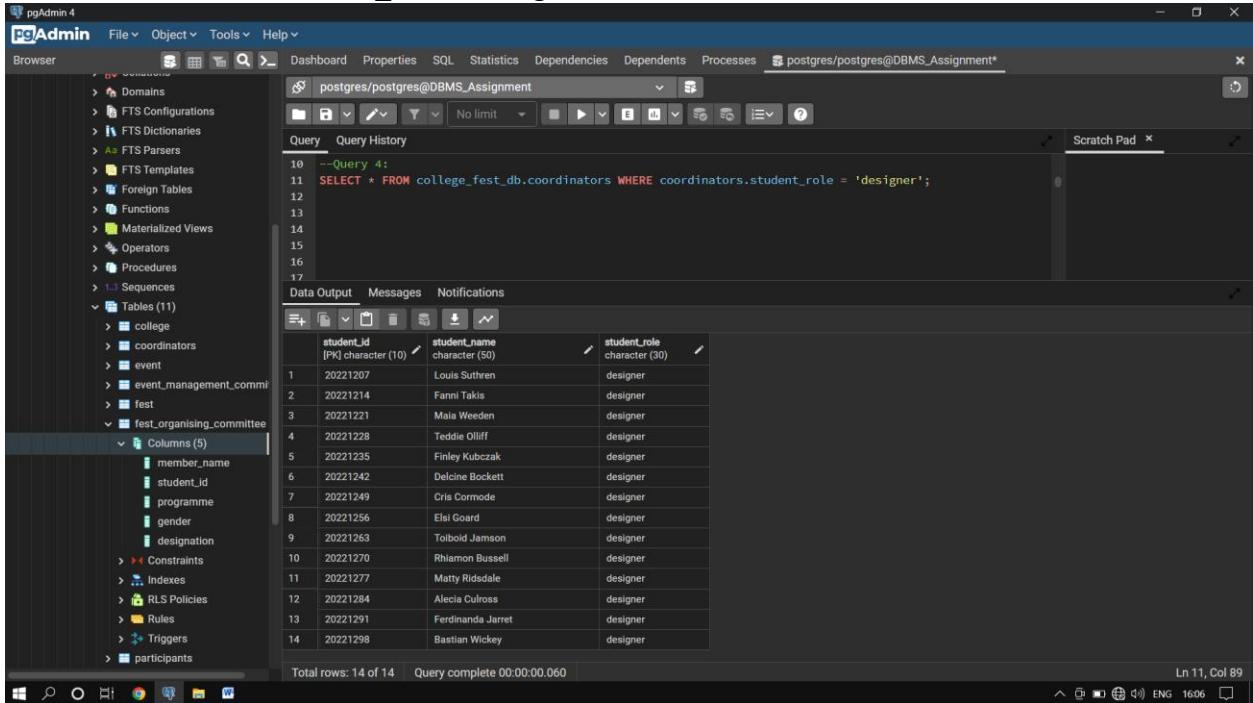


The screenshot shows the pgAdmin 4 interface. The left sidebar is the 'Browser' pane, showing the database structure with 'coordinators' selected. The main pane is the 'Query' pane, displaying the SQL query. The 'Data Output' pane shows the results of the query, listing 8 rows of coordinators whose names start with 'A'. The columns are student\_id, student\_name, and student\_role.

student_id	student_name	student_role
20221202	Amelita O'Fairy	vice president
20221215	Ashely Pim	security coordinator
20221227	Avril Millott	core member
20221246	Amerigo Skelhorne	social media manag...
20221250	Amelina Millin	security coordinator
20221267	Alain Pendrid	social media manag...
20221283	Allene Pool	core member
20221284	Alecia Culross	designer

4. English query: Select all the designers.

Query: `SELECT * FROM college_fest_db.coordinators WHERE coordinators.student_role = 'designer';`



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

- Browser:** The left sidebar shows the database structure with the `fest_organising_committee` table selected.
- Query Editor:** The main area contains the SQL query:

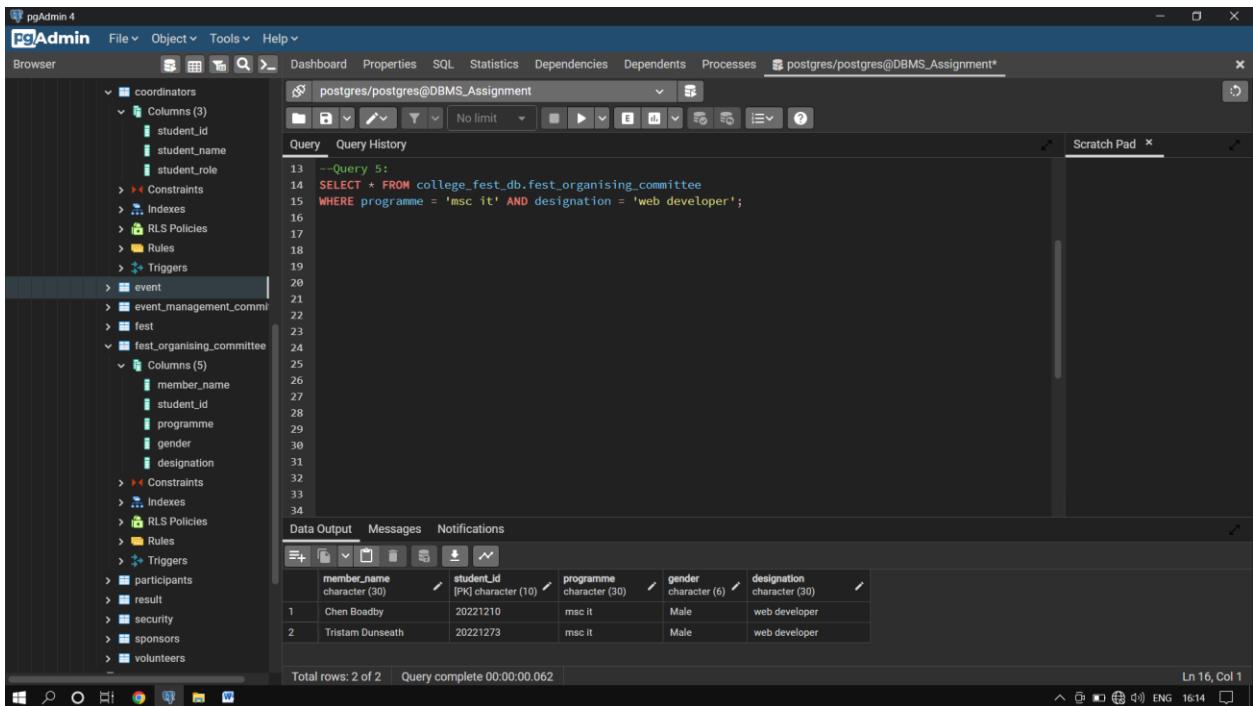
```
10 --Query 4:
11 SELECT * FROM college_fest_db.coordinators WHERE coordinators.student_role = 'designer';
12
13
14
15
16
17
```
- Data Output:** The results table shows 14 rows of data:

	student_id	student_name	student_role
1	20221207	Louis Sutren	designer
2	20221214	Fanni Takis	designer
3	20221221	Maia Weeden	designer
4	20221228	Teddie Olliff	designer
5	20221235	Finley Kubczak	designer
6	20221242	Decline Bockett	designer
7	20221249	Cris Cormode	designer
8	20221256	Eli Goard	designer
9	20221263	Tolbold Jamson	designer
10	20221270	Rhiamon Bussell	designer
11	20221277	Matty Ridsdale	designer
12	20221284	Alecia Culross	designer
13	20221291	Ferdinanda Jarret	designer
14	20221298	Bastian Wickey	designer

Total rows: 14 of 14 | Query complete 00:00:00.060

5. English query: Select all the Web Developers who are from MSc IT course.

Query: `SELECT * FROM college_fest_db.fest_organising_committee WHERE programme = 'msc it' AND designation = 'web developer';`



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

- Browser:** The left sidebar shows the database structure with the `fest_organising_committee` table selected.
- Query Editor:** The main area contains the SQL query:

```
13 --Query 5:
14 SELECT * FROM college_fest_db.fest_organising_committee
15 WHERE programme = 'msc it' AND designation = 'web developer';
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
```
- Data Output:** The results table shows 2 rows of data:

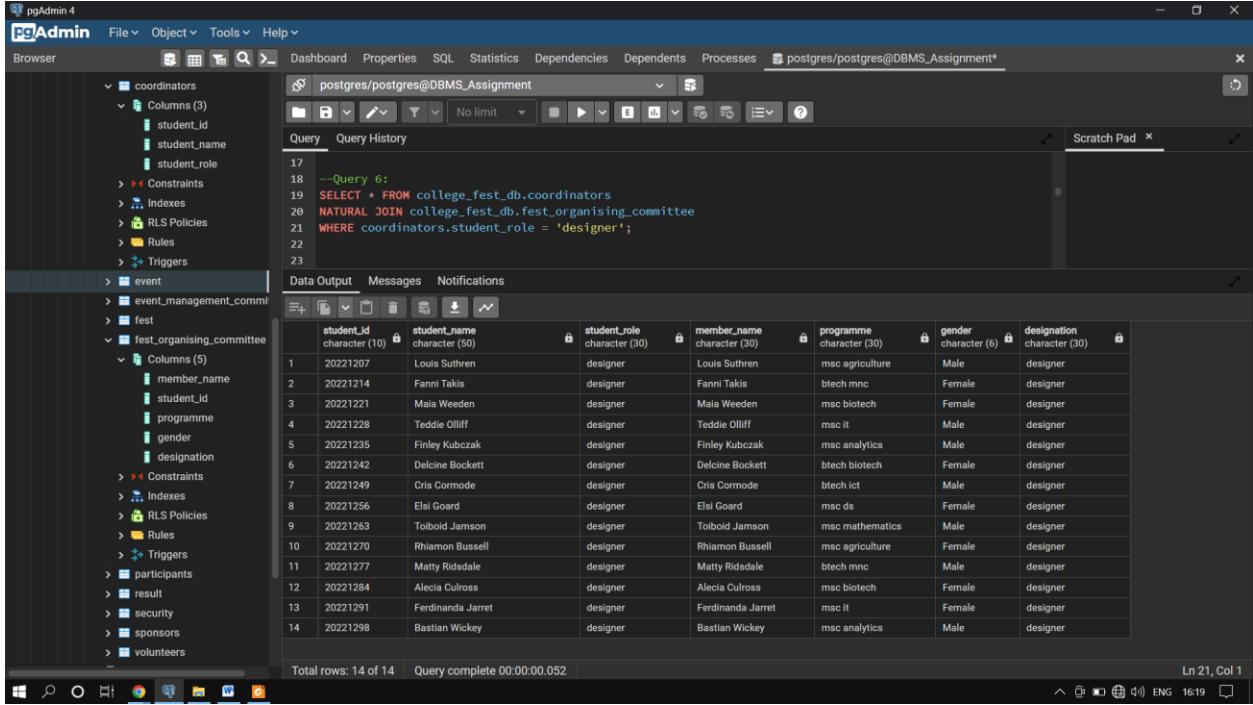
	member_name	student_id	programme	gender	designation
1	Chen Boardby	20221210	msc it	Male	web developer
2	Tristam Dunseath	20221273	msc it	Male	web developer

Total rows: 2 of 2 | Query complete 00:00:00.062

6. English query: Select all the designers who are female.

Query:

```
SELECT * FROM college_fest_db.coordinators NATURAL JOIN
college_fest_db.fest_organising_committee WHERE coordinators.student_role =
'designer';
```



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

- Browser:** The left sidebar shows the database schema structure for the 'college\_fest\_db' database, including tables like 'coordinators', 'fest\_organising\_committee', and 'event'.
- Query Editor:** The main window displays the SQL query and its execution results. The query is:

```
17
18 --Query 6:
19 SELECT * FROM college_fest_db.coordinators
20 NATURAL JOIN college_fest_db.fest_organising_committee
21 WHERE coordinators.student_role = 'designer';
22
23
```

- Data Output:** The results are presented in a table format:

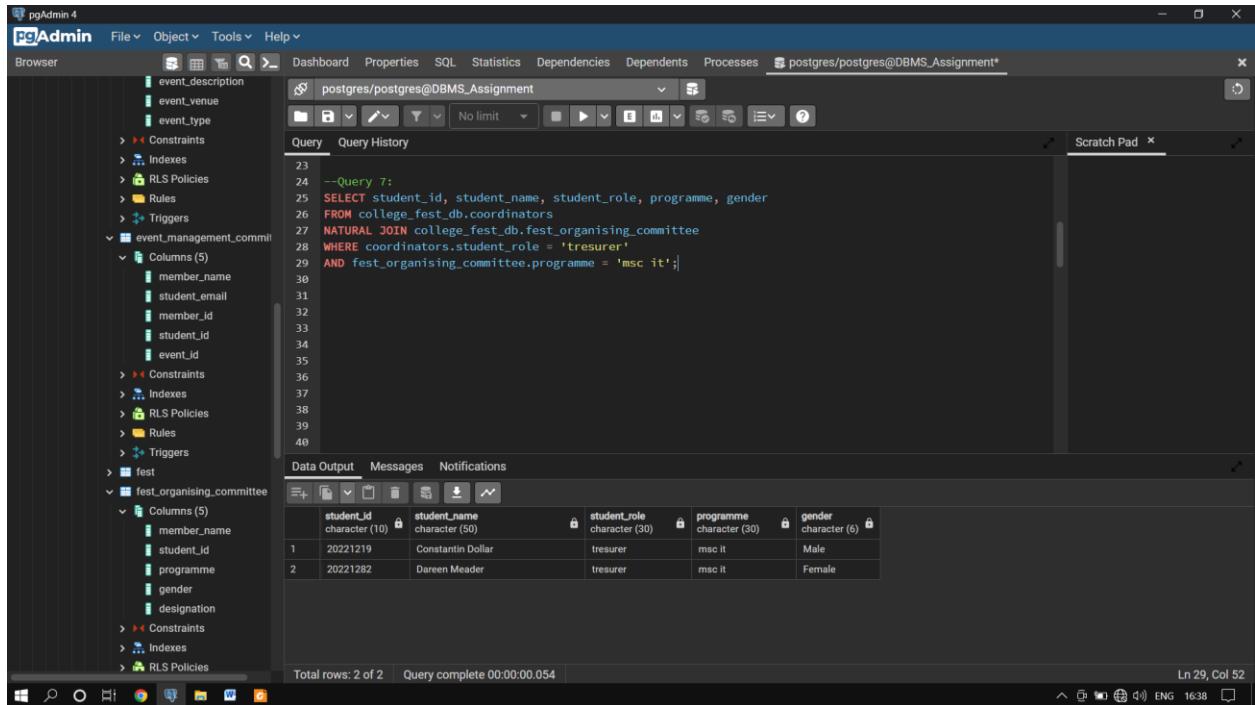
	student_id	student_name	student_role	member_name	programme	gender	designation
1	20221207	Louis Suthren	designer	Louis Suthren	msc agriculture	Male	designer
2	20221214	Fanni Takis	designer	Fanni Takis	btech mnc	Female	designer
3	20221221	Mala Weeden	designer	Mala Weeden	msc biotech	Female	designer
4	20221228	Teddie Olliff	designer	Teddie Olliff	msc it	Male	designer
5	20221235	Finley Kubczak	designer	Finley Kubczak	msc analytics	Male	designer
6	20221242	Delcine Bockett	designer	Delcine Bockett	btech biotech	Female	designer
7	20221249	Cris Cormode	designer	Cris Cormode	btech ict	Male	designer
8	20221256	Ela Goard	designer	Ela Goard	msc ds	Female	designer
9	20221263	Toibold Jamson	designer	Toibold Jamson	msc mathematics	Male	designer
10	20221270	Rhamon Bussell	designer	Rhamon Bussell	msc agriculture	Female	designer
11	20221277	Matty Riddale	designer	Matty Riddale	btech mnc	Male	designer
12	20221284	Alecia Culross	designer	Alecia Culross	msc biotech	Female	designer
13	20221291	Ferdinanda Jarret	designer	Ferdinanda Jarret	msc it	Female	designer
14	20221298	Bastian Wickey	designer	Bastian Wickey	msc analytics	Male	designer

- Messages:** The message bar at the bottom indicates "Total rows: 14 of 14" and "Query complete 00:00:00.052".

7. English query: Select personal information of students who have the role of Treasurer and are from MSc IT programme.

Query:

```
SELECT student_id, student_name, student_role, programme, gender
FROM college_fest_db.coordinators
NATURAL JOIN college_fest_db.fest_organising_committee
WHERE coordinators.student_role = 'treasurer'
AND fest_organising_committee.programme = 'msc it';
```



The screenshot shows the pgAdmin 4 interface with a query editor and a results table.

**Query Editor:**

```
23 --Query 7:
24
25 SELECT student_id, student_name, student_role, programme, gender
26 FROM college_fest_db.coordinators
27 NATURAL JOIN college_fest_db.fest_organising_committee
28 WHERE coordinators.student_role = 'treasurer'
29 AND fest_organising_committee.programme = 'msc it';
30
31
32
33
34
35
36
37
38
39
40
```

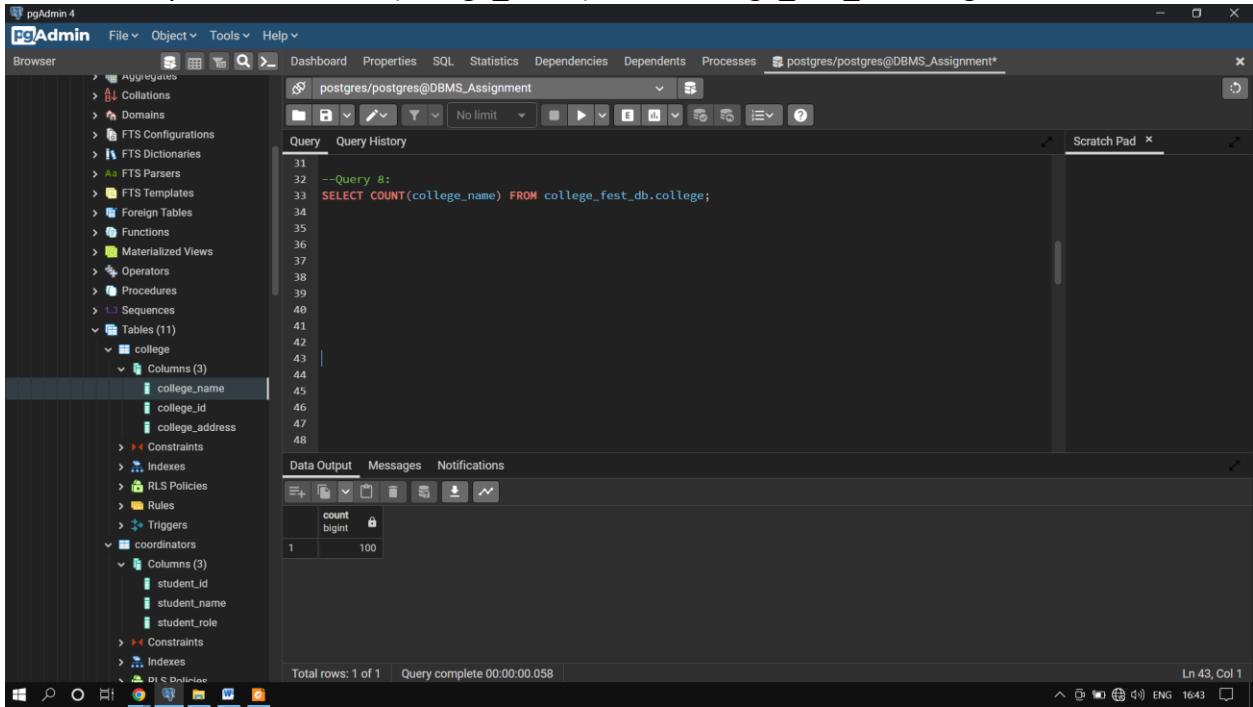
**Results Table:**

	student_id	student_name	student_role	programme	gender
1	20221219	Constantin Dollar	treasurer	msc it	Male
2	20221282	Dareen Meader	treasurer	msc it	Female

Total rows: 2 of 2 | Query complete 00:00:00.054 | Ln 29, Col 52

8. English query: Count the total number of participating colleges.

Query: `SELECT COUNT(college_name) FROM college_fest_db.college;`



The screenshot shows the pgAdmin 4 interface. The left sidebar (Browser) shows the database structure with the 'Tables (11)' node expanded, revealing 'college' and 'coordinators' tables, each with three columns. The 'college' table is selected. The main area (Query Editor) contains the following SQL query:

```
31
32 --Query 8:
33 SELECT COUNT(college_name) FROM college_fest_db.college;
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
```

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

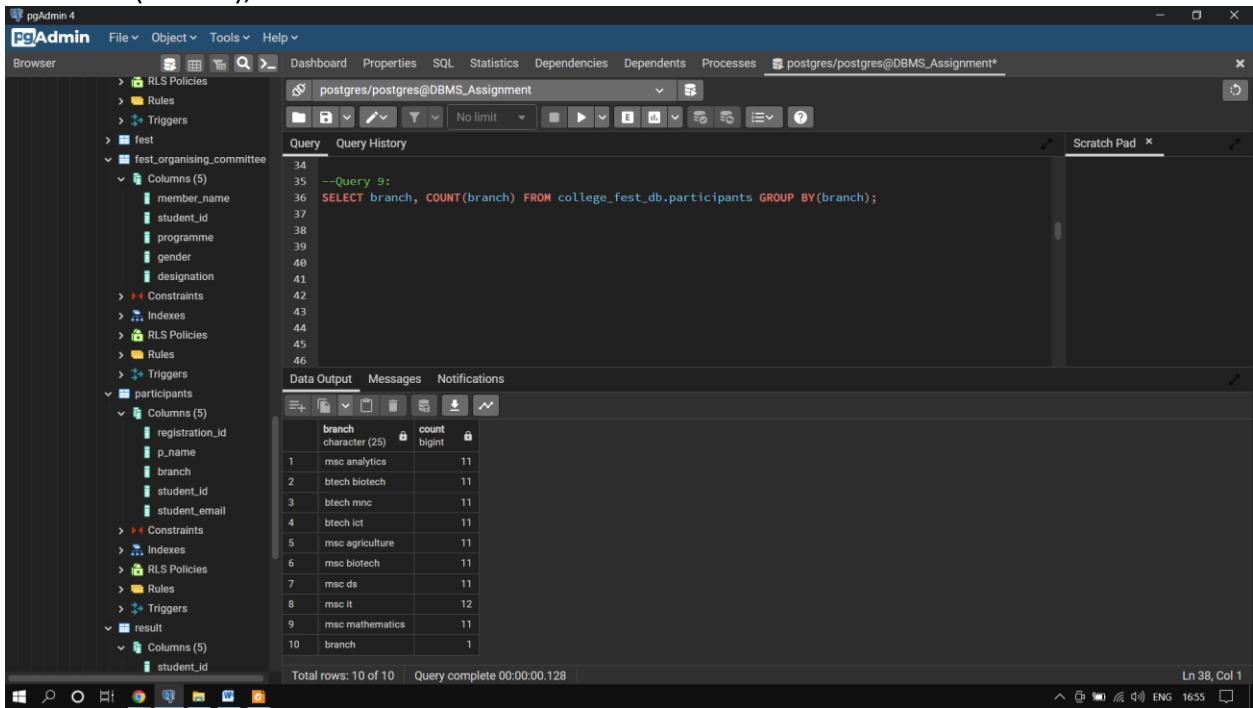
count	bigint
1	100

Below the table, the status bar indicates: 'Total rows: 1 of 1' and 'Query complete 00:00:00.058'.

9. English query: Count the total number of participants from each branch.

Query:

`SELECT branch, COUNT(branch) FROM college_fest_db.participants GROUP BY(branch);`



The screenshot shows the pgAdmin 4 interface. The left sidebar (Browser) shows the database structure with the 'fest' node expanded, revealing 'fest\_organising\_committee' and 'participants' tables, each with five columns. The 'participants' table is selected. The main area (Query Editor) contains the following SQL query:

```
34
35 --Query 9:
36 SELECT branch, COUNT(branch) FROM college_fest_db.participants GROUP BY(branch);
37
38
39
40
41
42
43
44
45
46
```

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

branch	character (25)	count	bigint
1	msc analytics	11	
2	btech biotech	11	
3	btech mnc	11	
4	btech ict	11	
5	msc agriculture	11	
6	msc biotech	11	
7	msc ds	11	
8	msc it	12	
9	msc mathematics	11	
10	branch	1	

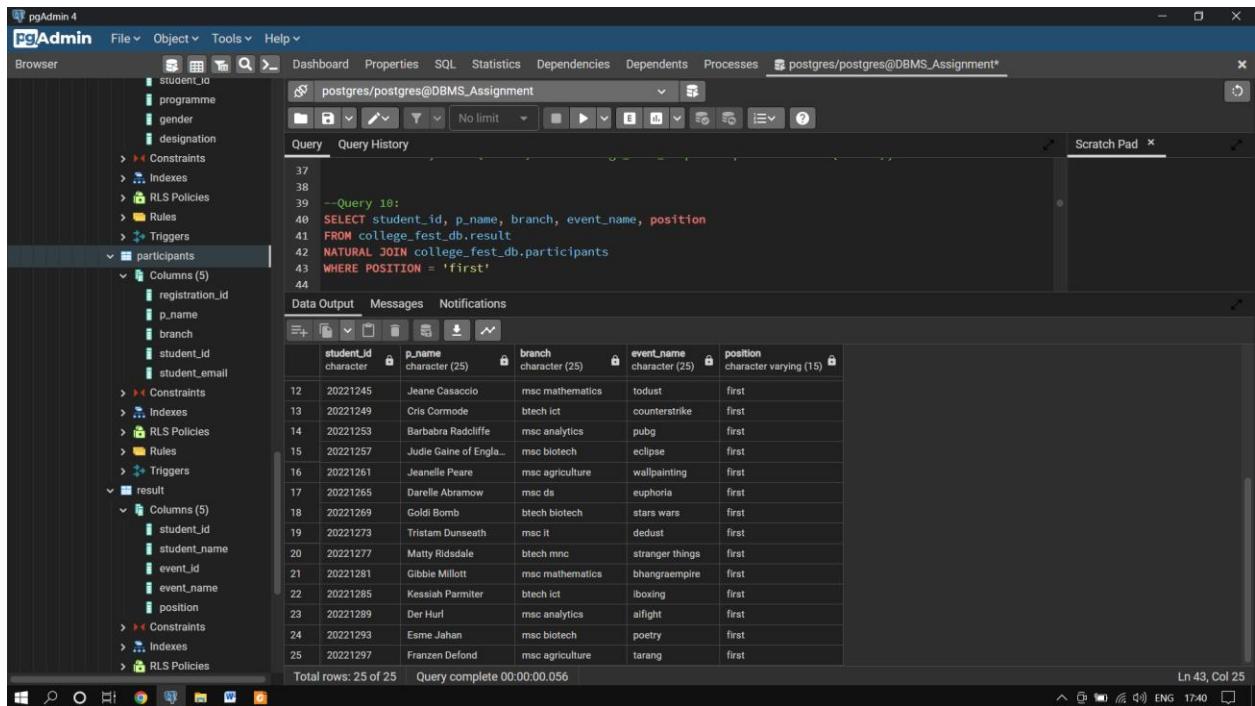
Below the table, the status bar indicates: 'Total rows: 10 of 10' and 'Query complete 00:00:00.128'.

10. English query: Display the following details

- Student id
- Student name
- Branch
- Event name
- Position

Of participants who secured first position.

Query: `SELECT student_id, p_name, branch, event_name, position  
FROM college_fest_db.result  
NATURAL JOIN college_fest_db.participants  
WHERE POSITION = 'first'`



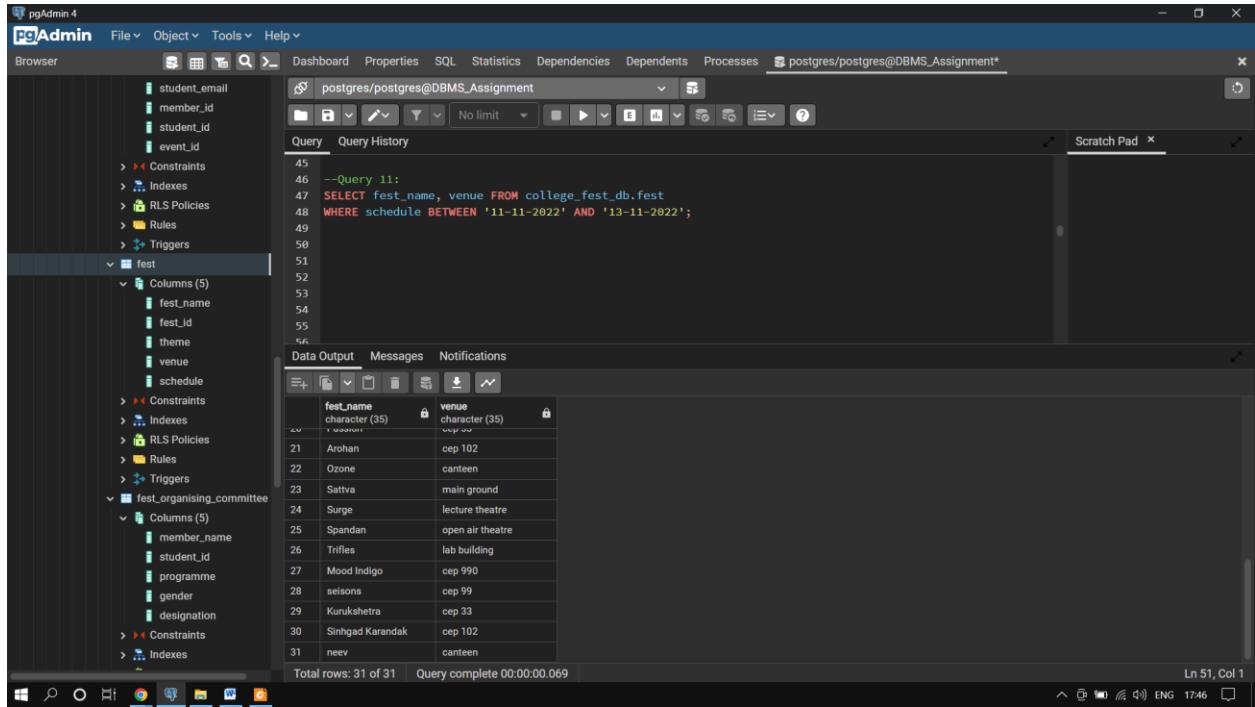
```
37  
38  
39 --Query 18:  
40 SELECT student_id, p_name, branch, event_name, position  
41 FROM college_fest_db.result  
42 NATURAL JOIN college_fest_db.participants  
43 WHERE POSITION = 'first'  
44
```

	student_id	p_name	branch	event_name	position
12	20221245	Jeane Casaccio	msc mathematics	todust	first
13	20221249	Cris Cormode	btech ict	counterstrike	first
14	20221253	Barbara Radcliffe	msc analytics	pubg	first
15	20221257	Judie Gaine of Engla...	msc biotech	eclipse	first
16	20221261	Jeanelle Peare	msc agriculture	wallpainting	first
17	20221265	Darelle Abramow	msc ds	euphoria	first
18	20221269	Goldi Bomb	btech biotech	stars wars	first
19	20221273	Tristam Dunseath	msc it	dedust	first
20	20221277	Matty Riddale	btech mnc	stranger things	first
21	20221281	Gibbs Millott	msc mathematics	bhangraempire	first
22	20221285	Kessiah Parmiter	btech ict	lboxing	first
23	20221289	Der Hurl	msc analytics	afight	first
24	20221293	Esmee Jahan	msc biotech	poetry	first
25	20221297	Franzen Defond	msc agriculture	tarang	first

11. English query: Display name and venue of all the fests that are between dates 11-11-2022 and 13-11-2022

Query:

```
SELECT * FROM college_fest_db.fest
WHERE schedule BETWEEN '11-11-2022' AND '13-11-2022';
```



The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database schema with tables like student\_email, member\_id, student\_id, event\_id, fest, fest\_organising\_committee, and their respective columns and constraints. The central area shows the SQL query being run:

```
--Query 11:
SELECT fest_name, venue FROM college_fest_db.fest
WHERE schedule BETWEEN '11-11-2022' AND '13-11-2022';
```

The results table shows the following data:

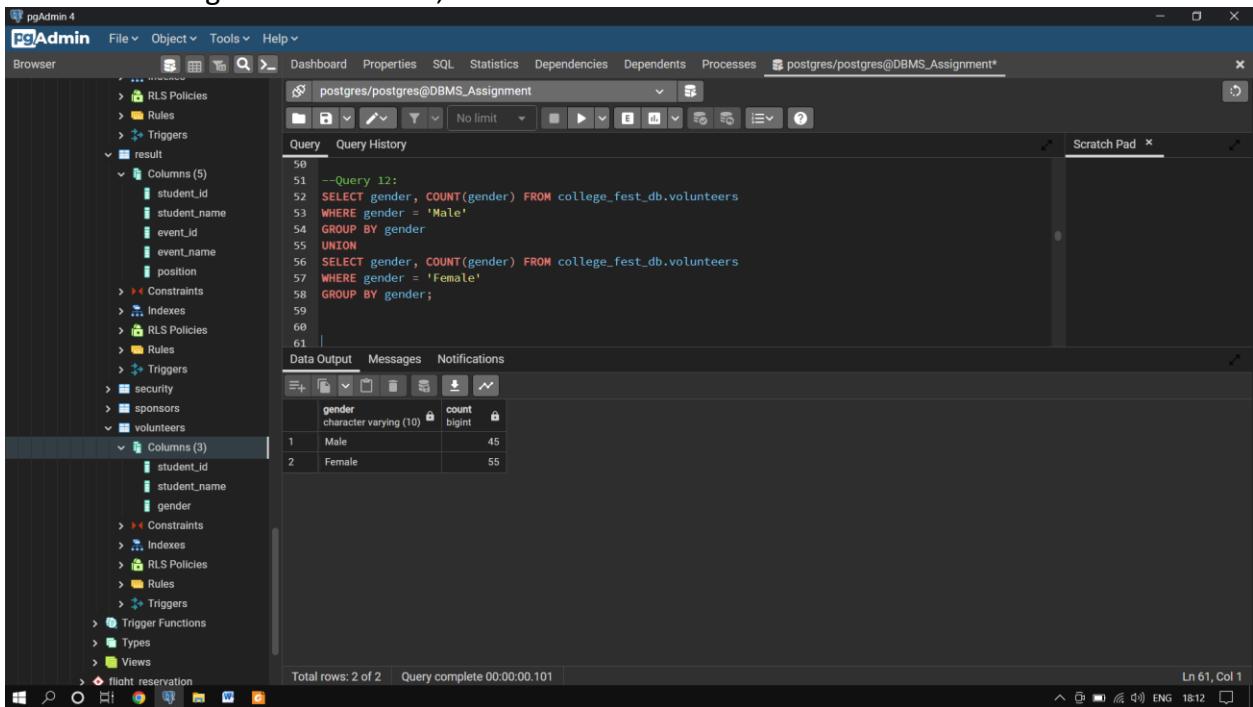
fest_name	venue
Arohan	cep 102
Ozone	canteen
Sattva	main ground
Surge	lecture theatre
Spandan	open air theatre
Trifles	lab building
Mood Indigo	cep 990
seisons	cep 99
Kurukshetra	cep 33
Sinhgad Karandak	cep 102
neev	canteen

Total rows: 31 of 31 Query complete 00:00:00.069

12. English query: Display the count of Male and Female volunteers

Query:

```
SELECT COUNT(gender) FROM college_fest_db.volunteers
WHERE gender = 'Male'
UNION
SELECT COUNT(gender) FROM college_fest_db.volunteers
WHERE gender = 'Female';
```



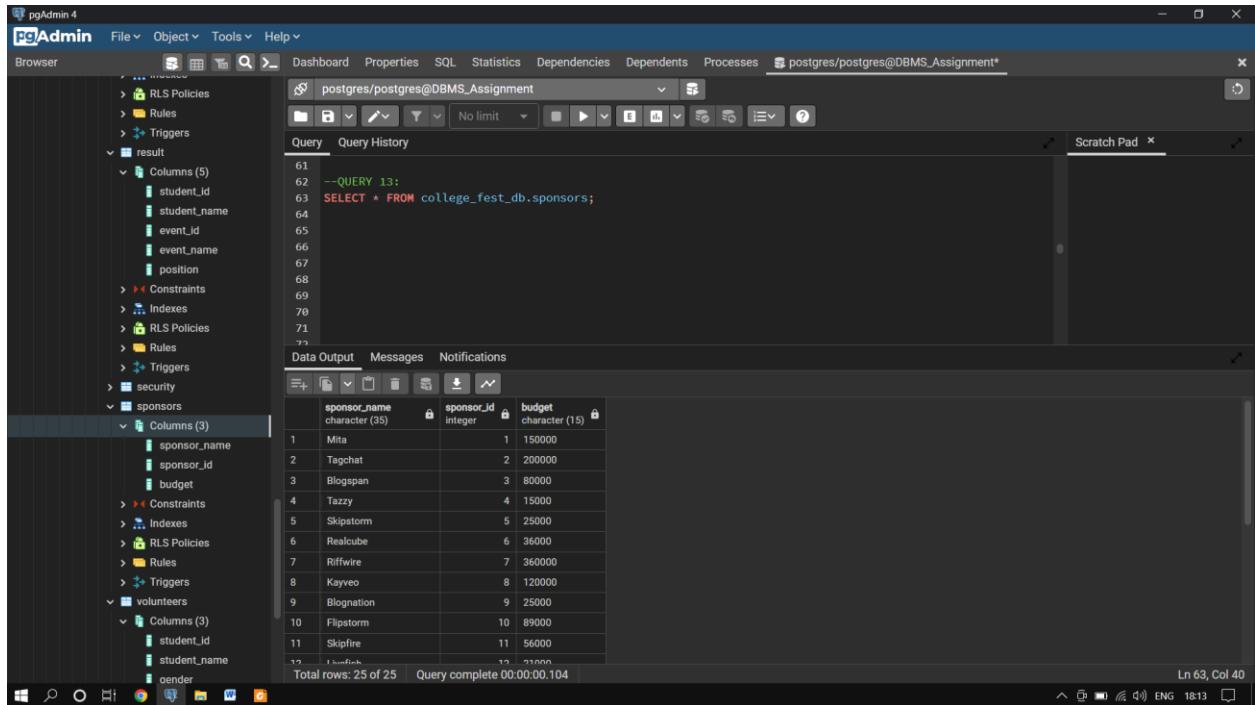
The screenshot shows the pgAdmin 4 interface with the following details:

- File Menu:** File, Object, Tools, Help
- Toolbar:** Standard toolbar with icons for New, Open, Save, Print, etc.
- Dashboard:** Shows the connection status: Postgres/postgres@DBMS\_Assignment\*
- Properties:** Shows the connection status: Postgres/postgres@DBMS\_Assignment\*
- Statistics:** Shows the connection status: Postgres/postgres@DBMS\_Assignment\*
- Dependencies:** Shows the connection status: Postgres/postgres@DBMS\_Assignment\*
- Dependents:** Shows the connection status: Postgres/postgres@DBMS\_Assignment\*
- Processes:** Shows the connection status: Postgres/postgres@DBMS\_Assignment\*
- Scratch Pad:** A tab labeled "Scratch Pad" is visible.
- Browser:** The left pane shows the database structure for the "college\_fest\_db" schema, including tables like "result", "volunteers", and "flight reservation".
- Query Editor:** The main pane contains the SQL query and its execution results.
  - Query:** The query is displayed in the editor.
  - Data Output:** The results are presented in a table format.
  - Table Headers:** gender (character varying (10)), count (bigint)
  - Table Data:**

gender	count
Male	45
Female	55
  - Message:** Total rows: 2 of 2 | Query complete 00:00:00.101 | Ln 61, Col 1

### 13. English query: Display all the sponsors.

Query: `SELECT * FROM college_fest_db.sponsors;`



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

- Browser:** Shows the schema structure of the database, including the `result` schema which contains the `sponsors` table.
- Query Editor:** Displays the SQL query: `--QUERY 13:  
SELECT * FROM college_fest_db.sponsors;`
- Data Output:** Shows the results of the query in a table format:

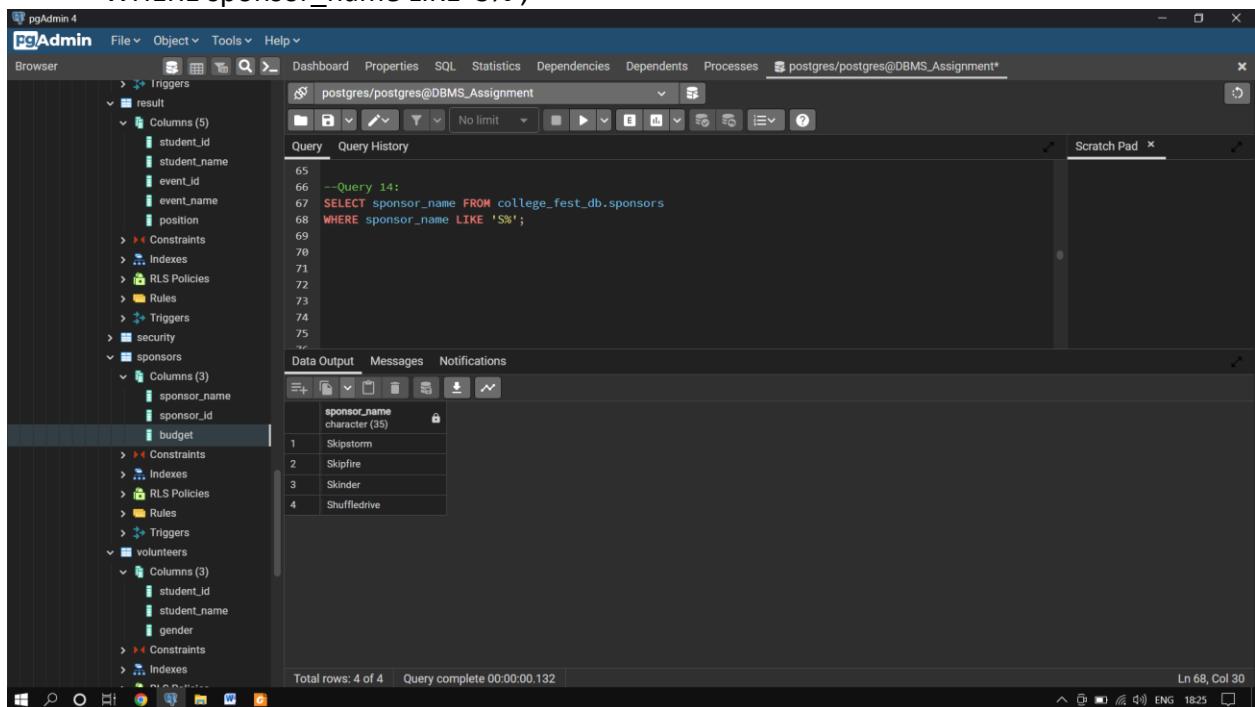
	sponsor_name	sponsor_id	budget
1	Mita	1	150000
2	Tagchat	2	200000
3	Blogspan	3	80000
4	Tazzy	4	15000
5	Skipatorm	5	25000
6	Realcube	6	36000
7	Riffwire	7	360000
8	Kayeo	8	120000
9	Blognation	9	25000
10	Flipstorm	10	89000
11	Skipfire	11	56000
12	Skinder	12	21000

Total rows: 25 of 25 | Query complete 00:00:00.104 | Ln 63, Col 40

### 14. English query: Display sponsors starting with S.

Query:

`SELECT sponsor_name FROM college_fest_db.sponsors  
WHERE sponsor_name LIKE 'S%';`



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

- Browser:** Shows the schema structure of the database, including the `result` schema which contains the `sponsors` table.
- Query Editor:** Displays the SQL query: `--Query 14:  
SELECT sponsor_name FROM college_fest_db.sponsors  
WHERE sponsor_name LIKE 'S%';`
- Data Output:** Shows the results of the query in a table format:

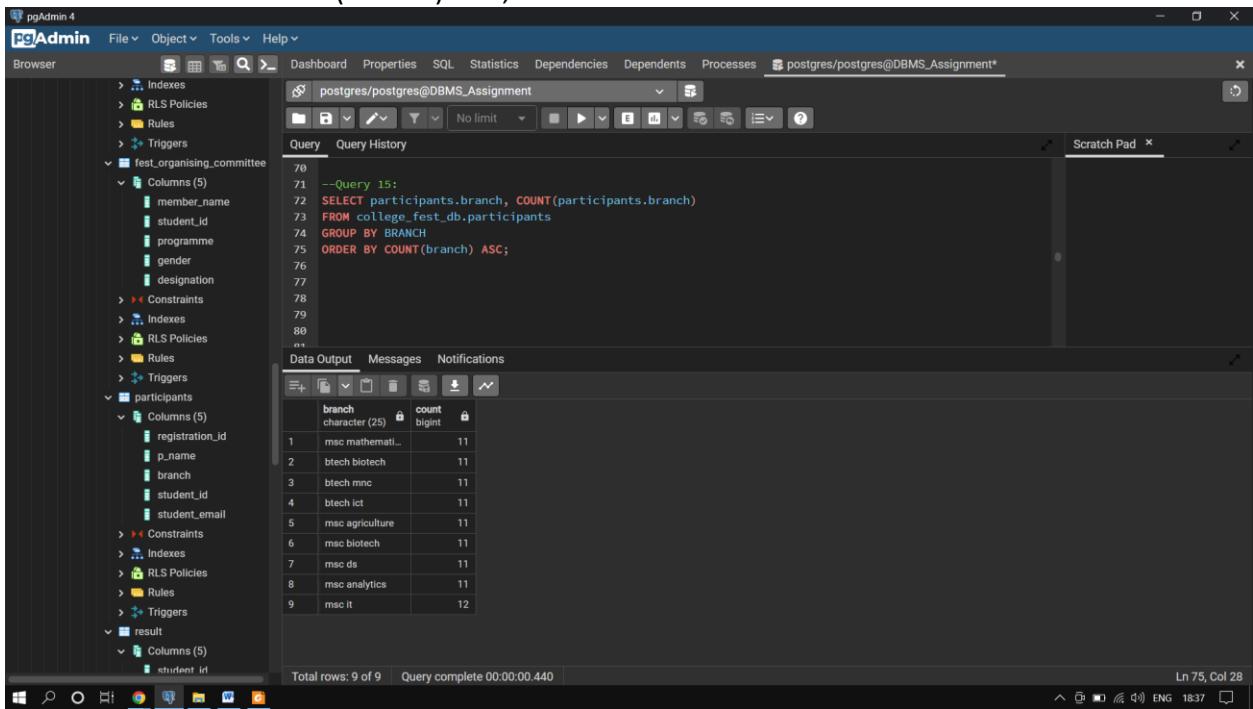
	sponsor_name
1	Skipatorm
2	Skipfire
3	Skinder
4	Shuffledrive

Total rows: 4 of 4 | Query complete 00:00:00.132 | Ln 68, Col 30

15. English query: Select the branch with total number of participations in ascending order of count.

Query:

```
SELECT participants.branch, COUNT(participants.branch)
FROM college_fest_db.participants
GROUP BY BRANCH
ORDER BY COUNT(branch) ASC;
```



The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure for the 'college\_fest\_db' schema, including the 'fest\_organising\_committee' and 'participants' tables. The 'participants' table has 5 columns: registration\_id, p\_name, branch, student\_id, and student\_email. The right pane shows the query editor with the following SQL code:

```
--Query 15:
SELECT participants.branch, COUNT(participants.branch)
FROM college_fest_db.participants
GROUP BY BRANCH
ORDER BY COUNT(branch) ASC;
```

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

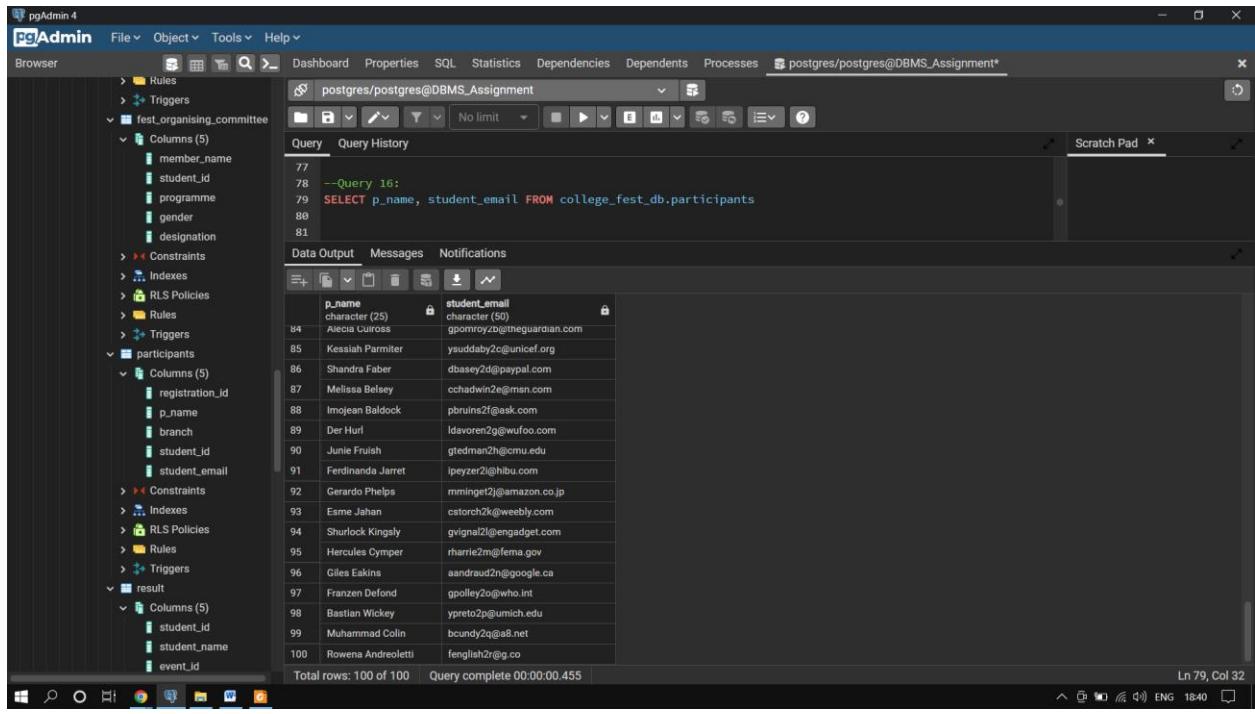
	branch	count
1	msc mathematics	11
2	btech biotech	11
3	btech mnc	11
4	btech ict	11
5	msc agriculture	11
6	msc biotech	11
7	msc ds	11
8	msc analytics	11
9	msc it	12

Total rows: 9 of 9 | Query complete 00:00:00.440 | Ln 75, Col 28

16. English query: Select the participant names and their email address.

Query:

```
SELECT p_name, student_email FROM college_fest_db.participants;
```



The screenshot shows the pgAdmin 4 interface with a query editor window. The query is:

```
--Query 16:  
SELECT p_name, student_email FROM college_fest_db.participants
```

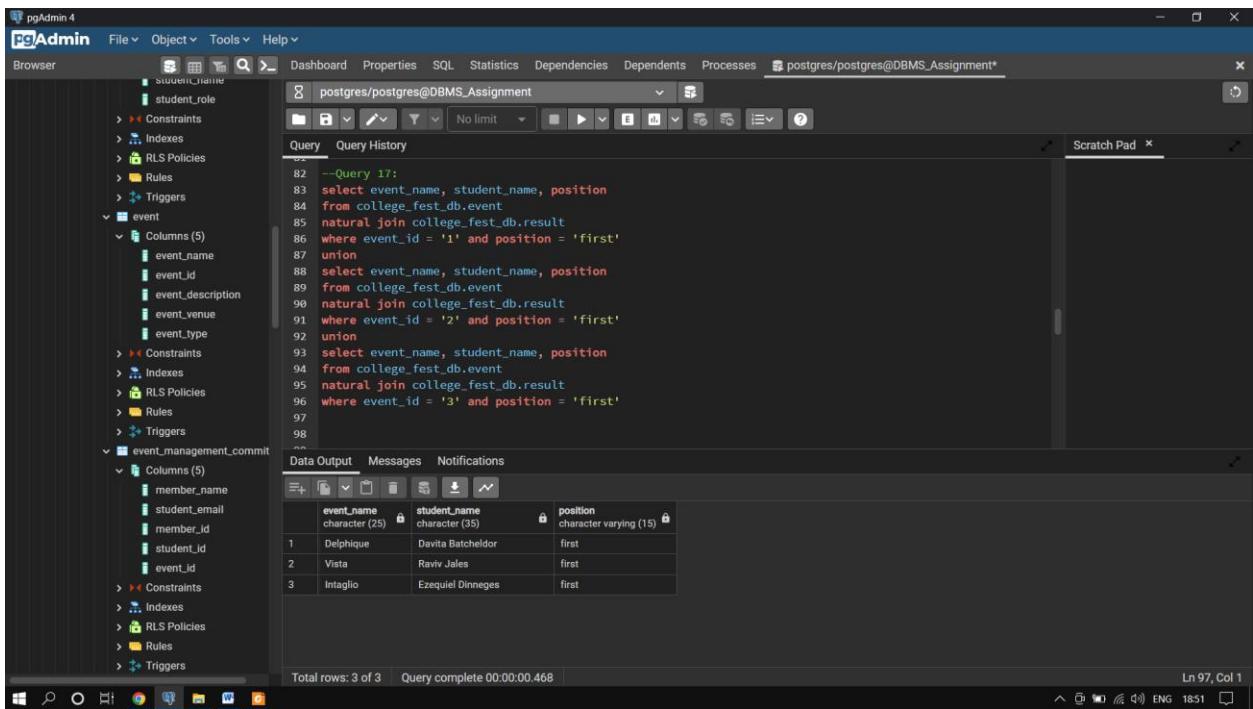
The results table displays 100 rows of data, each containing a participant's name and their email address. The columns are labeled 'p\_name' and 'student\_email'. The data includes names like Alecia Cuross, Kessiah Parmiter, Shandra Faber, Melissa Belsey, Imogen Baldock, Der Hurl, Junie Frish, Ferdinand Jarret, Gerardo Phelps, Esme Jahan, Sherlock Kingsley, Hercules Cypher, Giles Eakins, Franzen Defond, Bastian Wickey, Muhammad Coli, and Rowena Andreolitti, along with their respective email addresses.

	p_name	student_email
84	Alecia Cuross	gpmoroyzb@theguardian.com
85	Kessiah Parmiter	yauddaby2c@unicef.org
86	Shandra Faber	dbasey2d@paypal.com
87	Melissa Belsey	cchadwin2e@msn.com
88	Imogen Baldock	pbruins2f@ask.com
89	Der Hurl	ldsavoren2g@wufoo.com
90	Junie Frish	gtedman2h@cmu.edu
91	Ferdinand Jarret	lpeyzer2i@hbu.com
92	Gerardo Phelps	mminger2j@amazon.co.jp
93	Esme Jahan	cstorch2k@weebly.com
94	Sherlock Kingsley	gwignall2l@engadget.com
95	Hercules Cypher	rharris2m@fema.gov
96	Giles Eakins	aandraud2n@google.ca
97	Franzen Defond	gpolley2o@who.int
98	Bastian Wickey	ypreto2p@umich.edu
99	Muhammad Coli	bcundy2q@q8.net
100	Rowena Andreolitti	fenglish2r@g.co

17. English query: Select participants' names who secured first position from the top 3 events in the Event table.

Query:

```
select event_name, student_name, position
from college_fest_db.event
natural join college_fest_db.result
where event_id = '1' and position = 'first'
union
select event_name, student_name, position
from college_fest_db.event
natural join college_fest_db.result
where event_id = '2' and position = 'first'
union
select event_name, student_name, position
from college_fest_db.event
natural join college_fest_db.result
where event_id = '3' and position = 'first'
```



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

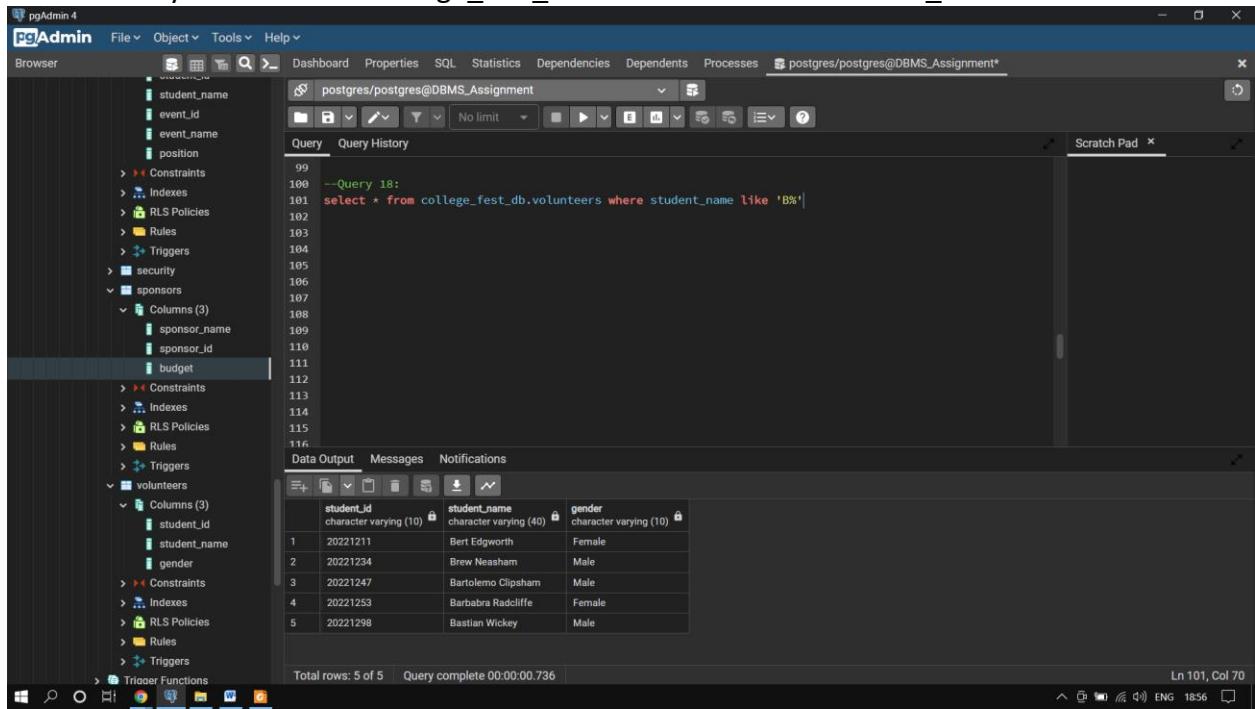
- Toolbar:** Includes File, Object, Tools, Help, Dashboard, Properties, SQL, Statistics, Dependencies, Dependents, Processes, and a connection dropdown for "postgres/postgres@DBMS\_Assignment".
- Browser:** Shows the schema structure for the "student\_fest" database, including tables like "student", "student\_role", "event", and "event\_management\_commit".
- Query Editor:** Displays the SQL query from the question, with line numbers 82 to 98.
- Data Output:** Shows the results of the query in a table format.

	event_name	student_name	position
1	Delphique	Daviv Batcheldor	first
2	Vista	Reviv Jales	first
3	Intaglio	Ezequiel Dinnenes	first

- Messages:** Shows "Total rows: 3 of 3" and "Query complete 00:00:00.468".
- System:** Shows the Windows taskbar at the bottom with icons for Start, Search, Task View, File Explorer, Edge, and File Explorer.

18. English query: Select volunteers with names starting with B

Query: select \* from college\_fest\_db.volunteers where student\_name like 'B%'



The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database schema with tables like student, event, sponsor, and volunteers. The volunteers table is currently selected. The main area shows a query editor with the following SQL code:

```
--Query 18:  
select * from college_fest_db.volunteers where student_name like 'B%'
```

Below the query, the results are displayed in a table:

student_id	student_name	gender
1	Bert Edgworth	Female
2	Brew Neasham	Male
3	Bartolemo Clipsham	Male
4	Barbara Radcliffe	Female
5	Bastian Wickey	Male

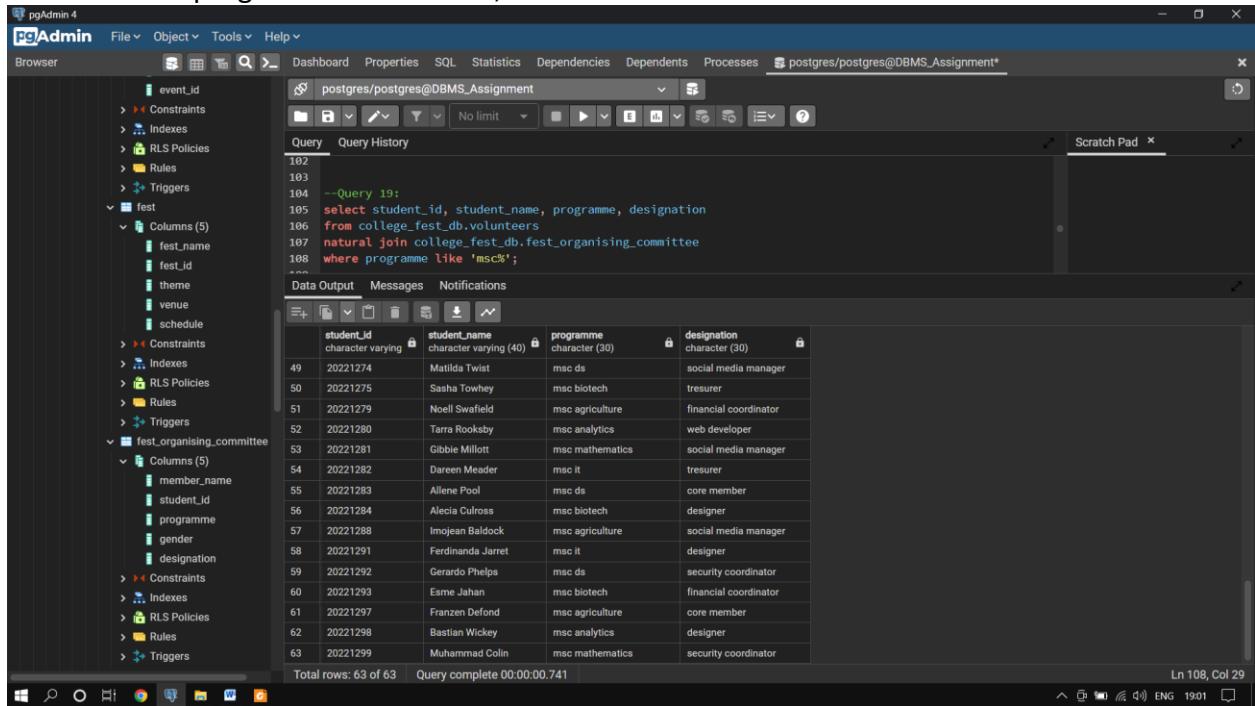
The status bar at the bottom indicates "Total rows: 5 of 5" and "Query complete 00:00:00.736".

19. English query: Display the following details of volunteers from MSc IT programme

- a. Student ID
- b. Student name
- c. Programme
- d. Designation

Query:

```
select student_id, student_name, programme, designation
from college_fest_db.volunteers
natural join college_fest_db.fest_organising_committee
where programme like 'msc%';
```



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

- Browser:** Shows the database structure for the 'college\_fest\_db' schema, including tables like 'fest', 'fest\_organising\_committee', and their respective columns and constraints.
- Query Editor:** The query is entered in the main query window:

```
102
103
104 --Query 19:
105 select student_id, student_name, programme, designation
106 from college_fest_db.volunteers
107 natural join college_fest_db.fest_organising_committee
108 where programme like 'msc%';
```
- Data Output:** The results are displayed in a table format:

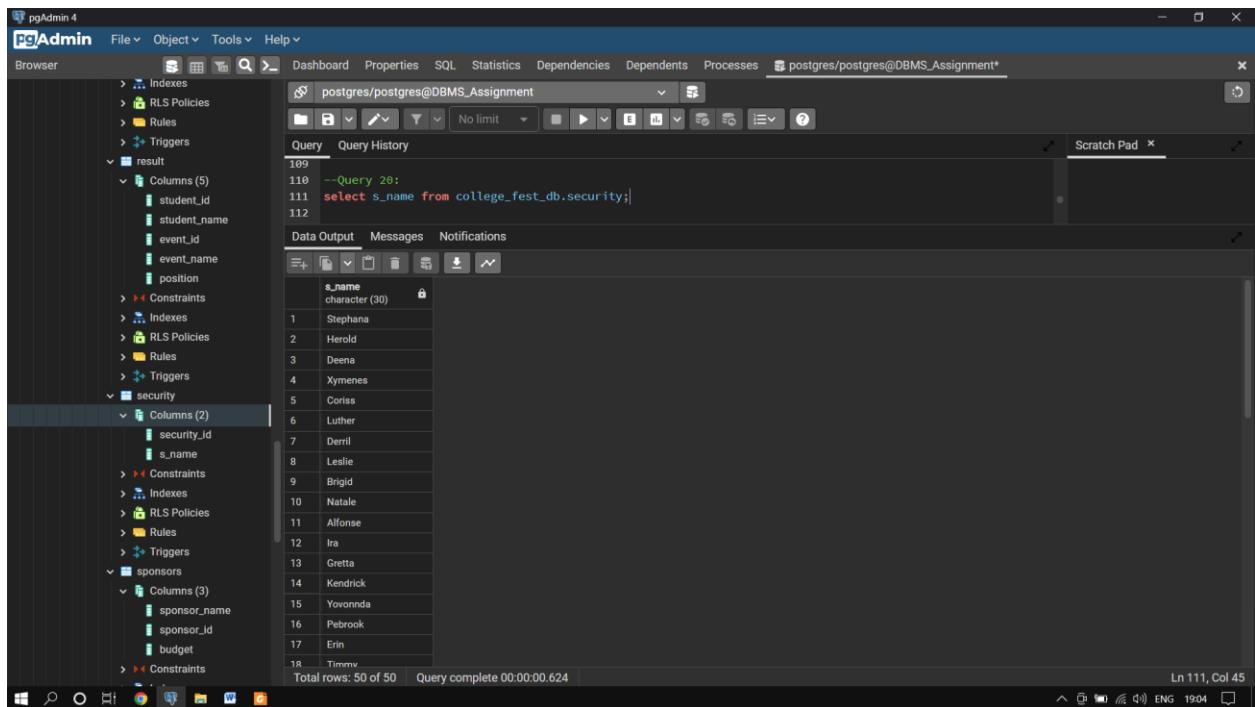
	student_id	student_name	programme	designation
49	20221274	Matilda Twist	msc ds	social media manager
50	20221275	Sasha Towhey	msc biotech	treasurer
51	20221279	Noell Swafield	msc agriculture	financial coordinator
52	20221280	Terra Rooksby	msc analytics	web developer
53	20221281	Gibbie Millott	msc mathematics	social media manager
54	20221282	Dereen Meader	msc it	treasurer
55	20221283	Allene Pool	msc ds	core member
56	20221284	Alecia Culross	msc biotech	designer
57	20221288	Imojean Baldock	msc agriculture	social media manager
58	20221291	Ferdinanda Jarret	msc it	designer
59	20221292	Gerardo Phelps	msc ds	security coordinator
60	20221293	Esme Jahan	msc biotech	financial coordinator
61	20221297	Franzen Defond	msc agriculture	core member
62	20221298	Bastian Wickey	msc analytics	designer
63	20221299	Muhammad Colin	msc mathematics	security coordinator

Total rows: 63 of 63    Query complete 00:00:00.741    Ln 108, Col 29

20. English query: Select all the Security members.

Query:

```
select s_name from college_fest_db.security;
```



The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure for the 'college\_fest\_db' schema, including tables like 'result', 'security', and 'sponsors'. The 'security' table is currently selected. The main window shows a query editor with the following SQL code:

```
189
110 --Query 28:
111 select s_name from college_fest_db.security;
112
```

The results pane displays 18 rows of data, each containing a name:

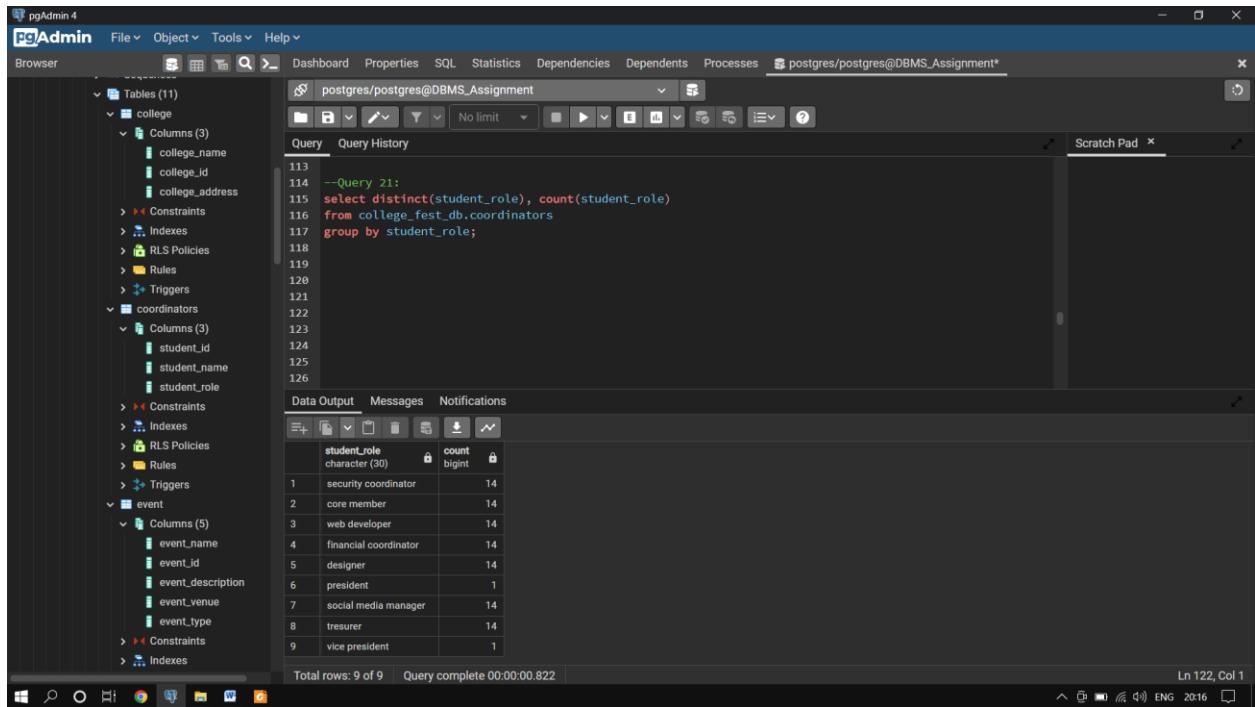
	s_name
1	Stephana
2	Herold
3	Deena
4	Xymenes
5	Coriss
6	Luther
7	Derril
8	Leslie
9	Brigid
10	Natale
11	Alfonse
12	Ira
13	Gretta
14	Kendrick
15	Yovonna
16	Pebrock
17	Erin
18	Timmv

Below the results, the status bar indicates "Total rows: 50 of 50" and "Query complete 00:00:00.624".

21. English query: Display total count of distinct Student roles.

Query:

```
select distinct(student_role), count(student_role)
from college_fest_db.coordinators
group by student_role;
```



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

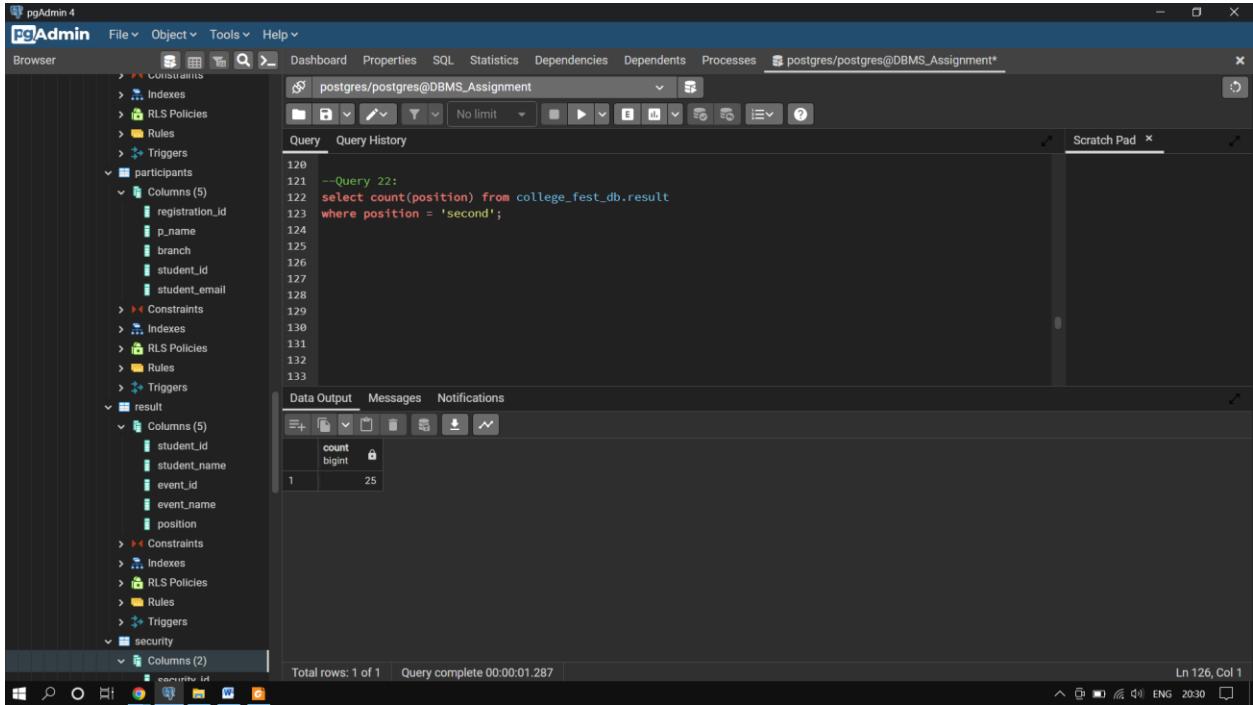
- Browser:** Shows the database structure with tables: college, coordinators, and event.
- Query Editor:** Contains the SQL query for displaying distinct student roles and their counts.
- Data Output:** Displays the results of the query in a table format.
- Table Data:**

	student_role	count
1	security coordinator	14
2	core member	14
3	web developer	14
4	financial coordinator	14
5	designer	14
6	president	1
7	social media manager	14
8	treasurer	14
9	vice president	1
- Messages:** Shows the message "Query complete 00:00:00.822".
- System Bar:** Shows the date and time as "2016-08-10 20:16".

22. English query: Count the total number of Second positions

Query:

```
select count(position) from college_fest_db.result
where position = 'second';
```



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

- Toolbar:** File, Object, Tools, Help.
- Browser:** Shows the database structure for 'college\_fest\_db'. It includes the 'participants' table (5 columns: registration\_id, p\_name, branch, student\_id, student\_email), the 'result' table (5 columns: student\_id, student\_name, event\_id, event\_name, position), and the 'security' table (2 columns: security\_id).
- Query Editor:** The query is entered in the 'Query' tab:

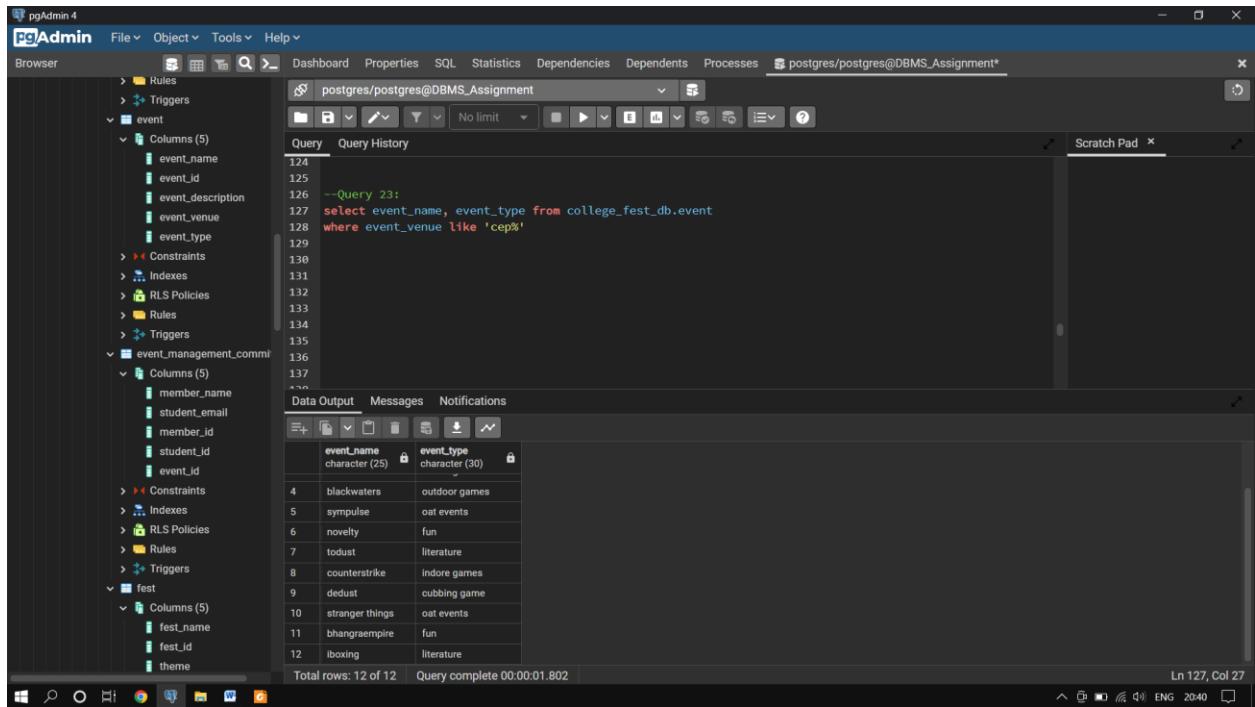
```
120
121 --Query 22:
122 select count(position) from college_fest_db.result
123 where position = 'second';
124
125
126
127
128
129
130
131
132
133
```
- Data Output:** The results are displayed in a table:

	count
1	25
- Messages:** Total rows: 1 of 1 | Query complete 00:00:01.287 | Ln 126, Col 1

23. English query: Display all the Event Name and Event Type which are hosted in CEP Block.

Query:

```
select event_name, event_type from college_fest_db.event
where event_venue like 'cep%';
```



The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure for the 'college\_fest\_db' schema, including tables like 'event', 'event\_management\_committee', and 'fest'. The main area shows the query editor with the following SQL code:

```
--Query 23:
select event_name, event_type from college_fest_db.event
where event_venue like 'cep%';
```

The results are displayed in a table:

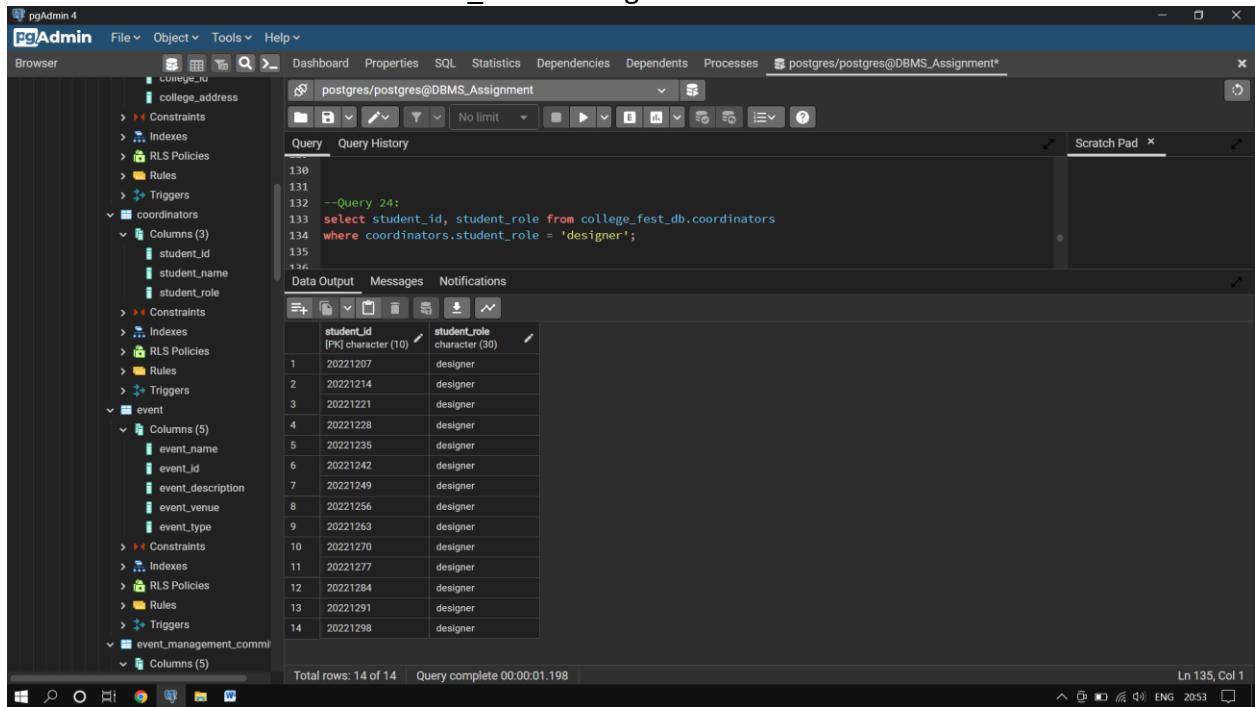
	event_name	event_type
4	blackwaters	outdoor games
5	sympulse	oat events
6	novelty	fun
7	todust	literature
8	counterstrike	indore games
9	dedust	cubbing game
10	stranger things	oat events
11	bhangraempire	fun
12	iboxing	literature

Total rows: 12 of 12 Query complete 00:00:01.802

#### 24. English query: Select all the Student IDs of Designers

Query:

```
select student_id, student_role from college_fest_db.coordinators
where coordinators.student_role = 'designer'
```



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

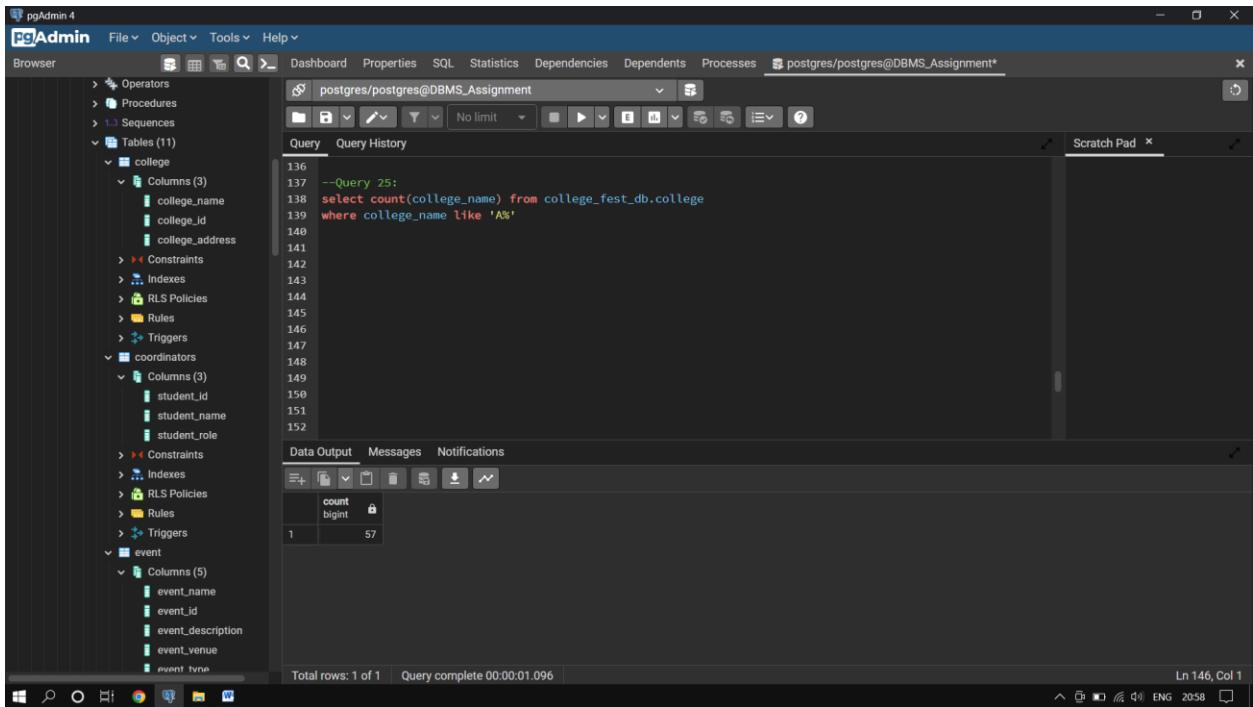
- Browser:** Displays the schema tree for the database "college\_fest\_db". The "coordinators" table is selected, showing its 3 columns: student\_id, student\_name, and student\_role.
- Query Editor:** The query "select student\_id, student\_role from college\_fest\_db.coordinators where coordinators.student\_role = 'designer';" is entered.
- Results:** A table shows the 14 rows returned by the query, with columns "student\_id" and "student\_role". All 14 rows have "student\_role" as "designer".
- Message Bar:** Shows "Total rows: 14 of 14" and "Query complete 00:00:01.198".
- Status Bar:** Shows "Ln 135, Col 1" and system icons.

student_id	student_role
1	designer
2	designer
3	designer
4	designer
5	designer
6	designer
7	designer
8	designer
9	designer
10	designer
11	designer
12	designer
13	designer
14	designer

25. English query: Display the count of participating colleges whose name start with A.

Query:

```
select count(college_name) from college_fest_db.college
where college_name like 'A%';
```



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

- Browser:** Shows the database structure with tables: college, coordinators, and event.
- Query Editor:** Displays the SQL query and its execution results.
- Results:** A table showing the count of colleges starting with 'A'.
- System Information:** Shows the total rows (1), query completion time (00:00:01.096), and line/character position (Ln 146, Col 1).

```
--Query 25:
select count(college_name) from college_fest_db.college
where college_name like 'A%'

Data Output
count
1          57

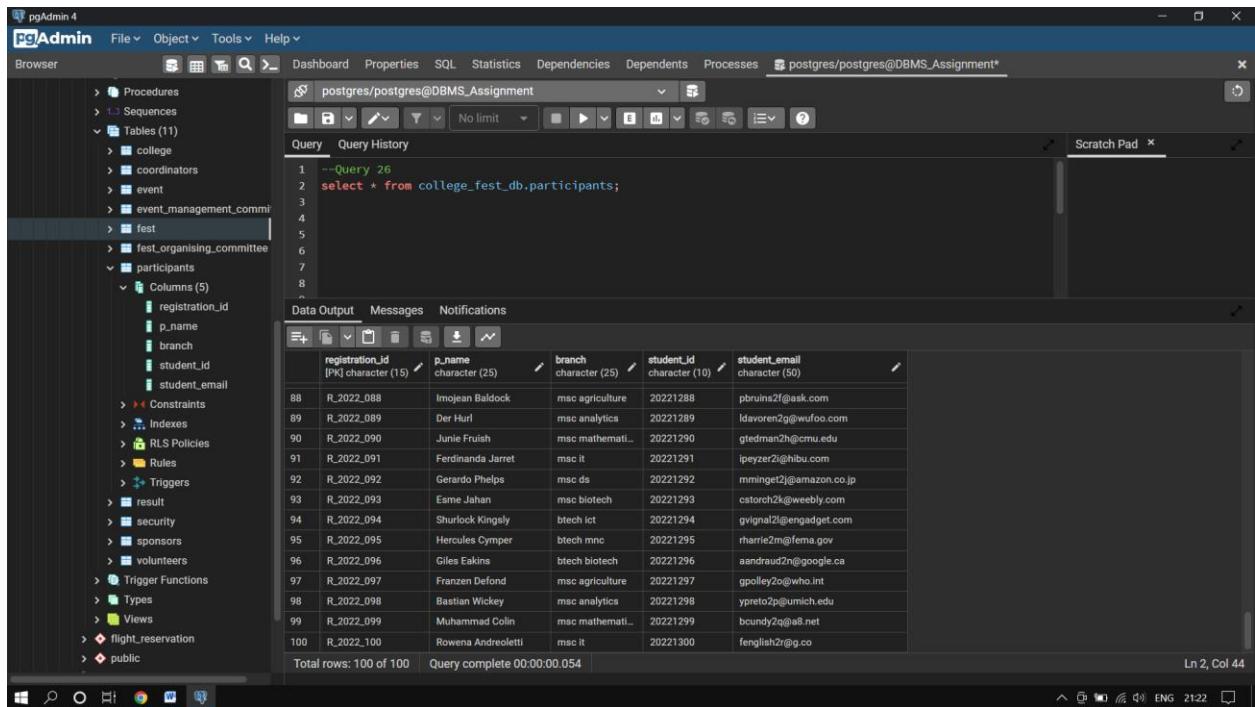
Total rows: 1 of 1  Query complete 00:00:01.096  Ln 146, Col 1
```

26. English query: Display the following participants details

- Registration ID
- Name
- Branch
- Student ID
- Email Address

Query:

```
select * from college_fest_db.participants;
```



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

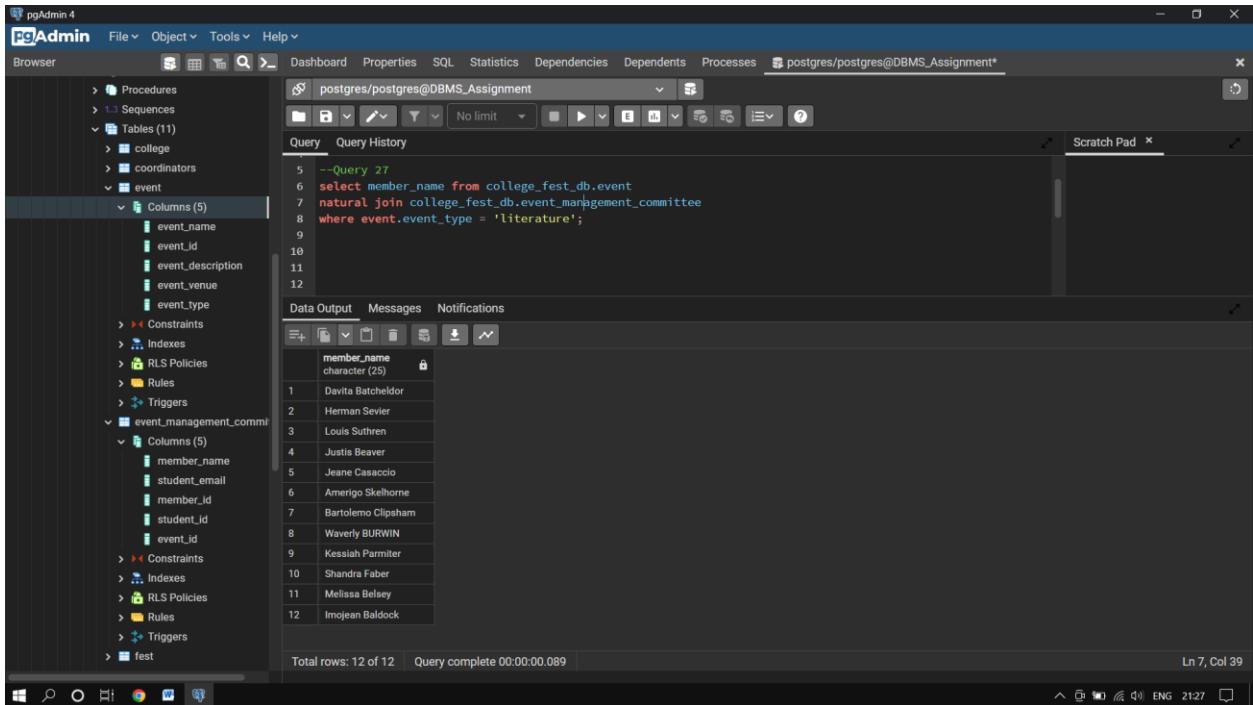
- File Menu:** File, Object, Tools, Help
- Toolbar:** Browser, Dashboard, Properties, SQL, Statistics, Dependencies, Dependents, Processes
- Connection:** postgres/postgres@DBMS\_Assignment\*
- Query Editor:** Shows the query: `select * from college_fest_db.participants;`
- Table Browser:** Shows the `participants` table with 100 rows.
- Data Output:** Shows the results of the query, including columns: `registration_id`, `p_name`, `branch`, `student_id`, and `student_email`.
- Messages:** Shows the message: "Query complete 0:00:00.054".
- Notifications:** Shows the message: "Ln 2, Col 44".

registration_id	p_name	branch	student_id	student_email
R_2022_088	Imojean Baldock	msc agriculture	20221288	pbruins2f@ask.com
R_2022_089	Der Hurl	msc analytics	20221289	ldavoren2g@wufoo.com
R_2022_090	Junie Fruish	msc mathemati...	20221290	gtedman2h@cmu.edu
R_2022_091	Ferdinanda Jarret	msc it	20221291	ipeyzer2i@hulu.com
R_2022_092	Gerardo Phelps	msc ds	20221292	mmlinget2j@amazon.co.jp
R_2022_093	Emme Jahan	msc biotech	20221293	cstoro2k@weebly.com
R_2022_094	Shurlock Kingsly	btech ic	20221294	gvignal2l@engadget.com
R_2022_095	Heracles Cypher	btech mne	20221295	rharrie2m@fema.gov
R_2022_096	Giles Eakins	btech biotech	20221296	aandraud2n@google.ca
R_2022_097	Franzen Defond	msc agriculture	20221297	gpolley2o@who.int
R_2022_098	Bastian Wickey	msc analytics	20221298	ypreto2p@umich.edu
R_2022_099	Muhammad Colin	msc mathemati...	20221299	bcundy2q@s8.net
R_2022_100	Rowena Andreotti	msc it	20221300	fenglieh2r@g.co

27. English query: Display the Member Names from the Management Committee of Literature event.

Query:

```
select member_name from college_fest_db.event
natural join college_fest_db.event_management_committee
where event.event_type = 'literature';
```



The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database schema with tables like college, coordinators, event, and event\_management\_committee. The event table has 5 columns: event\_name, event\_id, event\_description, event\_venue, and event\_type. The event\_management\_committee table also has 5 columns: member\_name, student\_email, member\_id, student\_id, and event\_id. The central pane shows the SQL query being run:

```
--Query 27
select member_name from college_fest_db.event
natural join college_fest_db.event_management_committee
where event.event_type = 'literature';
```

The results pane displays the member names:

member_name
Davita Batchelder
Herman Sevier
Louis Suthren
Justis Beaver
Jeane Casaccio
Amerigo Skelhorne
Bartolome Clipsham
Waverly BURWIN
Kesiah Parmiter
Shandria Faber
Melissa Belsey
Imojean Baldock

Total rows: 12 of 12 | Query complete 00:00:00.089 | Ln 7, Col 39

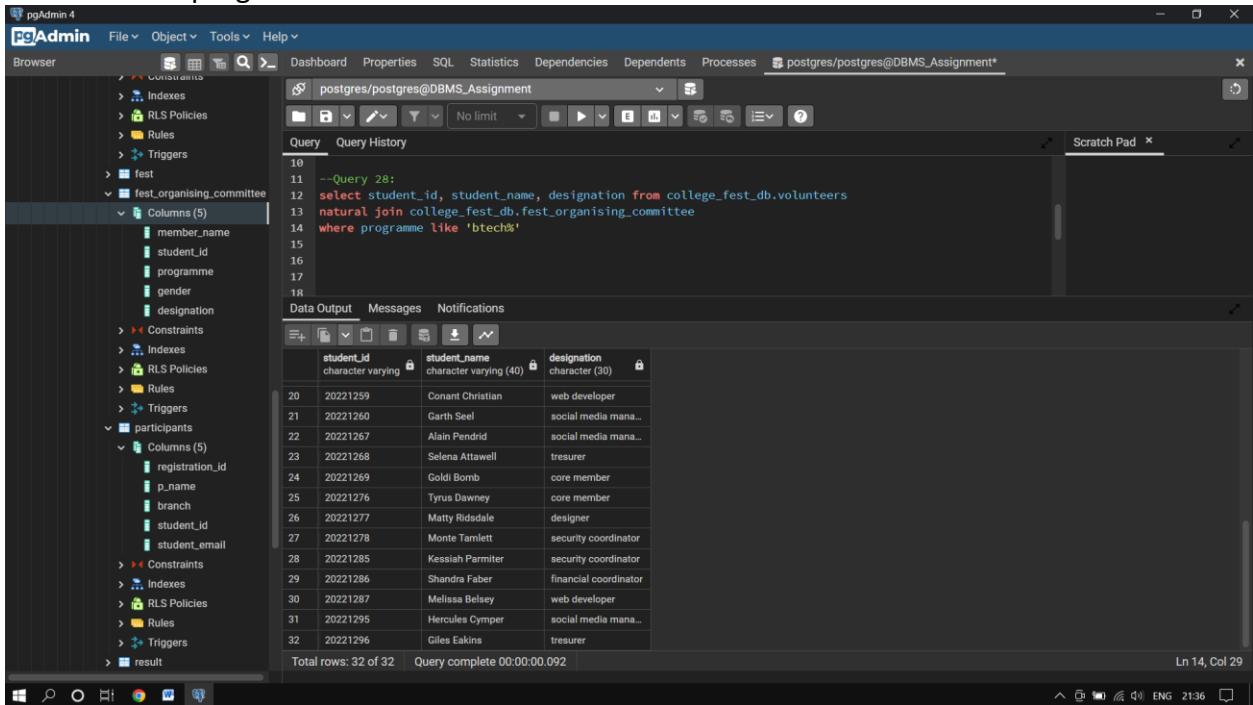
28. English query: Select the following details of Fest Organising Committee members

- Student ID
- Student Name
- Designation

Where the Programme is Btech.

Query:

```
select student_id, student_name, designation from college_fest_db.volunteers
natural join college_fest_db.fest_organising_committee
where programme like 'btech%'
```



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

- Browser:** The left sidebar shows the database structure for the 'fest' schema, including the 'fest\_organising\_committee' table with columns: member\_name, student\_id, programme, gender, and designation.
- Query Editor:** The main area contains the SQL query for selecting data from the volunteers and fest\_organising\_committee tables, filtered by programme like 'btech%'.
- Data Output:** The results table displays 32 rows of data, showing student\_id, student\_name, and designation. The data is as follows:

student_id	student_name	designation
20221259	Conant Christian	web developer
20221260	Garth Seel	social media mana...
20221267	Alain Pendrid	social media mana...
20221268	Selena Attawell	treasurer
20221269	Goldi Bomb	core member
20221276	Tyrus Dawney	core member
20221277	Matty Ridsdale	designer
20221278	Monte Tamlett	security coordinator
20221285	Kessiah Parmiter	security coordinator
20221286	Shandra Faber	financial coordinator
20221287	Melissa Belsey	web developer
20221295	Heracles Cyper	social media mana...
20221296	Giles Eakins	treasurer

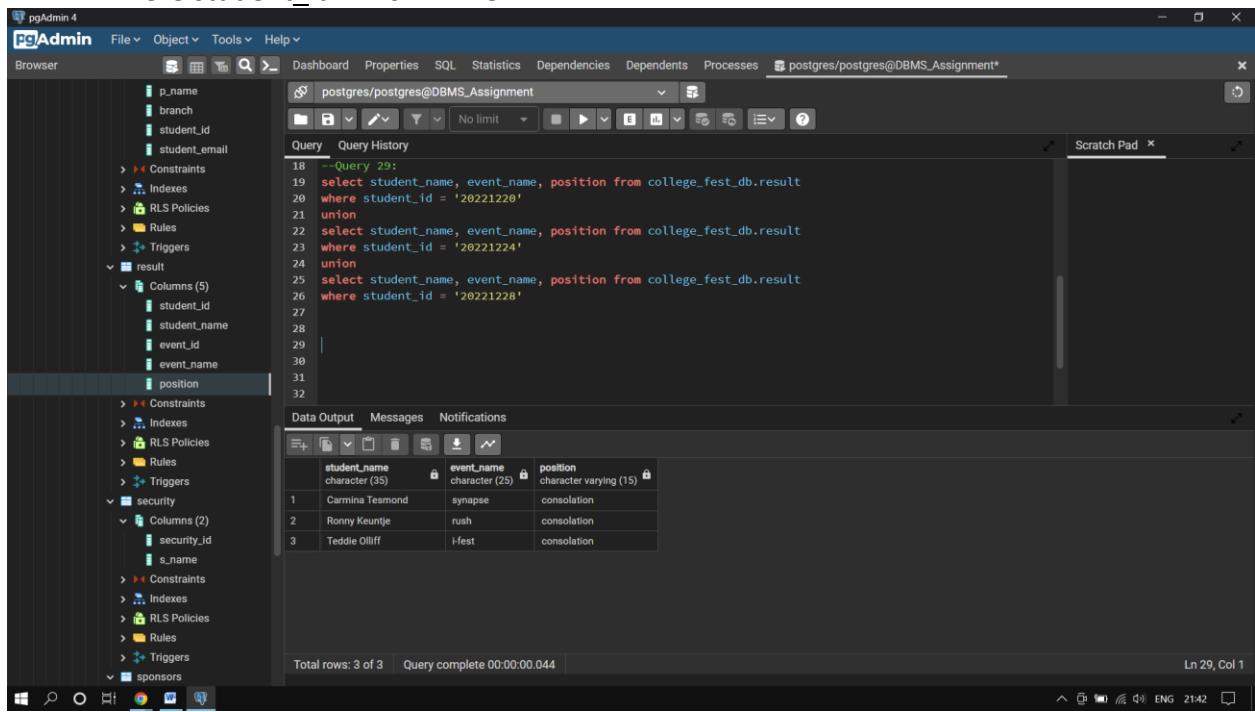
Total rows: 32 of 32 | Query complete 00:00:00.092 | Ln 14, Col 29

29. English query: Display the following details for Student IDs: 20221220, 20221224, 20221228

- a. Student Name
- b. Event Name
- c. Position

Query:

```
select student_name, event_name, position from college_fest_db.result
where student_id = '20221220'
union
select student_name, event_name, position from college_fest_db.result
where student_id = '20221224'
union
select student_name, event_name, position from college_fest_db.result
where student_id = '20221228'
```



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

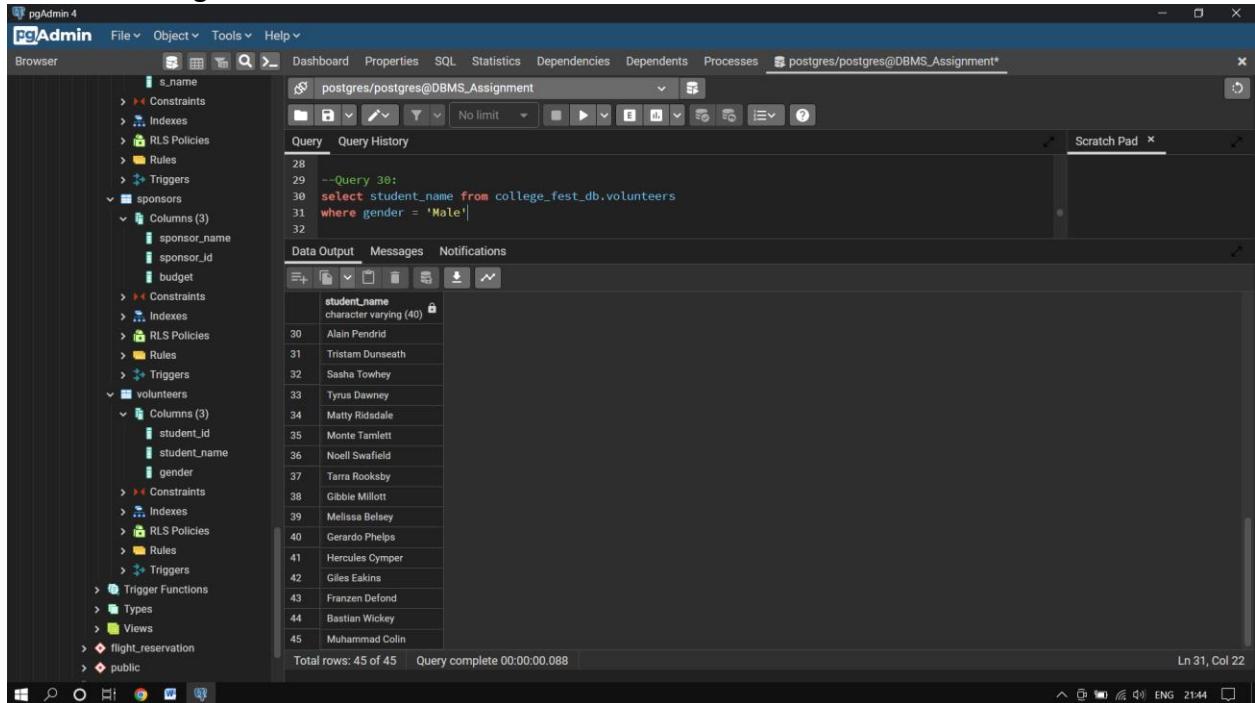
- Toolbar:** File, Object, Tools, Help.
- Browser:** Shows the schema structure of the database, including tables like p\_name, branch, student\_id, student\_email, result, security, and sponsors, along with their columns and constraints.
- Query Editor:** The query is pasted into the Query tab, showing the three SELECT statements with UNION operators.
- Data Output:** The results of the query are displayed in a table format.
- Table Data:**

	student_name	event_name	position
1	Carmina Tesmond	synapse	consolation
2	Ronny Keuntje	rush	consolation
3	Teddie Olliff	H-fest	consolation
- Message:** Total rows: 3 of 3 | Query complete 00:00:00.044 | Ln 29, Col 1

30. English query: Select names of Male volunteers.

Query:

```
select student_name from college_fest_db.volunteers
where gender = 'Male'
```



The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database schema with tables 'sponsors' and 'volunteers'. The 'volunteers' table has columns: student\_id, student\_name, gender. The main window shows a query editor with the following content:

```
28 --Query 30:
29
30 select student_name from college_fest_db.volunteers
31 where gender = 'Male'
32
```

The results pane displays the names of 45 male volunteers:

student_name
character varying (40)
Alain Pendrid
Tristam Dunseath
Sasha Towhey
Tyrus Dawney
Matty Riddale
Monte Tamlett
Noell Swafield
Tarra Rooksbys
Gibble Millott
Melissa Belsey
Gerardo Phelps
Hercules Cympre
Giles Eakins
Franzen Defond
Bastian Wickey
Muhammad Collin

At the bottom of the results pane, it says "Total rows: 45 of 45" and "Query complete 00:00:00.088".

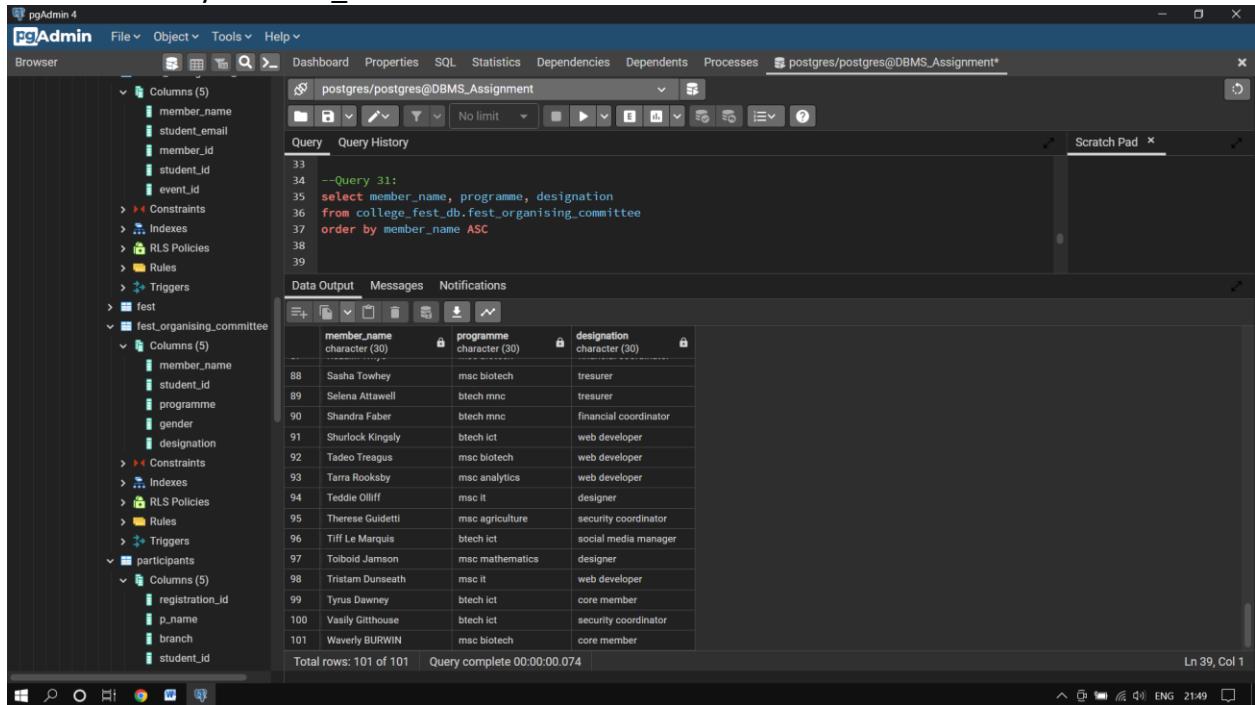
31. English query: Select the following details from the Fest Organising Committee

- Member name
- Programme
- Designation

And sort them in ascending order w.r.t member name.

Query:

```
select member_name, programme, designation
from college_fest_db.fest_organising_committee
order by member_name ASC
```



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

- Browser:** The left sidebar shows the database structure. Under the 'fest' schema, there is a table named 'fest\_organising\_committee' with columns: member\_name, programme, designation, registration\_id, p\_name, branch, and student\_id.
- Query Editor:** The main area contains the SQL query: 

```
33
34 --Query 31:
35 select member_name, programme, designation
36 from college_fest_db.fest_organising_committee
37 order by member_name ASC
38
39
```
- Data Output:** The results of the query are displayed in a table:

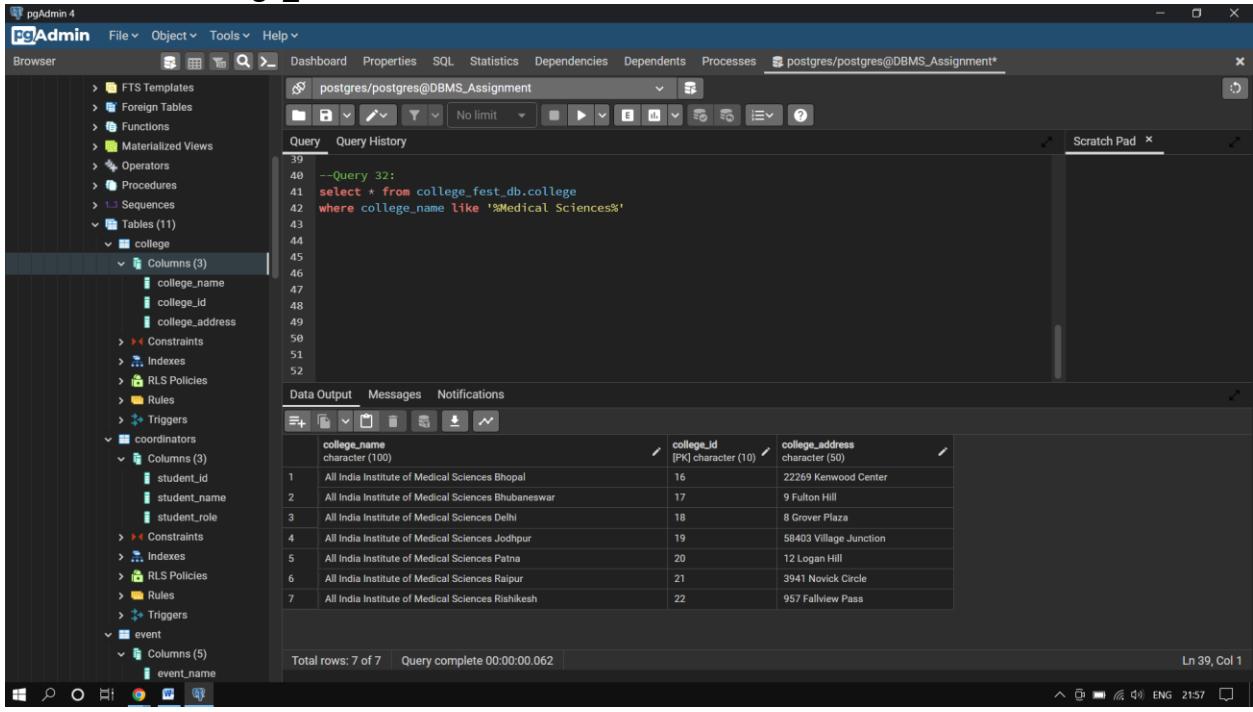
	member_name	programme	designation
88	Sasha Towhey	msc biotech	treasurer
89	Selena Attawell	btech mnc	treasurer
90	Shandra Faber	btech mnc	financial coordinator
91	Shurlock Kingsly	btech ict	web developer
92	Tadeo Treagus	msc biotech	web developer
93	Tara Rooksbay	msc analytics	web developer
94	Teddie Olliff	msc it	designer
95	Therese Guidetti	msc agriculture	security coordinator
96	Tiff Le Marquis	btech ict	social media manager
97	Tolbold Jamson	msc mathematics	designer
98	Tristan Dunseath	msc it	web developer
99	Tyrus Dawney	btech ict	core member
100	Vasily Githhouse	btech ict	security coordinator
101	Waverly BURWIN	msc biotech	core member

Total rows: 101 of 101 | Query complete 00:00:00.074 | Ln 39, Col 1

32. English query: Display all the participating Medical Sciences colleges.

Query:

```
select * from college_fest_db.college
where college_name like '%Medical Sciences%'
```



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

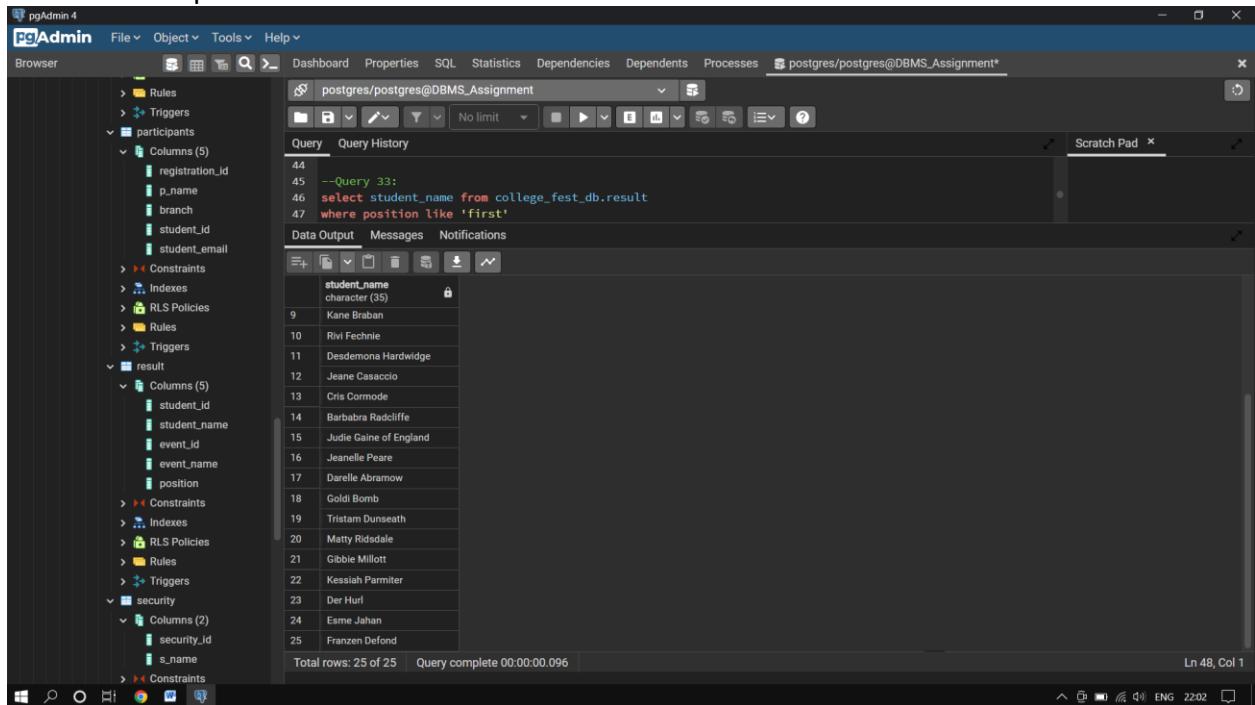
- Browser:** Shows the database structure for `college_fest_db`, specifically the `college` table which contains 3 columns: `college_name`, `college_id`, and `college_address`.
- Query Editor:** The query `select * from college_fest_db.college where college_name like '%Medical Sciences%'` is entered in the Query tab.
- Data Output:** The results are displayed in a table with 7 rows, showing the college name, ID, and address.
- Messages:** The message "Query complete 00:00:00.062" is shown at the bottom.
- System:** The bottom right shows system status: Ln 39, Col 1, ENG 2157.

	college_name	college_id	college_address
1	All India Institute of Medical Sciences Bhopal	16	22269 Kenwood Center
2	All India Institute of Medical Sciences Bhubaneswar	17	9 Fulton Hill
3	All India Institute of Medical Sciences Delhi	18	8 Grover Plaza
4	All India Institute of Medical Sciences Jodhpur	19	58402 Village Junction
5	All India Institute of Medical Sciences Patna	20	12 Logan Hill
6	All India Institute of Medical Sciences Raipur	21	3941 Novick Circle
7	All India Institute of Medical Sciences Rishikesh	22	957 Fallview Pass

33. English query: Display the names of all the students who secured first position

Query:

```
select student_name from college_fest_db.result  
where position like 'first'
```



The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure for the 'college\_fest\_db' schema, including tables like 'participants', 'result', and 'security'. The 'result' table is currently selected. The main pane shows the query results for the following SQL statement:

```
--Query 33:  
select student_name from college_fest_db.result  
where position like 'first'
```

The results table displays 25 rows of student names, each preceded by a row number. The names listed are:

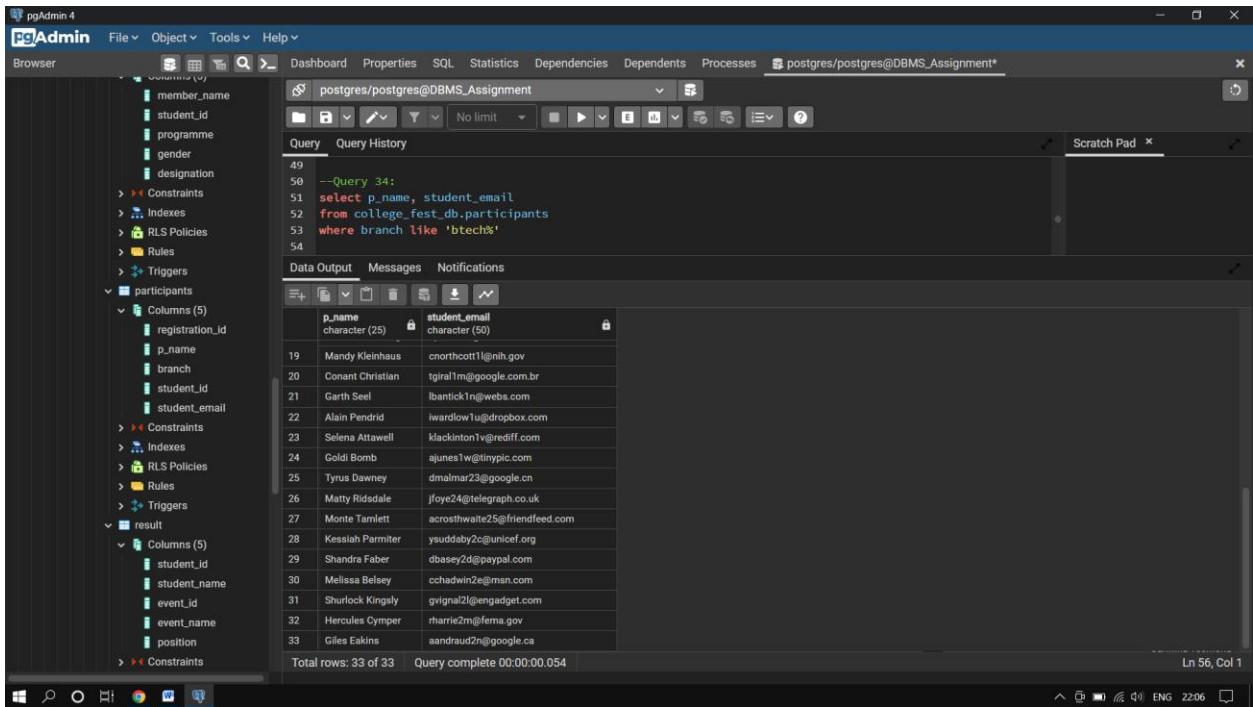
student_name
Kane Braban
Rivi Fecchine
Desdemona Hardwidge
Jeane Casaccio
Cris Cormode
Barbara Radcliffe
Judie Gaine of England
Jeanelle Pearn
Darelle Abramow
Goldi Bombs
Tristam Dunseath
Matty Riddale
Gibbie Millott
Kessiah Parmiter
Der Hurl
Esme Jahan
Franzen Defond

At the bottom of the results pane, it says 'Total rows: 25 of 25' and 'Query complete 00:00:00.096'.

34. English query: Display the name and email address of participants from Btech

Query:

```
select p_name, student_email
from college_fest_db.participants
where branch like 'btech%'
```



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

- File Bar:** File, Object, Tools, Help.
- Toolbar:** Contains icons for New, Open, Save, Copy, Paste, Find, and others.
- Connections:** Postgres/postgres@DBMS\_Assignment
- Browser:** Shows the schema structure:
  - College\_fest\_db (parent)
  - member\_name
  - student\_id
  - programme
  - gender
  - designation
  - Participants (parent)
    - Constraints
    - Indexes
    - RLS Policies
    - Rules
    - Triggers
    - Columns (5)
      - registration\_id
      - p\_name
      - branch
      - student\_id
      - student\_email
    - Constraints
    - Indexes
    - RLS Policies
    - Rules
    - Triggers
  - result (parent)
    - Columns (5)
      - student\_id
      - student\_name
      - event\_id
      - event\_name
      - position
    - Constraints
- Query Editor:** Contains the query text:

```
--Query 34:
select p_name, student_email
from college_fest_db.participants
where branch like 'btech%'
```

- Data Output:** Shows the results of the query in a table format:

	p_name	student_email
19	Mandy Kleinhuis	cnorthcott1@nih.gov
20	Conant Christian	tgirall1mp@google.com.br
21	Garth Seel	lbantick1n@webs.com
22	Alain Pendrid	lwardlow1u@dropbox.com
23	Selena Attawell	slackinton1v@rediff.com
24	Goldi Bomb	ajunes1w@tinyPic.com
25	Tyus Dawney	dmalmar23@google.cn
26	Matty Ridsdale	jfoye24@telegraph.co.uk
27	Monte Tamlett	acrostowalte25@friendfeed.com
28	Kessiah Parmiter	ysuddaby2c@unicef.org
29	Shandria Fabre	dbasey2d@paypal.com
30	Melissa Belsey	cchadwin2e@msn.com
31	Shurlock Kingaly	gvignal2f@engadget.com
32	Hercules Cypher	rharris2m@fema.gov
33	Giles Eakins	aandrau2n@google.ca

- Message Bar:** Total rows: 33 of 33, Query complete 00:00:00.054, Ln 56, Col 1.

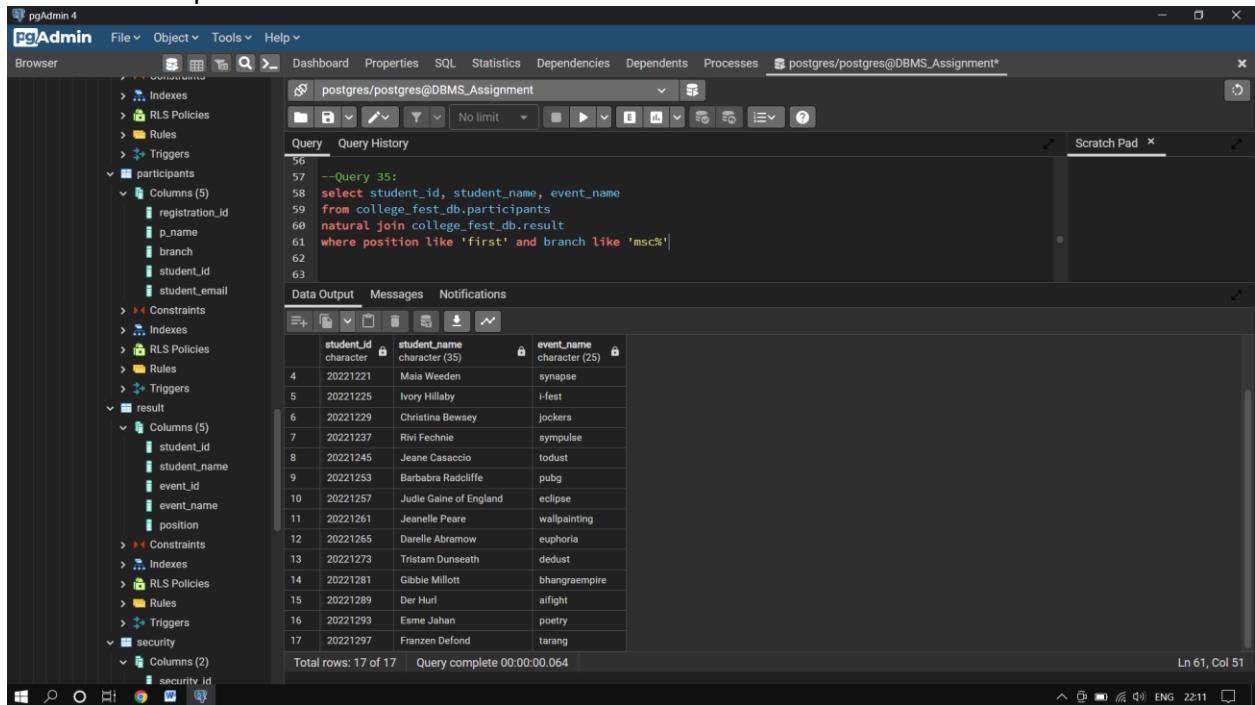
35. English query: Select the following details

- a. Student ID
- b. Student name
- c. Event name

Who are from MSc branch and secured first position

Query:

```
select student_id, student_name, event_name
from college_fest_db.participants
natural join college_fest_db.result
where position like 'first' and branch like 'msc%'
```



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

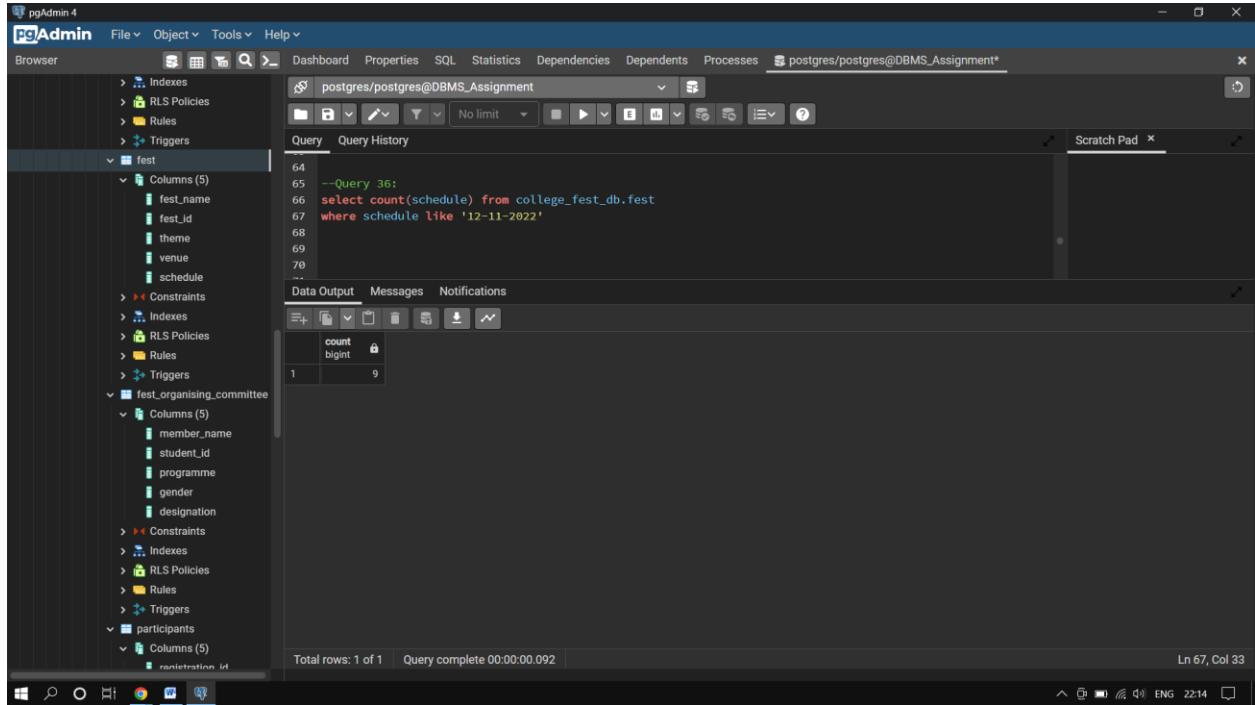
- Browser:** Shows the database schema structure for the 'college\_fest\_db' including 'participants' and 'result' tables.
- Query:** The SQL query is displayed in the query editor.
- Data Output:** The results of the query are shown in a table format.
- Table Headers:** student\_id, student\_name, event\_name
- Table Data:** 17 rows of data are listed, showing student IDs, names, and event names.
- Bottom Status:** Total rows: 17 of 17, Query complete 00:00:00.064, Ln 61, Col 51.

	student_id	student_name	event_name
4	20221221	Maia Weden	synapse
5	20221225	Ivory Hillaby	i-fest
6	20221229	Christina Bewsey	jokers
7	20221237	Rivi Fechnie	synpulse
8	20221245	Jeane Casaccio	todust
9	20221253	Barbabra Radcliffe	pubg
10	20221257	Judie Galine of England	eclipse
11	20221261	Jeanelle Pearse	wallpainting
12	20221265	Darelle Abramow	euphoria
13	20221273	Tristam Dunseath	dedust
14	20221281	Gibble Millott	bhangraempire
15	20221289	Der Hurl	alifight
16	20221293	Esme Jahan	poetry
17	20221297	Franzen Defond	tarang

36. English query: Display the total number of events hosted on date 12-11-2022

Query:

```
select count(schedule) from college_fest_db.fest
where schedule like '12-11-2022'
```



The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure with tables like 'fest' and 'fest\_organising\_committee'. The main query editor window contains the following SQL code:

```
--Query 36:
select count(schedule) from college_fest_db.fest
where schedule like '12-11-2022'
```

The results pane shows a single row of data:

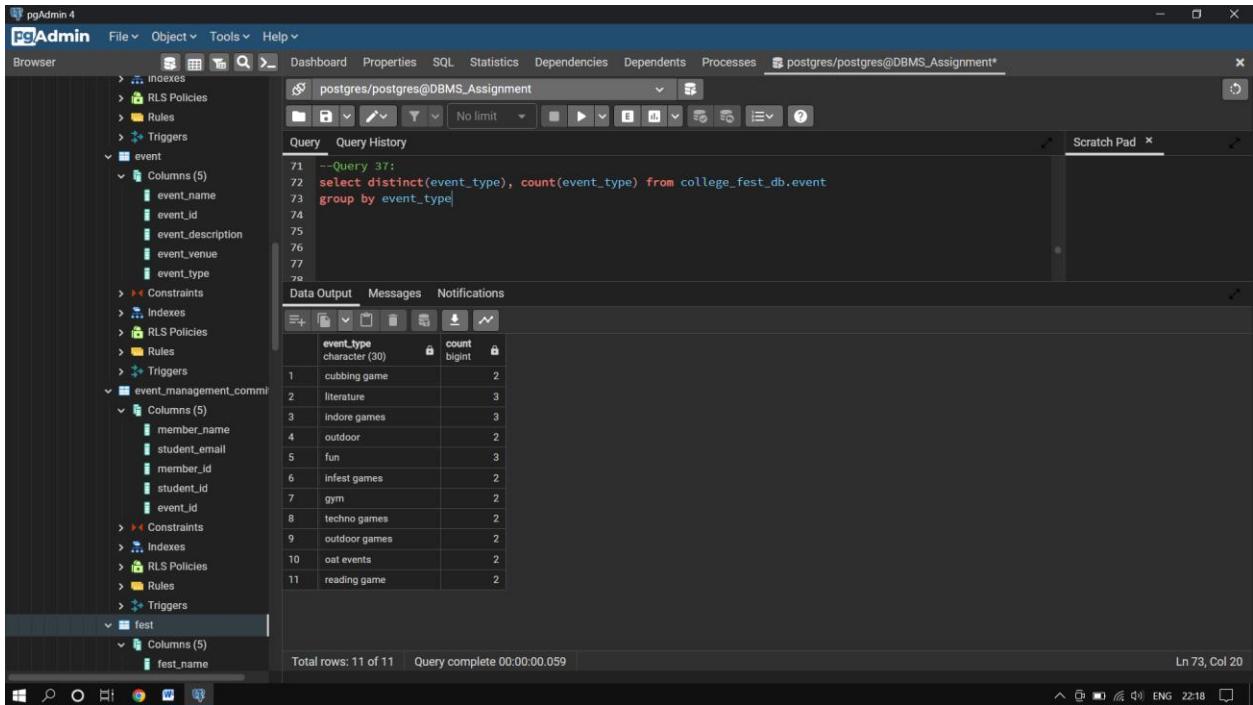
count	bigint
1	9

Below the results, the status bar indicates "Total rows: 1 of 1" and "Query complete 00:00:00.092".

37. English query: Display the number of each event type.

Query:

```
select distinct(event_type), count(event_type) from college_fest_db.event
group by event_type
```



The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure for the 'college\_fest\_db' schema, including tables like 'event', 'event\_management\_commission', and 'fest'. The main window shows a query editor with the following SQL code:

```
--Query 37:
select distinct(event_type), count(event_type) from college_fest_db.event
group by event_type
```

The 'Data Output' tab is selected, showing the results of the query:

event_type	count
cubbing game	2
literature	3
indore games	3
outdoor	2
fun	3
Infest games	2
gym	2
techno games	2
outdoor games	2
oat events	2
reading game	2

Total rows: 11 of 11 | Query complete 00:00:00.059 | Ln 73, Col 20

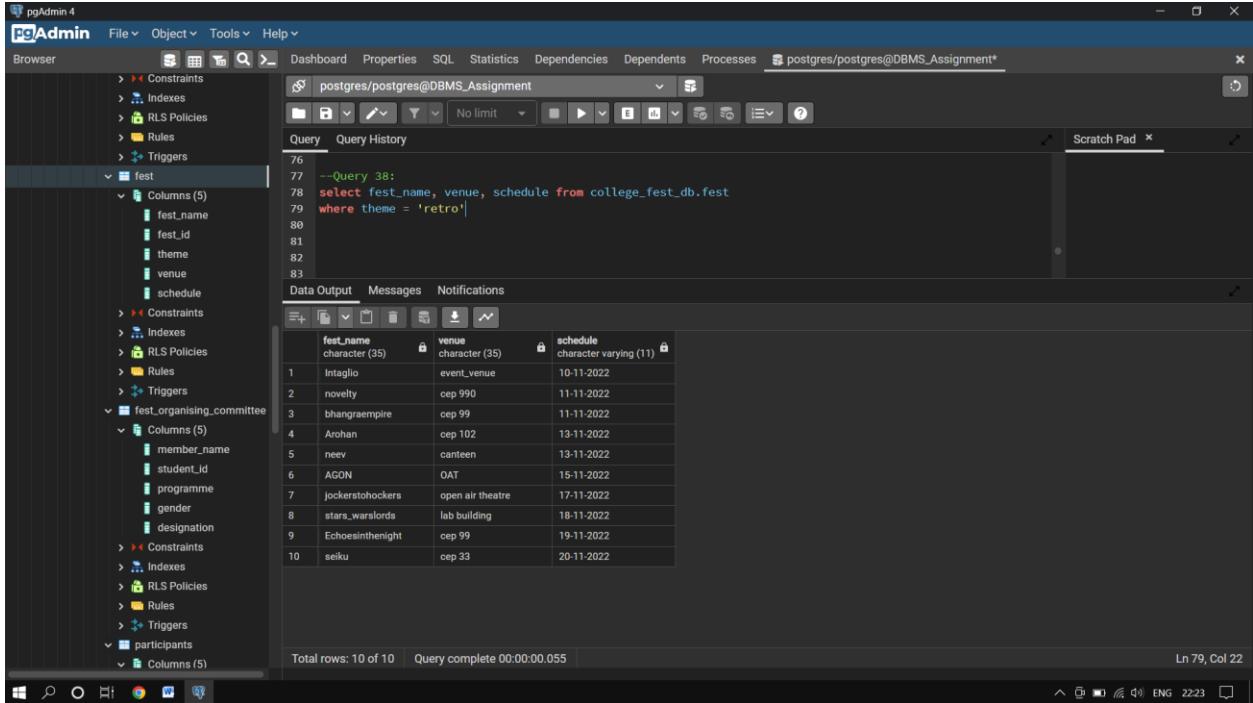
38. English query: Select the following details

- Fest name
- Venue
- Schedule

Where fest theme is Retro.

Query:

```
select fest_name, venue, schedule from college_fest_db.fest
where theme = 'retro'
```



The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure for the 'college\_fest\_db' schema, including the 'fest' table and its columns: fest\_name, fest\_id, theme, venue, and schedule. The 'fest' table is currently selected. The main query editor window contains the following SQL code:

```
--Query 38:
select fest_name, venue, schedule from college_fest_db.fest
where theme = 'retro'
```

The results pane shows a table with 10 rows of data, corresponding to the 10 entries in the 'fest' table where the theme is 'retro'. The columns are fest\_name, venue, and schedule. The data is as follows:

	fest_name	venue	schedule
1	Intaglio	event_venue	10-11-2022
2	novelty	cep 990	11-11-2022
3	bhangraempire	cep 99	11-11-2022
4	Arohan	cep 102	13-11-2022
5	neev	canteen	13-11-2022
6	AGON	OAT	15-11-2022
7	jockerstohockers	open air theatre	17-11-2022
8	stars_warslords	lab building	18-11-2022
9	Echoesintheneight	cep 99	19-11-2022
10	seiku	cep 33	20-11-2022

Total rows: 10 of 10 | Query complete 00:00:00.055 | Ln 79, Col 22

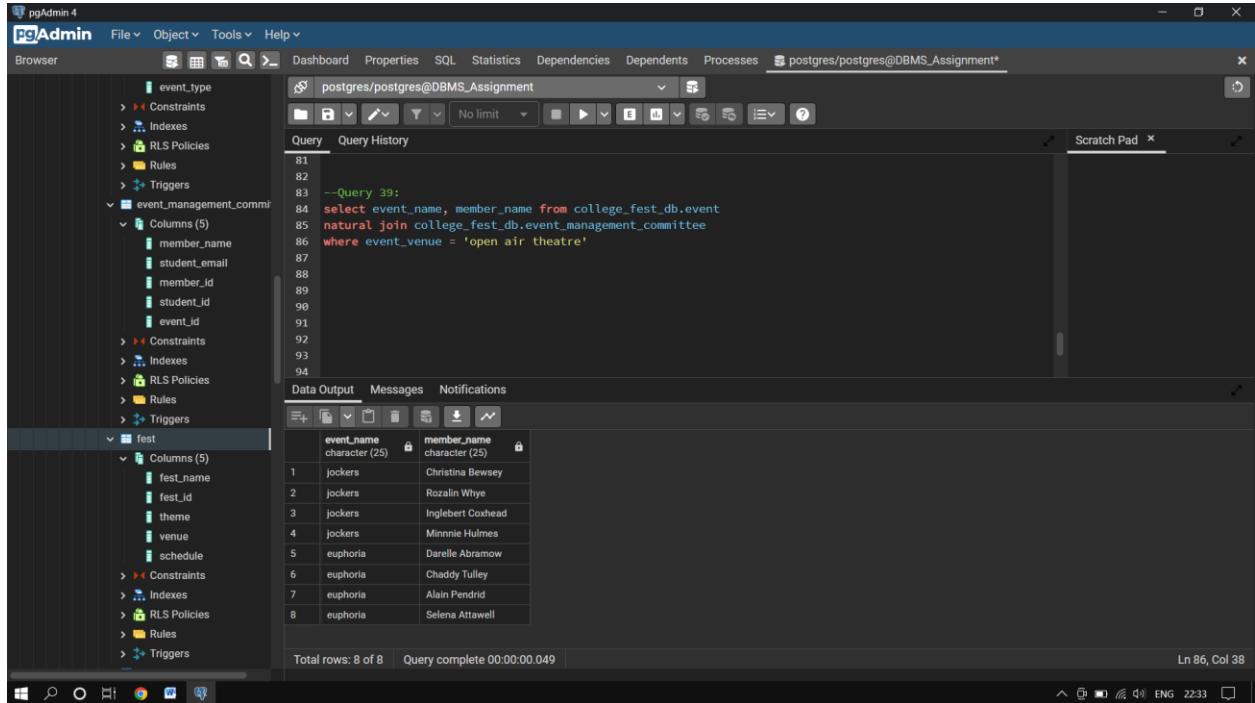
39. English query: Select the following details

- a. Event name
- b. Member name

For events that were hosted on Open Air Theatre

Query:

```
select event_name, member_name from college_fest_db.event
natural join college_fest_db.event_management_committee
where event_venue = 'open air theatre';
```



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

- Browser:** The left sidebar shows the database structure for the 'college\_fest\_db'. It includes the 'event\_type' table, the 'event\_management\_committee' table (which has 'member\_name', 'student\_email', 'member\_id', 'student\_id', and 'event\_id' columns), and the 'fest' table (which has 'fest\_name', 'fest\_id', 'theme', 'venue', and 'schedule' columns).
- Query Editor:** The main area shows the SQL query for question 39. The results are displayed in a table titled 'Data Output'.
- Data Output:** The table contains 8 rows of data, showing the event name and member name for each event.

	event_name	member_name
1	jokers	Christina Beussey
2	jokers	Rozalin Whye
3	jokers	Inglebert Coxhead
4	jokers	Minnie Hulmes
5	euphoria	Darrelle Abramow
6	euphoria	Chaddy Tully
7	euphoria	Alain Pendrid
8	euphoria	Selena Attawell

- Messages:** The message bar at the bottom indicates 'Query complete 00:00:00.049'.
- System:** The bottom right corner shows the system status: 'Ln 86, Col 38'.

40. English query: Display the following details

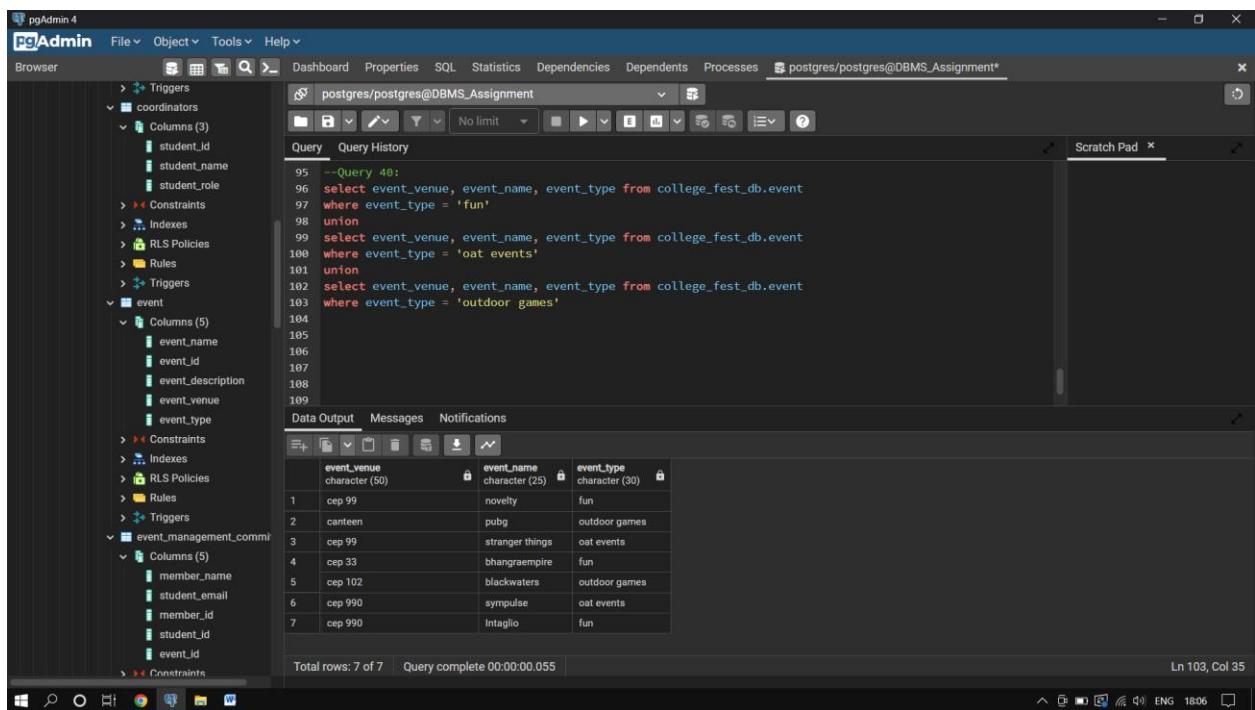
- Event venue
- Event name
- Event type

Of event that are of type

- Fun
- OAT Event
- Outdoor games

Query:

```
select event_venue, event_name, event_type from college_fest_db.event
where event_type = 'fun'
union
select event_venue, event_name, event_type from college_fest_db.event
where event_type = 'oat events'
union
select event_venue, event_name, event_type from college_fest_db.event
where event_type = 'outdoor games'
```



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

- Browser:** The left sidebar displays the database schema structure, including tables like coordinators, event, and event\_management\_com.
- Query Editor:** The main window shows the SQL query with line numbers 95 to 109.
- Data Output:** The bottom pane displays the results of the query, showing 7 rows of data:

	event_venue	event_name	event_type
1	cep 99	novelty	fun
2	canteen	pubg	outdoor games
3	cep 99	stranger things	oat events
4	cep 33	bhangraempire	fun
5	cep 102	blackwaters	outdoor games
6	cep 990	sympulse	oat events
7	cep 990	Intaglio	fun