

PostgreSQL Lab2

Mazen Abdeltawab Saad

Track Python Fayoum

```
su - postgres
psql
create database postgres_lab2 TEMPLATE postgres_lab1;
\l
\c postgres_lab2
```

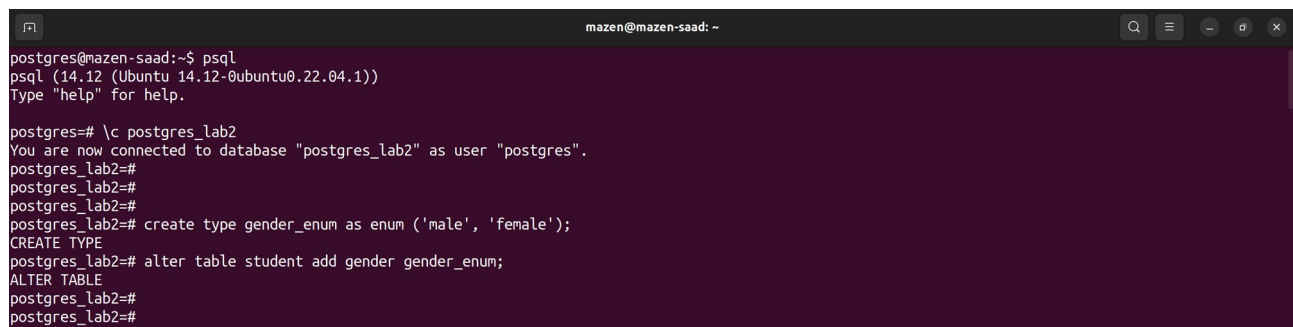
1. Add gender column for the student table[Enum]. It holds two value (male or female).

(solution 1)

```
create type gender_enum as enum ('male', 'female');
alter table student add gender gender_enum;
```

(solution 2)

```
alter table student add column gender enum('male', 'female');
```

A terminal window with a dark background and light text. The window title is 'mazen@mazen-saad: ~'. The terminal shows the following sequence of commands and output:

```
postgres@mazen-saad:~$ psql
psql (14.12 (Ubuntu 14.12-0ubuntu0.22.04.1))
Type "help" for help.

postgres=# \c postgres_lab2
You are now connected to database "postgres_lab2" as user "postgres".
postgres_lab2=#
postgres_lab2=#
postgres_lab2=#
postgres_lab2=# create type gender_enum as enum ('male', 'female');
CREATE TYPE
postgres_lab2=# alter table student add gender gender_enum;
ALTER TABLE
postgres_lab2=#
postgres_lab2=#
```

2. Add birth date column for the student table.

(solution)

```
alter table student add column birth_date date;
```

A terminal window with a dark background and light text. The window title is 'mazen@mazen-saad: ~'. The terminal shows a series of prompts 'postgres_lab2=#' and the command 'alter table student add column birth_date date;'. The command is followed by the output 'ALTER TABLE' on the next line.

```
mazen@mazen-saad: ~  
postgres_lab2=#  
postgres_lab2=#  
postgres_lab2=#  
postgres_lab2=#  
postgres_lab2=# alter table student add column birth_date date;  
ALTER TABLE  
postgres_lab2=#  
postgres_lab2=#
```

3. Delete the name column and replace it with two columns first name and last name.

(solution)

```
alter table student drop column name;  
alter table student add column first_name text;  
alter table student add column last_name text;
```

A terminal window with a dark purple background and white text. The window title is 'mazen@mazen-saad: ~'. The terminal shows a series of SQL commands being entered and executed in a PostgreSQL environment. The commands are: 'postgres_lab2=#', 'postgres_lab2=#', 'postgres_lab2=# alter table student drop column name;', 'ALTER TABLE', 'postgres_lab2=# alter table student add column first_name text;', 'ALTER TABLE', 'postgres_lab2=# alter table student add column last_name text;', 'ALTER TABLE', 'postgres_lab2=#', and 'postgres_lab2=#'.

```
mazen@mazen-saad: ~  
postgres_lab2=#  
postgres_lab2=#  
postgres_lab2=# alter table student drop column name;  
ALTER TABLE  
postgres_lab2=# alter table student add column first_name text;  
ALTER TABLE  
postgres_lab2=# alter table student add column last_name text;  
ALTER TABLE  
postgres_lab2=#  
postgres_lab2=#
```

4. Delete the address and email column and replace it with contact info (Address, email) as object/Composite Data type.

(solution)

```
alter table student drop column address;  
alter table student drop column email;
```

```
create type contact_info_obj as (email text, address text);  
alter table student add column contact_info contact_info_obj;
```

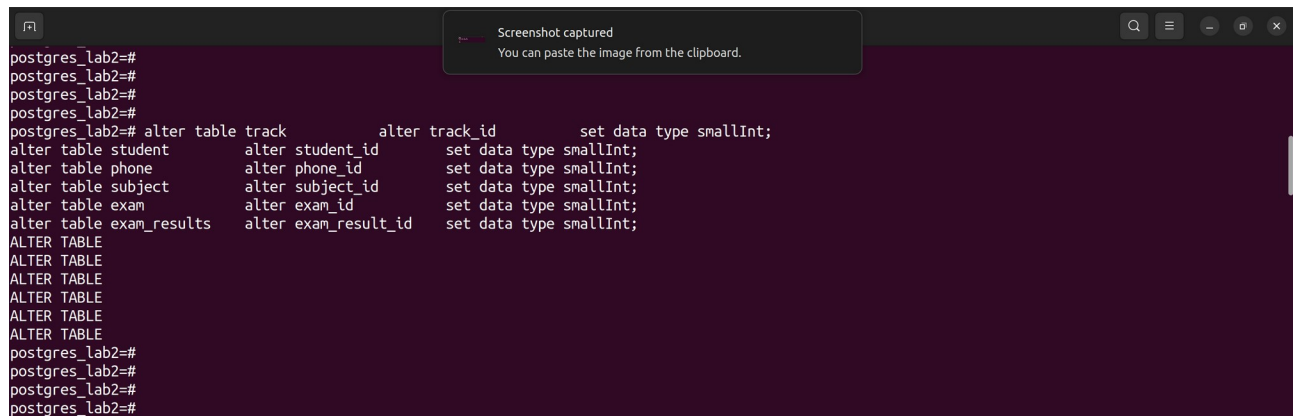
A terminal window with a dark background and light text. The window title is 'mazen@mazen-saad: ~'. The terminal shows a series of SQL commands being executed in a PostgreSQL environment. The commands are: 'postgres_lab2=#', 'postgres_lab2=#', 'postgres_lab2=# alter table student drop column address;', 'ALTER TABLE', 'postgres_lab2=# alter table student drop column email;', 'ALTER TABLE', 'postgres_lab2=# create type contact_info_obj as (email text, address text);', 'CREATE TYPE', 'postgres_lab2=# alter table student add column contact_info contact_info_obj;', 'ALTER TABLE', 'postgres_lab2=#', 'postgres_lab2=#', and 'postgres_lab2=#'.

```
mazen@mazen-saad: ~  
postgres_lab2=#  
postgres_lab2=#  
postgres_lab2=# alter table student drop column address;  
ALTER TABLE  
postgres_lab2=# alter table student drop column email;  
ALTER TABLE  
postgres_lab2=# create type contact_info_obj as (email text, address text);  
CREATE TYPE  
postgres_lab2=# alter table student add column contact_info contact_info_obj;  
ALTER TABLE  
postgres_lab2=#  
postgres_lab2=#  
postgres_lab2=#
```

5. Change any Serial Datatype at your tables to smallInt

(solution)

```
alter table track          alter track_id          set data type smallInt;
alter table student        alter student_id         set data type smallInt;
alter table phone          alter phone_id           set data type smallInt;
alter table subject        alter subject_id         set data type smallInt;
alter table exam           alter exam_id            set data type smallInt;
alter table exam_results   alter exam_result_id     set data type
smallInt;
```



The screenshot shows a terminal window with a dark background. At the top, there is a notification box that says "Screenshot captured" and "You can paste the image from the clipboard." Below this, the terminal displays a series of SQL commands. The first five lines are prompts from a PostgreSQL shell: postgres_lab2=#. The next five lines are the SQL commands to alter the serial IDs of the tables track, student, phone, subject, and exam to smallInt. These are followed by five lines of ALTER TABLE commands for the same tables. The terminal output shows the commands being executed, with some lines appearing to be repeated or truncated.

```
postgres_lab2=#
postgres_lab2=#
postgres_lab2=#
postgres_lab2=#
postgres_lab2=# alter table track          alter track_id          set data type smallInt;
alter table student        alter student_id         set data type smallInt;
alter table phone          alter phone_id           set data type smallInt;
alter table subject        alter subject_id         set data type smallInt;
alter table exam           alter exam_id            set data type smallInt;
alter table exam_results   alter exam_result_id     set data type smallInt;
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
ALTER TABLE
postgres_lab2=#
postgres_lab2=#
postgres_lab2=#
postgres_lab2=#
```

6. Add/Alter foreign key constraints in Your Tables.

(solution) s42

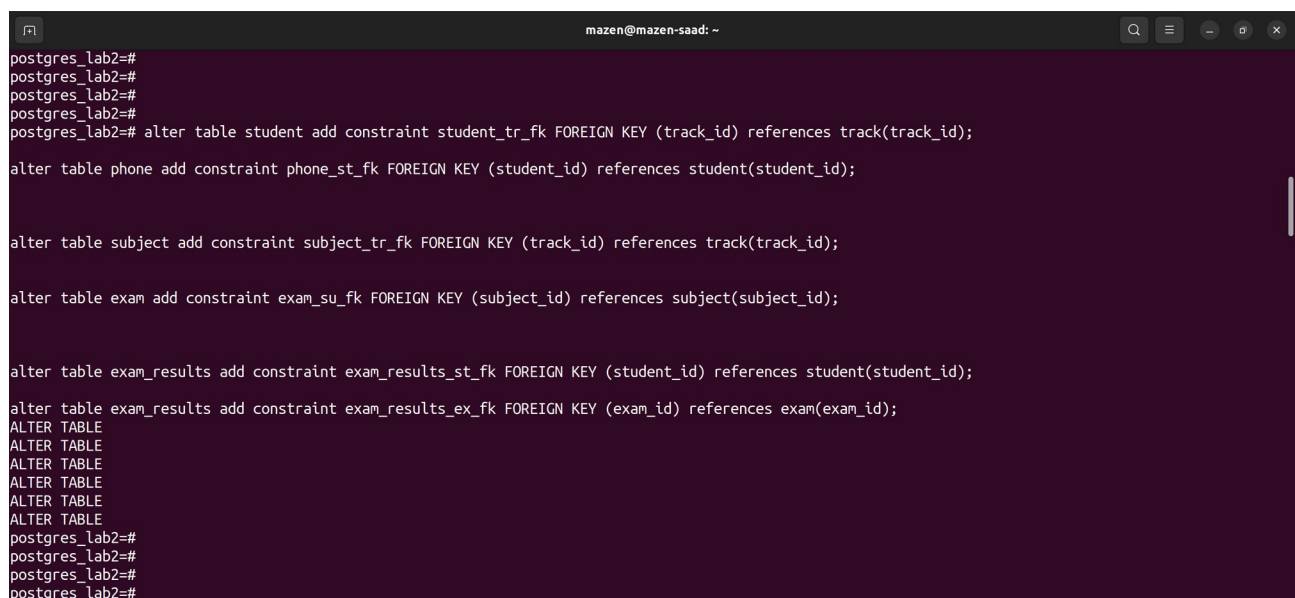
```
alter table student add constraint student_tr_fk FOREIGN KEY  
(track_id) references track(track_id);  
alter table phone add constraint phone_st_fk FOREIGN KEY  
(student_id) references student(student_id);
```

```
alter table subject add constraint subject_tr_fk FOREIGN KEY  
(track_id) references track(track_id);
```

```
alter table exam add constraint exam_su_fk FOREIGN KEY  
(subject_id) references subject(subject_id);
```

```
alter table exam_results add constraint exam_results_st_fk  
FOREIGN KEY (student_id) references student(student_id);
```

```
alter table exam_results add constraint exam_results_ex_fk  
FOREIGN KEY (exam_id) references exam(exam_id);
```

A terminal window with a dark purple background and white text. The window title is 'mazen@mazen-saad: ~'. It contains a series of SQL commands for adding foreign key constraints to a PostgreSQL database. The commands are: 'alter table student add constraint student_tr_fk FOREIGN KEY (track_id) references track(track_id);', 'alter table phone add constraint phone_st_fk FOREIGN KEY (student_id) references student(student_id);', 'alter table subject add constraint subject_tr_fk FOREIGN KEY (track_id) references track(track_id);', 'alter table exam add constraint exam_su_fk FOREIGN KEY (subject_id) references subject(subject_id);', 'alter table exam_results add constraint exam_results_st_fk FOREIGN KEY (student_id) references student(student_id);', and 'alter table exam_results add constraint exam_results_ex_fk FOREIGN KEY (exam_id) references exam(exam_id);'. There are also several 'ALTER TABLE' statements and 'postgres_lab2=#' prompts.

```
mazen@mazen-saad: ~  
postgres_lab2=#  
postgres_lab2=#  
postgres_lab2=#  
postgres_lab2=# alter table student add constraint student_tr_fk FOREIGN KEY (track_id) references track(track_id);  
  
alter table phone add constraint phone_st_fk FOREIGN KEY (student_id) references student(student_id);  
  
alter table subject add constraint subject_tr_fk FOREIGN KEY (track_id) references track(track_id);  
  
alter table exam add constraint exam_su_fk FOREIGN KEY (subject_id) references subject(subject_id);  
  
alter table exam_results add constraint exam_results_st_fk FOREIGN KEY (student_id) references student(student_id);  
  
alter table exam_results add constraint exam_results_ex_fk FOREIGN KEY (exam_id) references exam(exam_id);  
ALTER TABLE  
ALTER TABLE  
ALTER TABLE  
ALTER TABLE  
ALTER TABLE  
ALTER TABLE  
postgres_lab2=#  
postgres_lab2=#  
postgres_lab2=#  
postgres_lab2=#
```

7. Insert new data in all Tables.

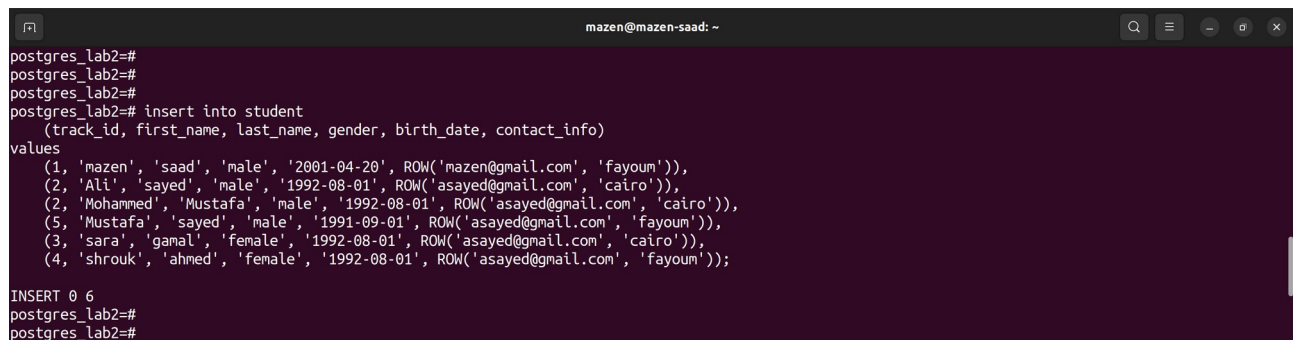
(solution)

insert into student

(track_id, first_name, last_name, gender, birth_date,
contact_info)

values

```
(1, 'mazen', 'saad', 'male', '2001-04-20',  
ROW('mazen@gmail.com', 'fayoum')),  
(2, 'Ali', 'sayed', 'male', '1992-08-01', ROW('asayed@gmail.com',  
'cairo')),  
(2, 'Mohammed', 'Mustafa', 'male', '1992-08-01',  
ROW('asayed@gmail.com', 'cairo')),  
(5, 'Mustafa', 'sayed', 'male', '1991-09-01',  
ROW('asayed@gmail.com', 'fayoum')),  
(3, 'sara', 'gamal', 'female', '1992-08-01',  
ROW('asayed@gmail.com', 'cairo')),  
(4, 'shrouk', 'ahmed', 'female', '1992-08-01',  
ROW('asayed@gmail.com', 'fayoum'));
```



The screenshot shows a terminal window with a dark background. The title bar at the top reads "mazen@mazen-saad: ~". The terminal content shows the execution of an SQL insert statement into a table named 'student'. The statement is preceded by a prompt "postgres_lab2=#". The output of the command is "INSERT 0 6", indicating that 6 rows were inserted. The prompt then changes to "postgres_lab2=#" again.

```
mazen@mazen-saad: ~  
postgres_lab2=#  
postgres_lab2=#  
postgres_lab2=# insert into student  
(track_id, first_name, last_name, gender, birth_date, contact_info)  
values  
(1, 'mazen', 'saad', 'male', '2001-04-20', ROW('mazen@gmail.com', 'fayoum')),  
(2, 'Ali', 'sayed', 'male', '1992-08-01', ROW('asayed@gmail.com', 'cairo')),  
(2, 'Mohammed', 'Mustafa', 'male', '1992-08-01', ROW('asayed@gmail.com', 'cairo')),  
(5, 'Mustafa', 'sayed', 'male', '1991-09-01', ROW('asayed@gmail.com', 'fayoum')),  
(3, 'sara', 'gamal', 'female', '1992-08-01', ROW('asayed@gmail.com', 'cairo')),  
(4, 'shrouk', 'ahmed', 'female', '1992-08-01', ROW('asayed@gmail.com', 'fayoum'));  
  
INSERT 0 6  
postgres_lab2=#  
postgres_lab2=#
```


8. Display all students' information.

(solution)

```
select * from student;
```

```
mazen@mazen-saad: ~  
postgres_lab2=#  
postgres_lab2=#  
postgres_lab2=#  
postgres_lab2=# select * from student;  
 student_id | track_id | gender | birth_date | first_name | last_name | contact_info  
-----  
          1 |         1 |        |            |            |            |  
          2 |         2 |        |            |            |            |  
          3 |         3 |        |            |            |            |  
          4 |         4 |        |            |            |            |  
          5 |         5 |        |            |            |            |  
          6 |         1 | male   | 2001-04-20 | mazen     | saad      | (mazen@gmail.com,fayoum)  
          7 |         2 | male   | 1992-08-01 | Ali       | sayed     | (asayed@gmail.com,cairo)  
          8 |         2 | male   | 1992-08-01 | Mohammed  | Mustafa   | (asayed@gmail.com,cairo)  
          9 |         5 | male   | 1991-09-01 | Mustafa   | sayed     | (asayed@gmail.com,fayoum)  
         10 |         3 | female | 1992-08-01 | sara      | gamal     | (asayed@gmail.com,cairo)  
         11 |         4 | female | 1992-08-01 | shrouk    | ahmed     | (asayed@gmail.com,fayoum)  
(11 rows)  
  
postgres_lab2=#  
postgres_lab2=#  
postgres_lab2=#
```

9. Display male students only.

(solution)

```
select * from student where gender = 'male';
```

```
mazen@mazen-saad: ~  
postgres_lab2=#  
postgres_lab2=#  
postgres_lab2=#  
postgres_lab2=# select * from student where gender = 'male';  
 student_id | track_id | gender | birth_date | first_name | last_name | contact_info  
-----  
          6 |         1 | male   | 2001-04-20 | mazen      | saad      | (mazen@gmail.com,fayoum)  
          7 |         2 | male   | 1992-08-01 | Ali        | sayed     | (asayed@gmail.com,cairo)  
          8 |         2 | male   | 1992-08-01 | Mohammed   | Mustafa   | (asayed@gmail.com,cairo)  
          9 |         5 | male   | 1991-09-01 | Mustafa    | sayed     | (asayed@gmail.com,fayoum)  
(4 rows)  
  
postgres_lab2=#  
postgres_lab2=#
```

10. Display the number of female students.

(solution)

```
select count(*) from student where gender = 'female';
```

A terminal window with a dark background and light text. The window title is 'mazen@mazen-saad: ~'. The terminal shows a PostgreSQL prompt 'postgres_lab2=#' followed by the SQL query 'select count(*) from student where gender = 'female';'. The output is a table with one column 'count' and one row with the value '2'. The prompt then returns to 'postgres_lab2=#'.

```
mazen@mazen-saad: ~
postgres_lab2=#
postgres_lab2=#
postgres_lab2=# select count(*) from student where gender = 'female';
 count
-----
      2
(1 row)

postgres_lab2=#
postgres_lab2=#
```

11. Display the students who are born before 1992-10-01.

(solution)

```
select * from student where birth_date < '1992-10-01';
```

```
mazen@mazen-saad: ~  
postgres_lab2=#  
postgres_lab2=#  
postgres_lab2=# select * from student where birth_date < '1992-10-01';  
 student_id | track_id | gender | birth_date | first_name | last_name | contact_info  
-----  
          7 |         2 | male   | 1992-08-01 | Ali        | sayed     | (asayed@gmail.com,cairo)  
          8 |         2 | male   | 1992-08-01 | Mohammed   | Mustafa   | (asayed@gmail.com,cairo)  
          9 |         5 | male   | 1991-09-01 | Mustafa    | sayed     | (asayed@gmail.com,fayoun)  
         10 |         3 | female | 1992-08-01 | sara       | gamal     | (asayed@gmail.com,cairo)  
         11 |         4 | female | 1992-08-01 | shrouk     | ahmed     | (asayed@gmail.com,fayoun)  
(5 rows)  
  
postgres_lab2=#  
postgres_lab2=#  
postgres_lab2=#
```

12. Display male students who are born before 1991-10-01.

(solution)

```
select * from student where gender = 'male' and birth_date < '1991-10-01';
```

```
mazen@mazen-saad: ~  
postgres_lab2=#  
postgres_lab2=#  
postgres_lab2=#  
postgres_lab2=# select * from student where gender = 'male' and birth_date < '1991-10-01';  
 student_id | track_id | gender | birth_date | first_name | last_name |      contact_info  
-----  
          9 |         5 | male   | 1991-09-01 | Mustafa   | sayed     | (asayed@gmail.com,fayoun)  
(1 row)  
  
postgres_lab2=#  
postgres_lab2=#
```

13. Display subjects and their max score sorted by max score.

(solution)

```
select * from subject order by max_score desc;
```

or

```
select subject_name, max_score from subject order by max_score  
desc;
```

```
mazen@mazen-saad: ~  
postgres_lab2=#  
postgres_lab2=#  
postgres_lab2=#  
postgres_lab2=# select * from subject order by max_score desc;  
subject_id | subject_name | description | max_score | track_id  
-----  
1 | C Programming | Introduction to C programming. | 100 | 1  
2 | C++ Programming | Advanced C++ concepts. | 100 | 2  
3 | Java | Introduction to Java programming. | 100 | 3  
4 | HTML | Basic HTML and web development. | 100 | 4  
5 | CSS | Basic CSS and web development. | 100 | 5  
(5 rows)  
  
postgres_lab2=#  
postgres_lab2=#
```

14. Display the subject with highest max score

(solution)

```
select * from subject order by max_score desc limit 1;
```

A terminal window with a dark background and light text. The window title is 'mazen@mazen-saad: ~'. The user is in a PostgreSQL shell. They enter a query to select the subject with the highest max_score. The result is displayed in a table format with columns: subject_id, subject_name, description, max_score, and track_id. The result shows one row: subject_id 1, subject_name C Programming, description Introduction to C programming., max_score 100, and track_id 1.

```
mazen@mazen-saad: ~  
postgres_lab2=#  
postgres_lab2=#  
postgres_lab2=# select * from subject order by max_score desc limit 1;  
 subject_id | subject_name |      description      | max_score | track_id  
-----+-----+-----+-----+-----  
          1 | C Programming | Introduction to C programming. |        100 |          1  
(1 row)  
  
postgres_lab2=#  
postgres_lab2=#
```

15. Display students' names that begin with A.

(solution) s26

```
select * from student where first_name like 'A%';
```

```
mazen@mazen-saad: ~  
postgres_lab2=#  
postgres_lab2=#  
postgres_lab2=# select * from student where first_name like 'A%';  
 student_id | track_id | gender | birth_date | first_name | last_name |      contact_info  
-----+-----+-----+-----+-----+-----+-----  
(1 row)  
      7 |      2 | male   | 1992-08-01 | Ali       | sayed    | (asayed@gmail.com,cairo)  
postgres_lab2=#  
postgres_lab2=#
```


16. Display the number of students' their name is "Mohammed"

(solution) s28

select count(*) from student where first_name like '%Mohammed';

or

select count(*) from student where first_name = 'Mohammed';

A terminal window with a dark background and light text. The window title is 'mazen@mazen-saad: ~'. The terminal shows a PostgreSQL prompt 'postgres_lab2=#' followed by the SQL query 'select count(*) from student where first_name = 'Mohammed';'. The output shows a single row with the value '1'.

```
mazen@mazen-saad: ~
postgres_lab2=#
postgres_lab2=#
postgres_lab2=# select count(*) from student where first_name = 'Mohammed';
 count
-----
      1
(1 row)

postgres_lab2=#
postgres_lab2=#
postgres_lab2=#
```

17. Display the number of males and females.

(solution) s28

```
select count(*) from student where gender = 'male' or gender = 'female';
```

or

```
select gender, count(gender) from student group by gender;
```

A terminal window with a dark purple background. The title bar shows 'mazen@mazen-saad: ~'. The prompt is 'postgres_lab2=#'. The user enters the SQL query 'select gender, count(gender) from student group by gender;'. The output shows a table with two columns: 'gender' and 'count'. The rows are 'female' with count 2, 'male' with count 4, and an empty row with count 0. The prompt then changes to 'postgres_lab2=#' again.

```
mazen@mazen-saad: ~
postgres_lab2=# select gender, count(gender) from student group by gender;
gender | count
-----+-----
female |     2
male   |     4
       |     0
(3 rows)

postgres_lab2=#
postgres_lab2=#
```

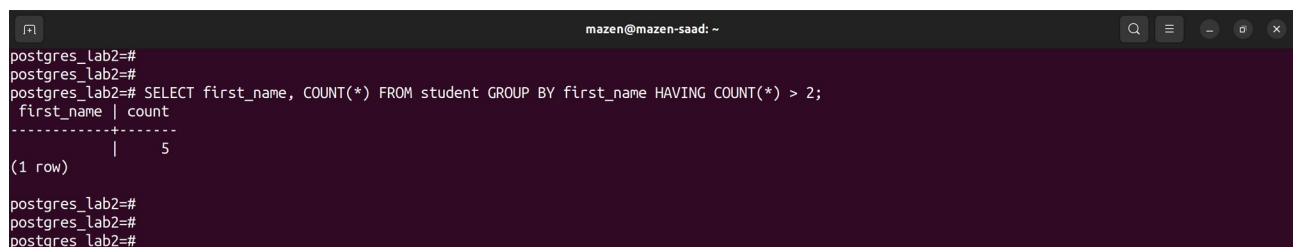
18. Display the repeated first names and their counts if higher than 2.

(solution) s29

```
select name from student group by name;  
select count(name) from student group by name;
```

or

```
SELECT first_name, COUNT(*) FROM student GROUP BY  
first_name HAVING COUNT(*) > 2;
```



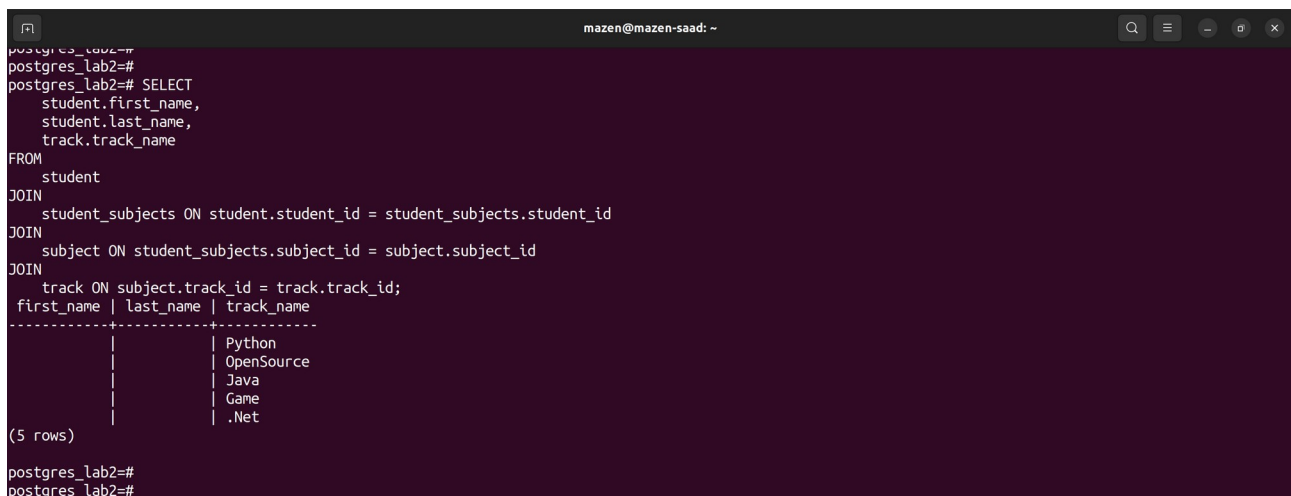
A terminal window titled 'mazen@mazen-saad: ~' showing a PostgreSQL session. The user enters 'postgres_lab2=#' and then the query 'SELECT first_name, COUNT(*) FROM student GROUP BY first_name HAVING COUNT(*) > 2;'. The output is a table with two columns: 'first_name' and 'count'. The first row shows '5' in the 'count' column. The terminal also shows '(1 row)' and the prompt 'postgres_lab2=#' repeated three times.

```
mazen@mazen-saad: ~  
postgres_lab2=#  
postgres_lab2=#  
postgres_lab2=# SELECT first_name, COUNT(*) FROM student GROUP BY first_name HAVING COUNT(*) > 2;  
 first_name | count  
-----+-----  
          | 5  
(1 row)  
postgres_lab2=#  
postgres_lab2=#  
postgres_lab2=#
```

19. Display the all Students and track name that belong to it

(solution)

```
SELECT
    student.first_name,
    student.last_name,
    track.track_name
FROM
    student
JOIN
    student_subjects ON student.student_id =
student_subjects.student_id
JOIN
    subject ON student_subjects.subject_id = subject.subject_id
JOIN
    track ON subject.track_id = track.track_id;
```

A terminal window with a dark background and light text. The window title is 'mazen@mazen-saad: ~'. The terminal shows a PostgreSQL prompt 'postgres=#' followed by a query. The query is a JOIN of student, student_subjects, subject, and track tables. The results are displayed in a table with columns first_name, last_name, and track_name. There are 5 rows of data. The prompt 'postgres lab2=#' appears twice at the bottom.

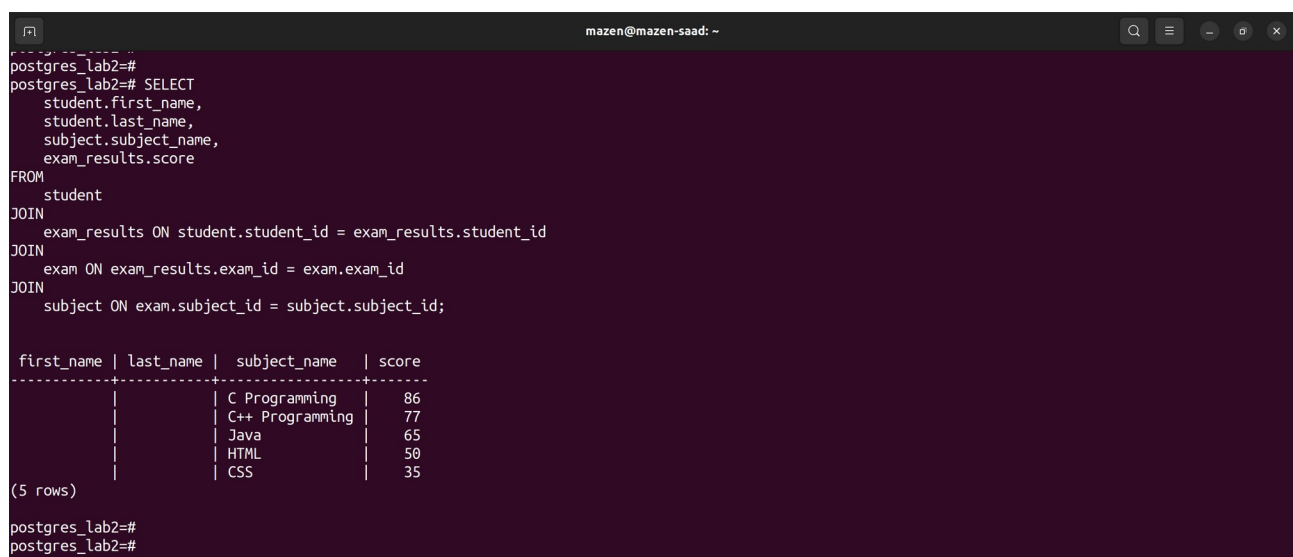
```
mazen@mazen-saad: ~
postgres=#
postgres_lab2=#
postgres_lab2=# SELECT
    student.first_name,
    student.last_name,
    track.track_name
FROM
    student
JOIN
    student_subjects ON student.student_id = student_subjects.student_id
JOIN
    subject ON student_subjects.subject_id = subject.subject_id
JOIN
    track ON subject.track_id = track.track_id;
 first_name | last_name | track_name
-----+-----+-----
           |           | Python
           |           | OpenSource
           |           | Java
           |           | Game
           |           | .Net
(5 rows)

postgres_lab2=#
postgres_lab2=#
```

20. (Bouns) Display students' names, their score and subject name.

(solution)

```
SELECT
    student.first_name,
    student.last_name,
    subject.subject_name,
    exam_results.score
FROM
    student
JOIN
    exam_results ON student.student_id = exam_results.student_id
JOIN
    exam ON exam_results.exam_id = exam.exam_id
JOIN
    subject ON exam.subject_id = subject.subject_id;
```

A terminal window with a dark background and light text. The window title is 'mazen@mazen-saad: ~'. The user has entered a SQL query into a PostgreSQL terminal. The query is a JOIN query that selects student names, subject names, and exam scores. The results are displayed in a table format with 5 rows. The first row shows a student with first_name 'Mazen' and last_name 'Saad' who scored 86 in 'C Programming'. The second row shows the same student scored 77 in 'C++ Programming'. The third row shows the student scored 65 in 'Java'. The fourth row shows the student scored 50 in 'HTML'. The fifth row shows the student scored 35 in 'CSS'. The terminal prompt is 'postgres_lab2=#' and the query is 'SELECT student.first_name, student.last_name, subject.subject_name, exam_results.score FROM student JOIN exam_results ON student.student_id = exam_results.student_id JOIN exam ON exam_results.exam_id = exam.exam_id JOIN subject ON exam.subject_id = subject.subject_id;'. The results are shown as a table with columns 'first_name', 'last_name', 'subject_name', and 'score'. The first row is 'Mazen', 'Saad', 'C Programming', '86'. The second row is 'Mazen', 'Saad', 'C++ Programming', '77'. The third row is 'Mazen', 'Saad', 'Java', '65'. The fourth row is 'Mazen', 'Saad', 'HTML', '50'. The fifth row is 'Mazen', 'Saad', 'CSS', '35'. Below the table, it says '(5 rows)'. The terminal prompt is 'postgres_lab2=#' and the query is 'SELECT student.first_name, student.last_name, subject.subject_name, exam_results.score FROM student JOIN exam_results ON student.student_id = exam_results.student_id JOIN exam ON exam_results.exam_id = exam.exam_id JOIN subject ON exam.subject_id = subject.subject_id;'. The results are shown as a table with columns 'first_name', 'last_name', 'subject_name', and 'score'. The first row is 'Mazen', 'Saad', 'C Programming', '86'. The second row is 'Mazen', 'Saad', 'C++ Programming', '77'. The third row is 'Mazen', 'Saad', 'Java', '65'. The fourth row is 'Mazen', 'Saad', 'HTML', '50'. The fifth row is 'Mazen', 'Saad', 'CSS', '35'. Below the table, it says '(5 rows)'. The terminal prompt is 'postgres_lab2=#' and the query is 'SELECT student.first_name, student.last_name, subject.subject_name, exam_results.score FROM student JOIN exam_results ON student.student_id = exam_results.student_id JOIN exam ON exam_results.exam_id = exam.exam_id JOIN subject ON exam.subject_id = subject.subject_id;'. The results are shown as a table with columns 'first_name', 'last_name', 'subject_name', and 'score'. The first row is 'Mazen', 'Saad', 'C Programming', '86'. The second row is 'Mazen', 'Saad', 'C++ Programming', '77'. The third row is 'Mazen', 'Saad', 'Java', '65'. The fourth row is 'Mazen', 'Saad', 'HTML', '50'. The fifth row is 'Mazen', 'Saad', 'CSS', '35'. Below the table, it says '(5 rows)'.

```
mazen@mazen-saad: ~
postgres_lab2=#
postgres_lab2=# SELECT
    student.first_name,
    student.last_name,
    subject.subject_name,
    exam_results.score
FROM
    student
JOIN
    exam_results ON student.student_id = exam_results.student_id
JOIN
    exam ON exam_results.exam_id = exam.exam_id
JOIN
    subject ON exam.subject_id = subject.subject_id;

 first_name | last_name | subject_name | score
-----
          |          | C Programming |    86
          |          | C++ Programming |    77
          |          | Java          |    65
          |          | HTML         |    50
          |          | CSS          |    35
(5 rows)

postgres_lab2=#
postgres_lab2=#
```

```
// lab3
su - postgres
psql
create database postgres_lab3 TEMPLATE postgres_lab2;
\l
\c postgres_lab3
```

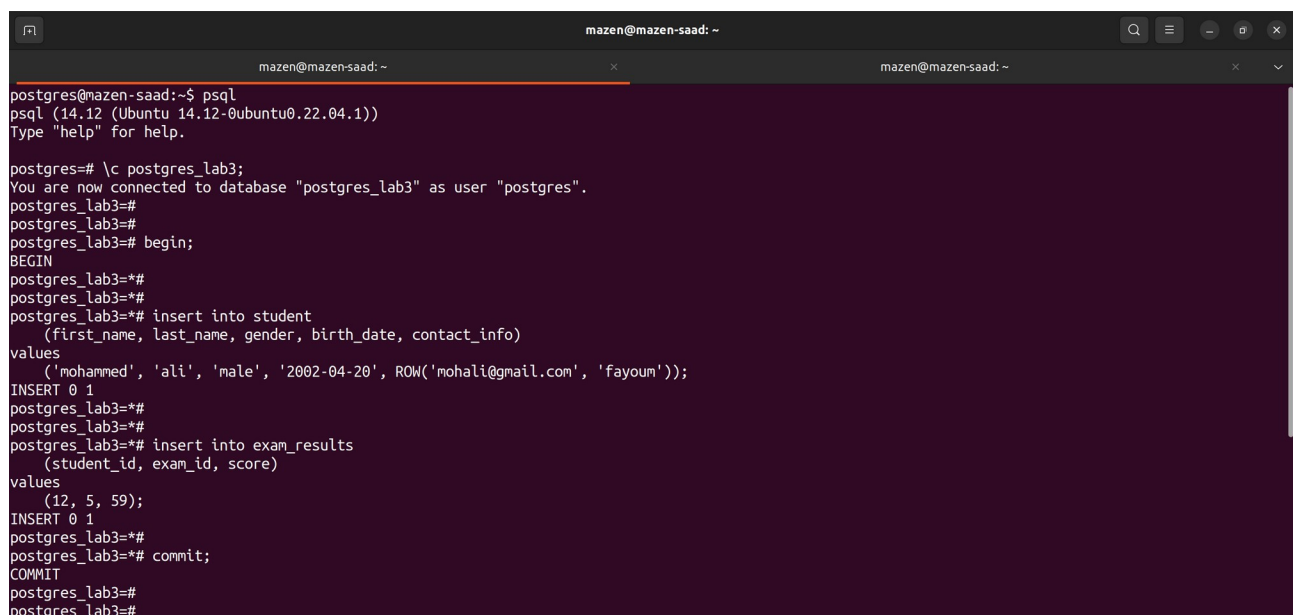
1. Insert new student and his score in exam in different subjects as transaction and save it.

```
begin;
```

```
insert into student
    (first_name, last_name, gender, birth_date, contact_info)
values
    ('mohammed', 'ali', 'male', '2002-04-20',
ROW('mohali@gmail.com', 'fayoum'));
```

```
insert into exam_results
    (student_id, exam_id, score)
values
    (6, 5, 59);
```

```
commit;
```



```
mazen@mazen-saad: ~
postgres@mazen-saad:~$ psql
psql (14.12 (Ubuntu 14.12-0ubuntu0.22.04.1))
Type "help" for help.

postgres=# \c postgres_lab3;
You are now connected to database "postgres_lab3" as user "postgres".
postgres_lab3=#
postgres_lab3=#
postgres_lab3=# begin;
BEGIN
postgres_lab3=*#
postgres_lab3=*#
postgres_lab3=*# insert into student
    (first_name, last_name, gender, birth_date, contact_info)
values
    ('mohammed', 'ali', 'male', '2002-04-20', ROW('mohali@gmail.com', 'fayoum'));
INSERT 0 1
postgres_lab3=*#
postgres_lab3=*# insert into exam_results
    (student_id, exam_id, score)
values
    (12, 5, 59);
INSERT 0 1
postgres_lab3=*#
postgres_lab3=*# commit;
COMMIT
postgres_lab3=#
postgres_lab3=#
```


2. Insert new students and his score in exam in different subjects as transaction and undo it.

```
begin;
```

```
insert into student
```

```
    (first_name, last_name, gender, birth_date, contact_info)
```

```
values
```

```
    ('jane', 'smith', 'male', '2003-04-20', ROW('jane@gmail.com',  
'fayoum'));
```

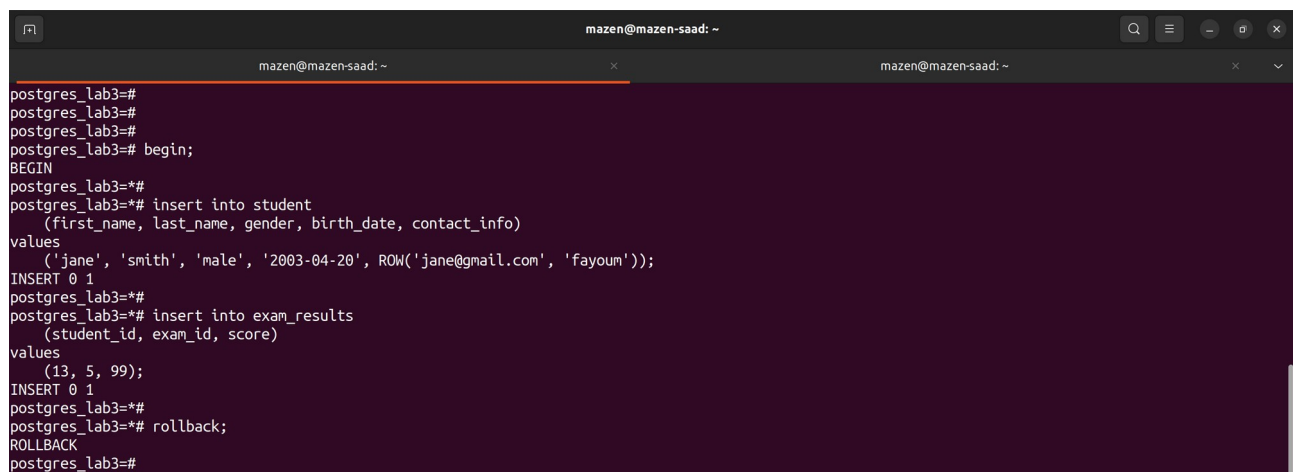
```
insert into exam_results
```

```
    (student_id, exam_id, score)
```

```
values
```

```
    (7, 5, 99);
```

```
rollback;
```



```
mazen@mazen-saad: ~  
postgres_lab3=#  
postgres_lab3=#  
postgres_lab3=#  
postgres_lab3=# begin;  
BEGIN  
postgres_lab3=#  
postgres_lab3=# insert into student  
    (first_name, last_name, gender, birth_date, contact_info)  
values  
    ('jane', 'smith', 'male', '2003-04-20', ROW('jane@gmail.com', 'fayoum'));  
INSERT 0 1  
postgres_lab3=#  
postgres_lab3=# insert into exam_results  
    (student_id, exam_id, score)  
values  
    (13, 5, 99);  
INSERT 0 1  
postgres_lab3=#  
postgres_lab3=# rollback;  
ROLLBACK  
postgres_lab3=#
```


8. Create user and give him all privileges.

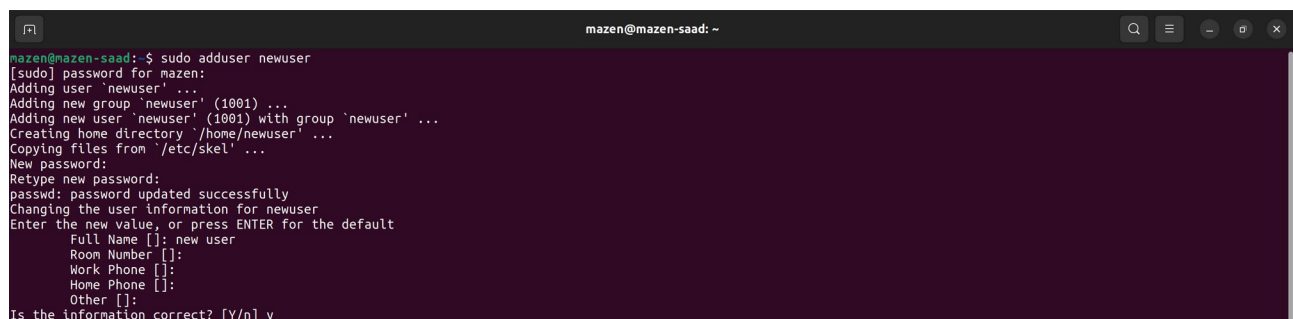
(solution)

```
sudo adduser newuser
sudo passwd newuser
su - postgres
psql
```

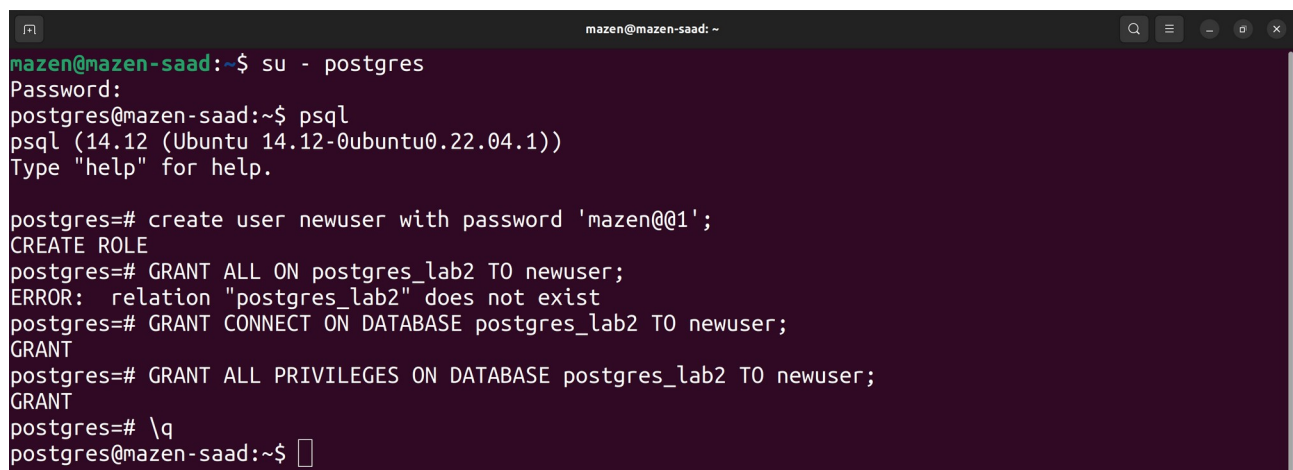
create user newuser with password 'mazen@@1';

GRANT CONNECT ON DATABASE postgres_lab2 TO newuser;

GRANT ALL PRIVILEGES ON DATABASE postgres_lab2 TO newuser;

A terminal window titled 'mazen@mazen-saad: ~' showing the command 'sudo adduser newuser'. The output shows the user being added with a group, a home directory, and a password. The user is prompted to enter their full name, room number, work phone, home phone, and other information. The information is entered as 'new user' for the full name, and the user confirms the information is correct with 'y'.

```
mazen@mazen-saad:~$ sudo adduser newuser
[sudo] password for mazen:
Adding user 'newuser' ...
Adding new group 'newuser' (1001) ...
Adding new user 'newuser' (1001) with group 'newuser' ...
Creating home directory '/home/newuser' ...
Copying files from '/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for newuser
Enter the new value, or press ENTER for the default
  Full Name []: new user
    Room Number []:
    Work Phone []:
    Home Phone []:
      Other []:
Is the information correct? [Y/n] y
```

A terminal window titled 'mazen@mazen-saad: ~' showing the command 'su - postgres'. The prompt changes to 'postgres@mazen-saad:~\$'. The user then enters 'psql', and the prompt changes to 'psql (14.12 (Ubuntu 14.12-0ubuntu0.22.04.1))'. The user enters 'Type "help" for help.' and then enters the following SQL commands: 'create user newuser with password 'mazen@@1';', 'CREATE ROLE', 'GRANT ALL ON postgres_lab2 TO newuser;', 'ERROR: relation "postgres_lab2" does not exist', 'GRANT CONNECT ON DATABASE postgres_lab2 TO newuser;', 'GRANT', 'GRANT ALL PRIVILEGES ON DATABASE postgres_lab2 TO newuser;', 'GRANT', and '\q'. The prompt returns to 'postgres@mazen-saad:~\$'.

```
mazen@mazen-saad:~$ su - postgres
Password:
postgres@mazen-saad:~$ psql
psql (14.12 (Ubuntu 14.12-0ubuntu0.22.04.1))
Type "help" for help.

postgres=# create user newuser with password 'mazen@@1';
CREATE ROLE
postgres=# GRANT ALL ON postgres_lab2 TO newuser;
ERROR:  relation "postgres_lab2" does not exist
postgres=# GRANT CONNECT ON DATABASE postgres_lab2 TO newuser;
GRANT
postgres=# GRANT ALL PRIVILEGES ON DATABASE postgres_lab2 TO newuser;
GRANT
postgres=# \q
postgres@mazen-saad:~$
```

9. Create another new user and make the authentication method is “trust” and give him all privileges if he login from his “local” server.

(solution)

```
sudo adduser localuser
```

```
sudo passwd localuser
```

```
su – postgres
```

```
psql
```

```
CREATE USER localuser WITH PASSWORD 'mazen@@1';
```

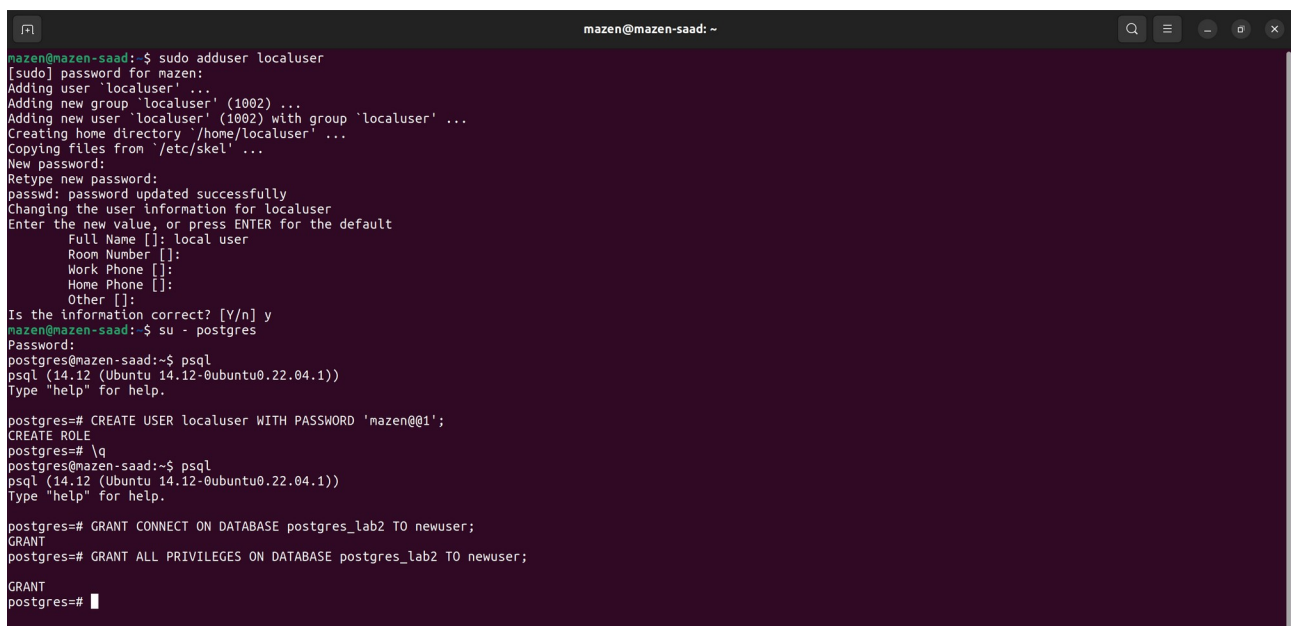
```
# edit file pg_hba.conf
```

```
sudo vim /var/lib/pg_hba.conf
```

```
local all localuser trust
```

```
GRANT CONNECT ON DATABASE postgres_lab2 TO newuser;
```

```
GRANT ALL PRIVILEGES ON DATABASE postgres_lab2 TO  
newuser;
```



```
mazen@mazen-saad: ~  
mazen@mazen-saad:~$ sudo adduser localuser  
[sudo] password for mazen:  
Adding user 'localuser' ...  
Adding new group 'localuser' (1002) ...  
Adding new user 'localuser' (1002) with group 'localuser' ...  
Creating home directory '/home/localuser' ...  
Copying files from '/etc/skel' ...  
New password:  
Retype new password:  
passwd: password updated successfully  
Changing the user information for localuser  
Enter the new value, or press ENTER for the default  
Full Name []: local user  
Room Number []:  
Work Phone []:  
Home Phone []:  
Other []:  
Is the information correct? [Y/n] y  
mazen@mazen-saad:~$ su - postgres  
Password:  
postgres@mazen-saad:~$ psql  
psql (14.12 (Ubuntu 14.12-0ubuntu0.22.04.1))  
Type "help" for help.  
  
postgres=# CREATE USER localuser WITH PASSWORD 'mazen@@1';  
CREATE ROLE  
postgres=# \q  
postgres@mazen-saad:~$ psql  
psql (14.12 (Ubuntu 14.12-0ubuntu0.22.04.1))  
Type "help" for help.  
  
postgres=# GRANT CONNECT ON DATABASE postgres_lab2 TO newuser;  
GRANT  
postgres=# GRANT ALL PRIVILEGES ON DATABASE postgres_lab2 TO newuser;  
GRANT  
postgres=#
```

