**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)

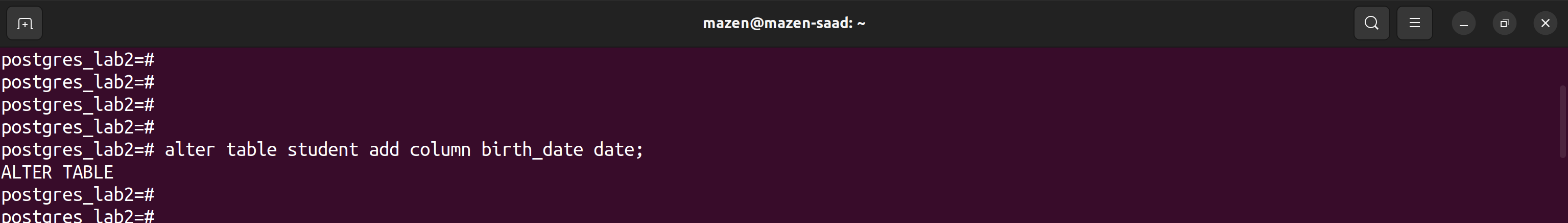
alter table student add column gender enum('male', 'female');



2. Add birth date column for the student table.

(solution)

alter table student add column birth\_date date;



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;



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;



5. Change any Serial Datatype at your tables to smallInt

(solution)

alter table track alter track\_id set data type smallInt;

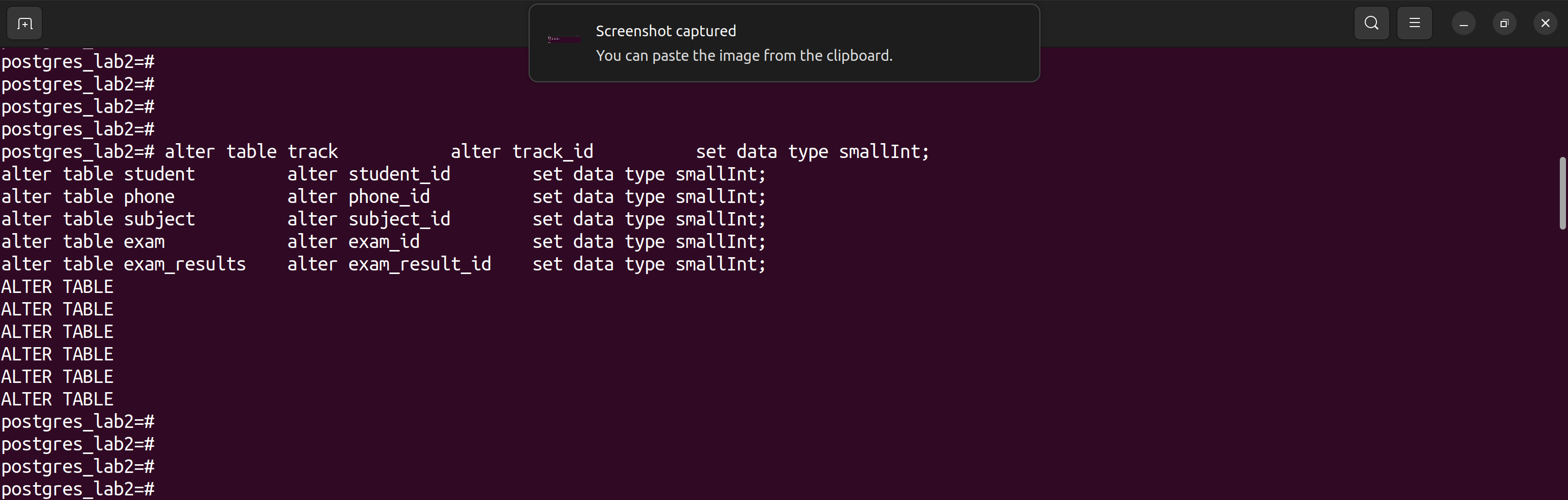
alter table student alter student\_id set data type smallInt;

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;



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

(solution) s42

alter table student add constraint student\_tr\_fk FOREIGN KEY (track\_id) references track(track\_id);

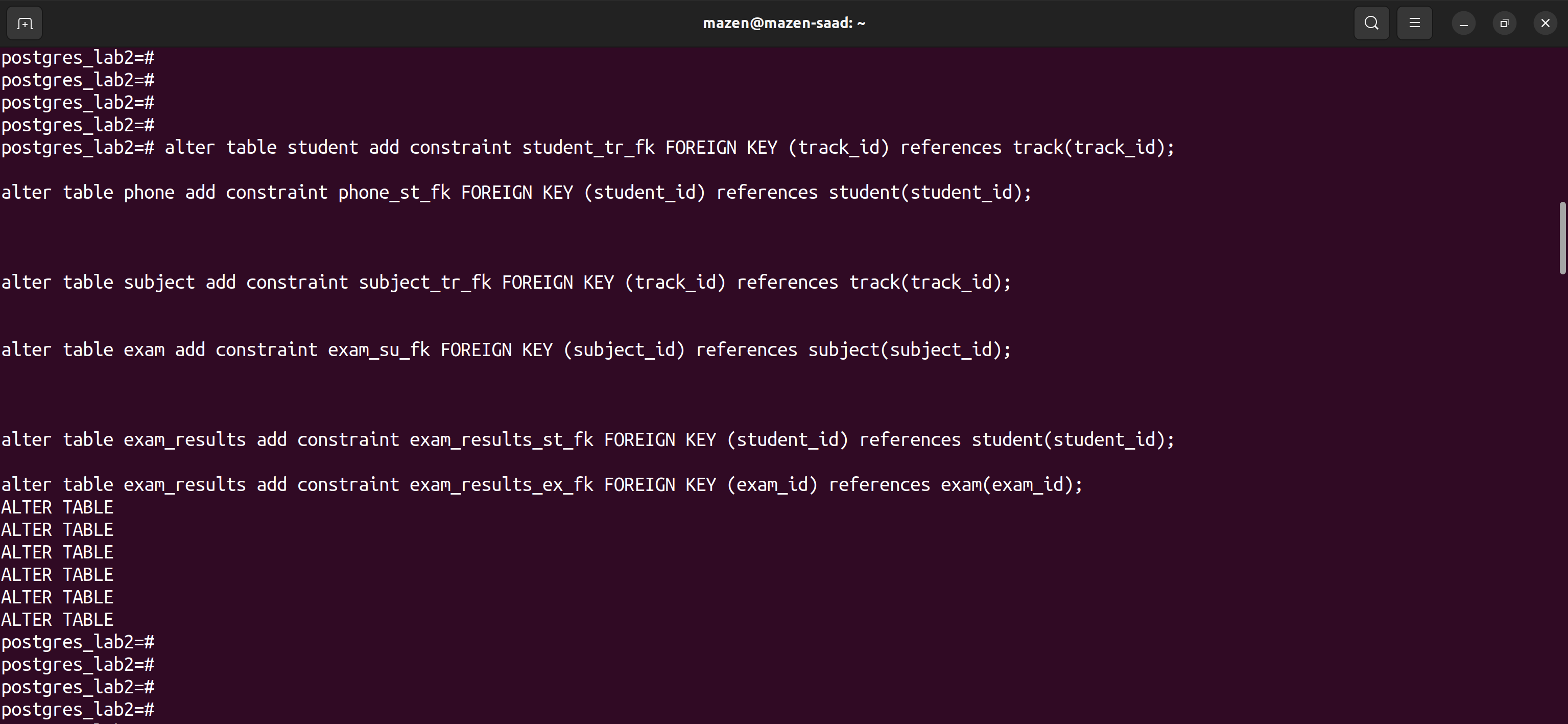
alter table phone add constraint phone\_st\_fk FOREIGN KEY (student\_id) references student(student\_id);

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);



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')),

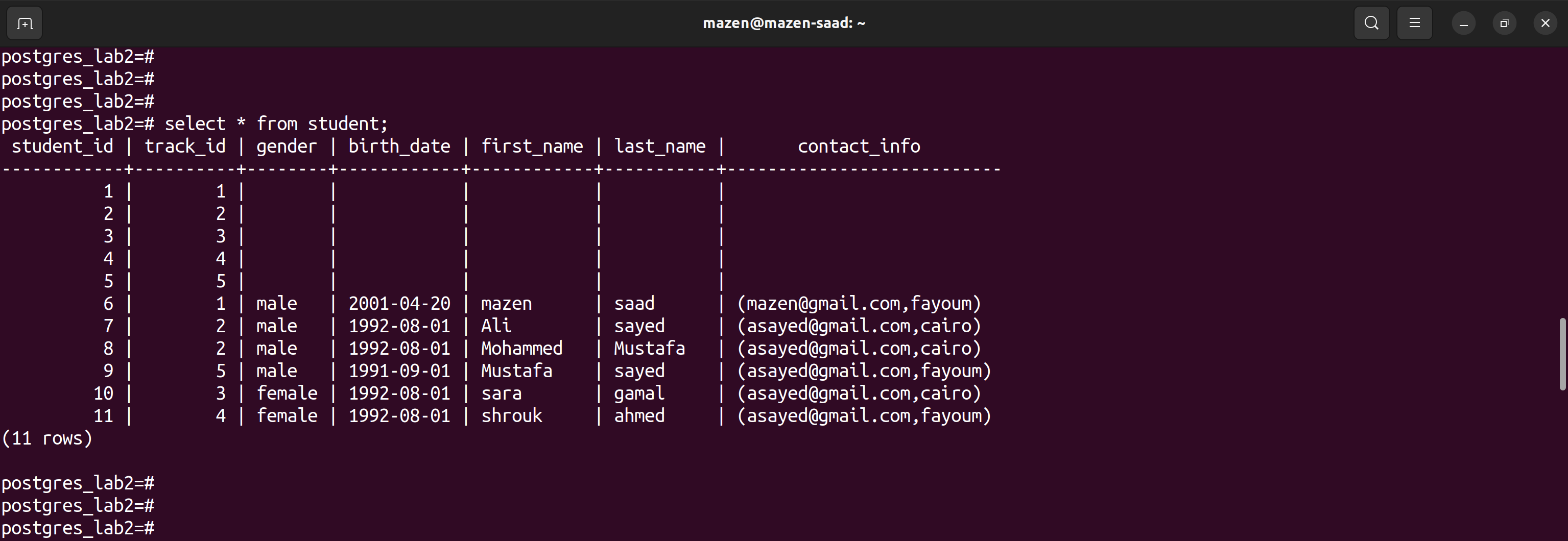
(4, 'shrouk', 'ahmed', 'female', '1992-08-01', ROW('asayed@gmail.com', 'fayoum'));



8. Display all students’ information.

(solution)

select \* from student;



9. Display male students only.

(solution)

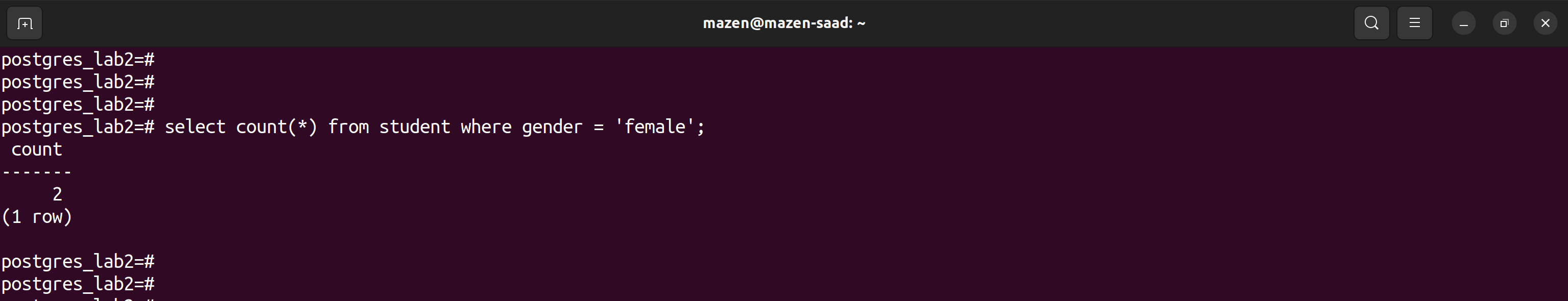
select \* from student where gender = 'male';



10. Display the number of female students.

(solution)

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



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

(solution)

select \* from student where birth\_date < '1992-10-01';



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

(solution)

select \* from student where gender = 'male' and birth\_date < '1991-10-01';



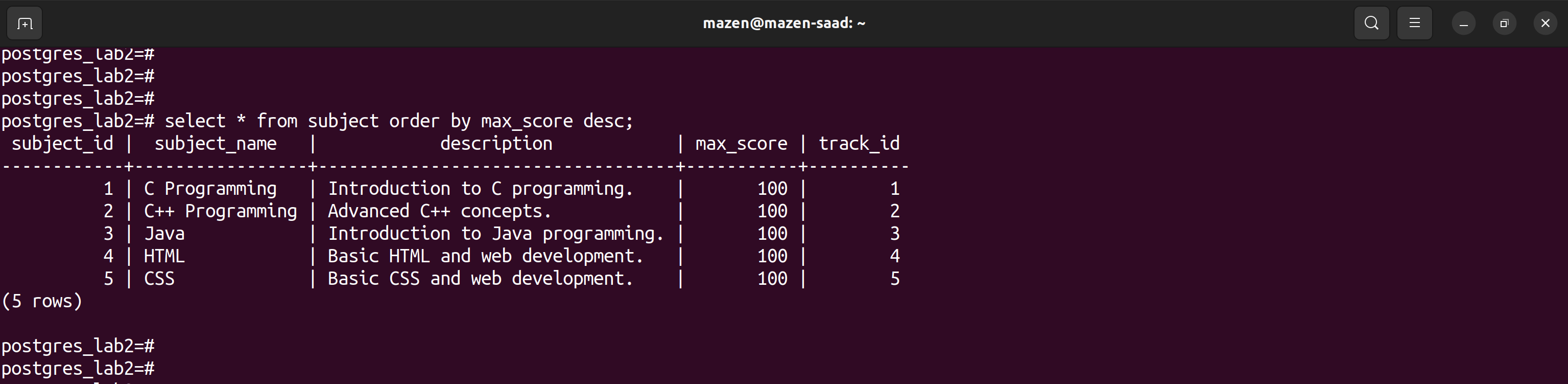
13. Display subjects and their max score sorted by max score.

(solution)

select \* from subject order by max\_score desc;

or

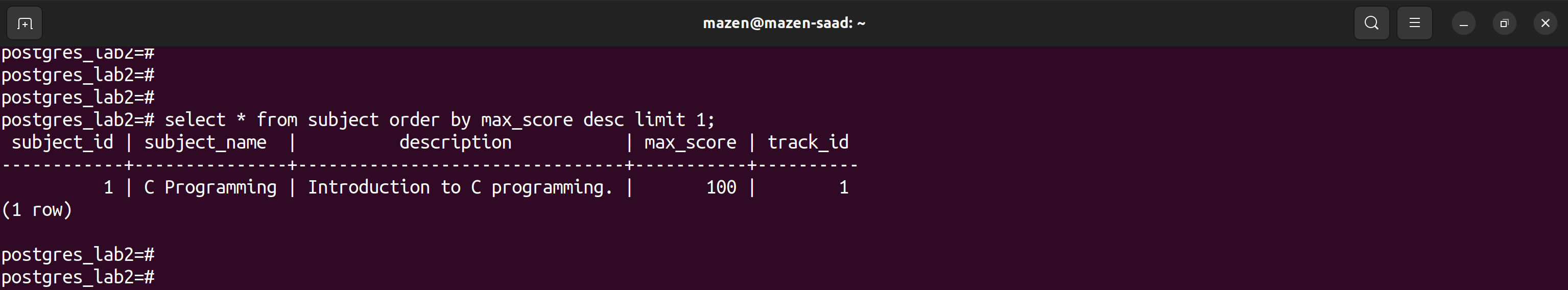
select subject\_name, max\_score from subject order by max\_score desc;



14. Display the subject with highest max score

(solution)

select \* from subject order by max\_score desc limit 1;



15. Display students’ names that begin with A.

(solution) s26

select \* from student where first\_name like 'A%';



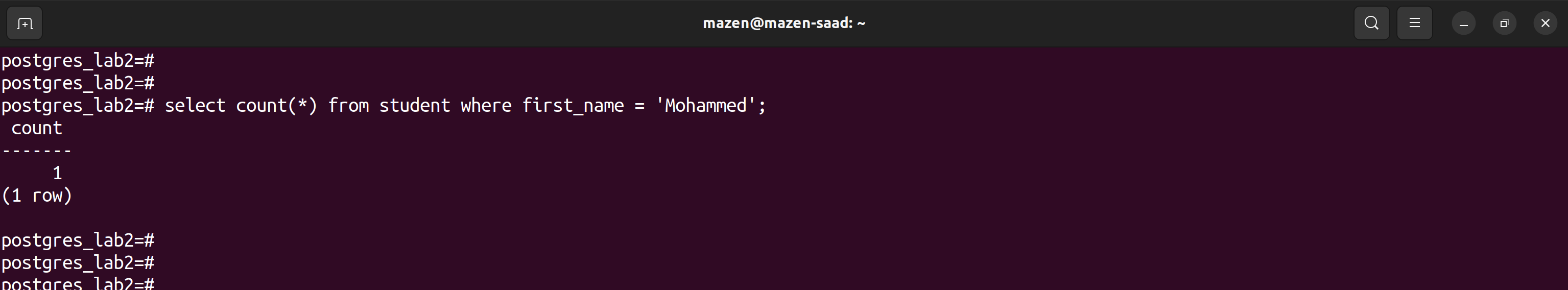
16. Display the number of students’ their name is “Mohammed”

(solution) s28

select count(\*) from student where first\_name like '%Mohammed';

or

select count(\*) from student where first\_name = 'Mohammed';



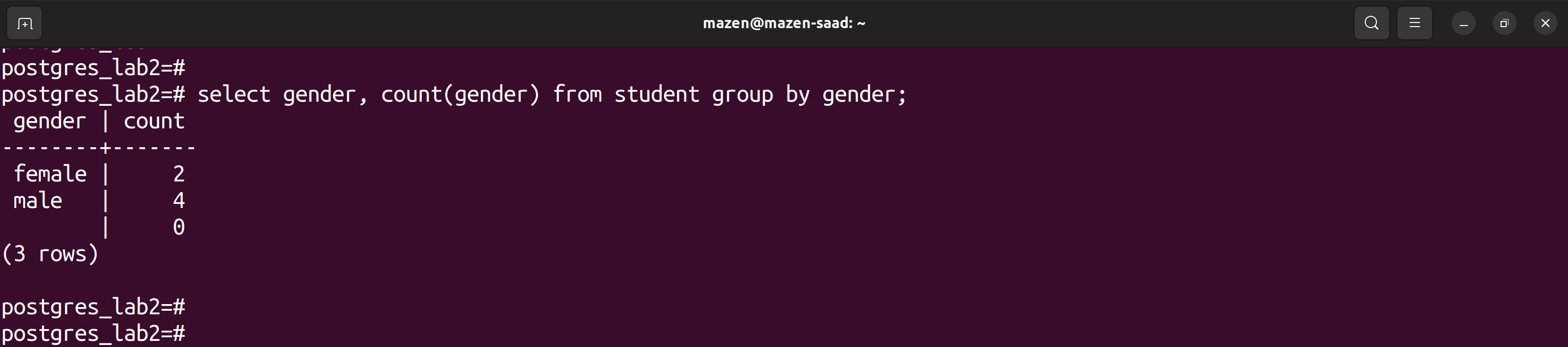
17. Display the number of males and females.

(solution) s28

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

or

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



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

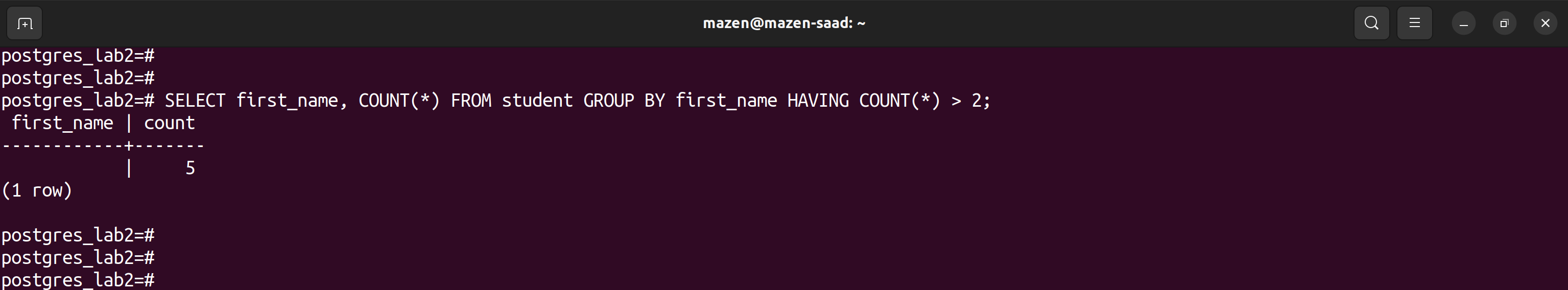
(solution) s29

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;



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

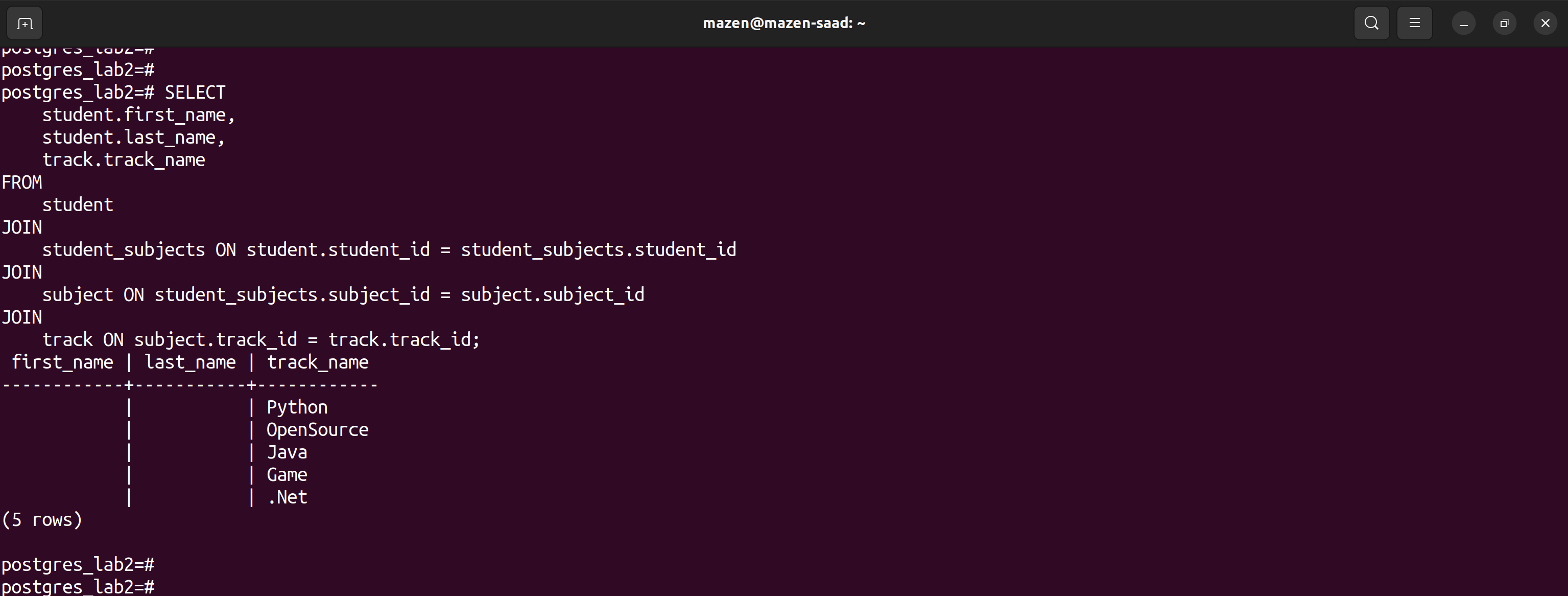
student\_subjects ON student.student\_id = student\_subjects.student\_id

JOIN

subject ON student\_subjects.subject\_id = subject.subject\_id

JOIN

track ON subject.track\_id = track.track\_id;



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

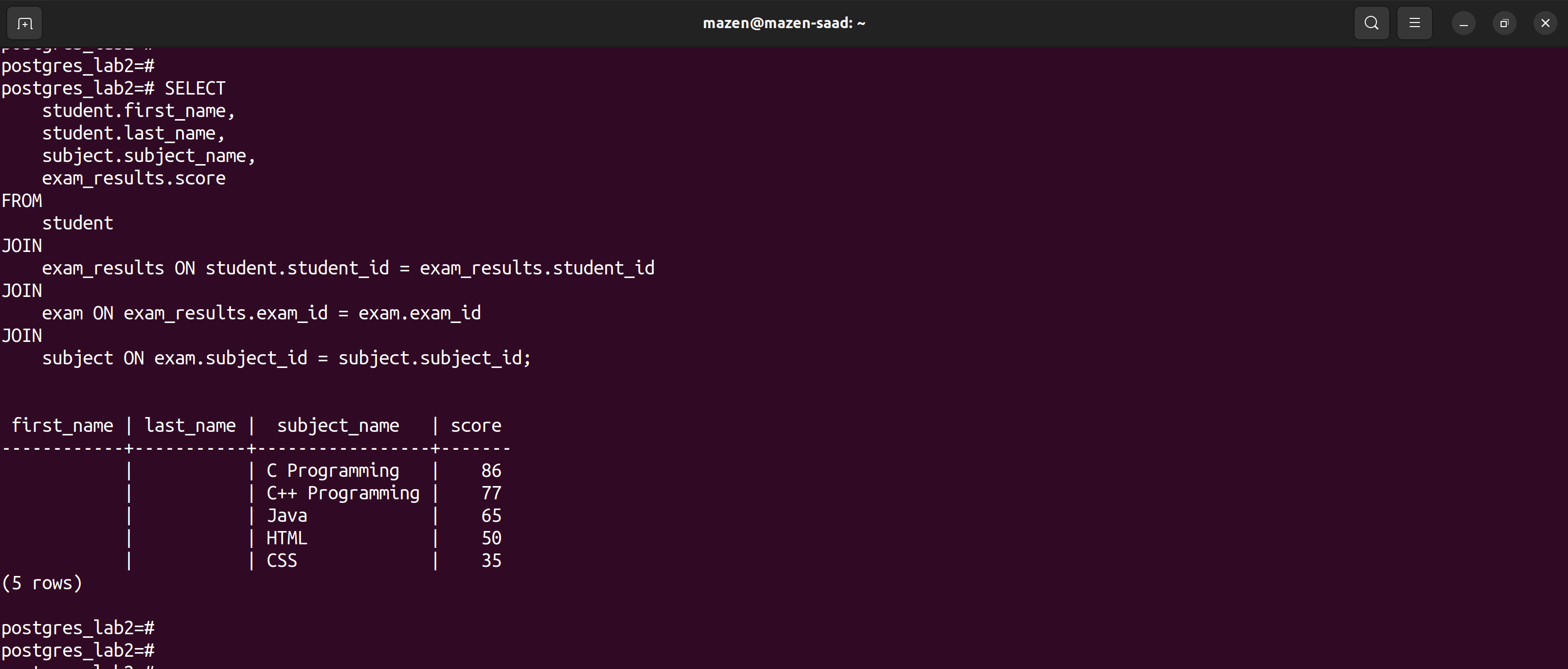
exam\_results ON student.student\_id = exam\_results.student\_id

JOIN

exam ON exam\_results.exam\_id = exam.exam\_id

JOIN

subject ON exam.subject\_id = subject.subject\_id;



// 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'));

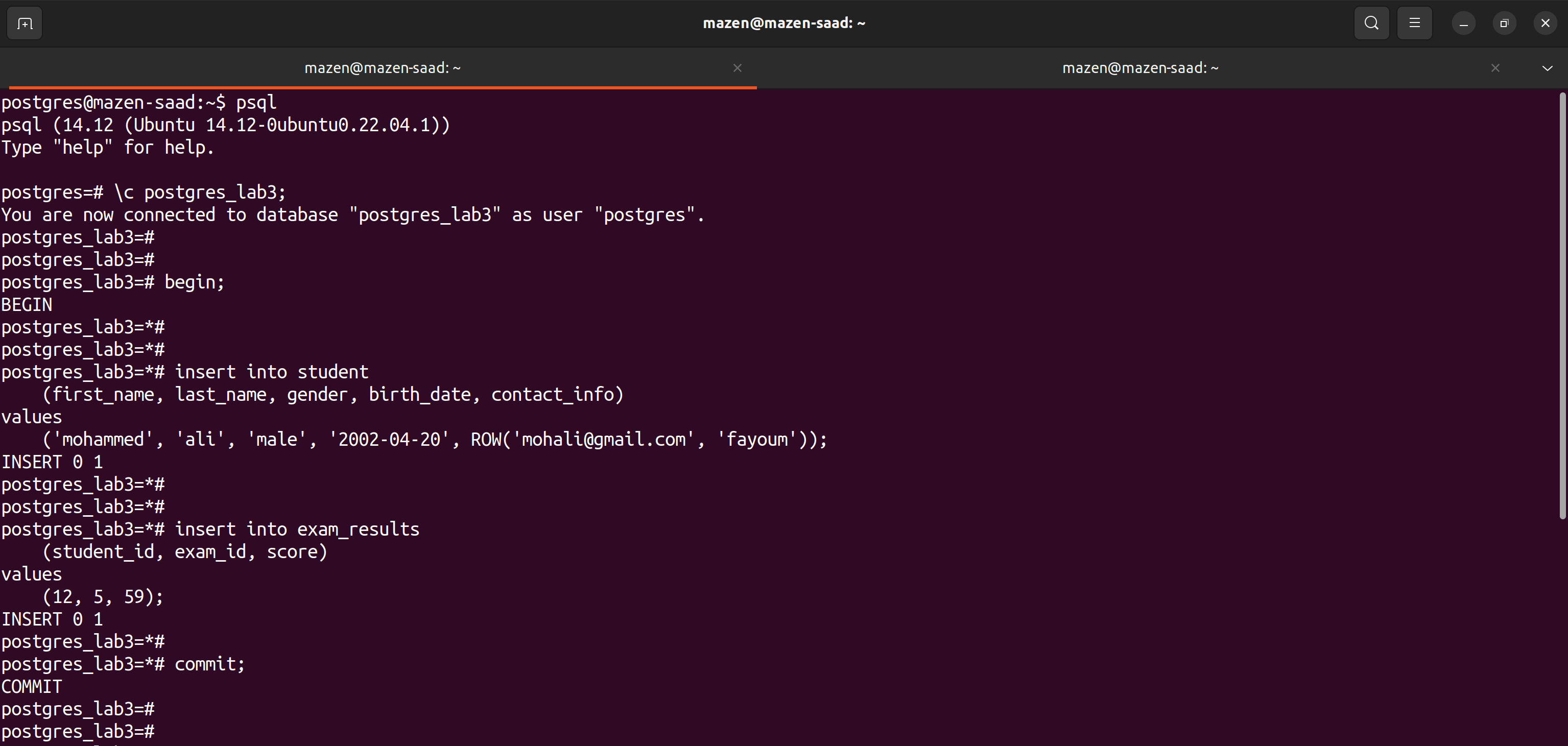
insert into exam\_results

(student\_id, exam\_id, score)

values

(6, 5, 59);

commit;



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'));

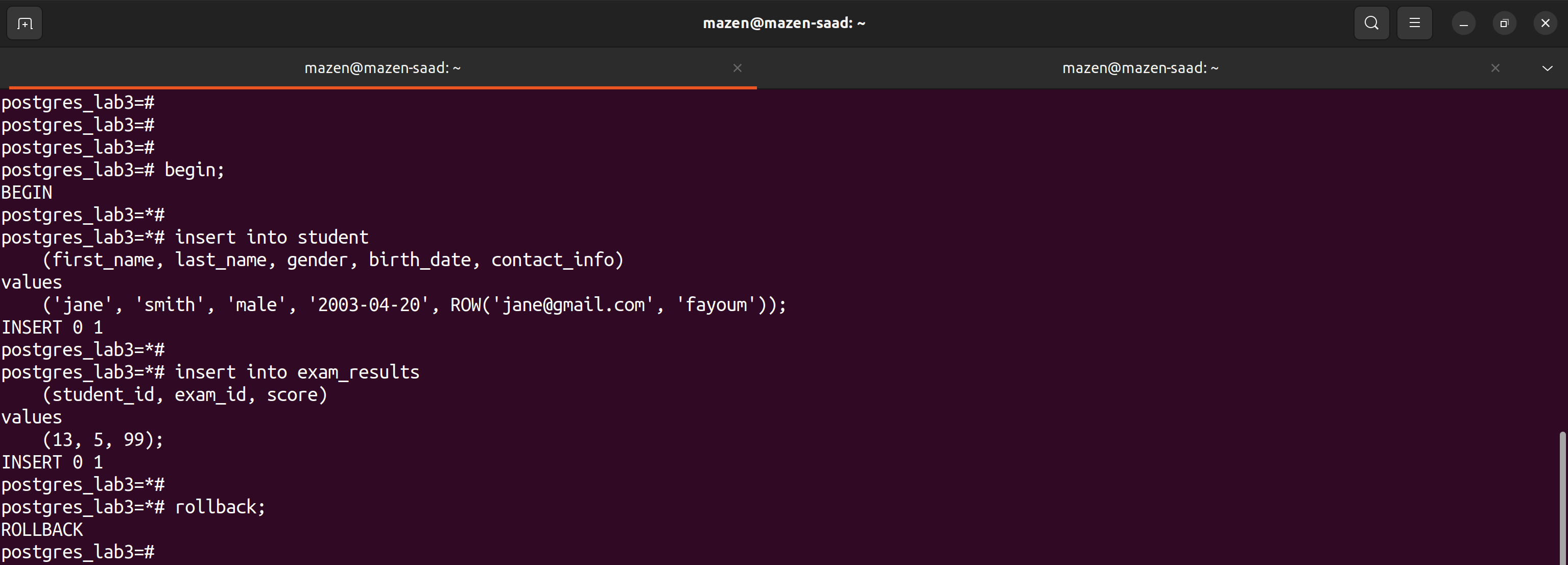
insert into exam\_results

(student\_id, exam\_id, score)

values

(7, 5, 99);

rollback;



8. Create user and give him all privileges.

(solution)

sudo adduser newuser

sudo passwd newuser

su – postgres

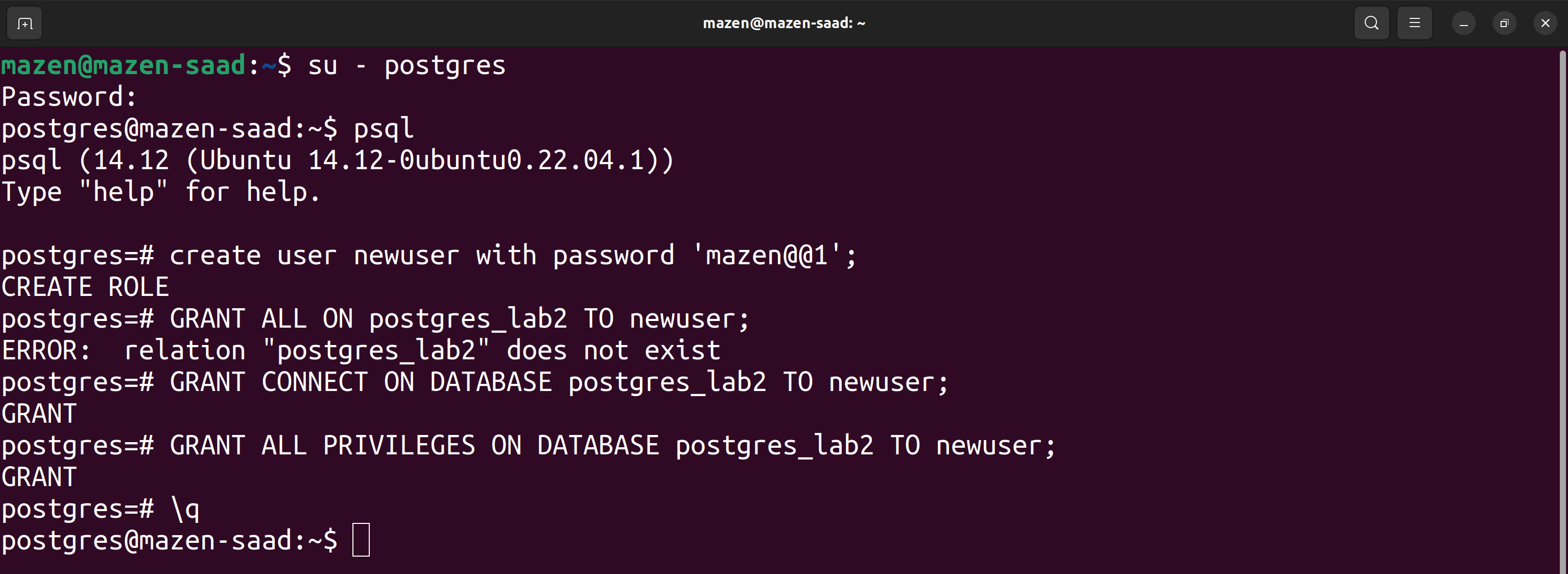
psql

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

GRANT CONNECT ON DATABASE postgres\_lab2 TO newuser;

GRANT ALL PRIVILEGES ON DATABASE postgres\_lab2 TO newuser;





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;

