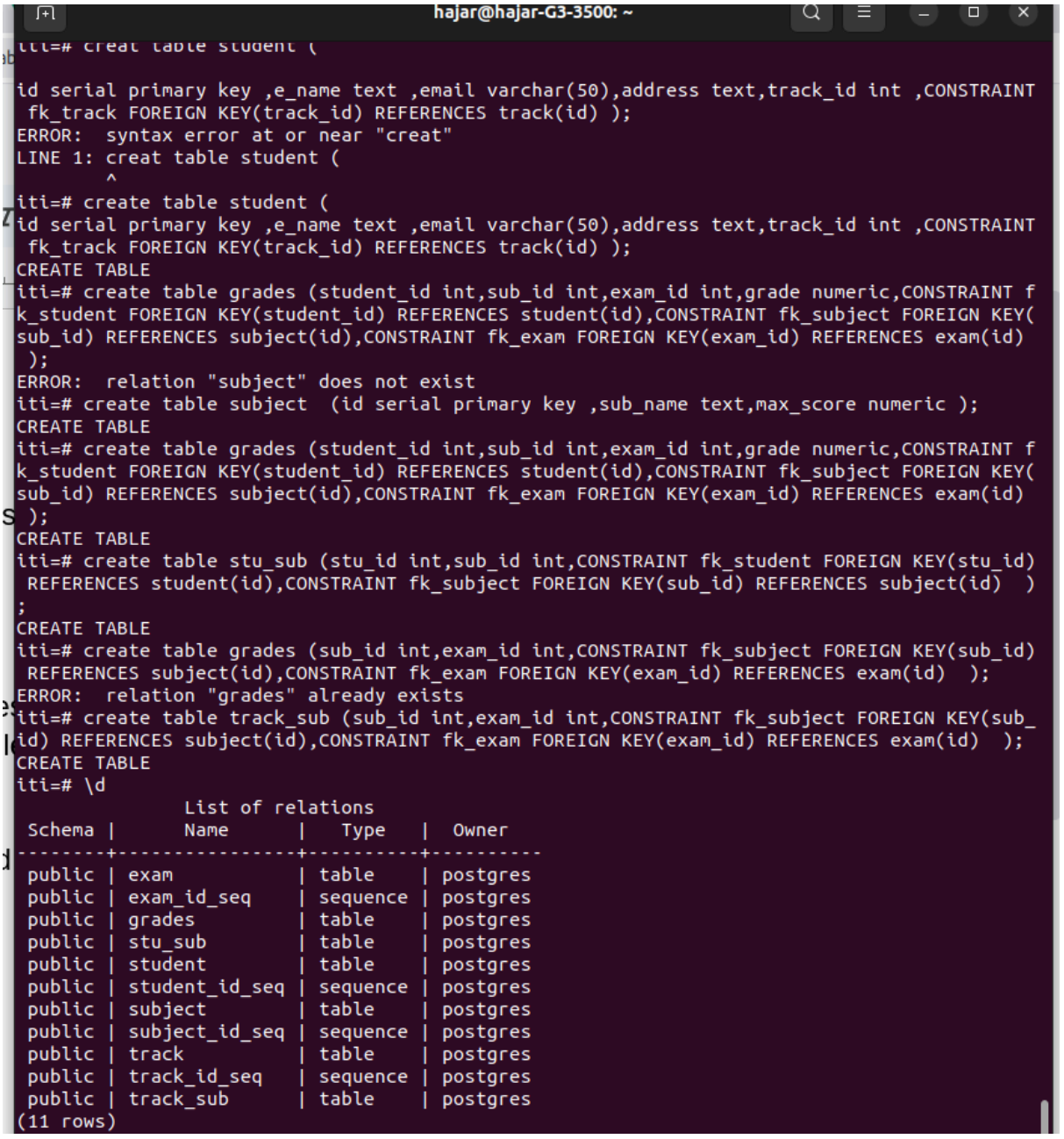
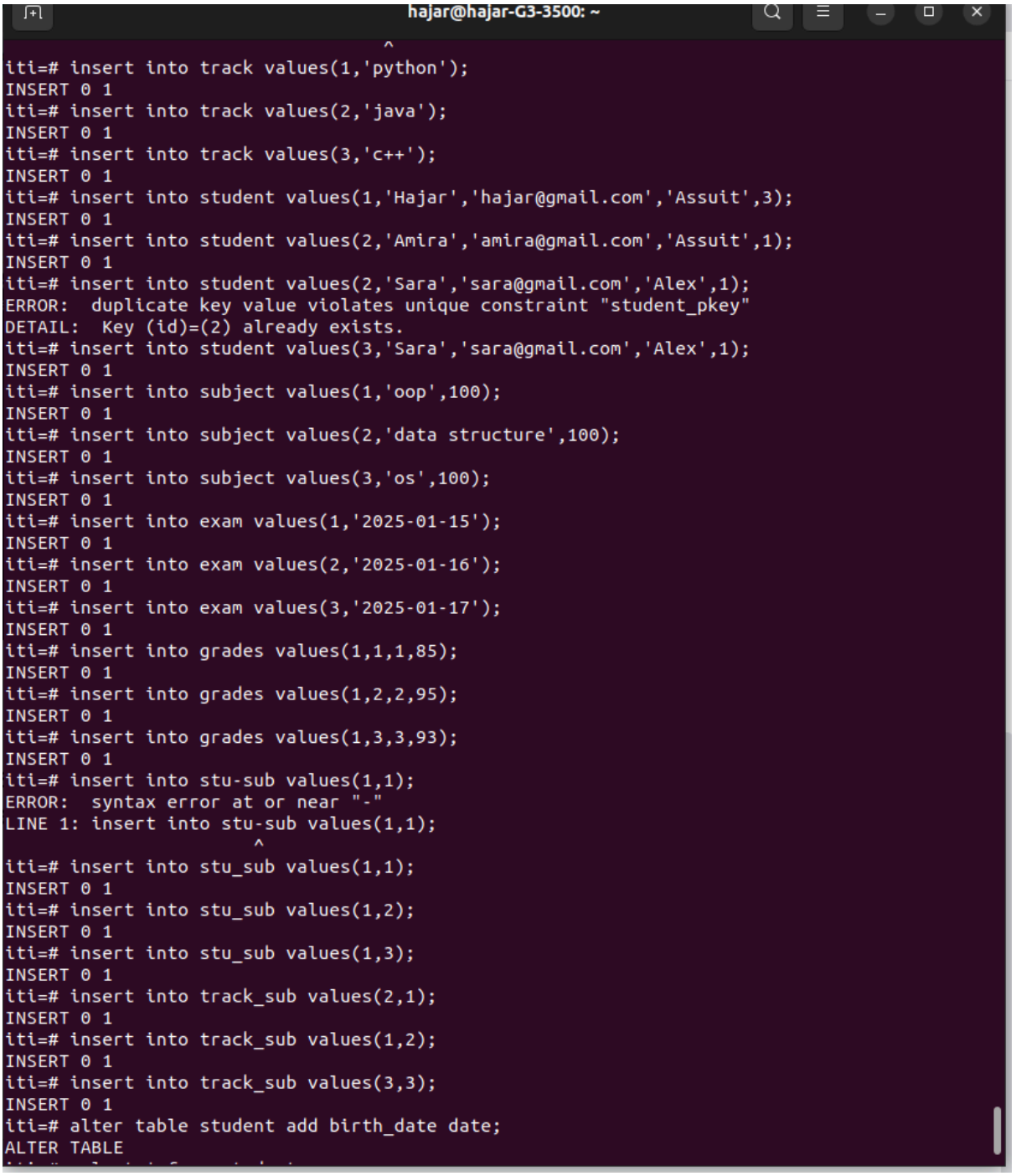
Lab 1

1. Create your tables with their columns in PostgreSQL.



2. Insert at minimum 3 Rows at each table.



3. Add birth date column for the student table.

4. Add gender column which hold only 2 values (Male or Female).

```
iti=# create type gender_type as enum('male','female');
CREATE TYPE
iti=# alter table student add gender gender_type;
ALTER TABLE
iti=# select * from student
;
 id | e_name |      email      | address | track_id | birth_date | gender
----+--------+-----------------+---------+----------+------------+--------
  1 | Hajar  | hajar@gmail.com | Assuit  |        3 |            |
  2 | Amira  | amira@gmail.com | Assuit  |        1 |            |
  3 | Sara   | sara@gmail.com  | Alex    |        1 |            |
(3 rows)

iti=#
```

5. Add/Alter foreign key constrains in your tables.

```
iti=# create table grades (student_id int,sub_id int,exam_id int,grade numeric,CONSTRAINT fk
_student FOREIGN KEY(student_id) REFERENCES student(id),CONSTRAINT fk_subject FOREIGN KEY(su
b_id) REFERENCES subject(id),CONSTRAINT fk_exam FOREIGN KEY(exam_id) REFERENCES exam(id)  );
```

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

```
iti=# select *
from student
where gender ='male' and birth_date < '1991-10-01';
 id | e_name | email | address | track_id | birth_date | gender
----+--------+-------+---------+----------+------------+--------
(0 rows)

iti=#
```

7. Display students' names that begin with A.

```
iti=# select e_name
iti-# from student
iti-# where e_name like 'A%';
 e_name
--------
 Amira
(1 row)

iti=#
```

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

```
iti=# select sub_name,max_score
iti-# from subject
iti-# order by max_score;
    sub_name    | max_score
----------------+-----------
 oop            |       100
 data structure |       100
 os             |       100
(3 rows)

iti=#
```

9. Display the subject with highest max score

```
iti=# select sub_name,max_score
iti-# from subject
iti-# order by max_score desc limit 1;
 sub_name | max_score
----------+-----------
 oop      |       100
(1 row)

iti=#
```

Lab 2

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

```
iti=# select count(*)
iti-# from student
iti-# where e_name = 'mohammed';
 count
-------
     0
(1 row)

iti=#
```

2. Display the number of males and females.

```
iti=# select gender,count(*)
from student
group by gender;
 gender | count
--------+-------
        |     3
(1 row)

iti=#
```

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

```
iti=# SELECT SPLIT_PART(e_name, ' ', 1) AS first_name, COUNT(*)
from student
group by first_name
having count(*) > 2;
 first_name | count
```

4. Display all students and track name they belong to.

```
iti=# select student.e_name, track.track_name
from student join track  on student.track_id = track.id;
 e_name | track_name
--------+------------
 Hajar  | c++
 Amira  | python
 Sara   | python
(3 rows)

iti=#
```

5. Display all students except those who are in OS track.

```
iti=# select student.*
iti-# from student join track on student.track_id = track.id
iti-# where track.track_name != 'os';
 id | e_name |      email      | address | track_id | birth_date | gender
----+--------+-----------------+---------+----------+------------+--------
  1 | Hajar  | hajar@gmail.com | Assuit  |        3 |            |
  2 | Amira  | amira@gmail.com | Assuit  |        1 |            |
  3 | Sara   | sara@gmail.com  | Alex    |        1 |            |
(3 rows)

iti=#
```