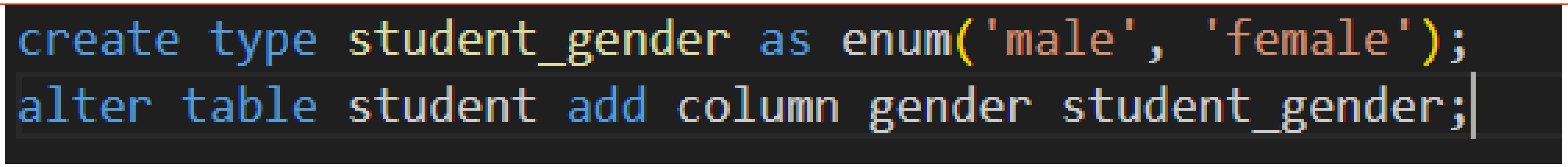
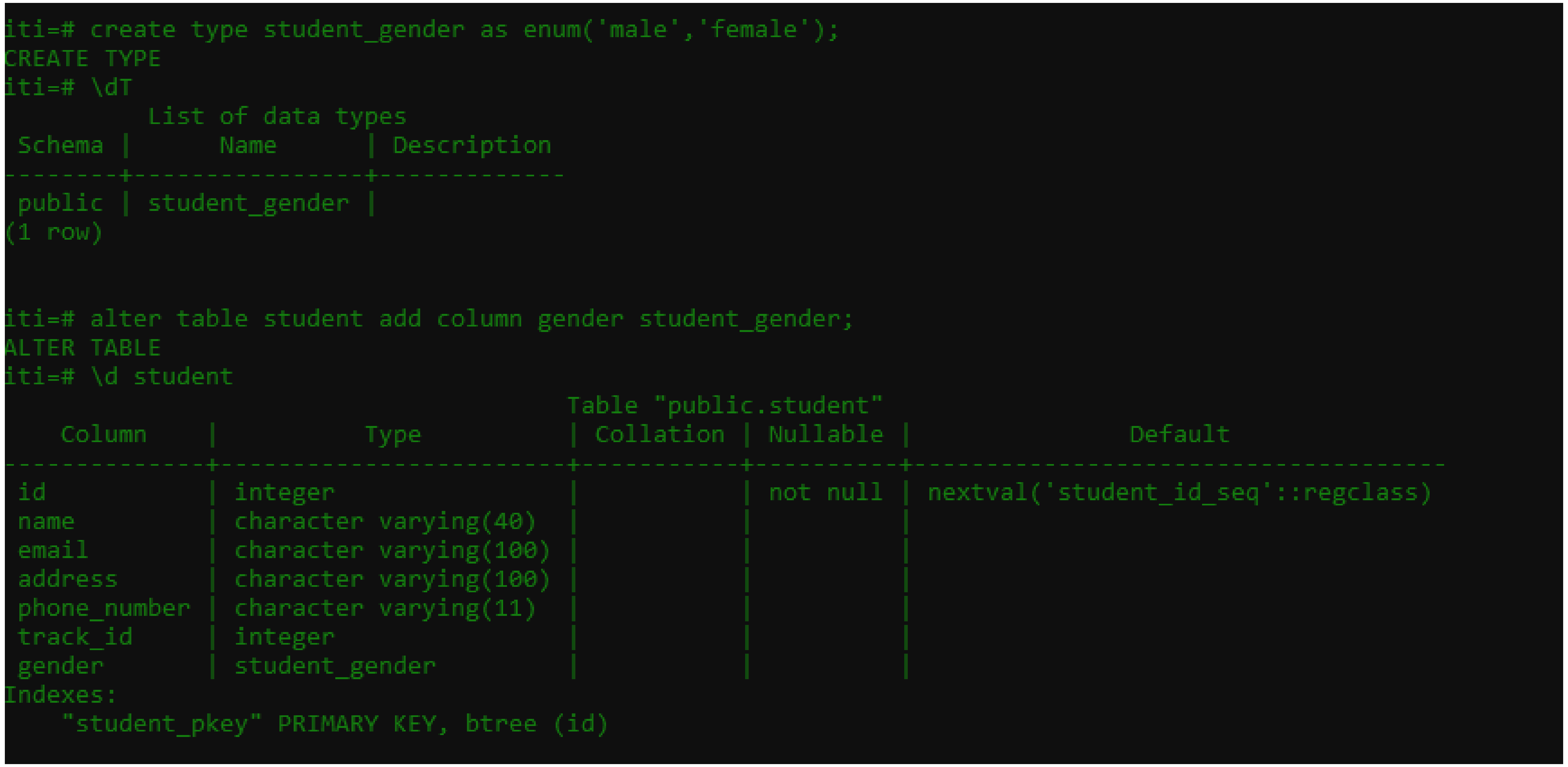
Day02 Assignment

postgreSQL

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



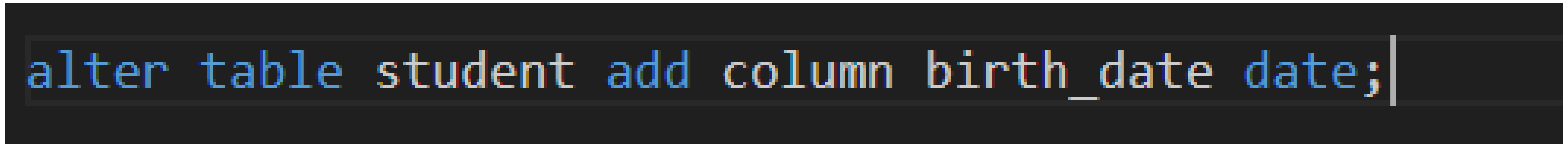
```
create type student_gender as enum('male', 'female');
alter table student add column gender student_gender;
```



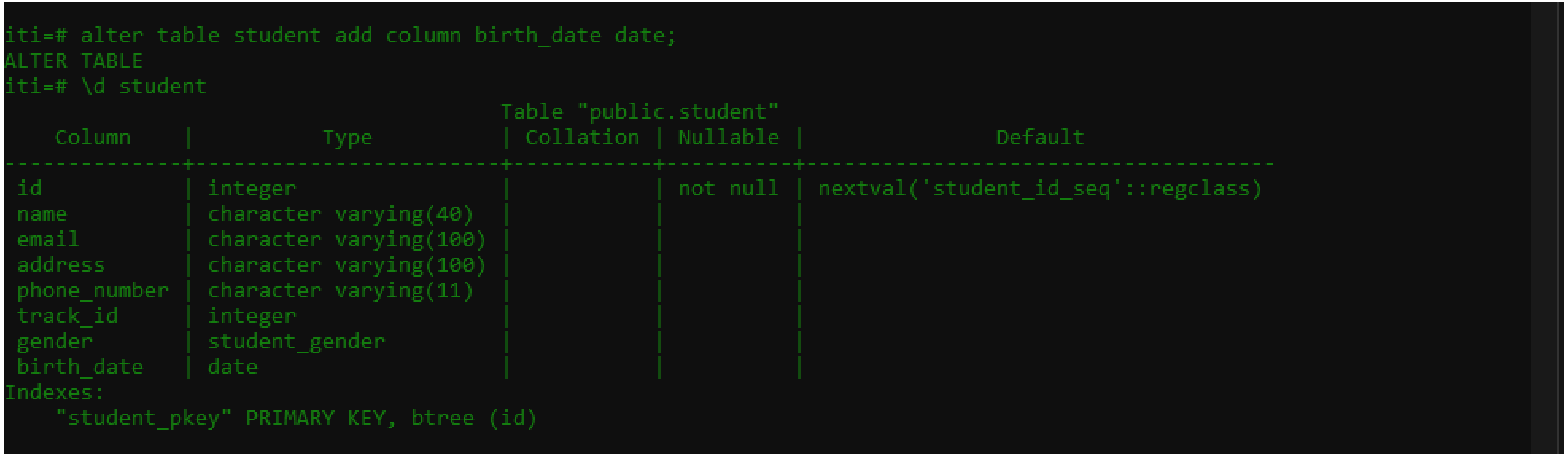
```
iti=# create type student_gender as enum('male','female');
CREATE TYPE
iti=# \dT
          List of data types
 Schema |      Name      | Description
--------+----------------+-------------
 public | student_gender |
(1 row)


iti=# alter table student add column gender student_gender;
ALTER TABLE
iti=# \d student
                                       Table "public.student"
    Column    |          Type          | Collation | Nullable |               Default
--------------+------------------------+-----------+----------+-------------------------------------
 id           | integer                |           | not null | nextval('student_id_seq'::regclass)
 name         | character varying(40)  |           |          |
 email        | character varying(100) |           |          |
 address      | character varying(100) |           |          |
 phone_number | character varying(11)  |           |          |
 track_id     | integer                |           |          |
 gender       | student_gender         |           |          |
Indexes:
    "student_pkey" PRIMARY KEY, btree (id)
```

2. Add birth date column for the student table

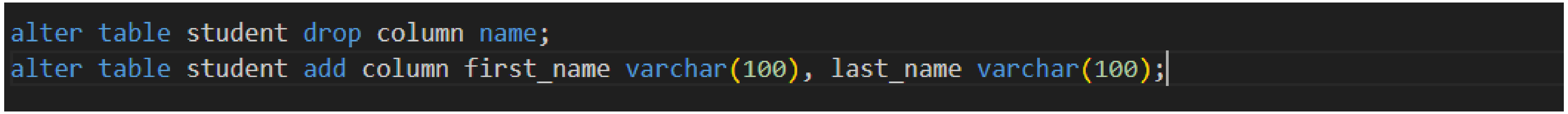


```
alter table student add column birth_date date;
```

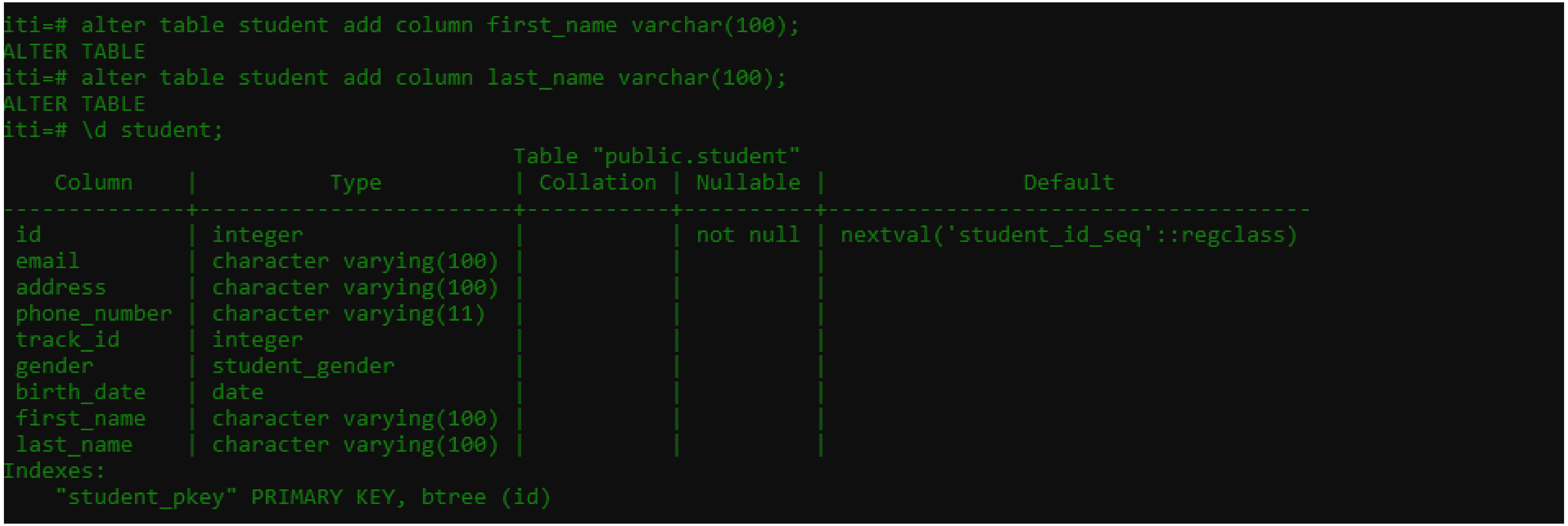


```
iti=# alter table student add column birth_date date;
ALTER TABLE
iti=# \d student
                                       Table "public.student"
    Column    |          Type          | Collation | Nullable |               Default
--------------+------------------------+-----------+----------+-------------------------------------
 id           | integer                |           | not null | nextval('student_id_seq'::regclass)
 name         | character varying(40)  |           |          |
 email        | character varying(100) |           |          |
 address      | character varying(100) |           |          |
 phone_number | character varying(11)  |           |          |
 track_id     | integer                |           |          |
 gender       | student_gender         |           |          |
 birth_date   | date                   |           |          |
Indexes:
    "student_pkey" PRIMARY KEY, btree (id)
```

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

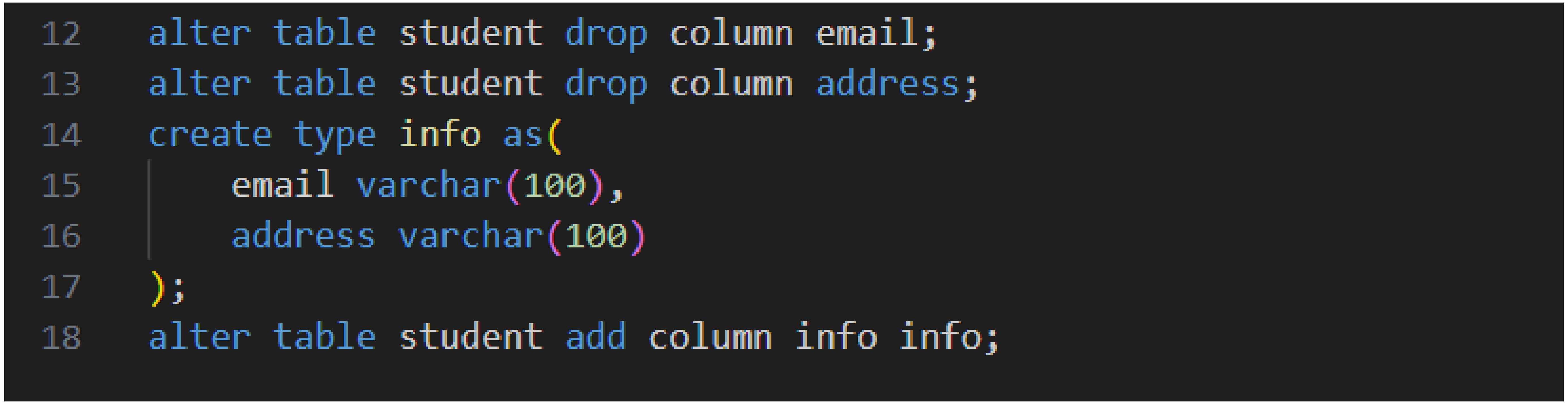


```
alter table student drop column name;
alter table student add column first_name varchar(100), last_name varchar(100);
```

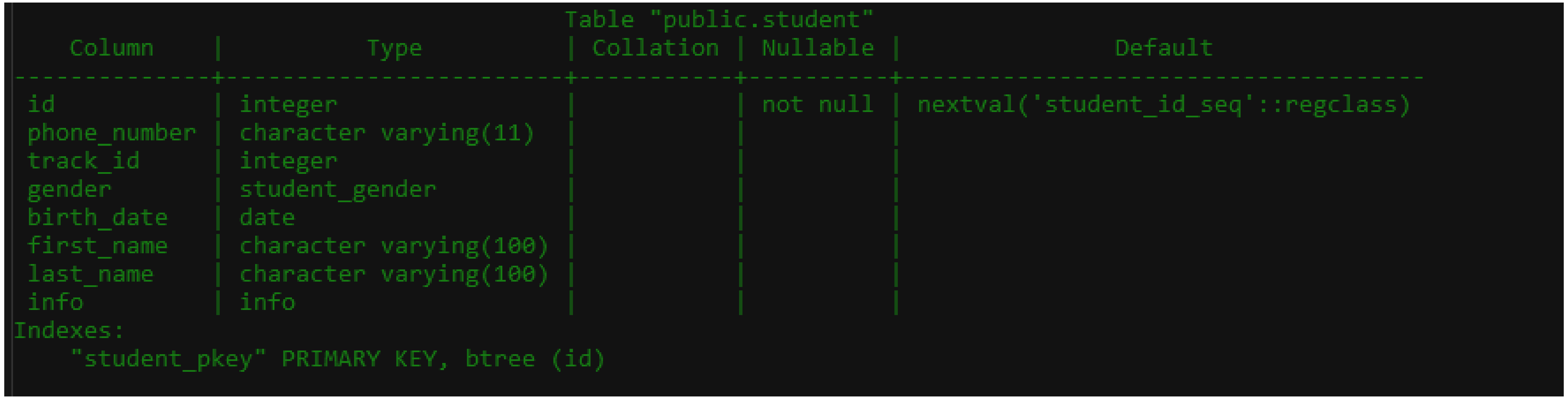


```
iti=# alter table student add column first_name varchar(100);
ALTER TABLE
iti=# alter table student add column last_name varchar(100);
ALTER TABLE
iti=# \d student;
                                       Table "public.student"
    Column    |          Type          | Collation | Nullable |                Default
--------------+------------------------+-----------+----------+-------------------------------------
 id           | integer                |           | not null | nextval('student_id_seq'::regclass)
 email        | character varying(100) |           |          |
 address      | character varying(100) |           |          |
 phone_number | character varying(11)  |           |          |
 track_id     | integer                |           |          |
 gender       | student_gender         |           |          |
 birth_date   | date                   |           |          |
 first_name   | character varying(100) |           |          |
 last_name    | character varying(100) |           |          |
Indexes:
    "student_pkey" PRIMARY KEY, btree (id)
```

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

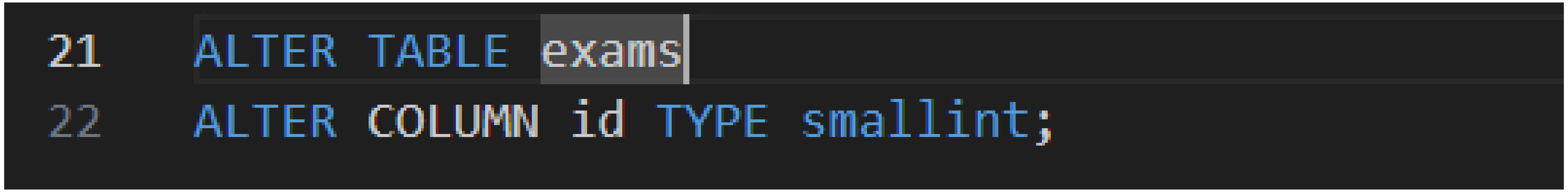


```
alter table student drop column email;
alter table student drop column address;
create type info as(
    email varchar(100),
    address varchar(100)
);
alter table student add column info info;
```

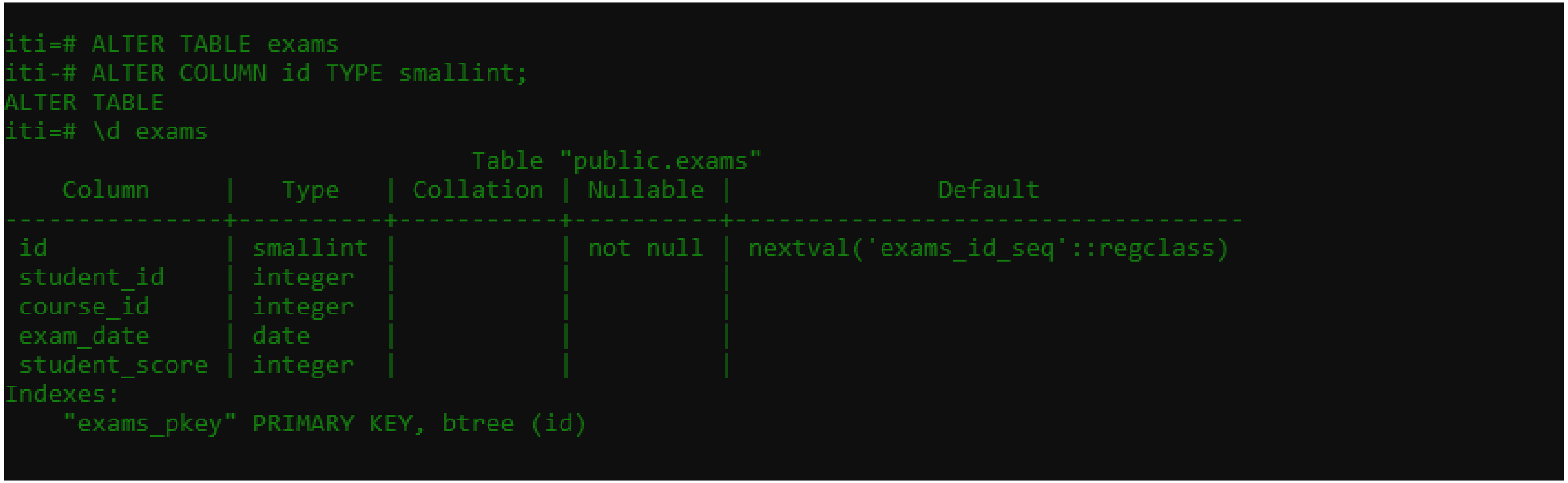


```
                                      Table "public.student"
    Column    |          Type          | Collation | Nullable |               Default
--------------+------------------------+-----------+----------+--------------------------------------
 id           | integer                |           | not null | nextval('student_id_seq'::regclass)
 phone_number | character varying(11)  |           |          |
 track_id     | integer                |           |          |
 gender       | student_gender         |           |          |
 birth_date   | date                   |           |          |
 first_name   | character varying(100) |           |          |
 last_name    | character varying(100) |           |          |
 info         | info                   |           |          |
Indexes:
    "student_pkey" PRIMARY KEY, btree (id)
```

5. Change any Serial Datatype at your tables to smallInt

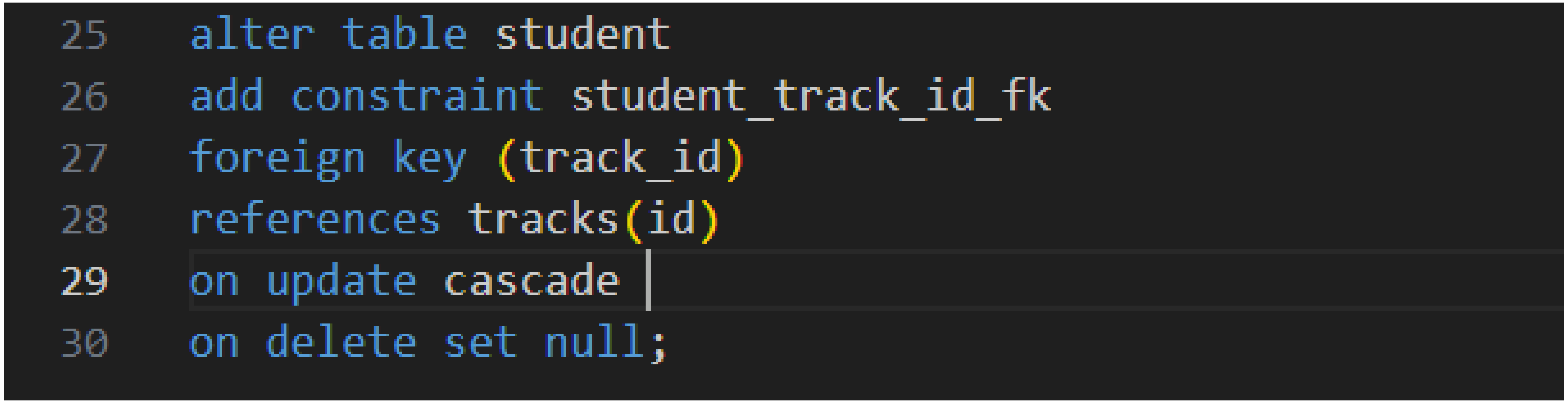


```
ALTER TABLE exams
ALTER COLUMN id TYPE smallint;
```

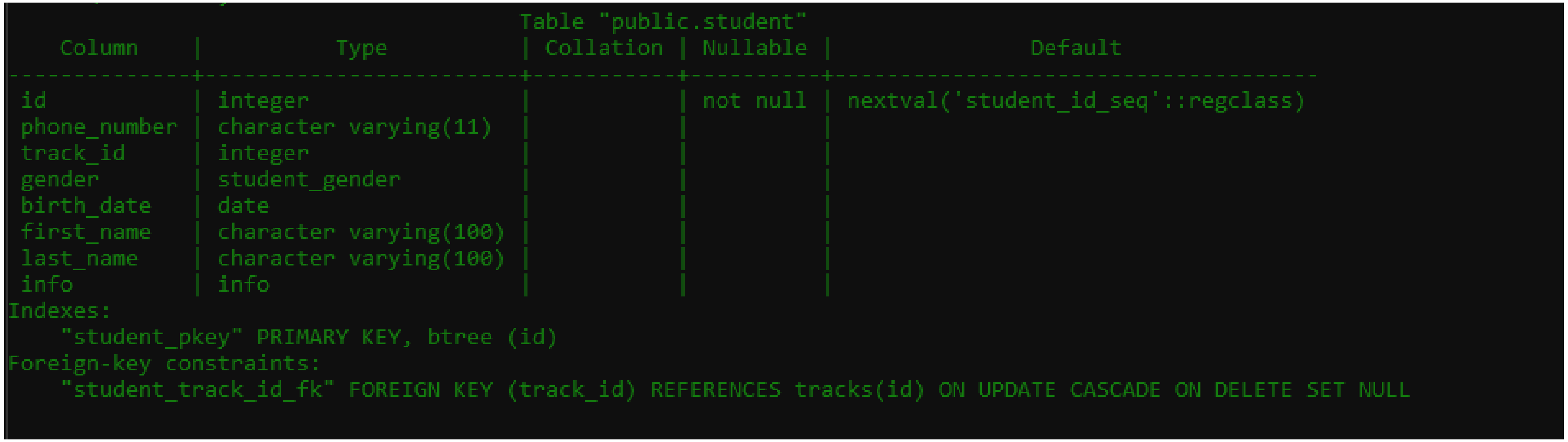


```
iti=# ALTER TABLE exams
iti-# ALTER COLUMN id TYPE smallint;
ALTER TABLE
iti=# \d exams
                                   Table "public.exams"
    Column     |   Type   | Collation | Nullable |              Default
---------------+----------+-----------+----------+-----------------------------------
 id            | smallint |           | not null | nextval('exams_id_seq'::regclass)
 student_id    | integer  |           |          |
 course_id     | integer  |           |          |
 exam_date     | date     |           |          |
 student_score | integer  |           |          |
Indexes:
    "exams_pkey" PRIMARY KEY, btree (id)
```

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

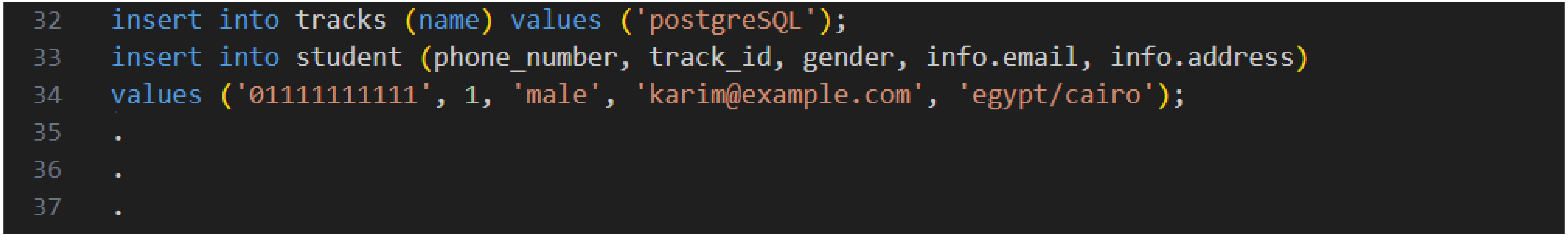


```sql
alter table student
add constraint student_track_id_fk
foreign key (track_id)
references tracks(id)
on update cascade
on delete set null;
```



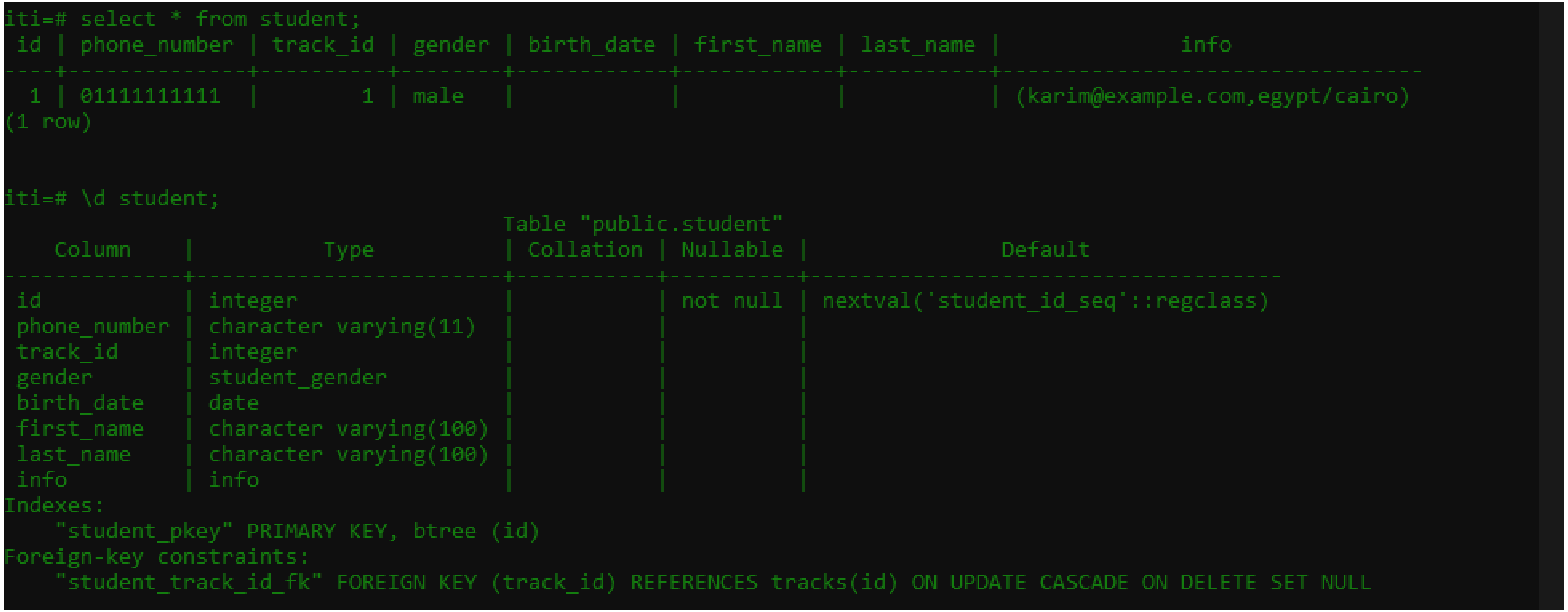
```
                                      Table "public.student"
    Column    |          Type          | Collation | Nullable |               Default
--------------+------------------------+-----------+----------+-------------------------------------
 id           | integer                |           | not null | nextval('student_id_seq'::regclass)
 phone_number | character varying(11)  |           |          |
 track_id     | integer                |           |          |
 gender       | student_gender         |           |          |
 birth_date   | date                   |           |          |
 first_name   | character varying(100) |           |          |
 last_name    | character varying(100) |           |          |
 info         | info                   |           |          |
Indexes:
    "student_pkey" PRIMARY KEY, btree (id)
Foreign-key constraints:
    "student_track_id_fk" FOREIGN KEY (track_id) REFERENCES tracks(id) ON UPDATE CASCADE ON DELETE SET NULL
```

7. Insert new data in all Tables.



```
insert into tracks (name) values ('postgreSQL');
insert into student (phone_number, track_id, gender, info.email, info.address)
values ('01111111111', 1, 'male', 'karim@example.com', 'egypt/cairo');
.
.
.
```

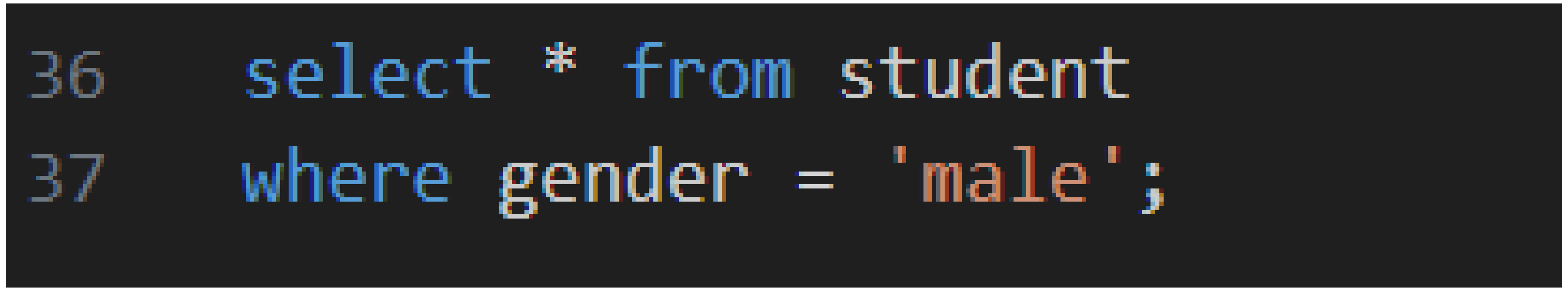
8. Display all students' information.



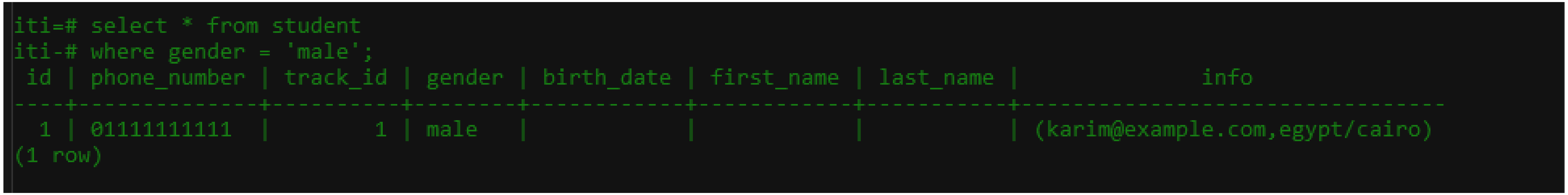
```
iti=# select * from student;
 id | phone_number | track_id | gender | birth_date | first_name | last_name |                info
----+--------------+----------+--------+------------+------------+-----------+---------------------------------
  1 | 01111111111  |        1 | male   |            |            |           | (karim@example.com,egypt/cairo)
(1 row)


iti=# \d student;
                                       Table "public.student"
    Column    |          Type          | Collation | Nullable |               Default
--------------+------------------------+-----------+----------+-------------------------------------
 id           | integer                |           | not null | nextval('student_id_seq'::regclass)
 phone_number | character varying(11)  |           |          |
 track_id     | integer                |           |          |
 gender       | student_gender         |           |          |
 birth_date   | date                   |           |          |
 first_name   | character varying(100) |           |          |
 last_name    | character varying(100) |           |          |
 info         | info                   |           |          |
Indexes:
    "student_pkey" PRIMARY KEY, btree (id)
Foreign-key constraints:
    "student_track_id_fk" FOREIGN KEY (track_id) REFERENCES tracks(id) ON UPDATE CASCADE ON DELETE SET NULL
```

9. Display male students only.

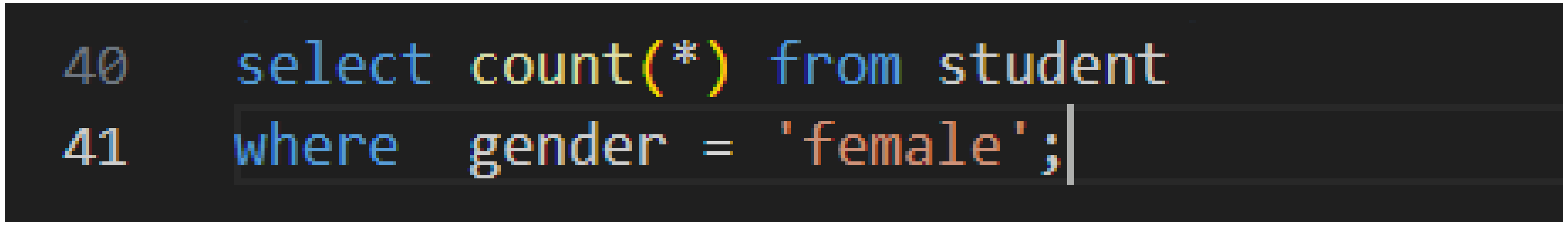


```
select * from student
where gender = 'male';
```

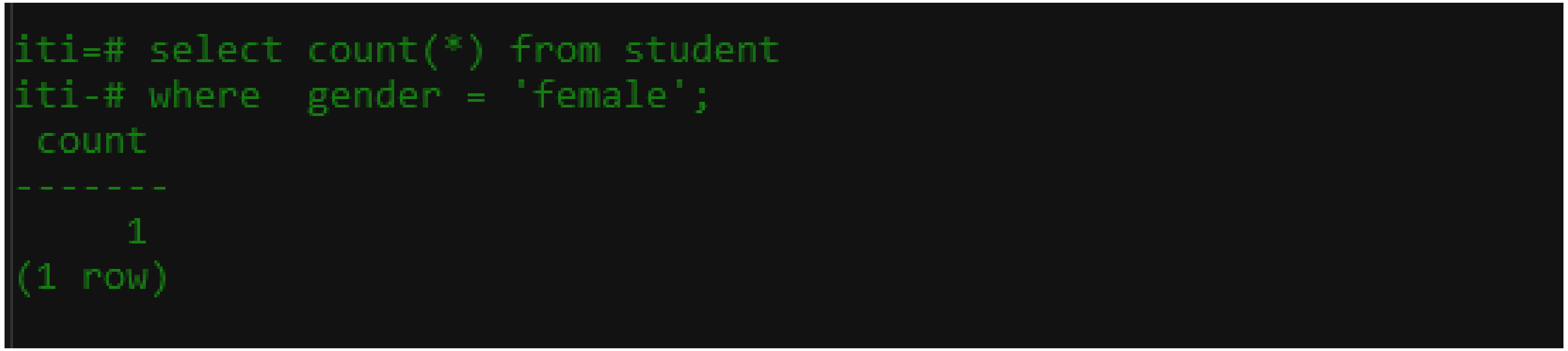


```
iti=# select * from student
iti-# where gender = 'male';
 id | phone_number | track_id | gender | birth_date | first_name | last_name |               info
----+--------------+----------+--------+------------+------------+-----------+-----------------------------------
  1 | 01111111111  |        1 | male   |            |            |           | (karim@example.com,egypt/cairo)
(1 row)
```

10. Display the number of female students.

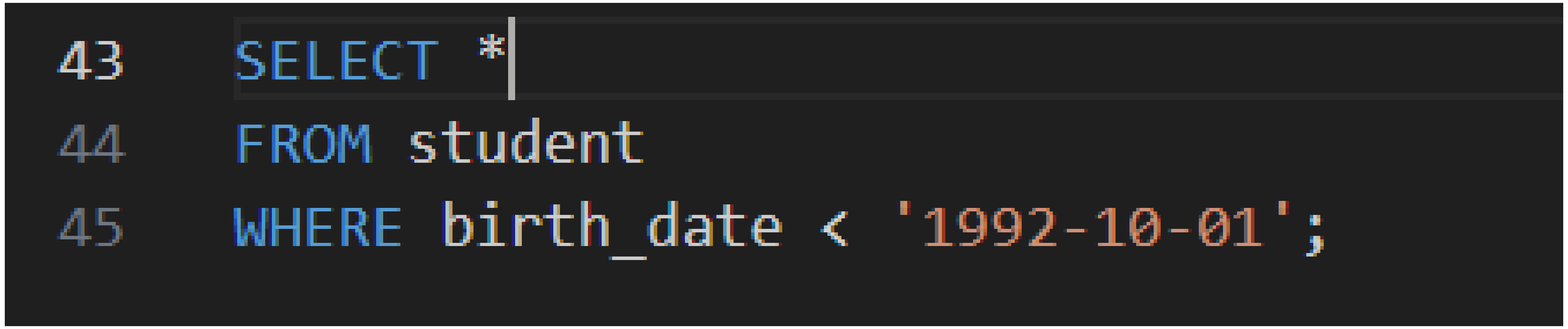


```sql
select count(*) from student
where  gender = 'female';
```

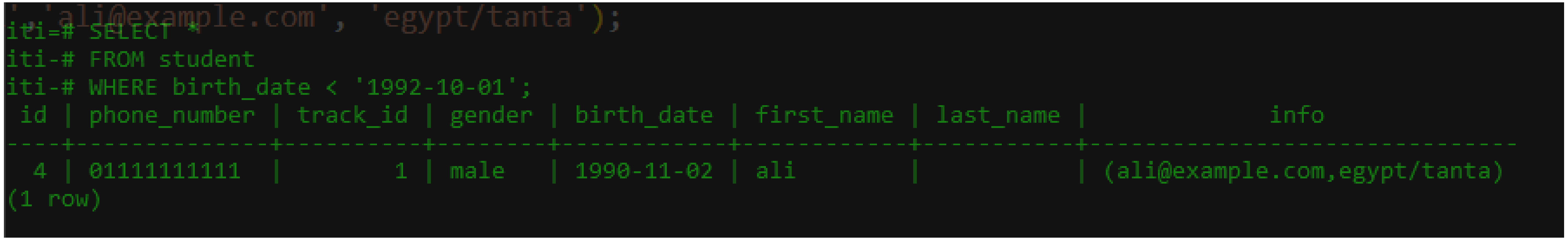


```
iti=# select count(*) from student
iti-# where  gender = 'female';
 count
-------
     1
(1 row)
```

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

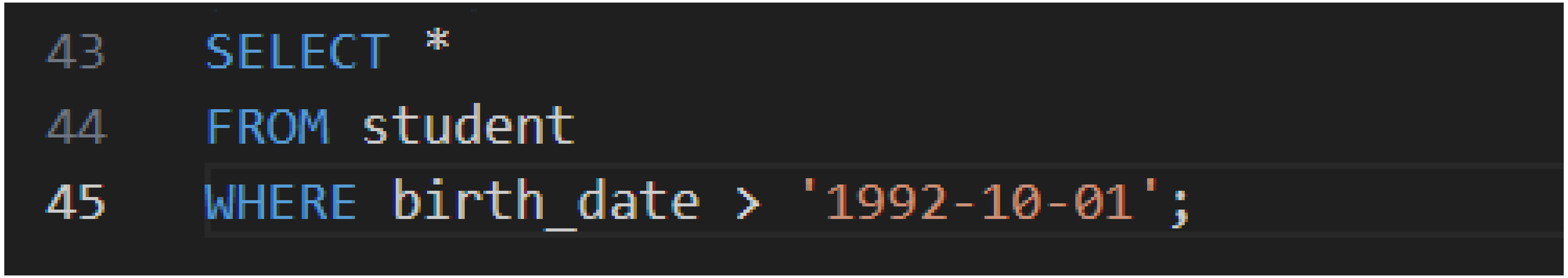


```sql
SELECT *
FROM student
WHERE birth_date < '1992-10-01';
```



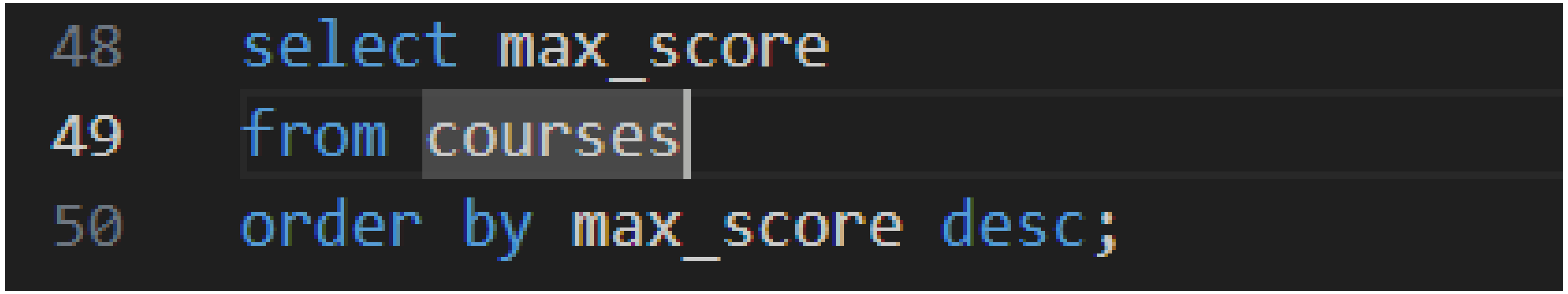
```
', 'ali@example.com', 'egypt/tanta');
iti=# SELECT *
iti-# FROM student
iti-# WHERE birth_date < '1992-10-01';
 id | phone_number | track_id | gender | birth_date | first_name | last_name |             info
----+--------------+----------+--------+------------+------------+-----------+-------------------------------
  4 | 01111111111  |        1 | male   | 1990-11-02 | ali        |           | (ali@example.com,egypt/tanta)
(1 row)
```

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

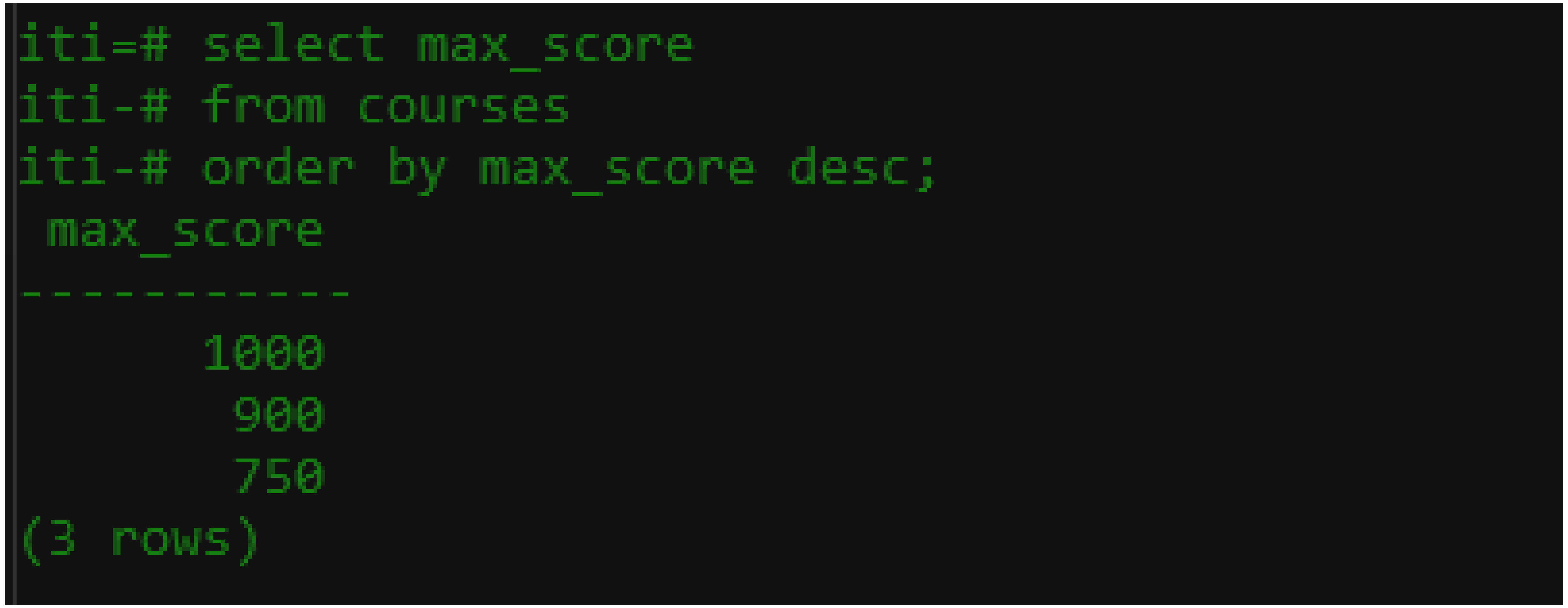


```sql
SELECT *
FROM student
WHERE birth_date > '1992-10-01';
```

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

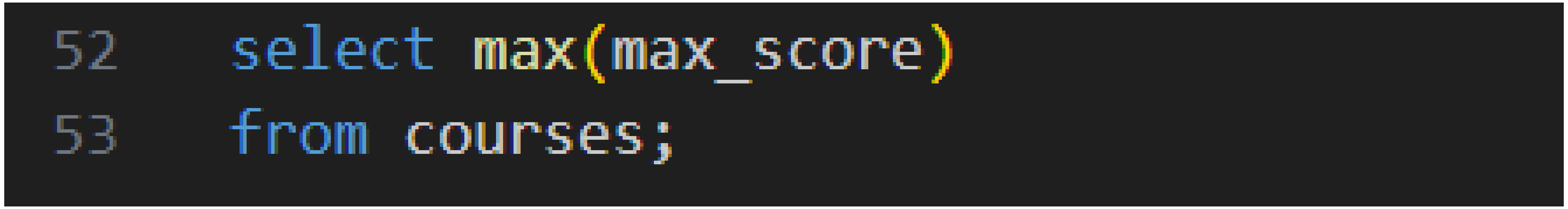


```
select max_score
from courses
order by max_score desc;
```

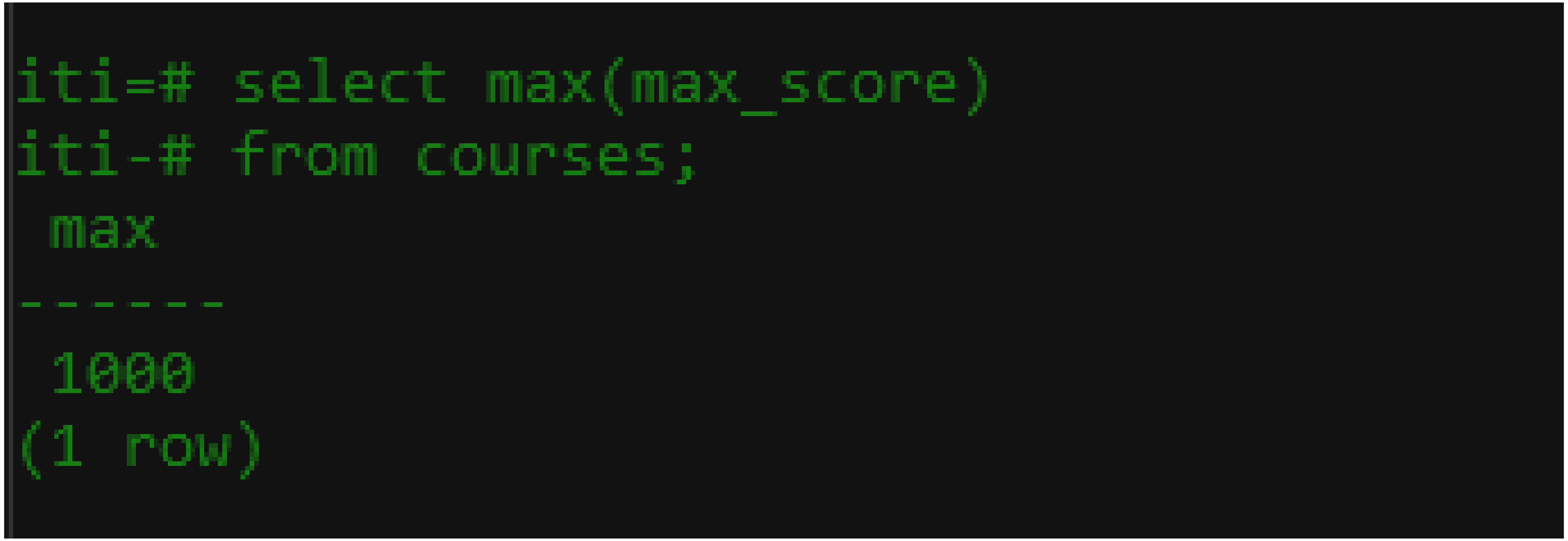


```
iti=# select max_score
iti-# from courses
iti-# order by max_score desc;
 max_score
-----------
      1000
       900
       750
(3 rows)
```

14. Display the subject with highest max score

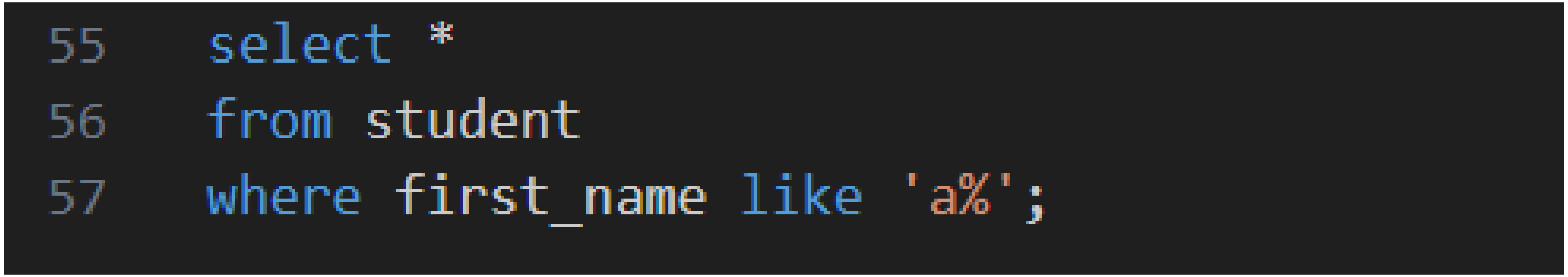


```sql
select max(max_score)
from courses;
```

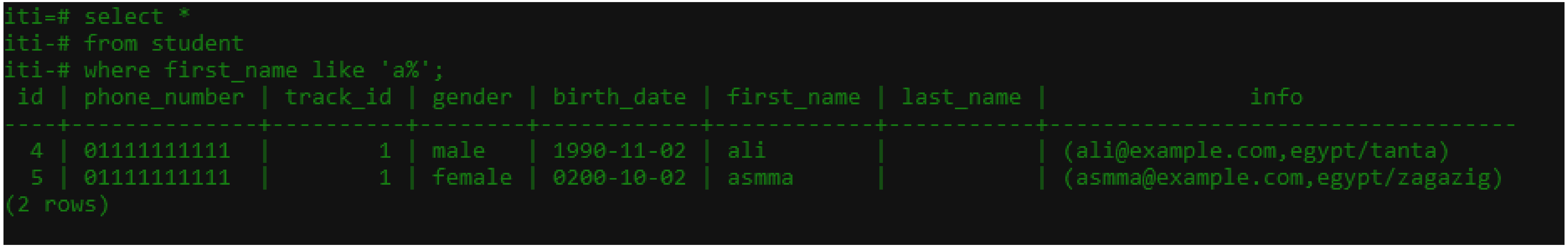


```
iti=# select max(max_score)
iti-# from courses;
 max
------
 1000
(1 row)
```

15. Display students' names that begin with A.

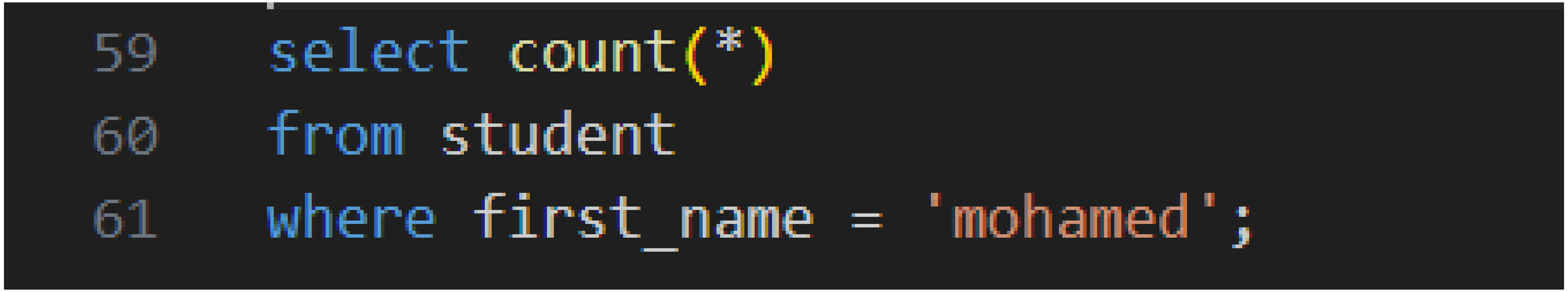


```sql
select *
from student
where first_name like 'a%';
```

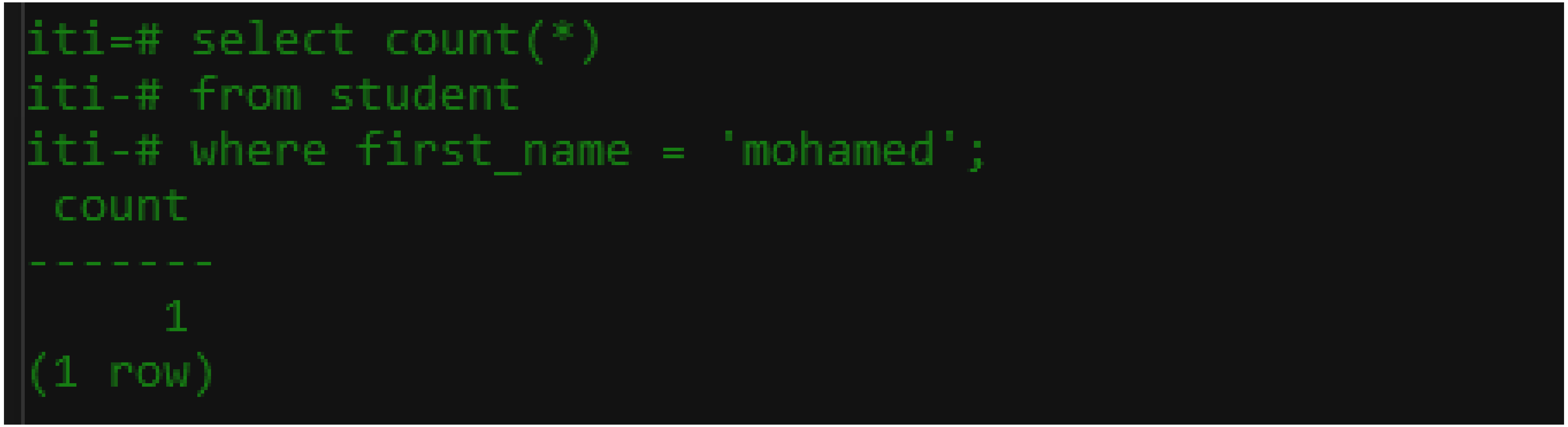


```
iti=# select *
iti-# from student
iti-# where first_name like 'a%';
 id | phone_number | track_id | gender | birth_date | first_name | last_name |               info
----+--------------+----------+--------+------------+------------+-----------+-----------------------------------
  4 | 01111111111  |        1 | male   | 1990-11-02 | ali        |           | (ali@example.com,egypt/tanta)
  5 | 01111111111  |        1 | female | 0200-10-02 | asmma      |           | (asmma@example.com,egypt/zagazig)
(2 rows)
```

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

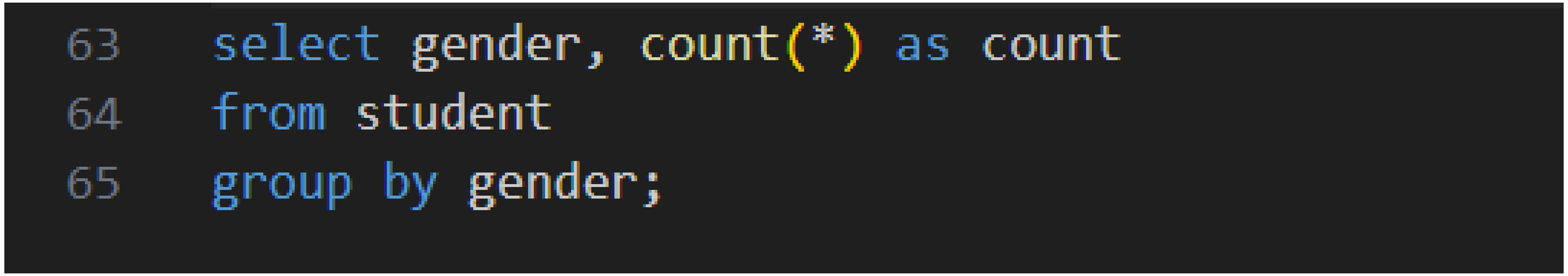


```
select count(*)
from student
where first_name = 'mohamed';
```

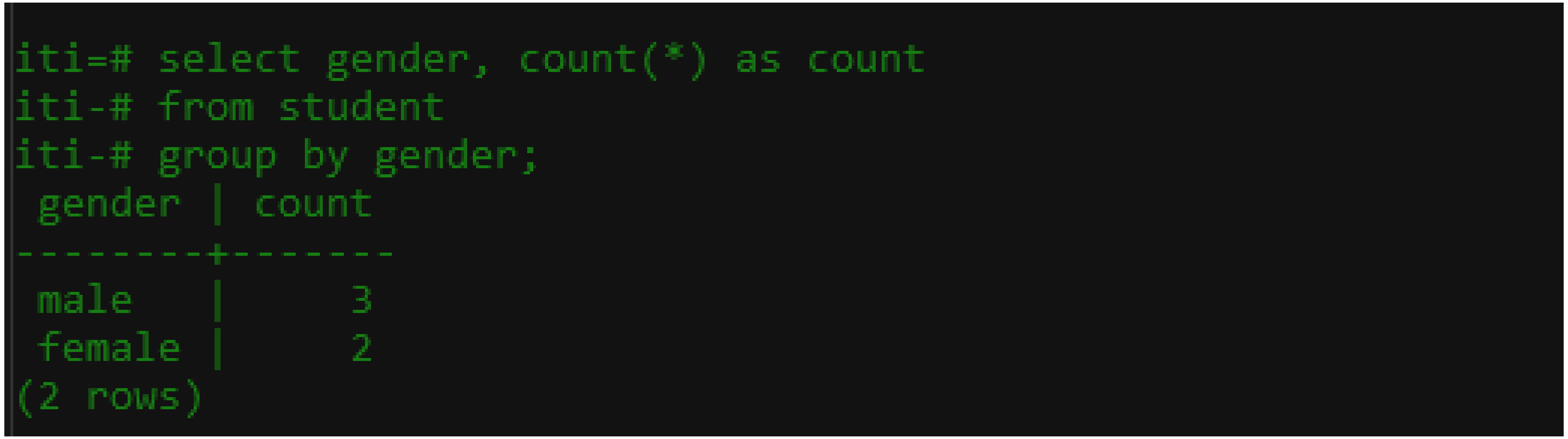


```
iti=# select count(*)
iti-# from student
iti-# where first_name = 'mohamed';
 count
-------
     1
(1 row)
```

17. Display the number of males and females.

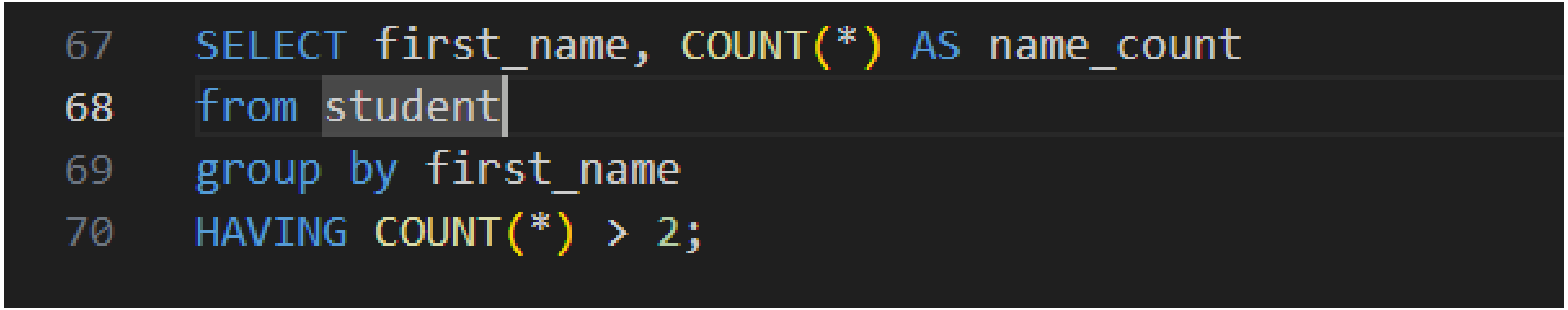


```sql
select gender, count(*) as count
from student
group by gender;
```

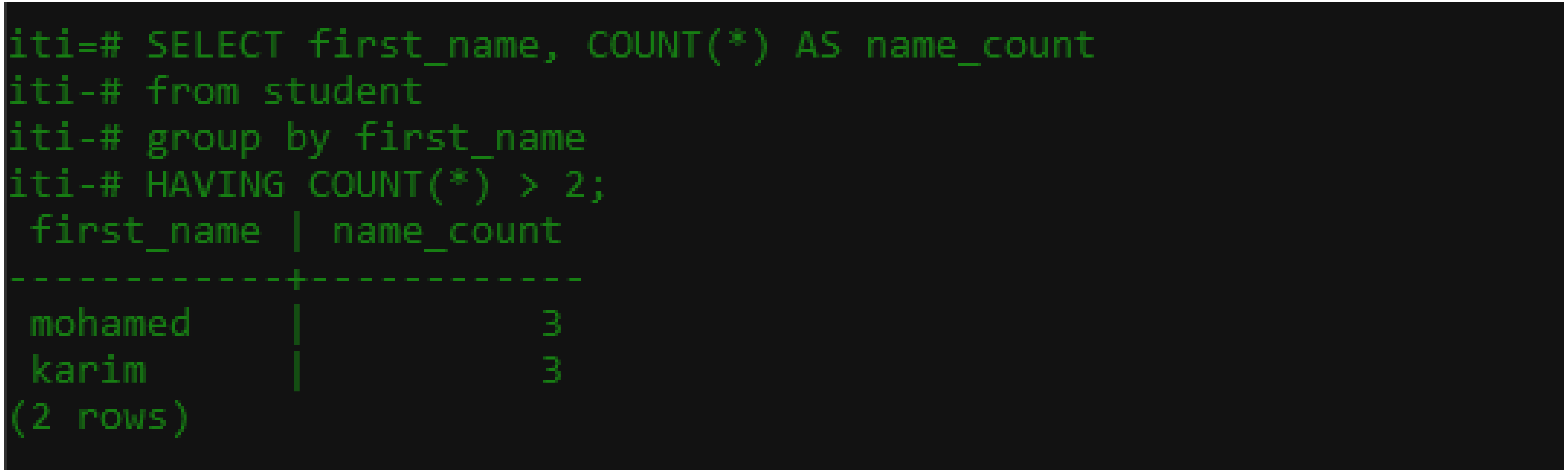


```
iti=# select gender, count(*) as count
iti-# from student
iti-# group by gender;
 gender | count
--------+-------
 male   |     3
 female |     2
(2 rows)
```

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

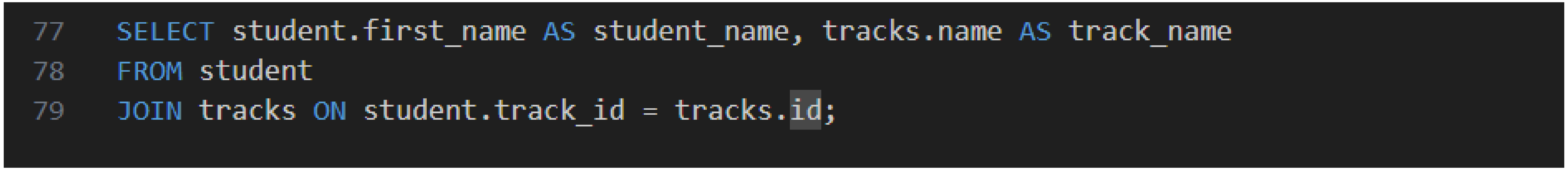


```sql
SELECT first_name, COUNT(*) AS name_count
from student
group by first_name
HAVING COUNT(*) > 2;
```

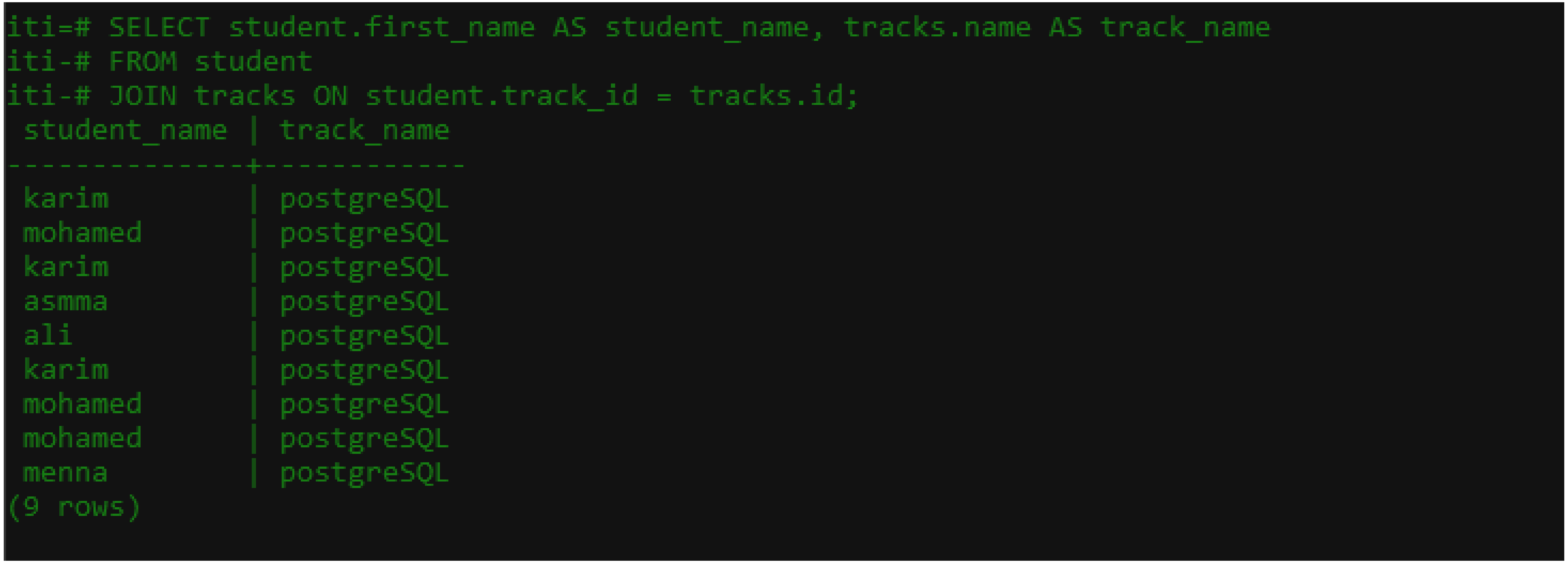


```
iti=# SELECT first_name, COUNT(*) AS name_count
iti-# from student
iti-# group by first_name
iti-# HAVING COUNT(*) > 2;
 first_name | name_count
------------+------------
 mohamed    |          3
 karim      |          3
(2 rows)
```

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

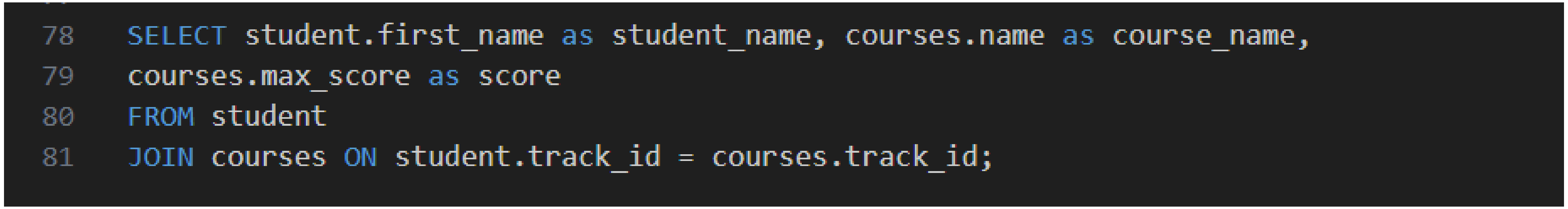


```
SELECT student.first_name AS student_name, tracks.name AS track_name
FROM student
JOIN tracks ON student.track_id = tracks.id;
```

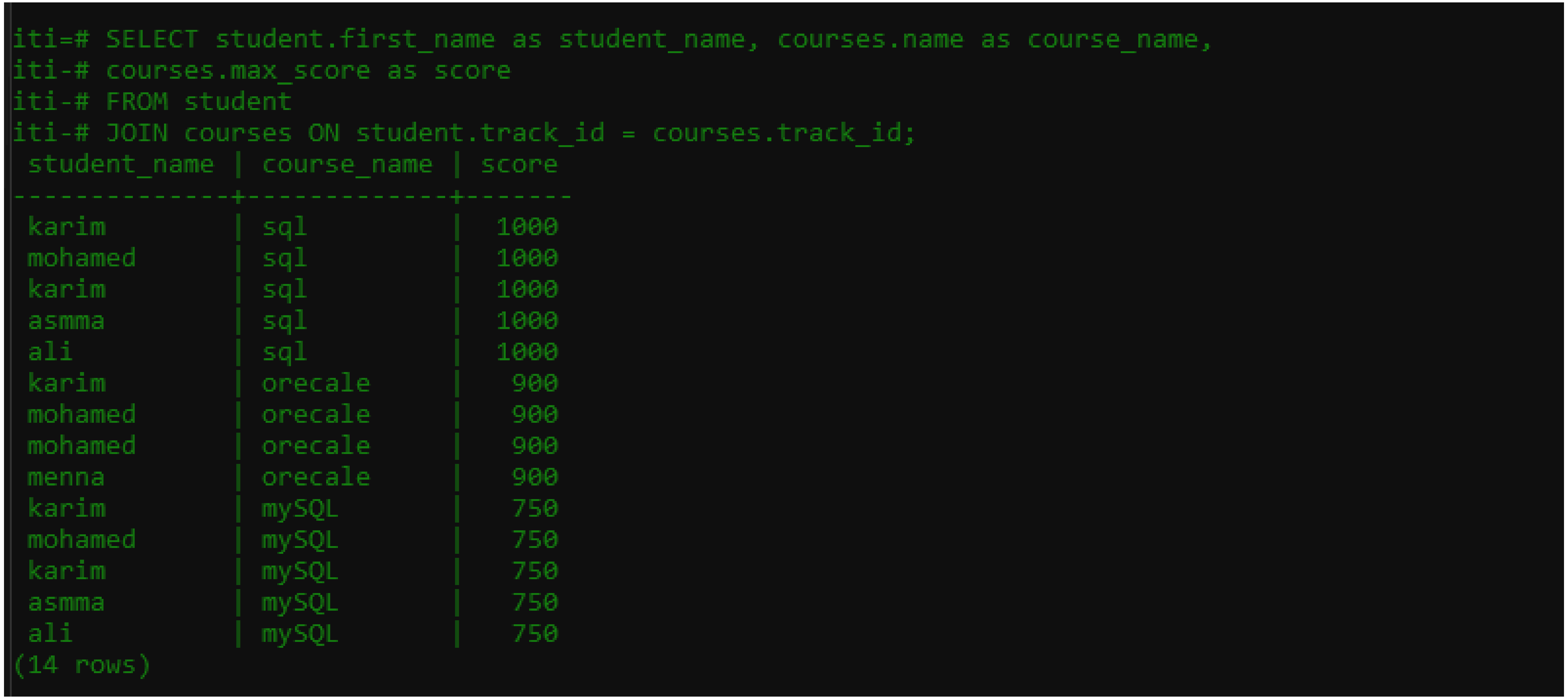


```
iti=# SELECT student.first_name AS student_name, tracks.name AS track_name
iti-# FROM student
iti-# JOIN tracks ON student.track_id = tracks.id;
 student_name | track_name
--------------+------------
 karim        | postgreSQL
 mohamed      | postgreSQL
 karim        | postgreSQL
 asmma        | postgreSQL
 ali          | postgreSQL
 karim        | postgreSQL
 mohamed      | postgreSQL
 mohamed      | postgreSQL
 menna        | postgreSQL
(9 rows)
```

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



```sql
SELECT student.first_name as student_name, courses.name as course_name,
courses.max_score as score
FROM student
JOIN courses ON student.track_id = courses.track_id;
```



```
iti=# SELECT student.first_name as student_name, courses.name as course_name,
iti-# courses.max_score as score
iti-# FROM student
iti-# JOIN courses ON student.track_id = courses.track_id;
 student_name | course_name | score
--------------+-------------+-------
 karim        | sql         |  1000
 mohamed      | sql         |  1000
 karim        | sql         |  1000
 asmma        | sql         |  1000
 ali          | sql         |  1000
 karim        | orecale     |   900
 mohamed      | orecale     |   900
 mohamed      | orecale     |   900
 menna        | orecale     |   900
 karim        | mySQL       |   750
 mohamed      | mySQL       |   750
 karim        | mySQL       |   750
 asmma        | mySQL       |   750
 ali          | mySQL       |   750
(14 rows)
```

Salford & Co.

THANK YOU

Kareim Magdy
Groub 3