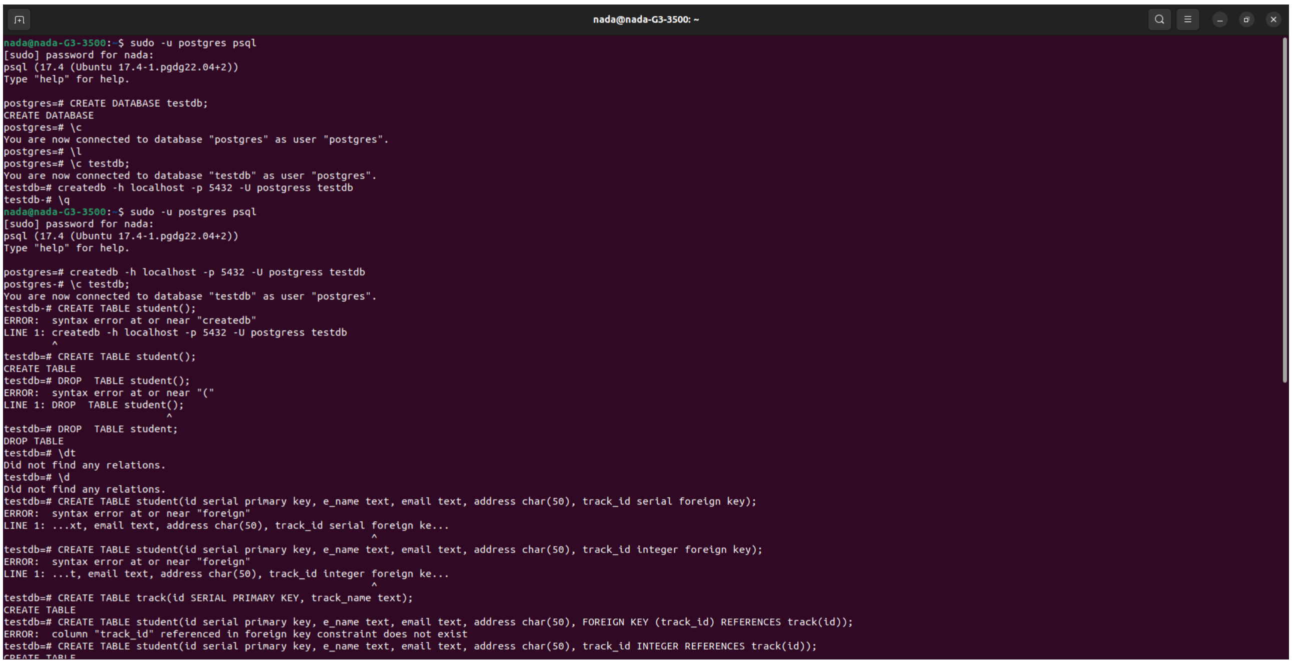
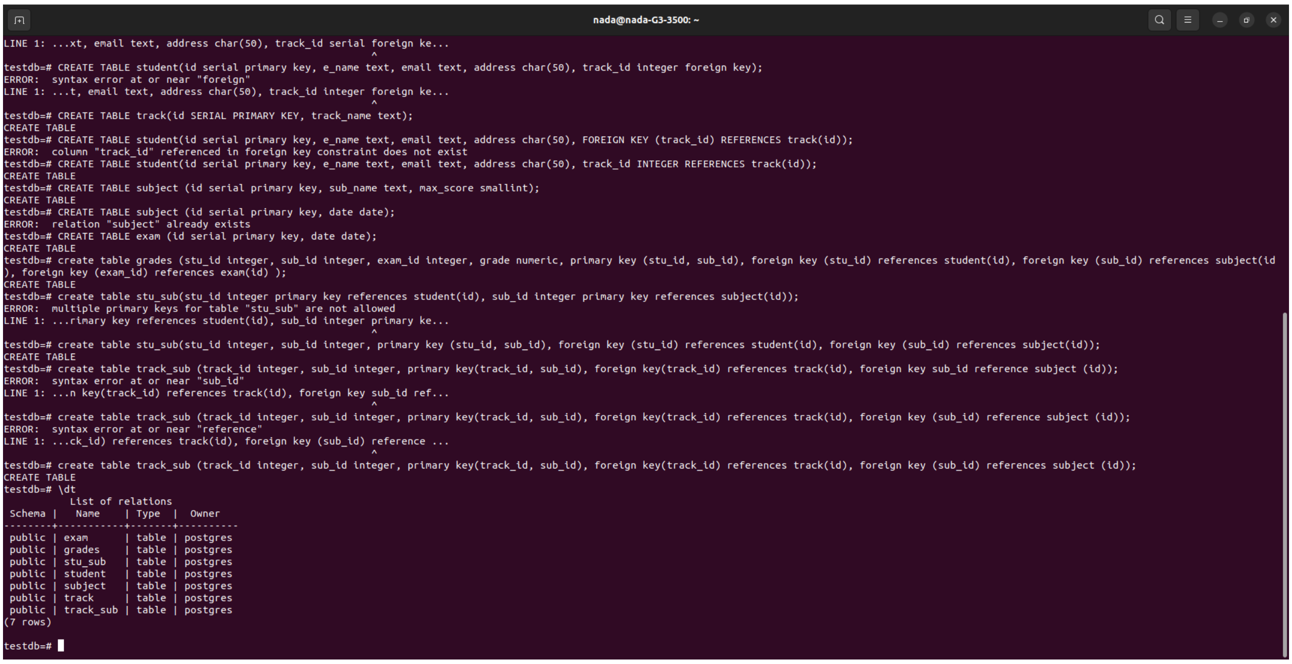
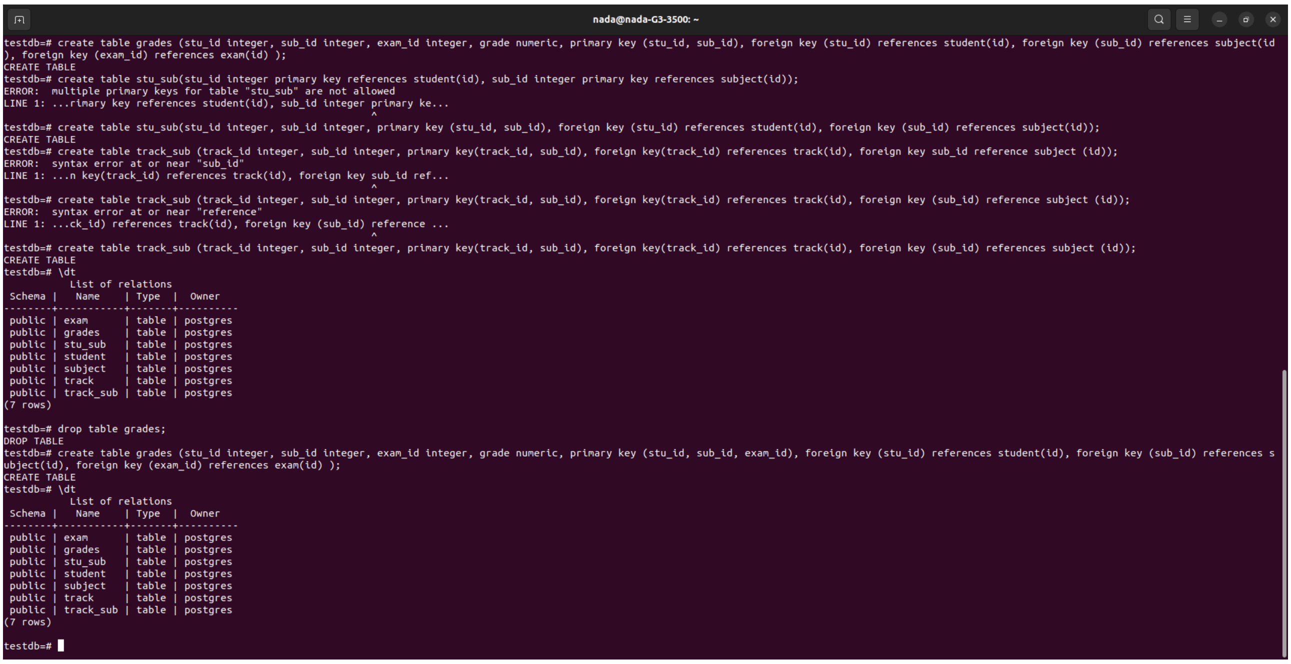
1.track (#id, track_name)
2.student (#id, e_name, email, address, *track_id)
3.subject (#id, sub_name, max_score)
4.exam (#id, date)
5.grades (#*stu_id, #*sub_id, *exam_id, grade)
6.stu_sub (#*stu_id, #*sub_id)
7.track_sub (#*track_id, #*sub_id)



```
nada@nada-G3-3500:~$ sudo -u postgres psql
[sudo] password for nada:
psql (17.4 (Ubuntu 17.4-1.pgdg22.04+2))
Type "help" for help.

postgres=# CREATE DATABASE testdb;
CREATE DATABASE
postgres=# \c
You are now connected to database "postgres" as user "postgres".
postgres=# \l
postgres=# \c testdb;
You are now connected to database "testdb" as user "postgres".
testdb=# createdb -h localhost -p 5432 -U postgress testdb
testdb-# \q
nada@nada-G3-3500:~$ sudo -u postgres psql
[sudo] password for nada:
psql (17.4 (Ubuntu 17.4-1.pgdg22.04+2))
Type "help" for help.

postgres=# createdb -h localhost -p 5432 -U postgress testdb
postgres-# \c testdb;
You are now connected to database "testdb" as user "postgres".
testdb-# CREATE TABLE student();
ERROR:  syntax error at or near "createdb"
LINE 1: createdb -h localhost -p 5432 -U postgress testdb
        ^
testdb=# CREATE TABLE student();
CREATE TABLE
testdb=# DROP  TABLE student();
ERROR:  syntax error at or near "("
LINE 1: DROP  TABLE student();
                           ^
testdb=# DROP  TABLE student;
DROP TABLE
testdb=# \dt
Did not find any relations.
testdb=# \d
Did not find any relations.
testdb=# CREATE TABLE student(id serial primary key, e_name text, email text, address char(50), track_id serial foreign key);
ERROR:  syntax error at or near "foreign"
LINE 1: ...xt, email text, address char(50), track_id serial foreign ke...
                                                             ^
testdb=# CREATE TABLE student(id serial primary key, e_name text, email text, address char(50), track_id integer foreign key);
ERROR:  syntax error at or near "foreign"
LINE 1: ...t, email text, address char(50), track_id integer foreign ke...
                                                             ^
testdb=# CREATE TABLE track(id SERIAL PRIMARY KEY, track_name text);
CREATE TABLE
testdb=# CREATE TABLE student(id serial primary key, e_name text, email text, address char(50), FOREIGN KEY (track_id) REFERENCES track(id));
ERROR:  column "track_id" referenced in foreign key constraint does not exist
testdb=# CREATE TABLE student(id serial primary key, e_name text, email text, address char(50), track_id INTEGER REFERENCES track(id));
CREATE TABLE
```

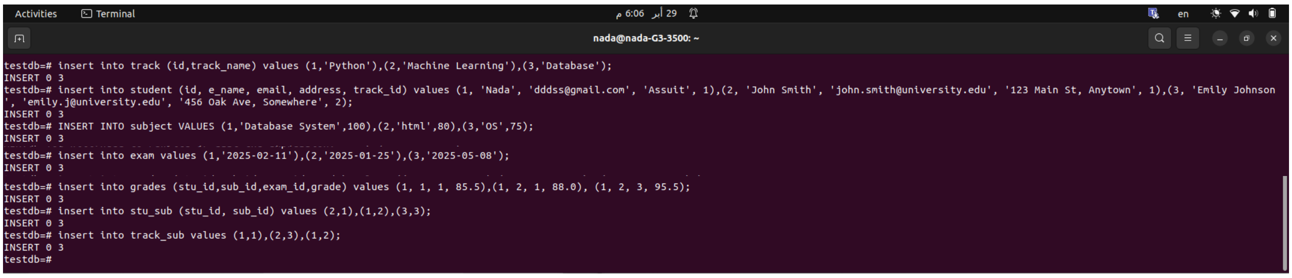


```
LINE 1: ...xt, email text, address char(50), track_id serial foreign ke...
                                                             ^
testdb=# CREATE TABLE student(id serial primary key, e_name text, email text, address char(50), track_id integer foreign key);
ERROR:  syntax error at or near "foreign"
LINE 1: ...t, email text, address char(50), track_id integer foreign ke...
                                                             ^
testdb=# CREATE TABLE track(id SERIAL PRIMARY KEY, track_name text);
CREATE TABLE
testdb=# CREATE TABLE student(id serial primary key, e_name text, email text, address char(50), FOREIGN KEY (track_id) REFERENCES track(id));
ERROR:  column "track_id" referenced in foreign key constraint does not exist
testdb=# CREATE TABLE student(id serial primary key, e_name text, email text, address char(50), track_id INTEGER REFERENCES track(id));
CREATE TABLE
testdb=# CREATE TABLE subject (id serial primary key, sub_name text, max_score smallint);
CREATE TABLE
testdb=# CREATE TABLE subject (id serial primary key, date date);
ERROR:  relation "subject" already exists
testdb=# CREATE TABLE exam (id serial primary key, date date);
CREATE TABLE
testdb=# create table grades (stu_id integer, sub_id integer, exam_id integer, grade numeric, primary key (stu_id, sub_id), foreign key (stu_id) references student(id), foreign key (sub_id) references subject(id
), foreign key (exam_id) references exam(id) );
CREATE TABLE
testdb=# create table stu_sub(stu_id integer primary key references student(id), sub_id integer primary key references subject(id));
ERROR:  multiple primary keys for table "stu_sub" are not allowed
LINE 1: ...rimary key references student(id), sub_id integer primary ke...
                                                             ^
testdb=# create table stu_sub(stu_id integer, sub_id integer, primary key (stu_id, sub_id), foreign key (stu_id) references student(id), foreign key (sub_id) references subject(id));
CREATE TABLE
testdb=# create table track_sub (track_id integer, sub_id integer, primary key(track_id, sub_id), foreign key(track_id) references track(id), foreign key sub_id reference subject (id));
ERROR:  syntax error at or near "sub_id"
LINE 1: ...n key(track_id) references track(id), foreign key sub_id ref...
                                                             ^
testdb=# create table track_sub (track_id integer, sub_id integer, primary key(track_id, sub_id), foreign key(track_id) references track(id), foreign key (sub_id) reference subject (id));
ERROR:  syntax error at or near "reference"
LINE 1: ...ck_id) references track(id), foreign key (sub_id) reference ...
                                                             ^
testdb=# create table track_sub (track_id integer, sub_id integer, primary key(track_id, sub_id), foreign key(track_id) references track(id), foreign key (sub_id) references subject (id));
CREATE TABLE
testdb=# \dt
           List of relations
 Schema |   Name    | Type  |  Owner
--------+-----------+-------+----------
 public | exam      | table | postgres
 public | grades    | table | postgres
 public | stu_sub   | table | postgres
 public | student   | table | postgres
 public | subject   | table | postgres
 public | track     | table | postgres
 public | track_sub | table | postgres
(7 rows)

testdb=# 
```



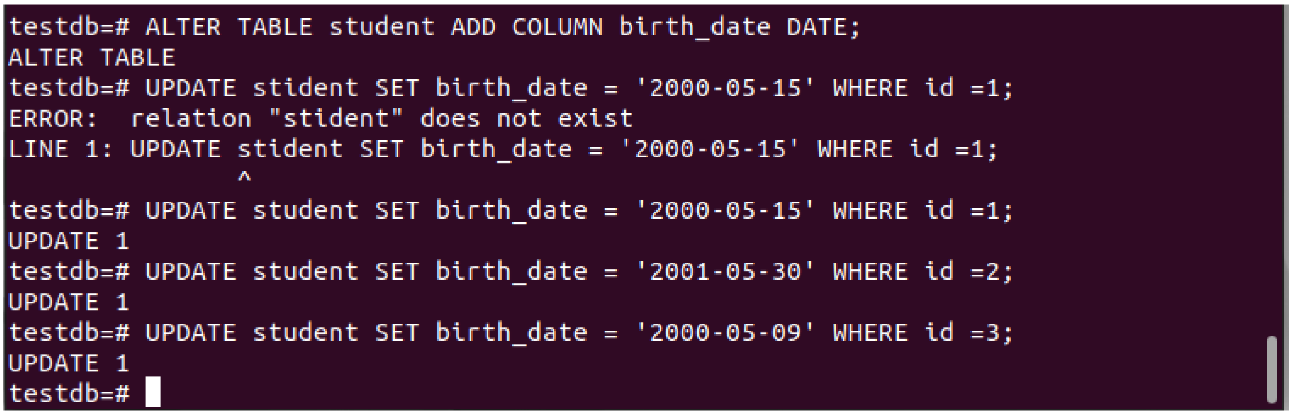
```
testdb=# create table grades (stu_id integer, sub_id integer, exam_id integer, grade numeric, primary key (stu_id, sub_id), foreign key (stu_id) references student(id), foreign key (sub_id) references subject(id
), foreign key (exam_id) references exam(id) );
CREATE TABLE
testdb=# create table stu_sub(stu_id integer primary key references student(id), sub_id integer primary key references subject(id));
ERROR:  multiple primary keys for table "stu_sub" are not allowed
LINE 1: ...rimary key references student(id), sub_id integer primary ke...
                                                             ^
testdb=# create table stu_sub(stu_id integer, sub_id integer, primary key (stu_id, sub_id), foreign key (stu_id) references student(id), foreign key (sub_id) references subject(id));
CREATE TABLE
testdb=# create table track_sub (track_id integer, sub_id integer, primary key(track_id, sub_id), foreign key(track_id) references track(id), foreign key sub_id reference subject (id));
ERROR:  syntax error at or near "sub_id"
LINE 1: ...n key(track_id) references track(id), foreign key sub_id ref...
                                                             ^
testdb=# create table track_sub (track_id integer, sub_id integer, primary key(track_id, sub_id), foreign key(track_id) references track(id), foreign key (sub_id) reference subject (id));
ERROR:  syntax error at or near "reference"
LINE 1: ...ck_id) references track(id), foreign key (sub_id) reference ...
                                                             ^
testdb=# create table track_sub (track_id integer, sub_id integer, primary key(track_id, sub_id), foreign key(track_id) references track(id), foreign key (sub_id) references subject (id));
CREATE TABLE
testdb=# \dt
          List of relations
 Schema |   Name    | Type  |  Owner
--------+-----------+-------+----------
 public | exam      | table | postgres
 public | grades    | table | postgres
 public | stu_sub   | table | postgres
 public | student   | table | postgres
 public | subject   | table | postgres
 public | track     | table | postgres
 public | track_sub | table | postgres
(7 rows)

testdb=# drop table grades;
DROP TABLE
testdb=# create table grades (stu_id integer, sub_id integer, exam_id integer, grade numeric, primary key (stu_id, sub_id, exam_id), foreign key (stu_id) references student(id), foreign key (sub_id) references s
ubject(id), foreign key (exam_id) references exam(id) );
CREATE TABLE
testdb=# \dt
          List of relations
 Schema |   Name    | Type  |  Owner
--------+-----------+-------+----------
 public | exam      | table | postgres
 public | grades    | table | postgres
 public | stu_sub   | table | postgres
 public | student   | table | postgres
 public | subject   | table | postgres
 public | track     | table | postgres
 public | track_sub | table | postgres
(7 rows)

testdb=# 
```

2. Insert at minimum 3 Rows at each table.



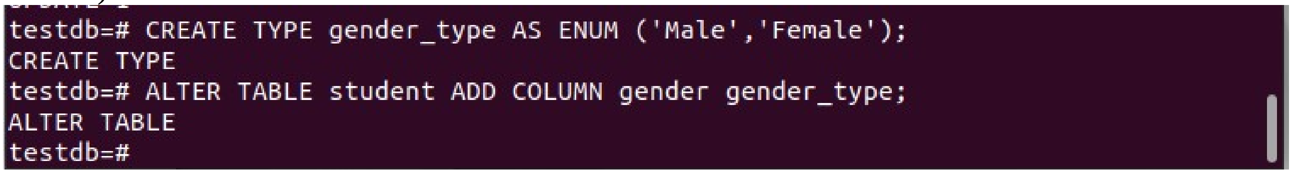
```
testdb=# insert into track (id,track_name) values (1,'Python'),(2,'Machine Learning'),(3,'Database');
INSERT 0 3
testdb=# insert into student (id, e_name, email, address, track_id) values (1, 'Nada', 'dddss@gmail.com', 'Assuit', 1),(2, 'John Smith', 'john.smith@university.edu', '123 Main St, Anytown', 1),(3, 'Emily Johnson
', 'emily.j@university.edu', '456 Oak Ave, Somewhere', 2);
INSERT 0 3
testdb=# INSERT INTO subject VALUES (1,'Database System',100),(2,'html',80),(3,'OS',75);
INSERT 0 3
testdb=# insert into exam values (1,'2025-02-11'),(2,'2025-01-25'),(3,'2025-05-08');
INSERT 0 3
testdb=# insert into grades (stu_id,sub_id,exam_id,grade) values (1, 1, 1, 85.5),(1, 2, 1, 88.0), (1, 2, 3, 95.5);
INSERT 0 3
testdb=# insert into stu_sub (stu_id, sub_id) values (2,1),(1,2),(3,3);
INSERT 0 3
testdb=# insert into track_sub values (1,1),(2,3),(1,2);
INSERT 0 3
testdb=#
```

3. Add birth date column for the student table.



```
testdb=# ALTER TABLE student ADD COLUMN birth_date DATE;
ALTER TABLE
testdb=# UPDATE stident SET birth_date = '2000-05-15' WHERE id =1;
ERROR:  relation "stident" does not exist
LINE 1: UPDATE stident SET birth_date = '2000-05-15' WHERE id =1;
               ^
testdb=# UPDATE student SET birth_date = '2000-05-15' WHERE id =1;
UPDATE 1
testdb=# UPDATE student SET birth_date = '2001-05-30' WHERE id =2;
UPDATE 1
testdb=# UPDATE student SET birth_date = '2000-05-09' WHERE id =3;
UPDATE 1
testdb=# 
```

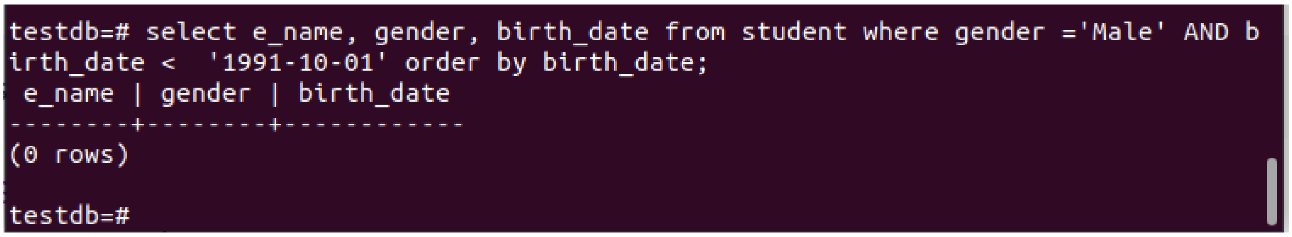
4. Add gender column which hold only 2 values (Male or Female).



```
testdb=# CREATE TYPE gender_type AS ENUM ('Male','Female');
CREATE TYPE
testdb=# ALTER TABLE student ADD COLUMN gender gender_type;
ALTER TABLE
testdb=#
```

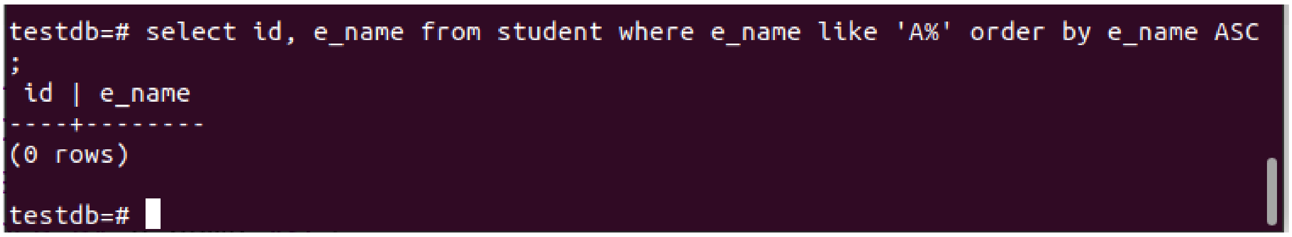
5. Add/Alter foreign key constrains in your tables.
Done in previous questions.

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



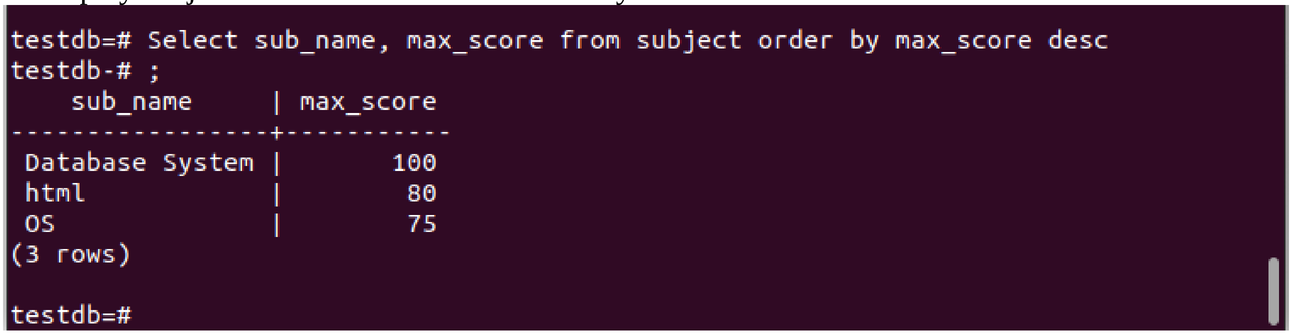
```
testdb=# select e_name, gender, birth_date from student where gender ='Male' AND b
irth_date <  '1991-10-01' order by birth_date;
 e_name | gender | birth_date
--------+--------+------------
(0 rows)

testdb=#
```

7.Display students’ names that begin with A.



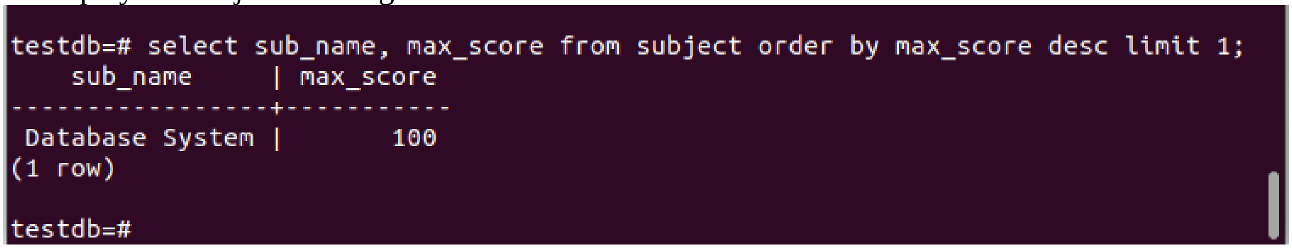
```
testdb=# select id, e_name from student where e_name like 'A%' order by e_name ASC
;
 id | e_name
----+--------
(0 rows)

testdb=#
```

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



```
testdb=# Select sub_name, max_score from subject order by max_score desc
testdb-# ;
    sub_name     | max_score
-----------------+-----------
 Database System |       100
 html            |        80
 OS              |        75
(3 rows)

testdb=#
```

9. Display the subject with highest max score



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

testdb=#
```