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).
5. Add/Alter foreign key constrains in your tables.
6. Display male students who are born before 1991-10-01.
7. Display students' names that begin with A.
8. Display subjects and their max score sorted by max score.
9. Display the subject with highest max

Lab 2

1. Display the number of students their name is "Mohammed"
2. Display the number of males and females.
3. Display the repeated first names and their counts if higher than 2.
4. Display all students and track name they belong to.
5. Display all students except those who are in OS track

```
ies  Terminal
ayooya@ayooya-HP-Pavilion-Laptop-14-ce3xxx:~$ sudo -u postgres psql
[sudo] password for ayooya:
could not change directory to "/home/ayooya": Permission denied
psql (14.17 (Ubuntu 14.17-0ubuntu0.22.04.1))
Type "help" for help.

postgres=# CREATE DATABASE student_tracking;
\c student_tracking
CREATE DATABASE
You are now connected to database "student_tracking" as user "postgres".
student_tracking=# CREATE TABLE track (
    id SERIAL PRIMARY KEY,
    track_name VARCHAR(50) NOT NULL
);
CREATE TABLE
student_tracking=# CREATE TABLE student (
    id SERIAL PRIMARY KEY,
    e_name VARCHAR(100) NOT NULL,
    email VARCHAR(100) UNIQUE,
    address TEXT,
    track_id INTEGER REFERENCES track(id)
);
CREATE TABLE
student_tracking=# CREATE TABLE subject (
    id SERIAL PRIMARY KEY,
    sub_name VARCHAR(100) NOT NULL,
    max_score INTEGER NOT NULL
);
CREATE TABLE
student_tracking=# CREATE TABLE subject (
    id SERIAL PRIMARY KEY,
    sub_name VARCHAR(100) NOT NULL,
    max_score INTEGER NOT NULL
);
ERROR:  relation "subject" already exists
student_tracking=# CREATE TABLE stu_sub (
    stu_id INTEGER REFERENCES student(id),
    sub_id INTEGER REFERENCES subject(id),
    PRIMARY KEY (stu_id, sub_id)
);
CREATE TABLE
student_tracking=# CREATE TABLE track_sub (
    track_id INTEGER REFERENCES track(id),
    sub_id INTEGER REFERENCES subject(id),
    PRIMARY KEY (track_id, sub_id)
);
CREATE TABLE
```

```
student_tracking=# INSERT INTO track (track_name) VALUES
('OS'), ('Database'), ('Programming'), ('Networking');
INSERT 0 4
student_tracking=# INSERT INTO student (e_name, email, address, track_id) VALUES
('Mohammed Ali', 'mohammed@example.com', '123 Main St', 1),
('Ahmed Hassan', 'ahmed@example.com', '456 Oak Ave', 2),
('Aisha Mohammed', 'aisha@example.com', '789 Pine Rd', 3),
('Fatima Ahmed', 'fatima@example.com', '321 Elm St', 1),
('Mohammed Omar', 'omar@example.com', '654 Maple Dr', 2),
('Ali Mohammed', 'ali@example.com', '987 Cedar Ln', 4);
INSERT 0 6
student_tracking=# INSERT INTO subject (sub_name, max_score) VALUES
('Database Systems', 100),
('Operating Systems', 90),
('Python Programming', 95),
('Computer Networks', 85);
INSERT 0 4
student_tracking=# INSERT INTO exam (date) VALUES
('2023-01-15'), ('2023-02-20'), ('2023-03-10');
ERROR:  relation "exam" does not exist
LINE 1: INSERT INTO exam (date) VALUES
                    ^
student_tracking=# CREATE TABLE exam (
    id SERIAL PRIMARY KEY,
    date DATE NOT NULL
);
CREATE TABLE
student_tracking=# INSERT INTO exam (date) VALUES
('2023-01-15'), ('2023-02-20'), ('2023-03-10');
INSERT 0 3
student_tracking=# INSERT INTO grades (stu_id, sub_id, exam_id, grade) VALUES
(1, 1, 1, 85), (1, 2, 1, 75), (2, 1, 1, 90),
(2, 3, 2, 88), (3, 4, 3, 82), (4, 2, 2, 78);
ERROR:  relation "grades" does not exist
LINE 1: INSERT INTO grades (stu_id, sub_id, exam_id, grade) VALUES
                    ^
student_tracking=# INSERT INTO stu_sub (stu_id, sub_id) VALUES
(1, 1), (1, 2), (2, 1), (2, 3), (3, 4), (4, 2);
INSERT 0 6
student_tracking=# INSERT INTO track_sub (track_id, sub_id) VALUES
(1, 2), (2, 1), (3, 3), (4, 4);
INSERT 0 4
student_tracking=# ALTER TABLE student ADD COLUMN birth_date DATE;
UPDATE student SET birth_date =
CASE id
    WHEN 1 THEN '1990-05-15'
    WHEN 2 THEN '1992-08-20'
    WHEN 3 THEN '1993-01-10'
    WHEN 4 THEN '1995-11-25'
    WHEN 5 THEN '1989-09-30'
    WHEN 6 THEN '1991-07-05'
END;
ALTER TABLE
```

```
ERROR:  column "birth_date" of relation "student" already exists
student_tracking=# UPDATE student SET birth_date =
CASE id
    WHEN 1 THEN '1990-05-15'::date
    WHEN 2 THEN '1992-08-20'::date
    WHEN 3 THEN '1993-01-10'::date
    WHEN 4 THEN '1995-11-25'::date
    WHEN 5 THEN '1989-09-30'::date
    WHEN 6 THEN '1991-07-05'::date
END;
UPDATE 6
student_tracking=# ALTER TABLE student ADD COLUMN gender VARCHAR(6) CHECK (gender IN ('Male', 'Female'));
ALTER TABLE
student_tracking=# UPDATE student SET gender =
CASE id
    WHEN 1 THEN 'Male'
    WHEN 2 THEN 'Male'
    WHEN 3 THEN 'Female'
    WHEN 4 THEN 'Female'
    WHEN 5 THEN 'Male'
    WHEN 6 THEN 'Male'
END;
UPDATE 6
student_tracking=# SELECT e_name, birth_date FROM student
WHERE gender = 'Male' AND birth_date < '1991-10-01';
    e_name     | birth_date
---------------+------------
 Mohammed Ali  | 1990-05-15
 Mohammed Omar | 1989-09-30
 Ali Mohammed  | 1991-07-05
(3 rows)

student_tracking=# SELECT e_name FROM student WHERE e_name LIKE 'A%';
     e_name
----------------
 Ahmed Hassan
 Aisha Mohammed
 Ali Mohammed
(3 rows)

student_tracking=# SELECT e_name FROM student WHERE e_name LIKE 'A%';
     e_name
----------------
 Ahmed Hassan
 Aisha Mohammed
 Ali Mohammed
(3 rows)
```

```
student_tracking=# SELECT sub_name, max_score FROM subject
ORDER BY max_score DESC LIMIT 1;
     sub_name     | max_score
------------------+-----------
 Database Systems |       100
(1 row)

student_tracking=# SELECT COUNT(*) FROM student WHERE e_name LIKE '%Mohammed%';
 count
-------
     4
(1 row)

student_tracking=# SELECT gender, COUNT(*) FROM student GROUP BY gender;
 gender | count
--------+-------
 Female |     2
 Male   |     4
(2 rows)

student_tracking=# SELECT
    split_part(e_name, ' ', 1) AS first_name,
    COUNT(*)
FROM student
GROUP BY first_name
HAVING COUNT(*) > 2;
 first_name | count
------------+-------
(0 rows)

student_tracking=# SELECT s.e_name, t.track_name
FROM student s JOIN track t ON s.track_id = t.id;
     e_name     | track_name
----------------+------------
 Fatima Ahmed   | OS
 Mohammed Ali   | OS
 Mohammed Omar  | Database
 Ahmed Hassan   | Database
 Aisha Mohammed | Programming
 Ali Mohammed   | Networking
(6 rows)

student_tracking=# SELECT s.e_name, t.track_name
FROM student s JOIN track t ON s.track_id = t.id;
     e_name     | track_name
----------------+------------
 Fatima Ahmed   | OS
 Mohammed Ali   | OS
 Mohammed Omar  | Database
 Ahmed Hassan   | Database
 Aisha Mohammed | Programming
 Ali Mohammed   | Networking
(6 rows)
```

```
student_tracking=# SELECT s.e_name, t.track_name
FROM student s JOIN track t ON s.track_id = t.id
WHERE t.track_name != 'OS';
     e_name     | track_name
----------------+-------------
 Mohammed Omar  | Database
 Ahmed Hassan   | Database
 Aisha Mohammed | Programming
 Ali Mohammed   | Networking
(4 rows)

student_tracking=# \d
              List of relations
 Schema |      Name      |   Type   |  Owner
--------+----------------+----------+----------
 public | exam           | table    | postgres
 public | exam_id_seq    | sequence | postgres
 public | stu_sub        | table    | postgres
 public | student        | table    | postgres
 public | student_id_seq | sequence | postgres
 public | subject        | table    | postgres
 public | subject_id_seq | sequence | postgres
 public | track          | table    | postgres
 public | track_id_seq   | sequence | postgres
 public | track_sub      | table    | postgres
(10 rows)

student_tracking=# \d table_name
```