Lab 1

1. Create your tables with their columns in PostgreSQL.

```
CREATE TABLE student (
    stu_id INT PRIMARY KEY,
    stu_name VARCHAR(20),
    email VARCHAR(40),
    address VARCHAR(40),
    phone VARCHAR(11),
    t_id INT
);
CREATE TABLE subject (
    sub_id INT PRIMARY KEY,
    sub_name VARCHAR(20),
    max_score INT
);
CREATE TABLE track (
    t_id INT PRIMARY KEY,
    t_name VARCHAR(20)
);
CREATE TABLE exam (
    e_id INT PRIMARY KEY,
    e_date DATE
);
CREATE TABLE grades (
    stu_id INT,
    sub_id INT,
    e_id INT,
    grade INT,
PRIMARY KEY(stu_id, sub_id)
);
CREATE TABLE stu_sub (
    stu_id INT,
    sub_id INT,
PRIMARY KEY(stu_id, sub_id)
);
CREATE TABLE track_sub (
    t_id INT,
    sub_id INT,
PRIMARY KEY(t_id, sub_id)
);
```

2. Insert at minimum 3 Rows at each table.

```
INSERT INTO student
VALUES (1, 'taghreed', 'taghreed@email', 'giza', '010', 1000),
(2, 'hanaa', 'hanaa@email', 'giza', '011', 2000),
(3, 'aya', 'aya@email', 'cairo', '012', 3000);
INSERT INTO subject
VALUES (100, 'ds', 50),
```

(200, 'os', 60),

(300, 'network', 70);

INSERT INTO track

VALUES (1000, 'python'),

(2000, '.net'),

(3000, 'php');

INSERT INTO exam

VALUES (11, '2024-2-25'),

(22, '2024-2-26'),

(33, '2024-2-27');

INSERT INTO grades

VALUES (1, 200, 22, 30),

(2, 100, 11, 50),

(3, 300, 33, 10);

INSERT INTO stu_sub

VALUES (1, 200),

(2, 100),

(3, 300);

INSERT INTO stu_sub

VALUES (1, 200),

(2, 100),

(3, 300);

3. Add gender column for the student table[Enum]. It holds two

value (male or female).

CREATE TYPE GENDER AS ENUM ('male', 'female');

ALTER TABLE student

ADD COLUMN gender GENDER;

UPDATE student

SET gender = 'female'

WHERE stu_id IN (1 2, 3);

4. Add birth date column for the student table.

ALTER TABLE student

ADD COLUMN birth_date DATE;

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

ALTER TABLE student DROP COLUMN address;

ALTER TABLE student DROP COLUMN email;

ALTER TABLE student

ADD COLUMN stu_contact_info CONTACT_INFO;

UPDATE student

SET stu_contact_info = ROW('giza', 'taghreed@email')

```
WHERE stu_id = 1;


UPDATE student
SET stu_contact_info = ROW('giza', 'hanaa@email')
WHERE stu_id = 2;


UPDATE student
SET stu_contact_info = ROW('cairo', 'aya@email')
WHERE stu_id = 3;
```

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

```
ALTER TABLE subject
ADD CONSTRAINT fk_student_track
FOREIGN KEY (t_id)
REFERENCES track (t_id);


ALTER TABLE stu_sub
ADD CONSTRAINT fk_subject_student
FOREIGN KEY (stu_id)
REFERENCES student (stu_id);


ALTER TABLE stu_sub
ADD CONSTRAINT fk_subject_student
FOREIGN KEY (sub_id)
REFERENCES subject (sub_id);


ALTER TABLE track_sub
ADD CONSTRAINT fk_track_subject
FOREIGN KEY (t_id )
REFERENCES track (t_id );


ALTER TABLE track_sub
ADD CONSTRAINT fk_suject_track
FOREIGN KEY (sub_id)
REFERENCES subject (sub_id);
```

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

```
SELECT sub_name, max_score
FROM subject
ORDER BY max_score;
```

8. Display the number of males and females.

```
SELECT COUNT(*), gender
FROM student
GROUP BY gender;
```

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

```
SELECT stu_name, COUNT(*)
FROM student
```

```
GROUP BY stu_name

HAVING COUNT(*) > 2;
```

10. Display all Students and track name they belong to.

```
SELECT stu_id, stu_name, t_name

FROM student s inner join track t ON s.t_id = t.t_id;
```