

PostgreSQL Lab3

Mazen Abdeltawab Saad

Track Python Fayoum

Lab3

su - postgres

mazen@@1

psql

create database postgres_lab3 TEMPLATE postgres_lab2;

\l

\c postgres_lab3

3. Create a view for student names with their Tracks names which is belong to it.

create view student_tracks_view

as select student.first_name, track.track_name

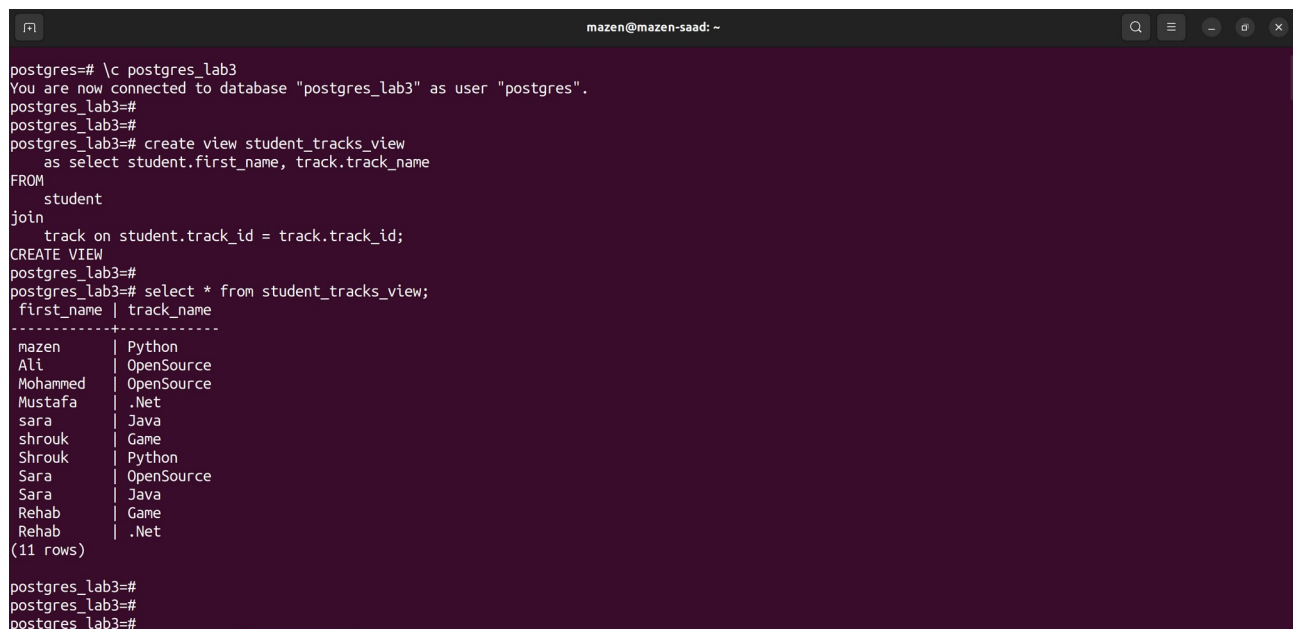
FROM

student

join

track on student.track_id = track.track_id;

select * from student_tracks_view;

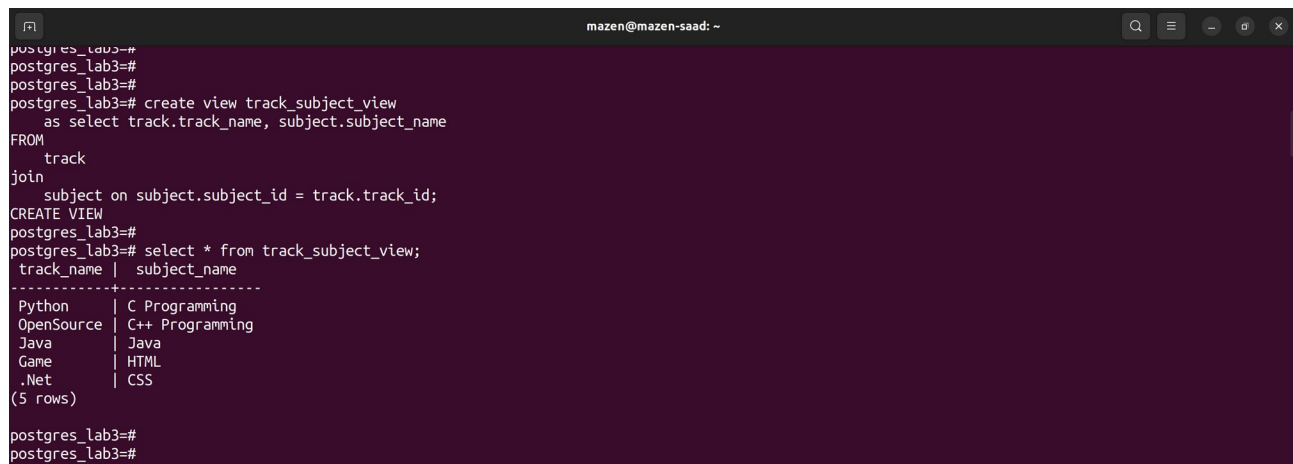


```
mazen@mazen-saad: ~  
postgres=# \c postgres_lab3  
You are now connected to database "postgres_lab3" as user "postgres".  
postgres_lab3=#  
postgres_lab3=#  
postgres_lab3=# create view student_tracks_view  
as select student.first_name, track.track_name  
FROM  
student  
join  
track on student.track_id = track.track_id;  
CREATE VIEW  
postgres_lab3=#  
postgres_lab3=# select * from student_tracks_view;  
first_name | track_name  
-----+-----  
mazen      | Python  
Ali        | OpenSource  
Mohammed   | OpenSource  
Mustafa    | .Net  
sara       | Java  
shrouk     | Game  
Shrouk     | Python  
Sara       | OpenSource  
Sara       | Java  
Rehab      | Game  
Rehab      | .Net  
(11 rows)  
  
postgres_lab3=#  
postgres_lab3=#  
postgres_lab3=#
```

4. Create a view for Tracks names and the subjects which is belong/study to it.

```
create view track_subject_view
as select track.track_name, subject.subject_name
FROM
    track
join
    subject on subject.subject_id = track.track_id;

select * from track_subject_view;
```



```
mazen@mazen-saad: ~
postgres_lab3=#
postgres_lab3=#
postgres_lab3=# create view track_subject_view
as select track.track_name, subject.subject_name
FROM
    track
join
    subject on subject.subject_id = track.track_id;
CREATE VIEW
postgres_lab3=#
postgres_lab3=# select * from track_subject_view;
 track_name | subject_name
-----+-----
 Python     | C Programming
 OpenSource | C++ Programming
 Java       | Java
 Game       | HTML
 .Net       | CSS
(5 rows)

postgres_lab3=#
postgres_lab3=#
```

5. Create a view for student names with their subject's names which will study.

```
create view student_subject_view
as select student.first_name, subject.subject_name
FROM
    student
join
    subject on subject.subject_id = subject.subject_id;

select * from student_subject_view;
```

```
mazen@mazen-saad: ~
postgres_lab3=#
postgres_lab3=#
postgres_lab3=#
postgres_lab3=#
postgres_lab3=# create view student_subject_view
postgres_lab3=# as select student.first_name, subject.subject_name
FROM
postgres_lab3=# student
join
postgres_lab3=# subject on subject.subject_id = subject.subject_id;
CREATE VIEW
postgres_lab3=#
postgres_lab3=# select * from student_subject_view;
postgres_lab3=#
postgres_lab3=#
```

```
mazen@mazen-saad: ~
first_name | subject_name
-----
mazen      | C Programming
mazen      | C++ Programming
mazen      | Java
mazen      | HTML
mazen      | CSS
Ali         | C Programming
Ali         | C++ Programming
Ali         | Java
Ali         | HTML
Ali         | CSS
Mohammed   | C Programming
Mohammed   | C++ Programming
Mohammed   | Java
Mohammed   | HTML
Mohammed   | CSS
Mustafa    | C Programming
Mustafa    | C++ Programming
Mustafa    | Java
Mustafa    | HTML
Mustafa    | CSS
sara       | C Programming
sara       | C++ Programming
sara       | Java
sara       | HTML
sara       | CSS
shrouk     | C Programming
shrouk     | C++ Programming
shrouk     | Java
shrouk     | HTML
shrouk     | CSS
mohammed   | C Programming
mohammed   | C++ Programming
mohammed   | Java
mohammed   | HTML
mohammed   | CSS
Shrouk     | C Programming
Shrouk     | C++ Programming
Shrouk     | Java
Shrouk     | HTML
Shrouk     | CSS
Sara       | C Programming
Sara       | C++ Programming
Sara       | Java
Sara       | HTML
Sara       | CSS
Sara       | C Programming
Sara       | C++ Programming
Sara       | Java
Sara       | HTML
Sara       | CSS
Rehab      | C Programming
Rehab      | C++ Programming
```

6. Create a view for all students name (Full Name) with their score in each subject and its date.

create view student_score_view

as select concat(student.first_name, ' ', student.last_name) as
FullName , subject.subject_name, exam_results.score,
exam.exam_date

FROM

student

join

subject on subject.subject_id = student.subject_id

join

exam_results on exam_results.student_id = student.student_id

join

exam on exam.exam_id = subject.subject_id

;

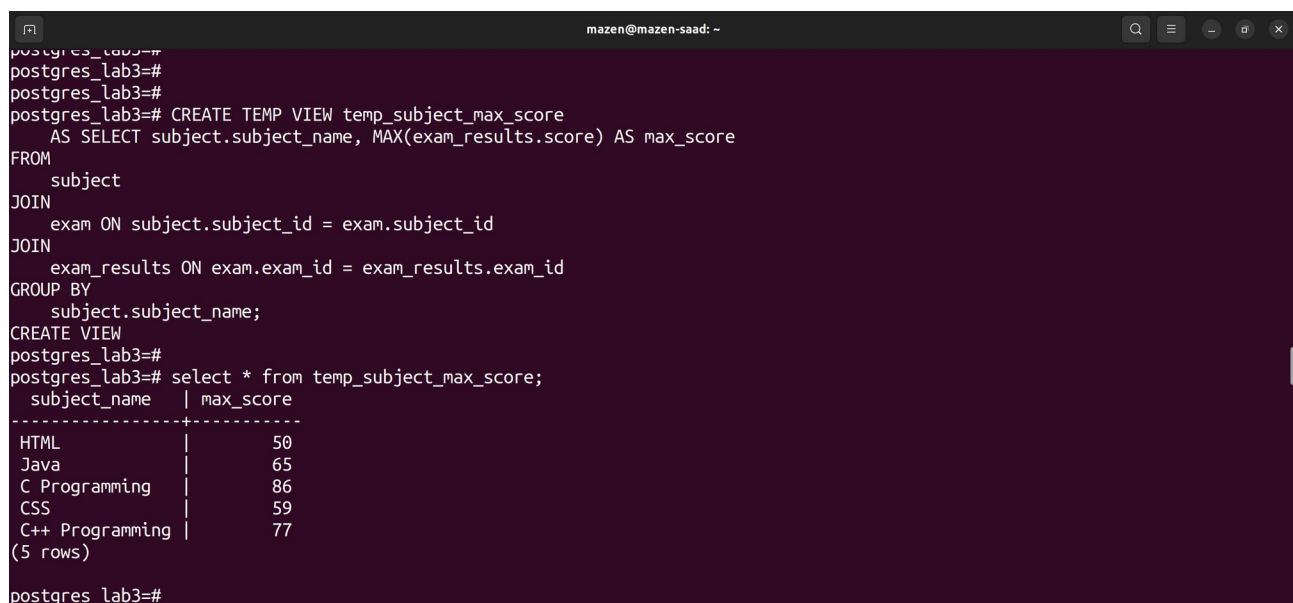
select * from student_score_view;

```
mazen@mazen-saad: ~  
postgres_lab3=# create view student_score_view  
as select concat(student.first_name, ' ', student.last_name) as FullName , subject.subject_name, exam_results.score, exam.exam_date  
FROM  
student  
join  
subject on subject.subject_id = student.subject_id  
join  
exam_results on exam_results.student_id = student.student_id  
join  
exam on exam.exam_id = subject.subject_id  
;  
CREATE VIEW  
postgres_lab3=#  
postgres_lab3=# select * from student_score_view;  
fullname | subject_name | score | exam_date  
-----  
Shrouk Ahmed | C Programming | 86 | 2024-07-01  
Shrouk Ahmed | C++ Programming | 86 | 2024-07-02  
Shrouk Ahmed | Java | 86 | 2024-07-03  
Shrouk Ahmed | HTML | 86 | 2024-07-04  
Shrouk Ahmed | CSS | 86 | 2024-07-05  
Sara Ahmed | C Programming | 77 | 2024-07-01  
Sara Ahmed | C++ Programming | 77 | 2024-07-02  
Sara Ahmed | Java | 77 | 2024-07-03  
Sara Ahmed | HTML | 77 | 2024-07-04  
Sara Ahmed | CSS | 77 | 2024-07-05  
Sara 'GanaL | C Programming | 65 | 2024-07-01  
Sara 'GanaL | C++ Programming | 65 | 2024-07-02  
Sara 'GanaL | Java | 65 | 2024-07-03  
Sara 'GanaL | HTML | 65 | 2024-07-04  
Sara 'GanaL | CSS | 65 | 2024-07-05  
Rehab Ahmed | C Programming | 58 | 2024-07-01  
Rehab Ahmed | C++ Programming | 58 | 2024-07-02  
Rehab Ahmed | Java | 58 | 2024-07-03  
Rehab Ahmed | HTML | 58 | 2024-07-04  
Rehab Ahmed | CSS | 58 | 2024-07-05  
Rehab ezza | C Programming | 35 | 2024-07-01  
Rehab ezza | C++ Programming | 35 | 2024-07-02  
Rehab ezza | Java | 35 | 2024-07-03  
Rehab ezza | HTML | 35 | 2024-07-04  
Rehab ezza | CSS | 35 | 2024-07-05  
mazen saad | C Programming | 59 | 2024-07-01  
mazen saad | C++ Programming | 59 | 2024-07-02  
mazen saad | Java | 59 | 2024-07-03  
mazen saad | HTML | 59 | 2024-07-04  
mazen saad | CSS | 59 | 2024-07-05  
(30 rows)
```

7. Create a temporary view for all subjects with their max_score.

```
CREATE TEMP VIEW temp_subject_max_score
  AS SELECT subject.subject_name, MAX(exam_results.score) AS
max_score
FROM
  subject
JOIN
  exam ON subject.subject_id = exam.subject_id
JOIN
  exam_results ON exam.exam_id = exam_results.exam_id
GROUP BY
  subject.subject_name;

select * from temp_subject_max_score;
```



```
mazen@mazen-saad: ~
postgres=#
postgres_lab3=#
postgres_lab3=#
postgres_lab3=# CREATE TEMP VIEW temp_subject_max_score
AS SELECT subject.subject_name, MAX(exam_results.score) AS max_score
FROM
  subject
JOIN
  exam ON subject.subject_id = exam.subject_id
JOIN
  exam_results ON exam.exam_id = exam_results.exam_id
GROUP BY
  subject.subject_name;
CREATE VIEW
postgres_lab3=#
postgres_lab3=# select * from temp_subject_max_score;
 subject_name | max_score
-----+-----
HTML          |         50
Java          |         65
C Programming |         86
CSS           |         59
C++ Programming |        77
(5 rows)
postgres_lab3=#
```

10. (from Q.6) Display the date of exam as the following: day 'month name' year.

SELECT

FullName, subject_name, score, TO_CHAR(exam_date, 'Day FMMonth YYYY') AS formatted_date

FROM

student_score_view;

```
mazen@mazen-saad: ~
postgres_lab3=# SELECT
postgres_lab3=#     FullName, subject_name, score, TO_CHAR(exam_date, 'Day FMMonth YYYY') AS formatted_date
postgres_lab3=# FROM
postgres_lab3=#     student_score_view;
 fullname | subject_name | score | formatted_date
-----
 Shrouk Ahmed | C Programming | 86 | Monday July 2024
 Shrouk Ahmed | C++ Programming | 86 | Tuesday July 2024
 Shrouk Ahmed | Java | 86 | Wednesday July 2024
 Shrouk Ahmed | HTML | 86 | Thursday July 2024
 Shrouk Ahmed | CSS | 86 | Friday July 2024
 Sara Ahmed | C Programming | 77 | Monday July 2024
 Sara Ahmed | C++ Programming | 77 | Tuesday July 2024
 Sara Ahmed | Java | 77 | Wednesday July 2024
 Sara Ahmed | HTML | 77 | Thursday July 2024
 Sara Ahmed | CSS | 77 | Friday July 2024
 Sara 'Gamal | C Programming | 65 | Monday July 2024
 Sara 'Gamal | C++ Programming | 65 | Tuesday July 2024
 Sara 'Gamal | Java | 65 | Wednesday July 2024
 Sara 'Gamal | HTML | 65 | Thursday July 2024
 Sara 'Gamal | CSS | 65 | Friday July 2024
 Rehab Ahmed | C Programming | 50 | Monday July 2024
 Rehab Ahmed | C++ Programming | 50 | Tuesday July 2024
 Rehab Ahmed | Java | 50 | Wednesday July 2024
 Rehab Ahmed | HTML | 50 | Thursday July 2024
 Rehab Ahmed | CSS | 50 | Friday July 2024
 Rehab ezza | C Programming | 35 | Monday July 2024
 Rehab ezza | C++ Programming | 35 | Tuesday July 2024
 Rehab ezza | Java | 35 | Wednesday July 2024
 Rehab ezza | HTML | 35 | Thursday July 2024
 Rehab ezza | CSS | 35 | Friday July 2024
 mazen saad | C Programming | 59 | Monday July 2024
 mazen saad | C++ Programming | 59 | Tuesday July 2024
 mazen saad | Java | 59 | Wednesday July 2024
 mazen saad | HTML | 59 | Thursday July 2024
 mazen saad | CSS | 59 | Friday July 2024
(30 rows)
```

11. Display name and age of each students

SELECT

concat(student.first_name, ' ', student.last_name) as FullName,
age(birth_date) AS Age

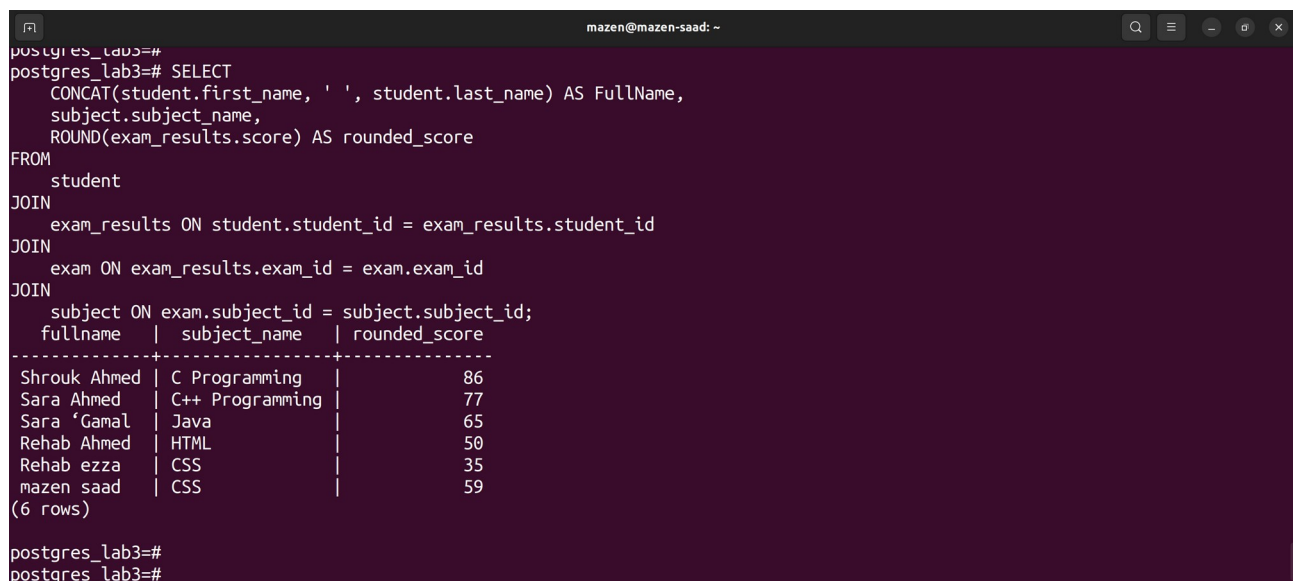
FROM

student;

```
mazen@mazen-saad: ~  
postgres=#  
postgres_lab3=#  
postgres_lab3=# SELECT  
concat(student.first_name, ' ', student.last_name) as FullName, age(birth_date) AS Age  
FROM  
student;  
  fullname |      age  
-----+-----  
mazen saad | 23 years 3 mons 11 days  
Ali sayed  | 32 years  
Mohammed Mustafa | 32 years  
Mustafa sayed | 32 years 11 mons  
sara gamal  | 32 years  
shrouk ahmed | 32 years  
mohammed ali | 22 years 3 mons 11 days  
Shrouk Ahmed | 23 years 2 mons 17 days  
Sara Ahmed   | 21 years 11 mons 10 days  
Sara 'Gamal  | 23 years 2 mons 17 days  
Rehab Ahmed  | 33 years 7 mons 2 days  
Rehab ezza   | 22 years 7 mons 2 days  
(12 rows)  
  
postgres_lab3=#  
postgres_lab3=#
```


12. Display the name of students with their Rounded score in each subject

```
SELECT
    CONCAT(student.first_name, ' ', student.last_name) AS
FullName,
    subject.subject_name,
    ROUND(exam_results.score) AS rounded_score
FROM
    student
JOIN
    exam_results ON student.student_id = exam_results.student_id
JOIN
    exam ON exam_results.exam_id = exam.exam_id
JOIN
    subject ON exam.subject_id = subject.subject_id;
```

A terminal window with a dark purple background and light green text. The window title is 'mazen@mazen-saad: ~'. It shows the execution of a SQL query in PostgreSQL. The query joins student, exam_results, exam, and subject tables to display student names, subject names, and rounded scores. The output is a table with 6 rows and 3 columns: fullname, subject_name, and rounded_score. The results are: Shrouk Ahmed | C Programming | 86, Sara Ahmed | C++ Programming | 77, Sara 'Gamal | Java | 65, Rehab Ahmed | HTML | 50, Rehab ezza | CSS | 35, and mazen saad | CSS | 59. The terminal prompt 'postgres=#' is visible at the top and bottom.

```
mazen@mazen-saad: ~
postgres=#
postgres_lab3=# SELECT
    CONCAT(student.first_name, ' ', student.last_name) AS FullName,
    subject.subject_name,
    ROUND(exam_results.score) AS rounded_score
FROM
    student
JOIN
    exam_results ON student.student_id = exam_results.student_id
JOIN
    exam ON exam_results.exam_id = exam.exam_id
JOIN
    subject ON exam.subject_id = subject.subject_id;
 fullname | subject_name | rounded_score
-----+-----+-----
 Shrouk Ahmed | C Programming |          86
  Sara Ahmed | C++ Programming |          77
  Sara 'Gamal | Java |          65
  Rehab Ahmed | HTML |          50
  Rehab ezza | CSS |          35
  mazen saad | CSS |          59
(6 rows)

postgres_lab3=#
postgres_lab3=#
```

13. Display the name of students with the year of Birthdate

SELECT

concat(student.first_name, ' ', student.last_name) as full_name,
date_part('year',birth_date) AS birth_year

FROM

student;

```
mazen@mazen-saad: ~  
postgres_lab3=#  
postgres_lab3=# SELECT  
concat(student.first_name, ' ', student.last_name) as full_name, date_part('year',birth_date) AS birth_year  
FROM  
student;  
 full_name | birth_year  
-----+-----  
mazen saad | 2001  
Ali sayed | 1992  
Mohammed Mustafa | 1992  
Mustafa sayed | 1991  
sara gamal | 1992  
shrouk ahmed | 1992  
mohammed ali | 2002  
Shrouk Ahmed | 2001  
Sara Ahmed | 2002  
Sara 'Gamal | 2001  
Rehab Ahmed | 1990  
Rehab ezza | 2001  
(12 rows)  
  
postgres_lab3=#  
postgres_lab3=#
```

14. Add new exam result, in date column use NOW() function;

```
INSERT INTO
    exam (exam_date, subject_id)
VALUES
    (NOW(), 3);
```

A terminal window with a dark background and light text. The window title bar shows 'mazen@mazen-saad: ~'. The terminal content shows a PostgreSQL prompt 'postgres_lab3=#' followed by the SQL command 'INSERT INTO exam (exam_date, subject_id) VALUES (NOW(), 3);'. The command is executed successfully, returning 'INSERT 0 1'. The prompt then changes to 'postgres_lab3=#' again.

```
mazen@mazen-saad: ~
postgres_lab3=#
postgres_lab3=# INSERT INTO
    exam (exam_date, subject_id)
VALUES
    (NOW(), 3);
INSERT 0 1
postgres_lab3=#
postgres_lab3=#
```

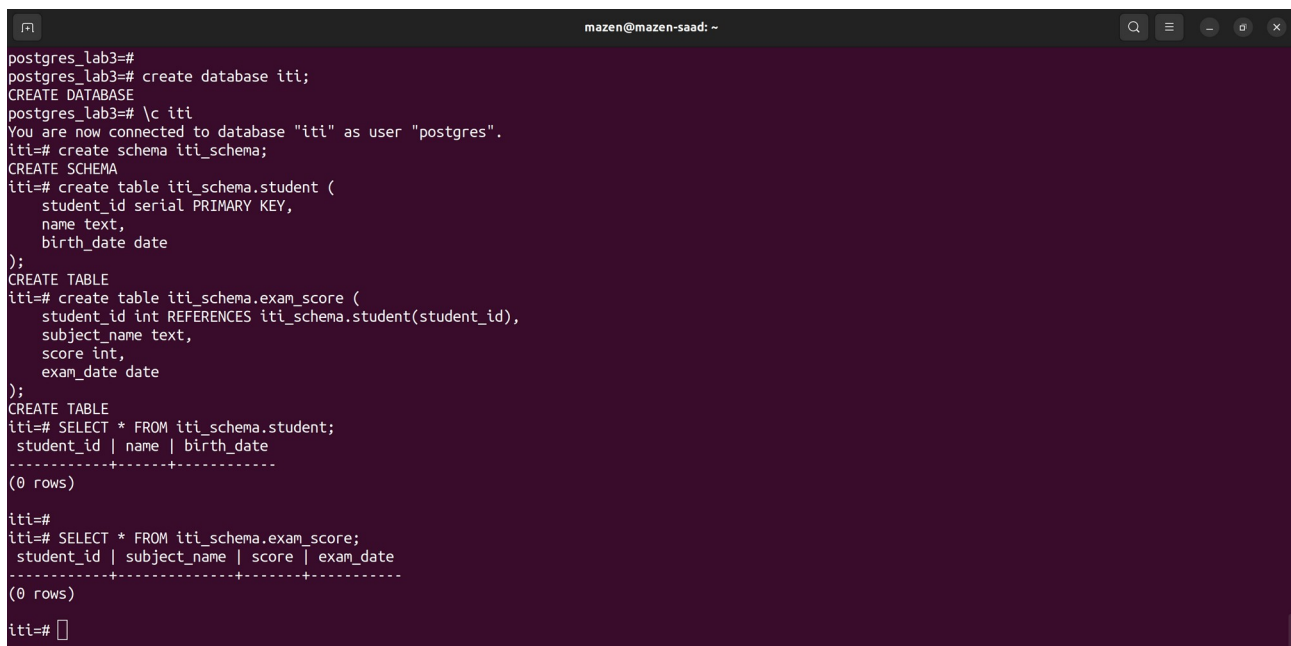
15. Create database called ITI, and create different schema and Tables inside this schema

```
create database iti;  
\c iti
```

```
create schema iti_schema;
```

```
create table iti_schema.student (  
    student_id serial PRIMARY KEY,  
    name text,  
    birth_date date  
);  
create table iti_schema.exam_score (  
    student_id int REFERENCES iti_schema.student(student_id),  
    subject_name text,  
    score int,  
    exam_date date  
);
```

```
SELECT * FROM iti_schema.student;  
SELECT * FROM iti_schema.exam_score;
```



```
mazen@mazen-saad: ~  
postgres_lab3=#  
postgres_lab3=# create database iti;  
CREATE DATABASE  
postgres_lab3=# \c iti  
You are now connected to database "iti" as user "postgres".  
iti=# create schema iti_schema;  
CREATE SCHEMA  
iti=# create table iti_schema.student (  
    student_id serial PRIMARY KEY,  
    name text,  
    birth_date date  
);  
CREATE TABLE  
iti=# create table iti_schema.exam_score (  
    student_id int REFERENCES iti_schema.student(student_id),  
    subject_name text,  
    score int,  
    exam_date date  
);  
CREATE TABLE  
iti=# SELECT * FROM iti_schema.student;  
 student_id | name | birth_date  
-----+-----+-----  
(0 rows)  
  
iti=#  
iti=# SELECT * FROM iti_schema.exam_score;  
 student_id | subject_name | score | exam_date  
-----+-----+-----+-----  
(0 rows)  
  
iti=#
```

lab simp

1. Create trigger to prevent insert new Course with name length greater than 20 chars;

```
su - postgres
```

```
mazen@@1
```

```
psql
```

```
create database postgres_lab_simp;
```

```
\l
```

```
\c postgres_lab_simp
```

```
create table courses (  
    course_id serial primary key,  
    course_name text  
);
```

```
insert into courses (course_name) values ('html');
```

```
insert into courses (course_name) values ('css');
```

```
insert into courses (course_name) values ('js');
```

```
\i /tmp/triggerone.sql
```

```
CREATE OR REPLACE FUNCTION check_course_name_length()
```

```
RETURNS TRIGGER AS $$
```

```
BEGIN
```

```
    IF LENGTH(NEW.name) > 20 THEN
```

```
        RAISE EXCEPTION 'Course name cannot exceed 20  
characters';
```

```
    END IF;
```

```
    RETURN NEW;
```

```
END;
```

```
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER course_name_length_trigger
```

BEFORE INSERT ON courses
FOR EACH ROW
EXECUTE FUNCTION check_course_name_length();

INSERT INTO
Courses (course_name)
VALUES
('ThisCourseNameIsWayTooLongForTheSystem');

```
mazen@mazen-saad: ~  
iti=#  
iti=# create database postgres_lab_simp;  
CREATE DATABASE  
iti=# \c postgres_lab_simp  
You are now connected to database "postgres_lab_simp" as user "postgres".  
postgres_lab_simp=# create table courses (  
    course_id serial primary key,  
    course_name text  
);  
insert into courses (course_name) values ('html');  
insert into courses (course_name) values ('css');  
insert into courses (course_name) values ('js');  
CREATE TABLE  
INSERT 0 1  
INSERT 0 1  
INSERT 0 1  
postgres_lab_simp=# \i /tmp/triggerone.sql  
CREATE FUNCTION  
CREATE TRIGGER  
postgres_lab_simp=# INSERT INTO  
    Courses (course_name)  
VALUES  
    ('ThisCourseNameIsWayTooLongForTheSystem');  
ERROR: record "new" has no field "name"  
CONTEXT: SQL expression "LENGTH(NEW.name) > 20"  
PL/pgSQL function check_course_name_length() line 3 at IF  
postgres_lab_simp=#
```

2. Create trigger to prevent user to insert or update Exam with Score greater than 100 or less than zero

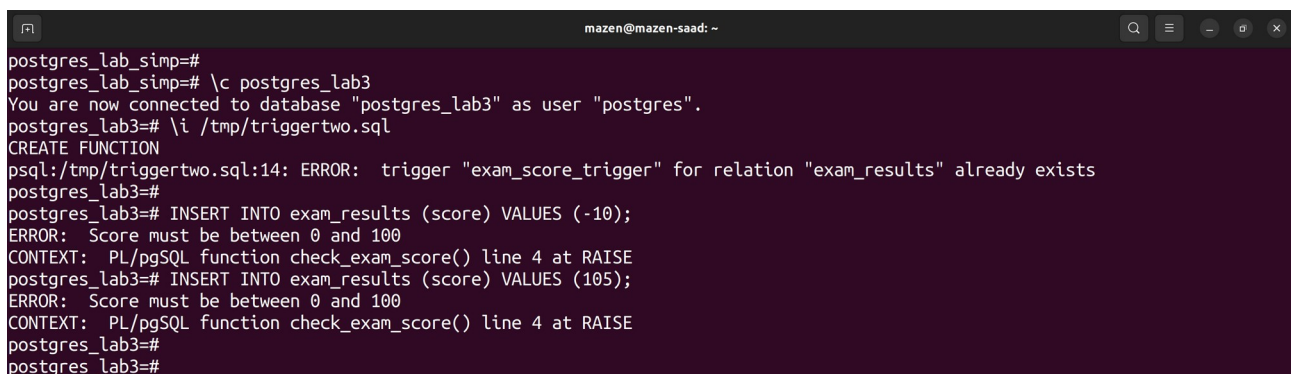
```
\c postgres_lab3
```

```
\i /tmp/triggertwo.sql
```

```
CREATE OR REPLACE FUNCTION check_exam_score()  
RETURNS TRIGGER AS $$  
BEGIN  
    IF NEW.score < 0 OR NEW.score > 100 THEN  
        RAISE EXCEPTION 'Score must be between 0 and 100';  
    END IF;  
    RETURN NEW;  
END;  
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER exam_score_trigger  
BEFORE INSERT OR UPDATE ON exam_results  
FOR EACH ROW  
EXECUTE FUNCTION check_exam_score();
```

```
INSERT INTO exam_results (score) VALUES (-10);  
INSERT INTO exam_results (score) VALUES (105);  
INSERT INTO exam_results (score) VALUES (85);
```



A terminal window titled 'mazen@mazen-saad: ~' showing the execution of PostgreSQL commands. The user connects to the 'postgres_lab3' database. They attempt to create a function, but it fails because a trigger with the same name already exists. Then, they attempt to insert values -10 and 105 into the 'exam_results' table, which both fail due to the 'check_exam_score' trigger enforcing the 0-100 score range. The final successful command is an insert of 85.

```
mazen@mazen-saad: ~  
postgres_lab_simp=#  
postgres_lab_simp=# \c postgres_lab3  
You are now connected to database "postgres_lab3" as user "postgres".  
postgres_lab3=# \i /tmp/triggertwo.sql  
CREATE FUNCTION  
psql:/tmp/triggertwo.sql:14: ERROR:  trigger "exam_score_trigger" for relation "exam_results" already exists  
postgres_lab3=#  
postgres_lab3=# INSERT INTO exam_results (score) VALUES (-10);  
ERROR:  Score must be between 0 and 100  
CONTEXT:  PL/pgSQL function check_exam_score() line 4 at RAISE  
postgres_lab3=# INSERT INTO exam_results (score) VALUES (105);  
ERROR:  Score must be between 0 and 100  
CONTEXT:  PL/pgSQL function check_exam_score() line 4 at RAISE  
postgres_lab3=#  
postgres_lab3=#
```

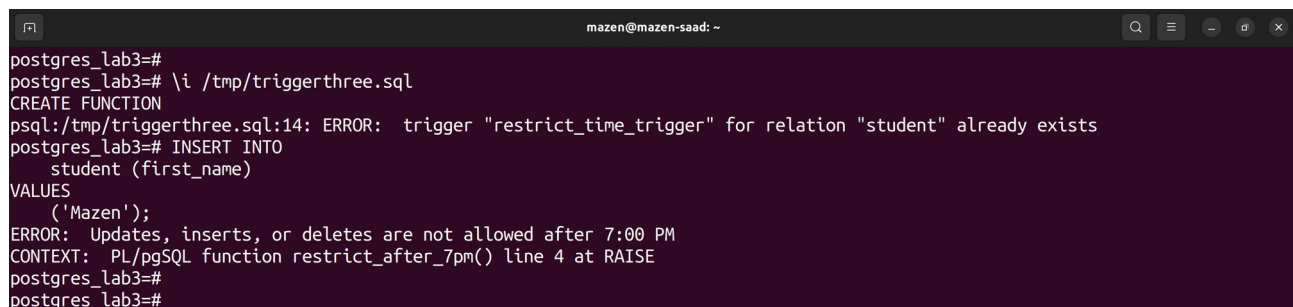
3. (bonus) Create trigger to prevent any user to update/insert/delete to all tables (Students, Exams, Tracks,...) after 7:00 PM

```
\i /tmp/triggerthree.sql
```

```
CREATE OR REPLACE FUNCTION restrict_after_7pm()  
RETURNS TRIGGER AS $$  
BEGIN  
    IF EXTRACT(HOUR FROM CURRENT_TIME) >= 19 THEN  
        RAISE EXCEPTION 'Updates, inserts, or deletes are not  
allowed after 7:00 PM';  
    END IF;  
    RETURN NEW;  
END;  
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER restrict_time_trigger  
BEFORE INSERT OR UPDATE OR DELETE ON student  
FOR EACH ROW  
EXECUTE FUNCTION restrict_after_7pm();
```

```
INSERT INTO  
    student (first_name)  
VALUES  
    ('Mazen');
```

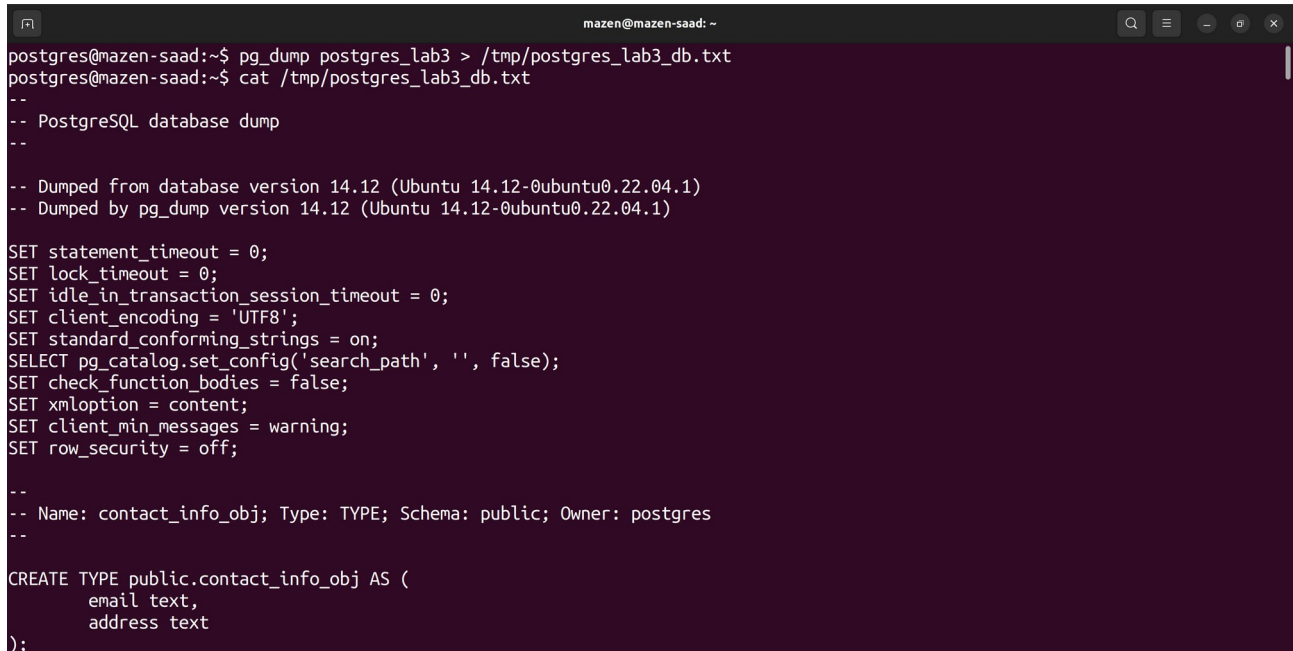


A terminal window titled 'mazen@mazen-saad: ~' with a dark purple background. It shows the execution of SQL commands in a PostgreSQL shell. The commands are: `postgres_lab3=#`, `postgres_lab3=# \i /tmp/triggerthree.sql`, `CREATE FUNCTION`, `psql:/tmp/triggerthree.sql:14: ERROR: trigger "restrict_time_trigger" for relation "student" already exists`, `postgres_lab3=# INSERT INTO`, `student (first_name)`, `VALUES`, `('Mazen');`, `ERROR: Updates, inserts, or deletes are not allowed after 7:00 PM`, `CONTEXT: PL/pgSQL function restrict_after_7pm() line 4 at RAISE`, `postgres_lab3=#`, and `postgres_lab3=#`.

```
mazen@mazen-saad: ~  
postgres_lab3=#  
postgres_lab3=# \i /tmp/triggerthree.sql  
CREATE FUNCTION  
psql:/tmp/triggerthree.sql:14: ERROR: trigger "restrict_time_trigger" for relation "student" already exists  
postgres_lab3=# INSERT INTO  
student (first_name)  
VALUES  
('Mazen');  
ERROR: Updates, inserts, or deletes are not allowed after 7:00 PM  
CONTEXT: PL/pgSQL function restrict_after_7pm() line 4 at RAISE  
postgres_lab3=#  
postgres_lab3=#
```


4. Backup your Database to external file

`pg_dump postgres_lab3 > /tmp/postgres_lab3_db.txt`

A terminal window with a dark background and light text. The window title is "mazen@mazen-saad: ~". The command "pg_dump postgres_lab3 > /tmp/postgres_lab3_db.txt" has been executed. The output shows the dump of a PostgreSQL database. It includes version information, settings like "SET statement_timeout = 0;", and the definition of a new type "public.contact_info_obj".

```
mazen@mazen-saad: ~  
postgres@mazen-saad:~$ pg_dump postgres_lab3 > /tmp/postgres_lab3_db.txt  
postgres@mazen-saad:~$ cat /tmp/postgres_lab3_db.txt  
--  
-- PostgreSQL database dump  
--  
  
-- Dumped from database version 14.12 (Ubuntu 14.12-0ubuntu0.22.04.1)  
-- Dumped by pg_dump version 14.12 (Ubuntu 14.12-0ubuntu0.22.04.1)  
  
SET statement_timeout = 0;  
SET lock_timeout = 0;  
SET idle_in_transaction_session_timeout = 0;  
SET client_encoding = 'UTF8';  
SET standard_conforming_strings = on;  
SELECT pg_catalog.set_config('search_path', '', false);  
SET check_function_bodies = false;  
SET xmloption = content;  
SET client_min_messages = warning;  
SET row_security = off;  
  
--  
-- Name: contact_info_obj; Type: TYPE; Schema: public; Owner: postgres  
--  
  
CREATE TYPE public.contact_info_obj AS (  
    email text,  
    address text  
);
```

5. Backup your Student table to external file

su - postgres

mazen@@1

psql

\c postgres_lab3

copy student to '/tmp/student_table.txt';

A terminal window with a dark purple background. The title bar shows 'mazen@mazen-saad: ~'. The terminal content shows the following sequence of commands and output:

```
postgres_lab3=#
postgres_lab3=#
postgres_lab3=#
postgres_lab3=# copy student to '/tmp/student_table.txt';
COPY 12
postgres_lab3=#
postgres_lab3=#
```