PostgreSQL Lab1

Mazen Abdeltawab Saad

Track Python Fayoum

1. Install PostgreSQL DBMS.
The scenario is that as an instructor in ITI, you have grade-keeping responsibilities.
You want to convert the grading process from a manual operation using a gradebook to an electronic representation using database.
In this case:

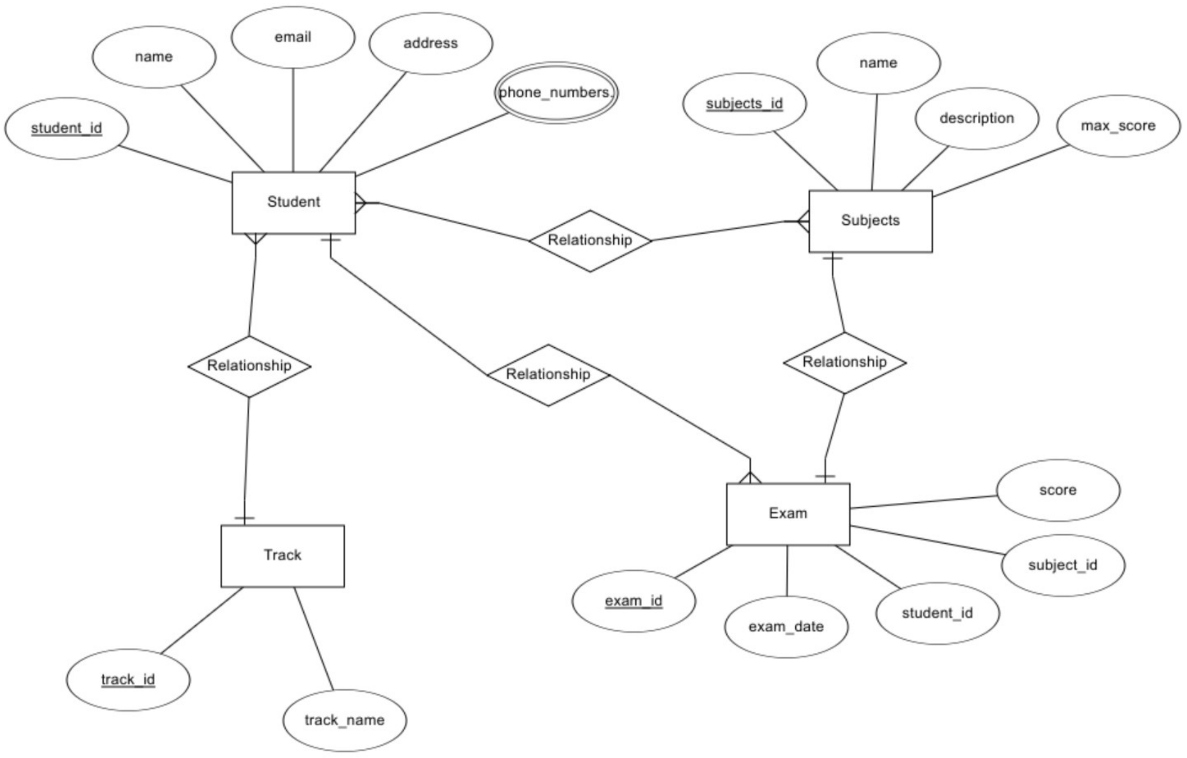
- For each student, you keep track his name, contact info (email, address), and multiple phone numbers.
- Each Student belong to Track (Telecom, OpenSource, Java, Game,..), and each track have different students
- Each track have different subjects/courses such as (C, CPP, HTML, ...), and each student study subjects/courses based on the track that he belong to it
- For each subject, you need to define the name and the description and max score (total grade 100).
- The students achieve score in subject by exam result.
- For each exam which taken by student you must store Exam date, student score in exam (such as 75).

- Keep track of Students and Track which he belong to it, and subjects owned by Tack , and Subjects which will studied by each student

(Solution)
Student(#'student_id', name, email, address, phone_numbers)
Track (#'track_id', track_name)
Subject (#'subject_id', subject_name, description, max_score)
Exam (#"exam_id", exam_date, *student_id, *subject_id, score)

2. Design ERD and write down the mapping schema.

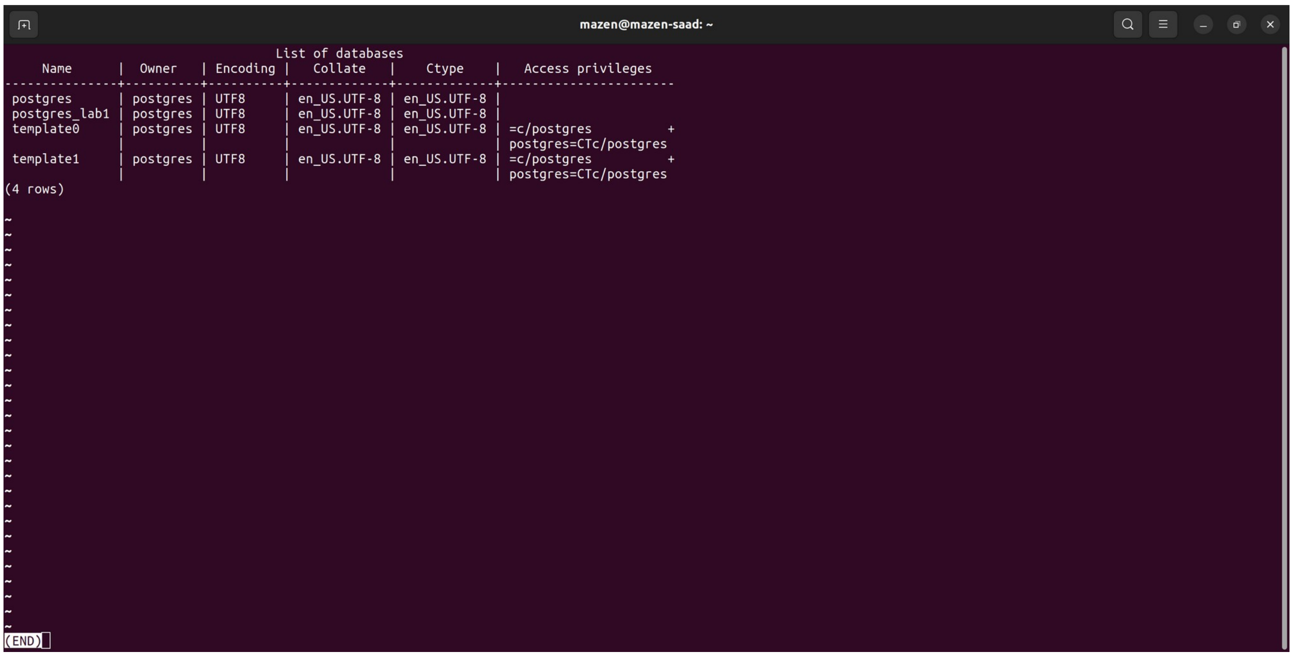
(solution)



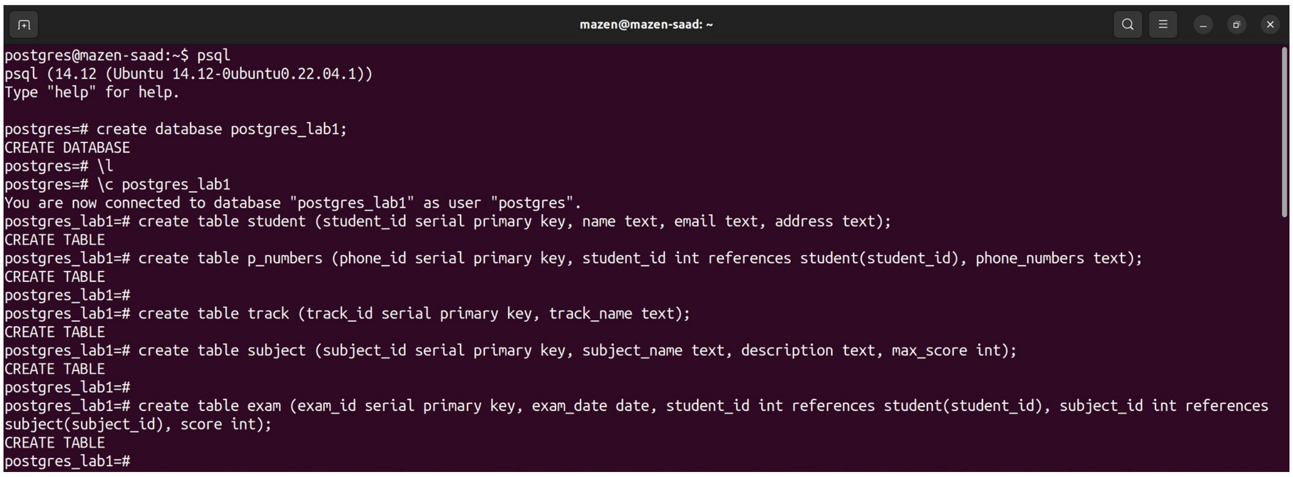
3. Create your mapped tables with their columns in PostgreSQL.

(solution)

```
su - postgres
psql
create database postgres_lab1;
\l
\c postgres_lab1


create table student (student_id serial primary key, name text, email
text, address text);

create table p_numbers (phone_id serial primary key, student_id int
references student(student_id), phone_numbers text);

create table track (track_id serial primary key, track_name text);

create table subject (subject_id serial primary key, subject_name
text, description text, max_score int);

create table exam (exam_id serial primary key, exam_date date,
student_id int references student(student_id), subject_id int
references subject(subject_id), score int);
```



```
                                  List of databases
     Name      |  Owner   | Encoding |   Collate   |    Ctype    |   Access privileges
---------------+----------+----------+-------------+-------------+-----------------------
 postgres      | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 postgres_lab1 | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 template0     | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/postgres          +
               |          |          |             |             | postgres=CTc/postgres
 template1     | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/postgres          +
               |          |          |             |             | postgres=CTc/postgres
(4 rows)

~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
(END)
```



```
postgres@mazen-saad:~$ psql
psql (14.12 (Ubuntu 14.12-0ubuntu0.22.04.1))
Type "help" for help.

postgres=# create database postgres_lab1;
CREATE DATABASE
postgres=# \l
postgres=# \c postgres_lab1
You are now connected to database "postgres_lab1" as user "postgres".
postgres_lab1=# create table student (student_id serial primary key, name text, email text, address text);
CREATE TABLE
postgres_lab1=# create table p_numbers (phone_id serial primary key, student_id int references student(student_id), phone_numbers text);
CREATE TABLE
postgres_lab1=#
postgres_lab1=# create table track (track_id serial primary key, track_name text);
CREATE TABLE
postgres_lab1=# create table subject (subject_id serial primary key, subject_name text, description text, max_score int);
CREATE TABLE
postgres_lab1=#
postgres_lab1=# create table exam (exam_id serial primary key, exam_date date, student_id int references student(student_id), subject_id int references
subject(subject_id), score int);
CREATE TABLE
postgres_lab1=#
```

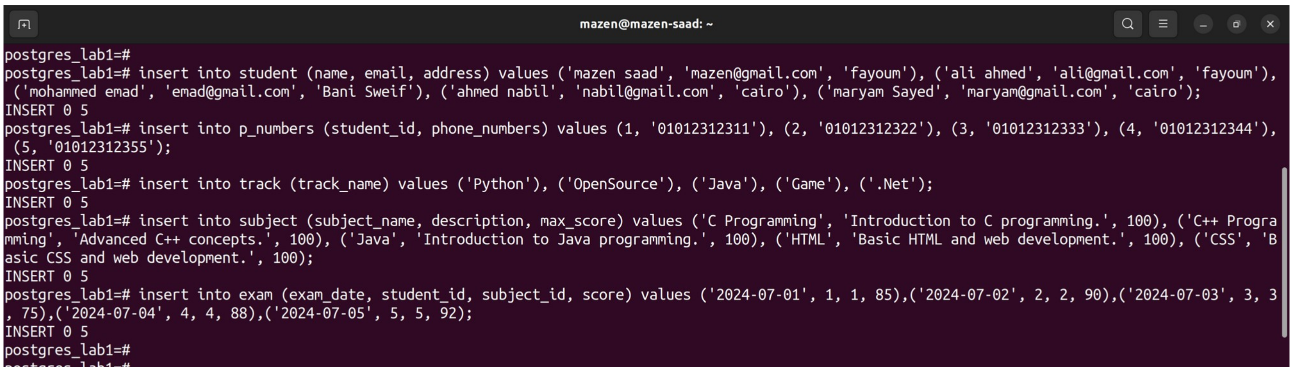
4. Insert at minimum 5 Rows at each table

(solution)

```
insert into student (name, email, address)
        values
        ('mazen saad', 'mazen@gmail.com', 'fayoum'),
        ('ali ahmed', 'ali@gmail.com', 'fayoum'),
        ('mohammed emad', 'emad@gmail.com', 'Bani Sweif'),
        ('ahmed nabil', 'nabil@gmail.com', 'cairo'),
        ('maryam Sayed', 'maryam@gmail.com', 'cairo');

insert into p_numbers (student_id, phone_numbers)
        values
        (1, '01012312311'),
        (2, '01012312322'),
        (3, '01012312333'),
        (4, '01012312344'),
        (5, '01012312355');

insert into track (track_name)
        values
        ('Python'),
        ('OpenSource'),
        ('Java'),
        ('Game'),
        ('.Net');

insert into subject (subject_name, description, max_score)
        values
        ('C Programming', 'Introduction to C programming.', 100),
        ('C++ Programming', 'Advanced C++ concepts.', 100),
        ('Java', 'Introduction to Java programming.', 100),
        ('HTML', 'Basic HTML and web development.', 100),
        ('CSS', 'Basic CSS and web development.', 100);

insert into exam (exam_date, student_id, subject_id, score)
        values
        ('2024-07-01', 1, 1, 85),
        ('2024-07-02', 2, 2, 90),
        ('2024-07-03', 3, 3, 75),
        ('2024-07-04', 4, 4, 88),
        ('2024-07-05', 5, 5, 92);
```

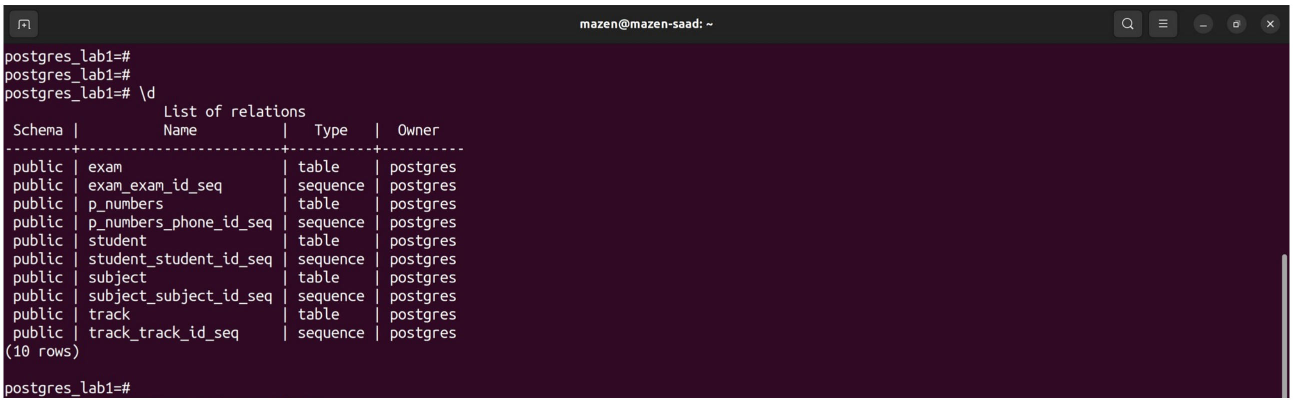
```
\d
```

```
select * from student;
select * from p_numbers;
select * from track;
select * from subject;
select * from exam;
```



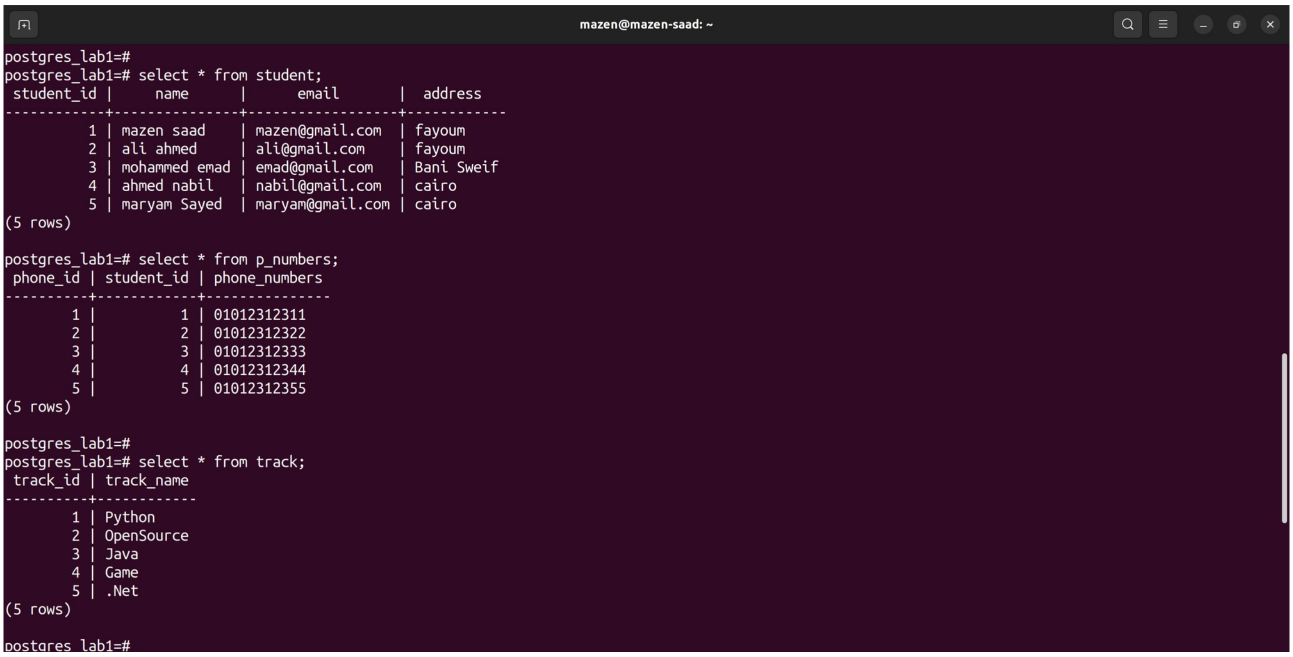
mazen@mazen-saad: ~

```
postgres_lab1=#
postgres_lab1=# insert into student (name, email, address) values ('mazen saad', 'mazen@gmail.com', 'fayoum'), ('ali ahmed', 'ali@gmail.com', 'fayoum'),
 ('mohammed emad', 'emad@gmail.com', 'Bani Sweif'), ('ahmed nabil', 'nabil@gmail.com', 'cairo'), ('maryam Sayed', 'maryam@gmail.com', 'cairo');
INSERT 0 5
postgres_lab1=# insert into p_numbers (student_id, phone_numbers) values (1, '01012312311'), (2, '01012312322'), (3, '01012312333'), (4, '01012312344'),
 (5, '01012312355');
INSERT 0 5
postgres_lab1=# insert into track (track_name) values ('Python'), ('OpenSource'), ('Java'), ('Game'), ('.Net');
INSERT 0 5
postgres_lab1=# insert into subject (subject_name, description, max_score) values ('C Programming', 'Introduction to C programming.', 100), ('C++ Progra
mming', 'Advanced C++ concepts.', 100), ('Java', 'Introduction to Java programming.', 100), ('HTML', 'Basic HTML and web development.', 100), ('CSS', 'B
asic CSS and web development.', 100);
INSERT 0 5
postgres_lab1=# insert into exam (exam_date, student_id, subject_id, score) values ('2024-07-01', 1, 1, 85),('2024-07-02', 2, 2, 90),('2024-07-03', 3, 3
, 75),('2024-07-04', 4, 4, 88),('2024-07-05', 5, 5, 92);
INSERT 0 5
postgres_lab1=#
```

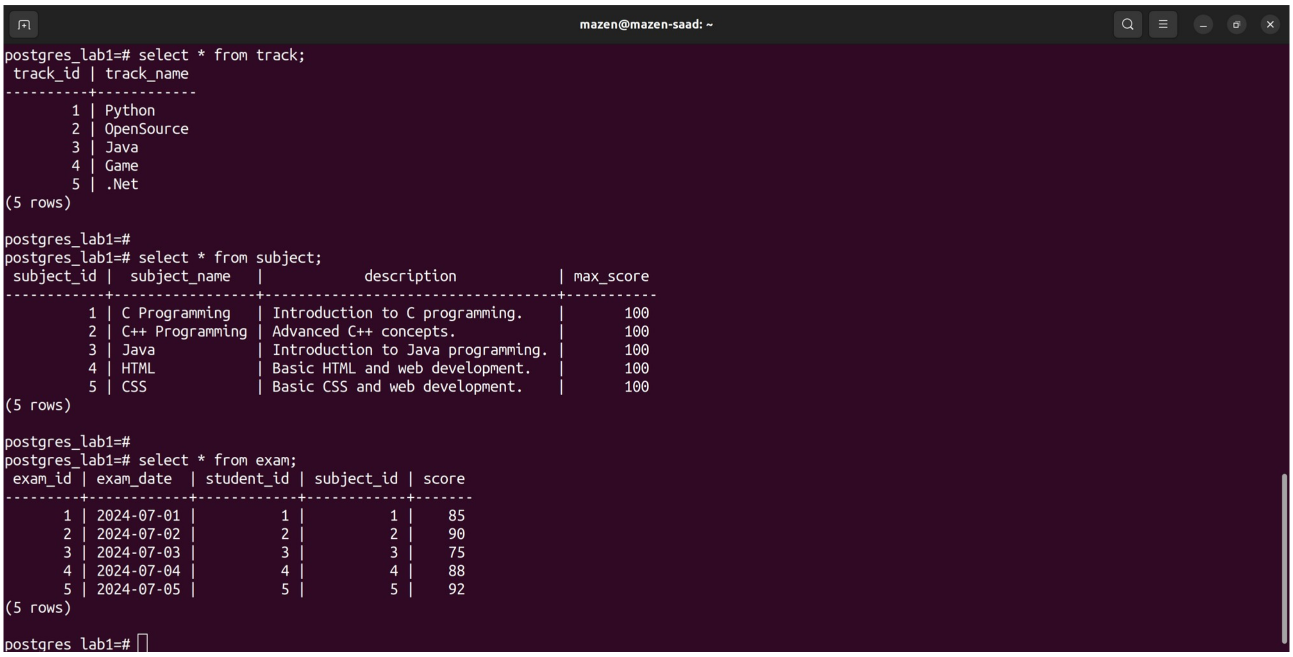


mazen@mazen-saad: ~

```
postgres_lab1=#
postgres_lab1=#
postgres_lab1=# \d
                   List of relations
 Schema |          Name          |   Type   |  Owner
--------+------------------------+----------+----------
 public | exam                   | table    | postgres
 public | exam_exam_id_seq       | sequence | postgres
 public | p_numbers              | table    | postgres
 public | p_numbers_phone_id_seq | sequence | postgres
 public | student                | table    | postgres
 public | student_student_id_seq | sequence | postgres
 public | subject                | table    | postgres
 public | subject_subject_id_seq | sequence | postgres
 public | track                  | table    | postgres
 public | track_track_id_seq     | sequence | postgres
(10 rows)

postgres_lab1=#
```



```
postgres_lab1=#
postgres_lab1=# select * from student;
 student_id |     name      |      email       |  address
------------+---------------+------------------+------------
          1 | mazen saad    | mazen@gmail.com  | fayoum
          2 | ali ahmed     | ali@gmail.com    | fayoum
          3 | mohammed emad | emad@gmail.com   | Bani Sweif
          4 | ahmed nabil   | nabil@gmail.com  | cairo
          5 | maryam Sayed  | maryam@gmail.com | cairo
(5 rows)

postgres_lab1=# select * from p_numbers;
 phone_id | student_id | phone_numbers
----------+------------+---------------
        1 |          1 | 01012312311
        2 |          2 | 01012312322
        3 |          3 | 01012312333
        4 |          4 | 01012312344
        5 |          5 | 01012312355
(5 rows)

postgres_lab1=#
postgres_lab1=# select * from track;
 track_id | track_name
----------+------------
        1 | Python
        2 | OpenSource
        3 | Java
        4 | Game
        5 | .Net
(5 rows)

postgres_lab1=#
```



```
postgres_lab1=# select * from track;
 track_id | track_name
----------+------------
        1 | Python
        2 | OpenSource
        3 | Java
        4 | Game
        5 | .Net
(5 rows)

postgres_lab1=#
postgres_lab1=# select * from subject;
 subject_id |  subject_name   |            description             | max_score
------------+-----------------+------------------------------------+-----------
          1 | C Programming   | Introduction to C programming.     |       100
          2 | C++ Programming | Advanced C++ concepts.             |       100
          3 | Java            | Introduction to Java programming.  |       100
          4 | HTML            | Basic HTML and web development.    |       100
          5 | CSS             | Basic CSS and web development.     |       100
(5 rows)

postgres_lab1=#
postgres_lab1=# select * from exam;
 exam_id | exam_date  | student_id | subject_id | score
---------+------------+------------+------------+-------
       1 | 2024-07-01 |          1 |          1 |    85
       2 | 2024-07-02 |          2 |          2 |    90
       3 | 2024-07-03 |          3 |          3 |    75
       4 | 2024-07-04 |          4 |          4 |    88
       5 | 2024-07-05 |          5 |          5 |    92
(5 rows)

postgres_lab1=#
```

Reports :-
1. What is NoSQL?

(solution)
NoSQL refers to a variety of database systems that are designed to handle unstructured data and provide flexible schema structures. Unlike traditional relational databases, NoSQL databases are often used for applications requiring horizontal scaling, high performance, and the ability to handle large volumes of diverse data types.

2. DBMSs Types (such as: Hierarchical,...) and brief about each type? Reports   Reports

(solution)
1- Hierarchical DBMS: Data is organized in a tree-like structure (e.g., IBM’s Information Management System). It’s useful for applications with a clear hierarchical relationship, but lacks flexibility in querying data.

2- Network DBMS: Data is structured as a graph, with nodes and relationships (e.g., Integrated Data Store). It supports more complex relationships than hierarchical DBMS but can be harder to manage.

3- Relational DBMS (RDBMS): Data is organized in tables with rows and columns. Tables can be related to each other through foreign keys. Examples include PostgreSQL, MySQL, and Oracle. It is widely used due to its support for complex queries and data integrity.

4- Object-Oriented DBMS: Stores data as objects, similar to object-oriented programming. Examples include ObjectDB and db4o. It’s used in applications requiring complex data modeling and relationships.

5- NoSQL DBMS: Designed for unstructured data and horizontal scaling, including document, key-value, column-family, and graph stores (e.g., MongoDB, Redis, Cassandra, Neo4j).