**DBMS Assignment 3**

**PESUVariance - Learn, Grow & Challenge Yourself!**

**Team ID: H1**

**Names and respective SRN’s:**

**Shrikar Madhu: PES1UG19CS470**

**Sravya Yepuri: PES1UG19CS502**

**Sri Ramya Priya Vedula: PES1UG19CS504**

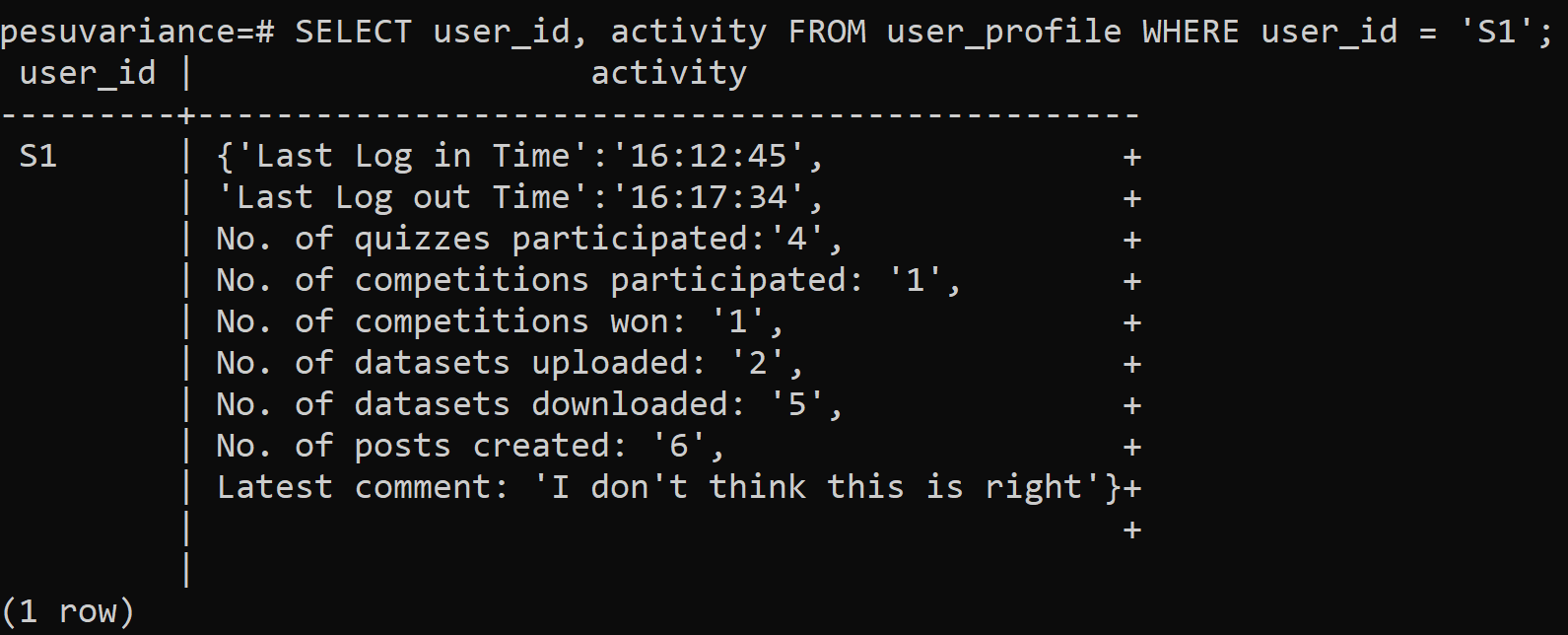
**Yousha Mahamuni: PES2UG19CS468**

**Semester: 5, Section: H**

**Simple Queries:**

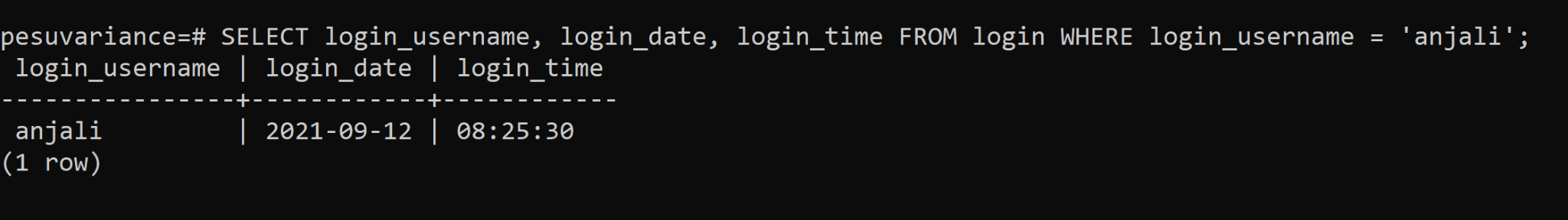
1. **Retrieve the user activities of student user with User ID as S1. (User)**

SELECT user\_id, activity FROM user\_profile WHERE user\_id = 'S1';



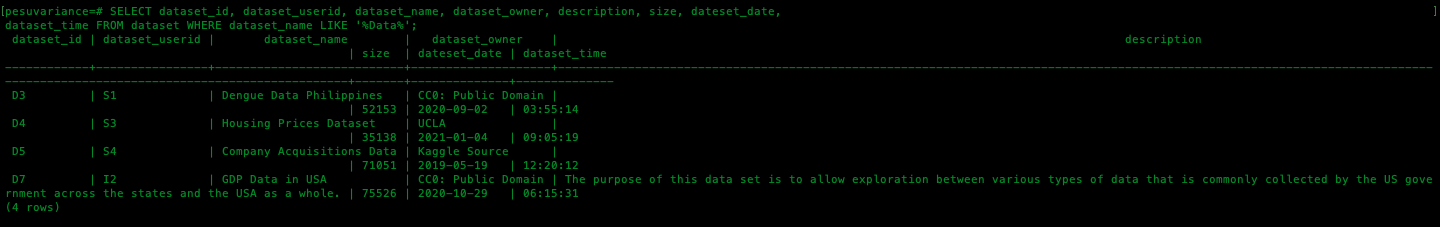
1. **Show the login date and time for a user with username as anjali. (Login)**

SELECT login\_username, login\_date, login\_time FROM login WHERE login\_username = 'anjali';



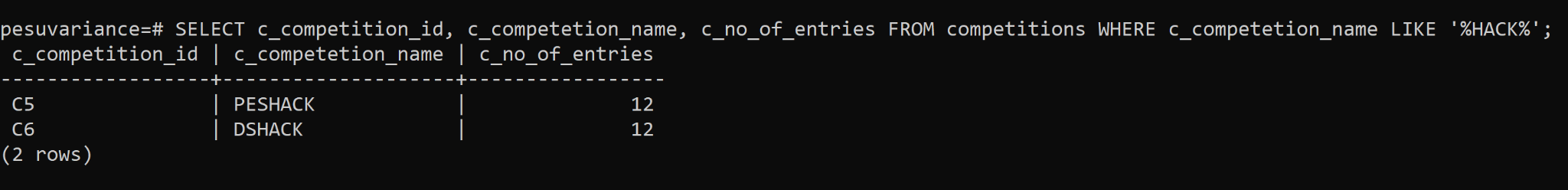
1. **Get all the details of the dataset where the name of the dataset has the word ‘Data’ in it. (Dataset)**

SELECT dataset\_id, dataset\_userid, dataset\_name, dataset\_owner, description, size, dateset\_date, dataset\_time FROM dataset WHERE dataset\_name LIKE '%Data%';



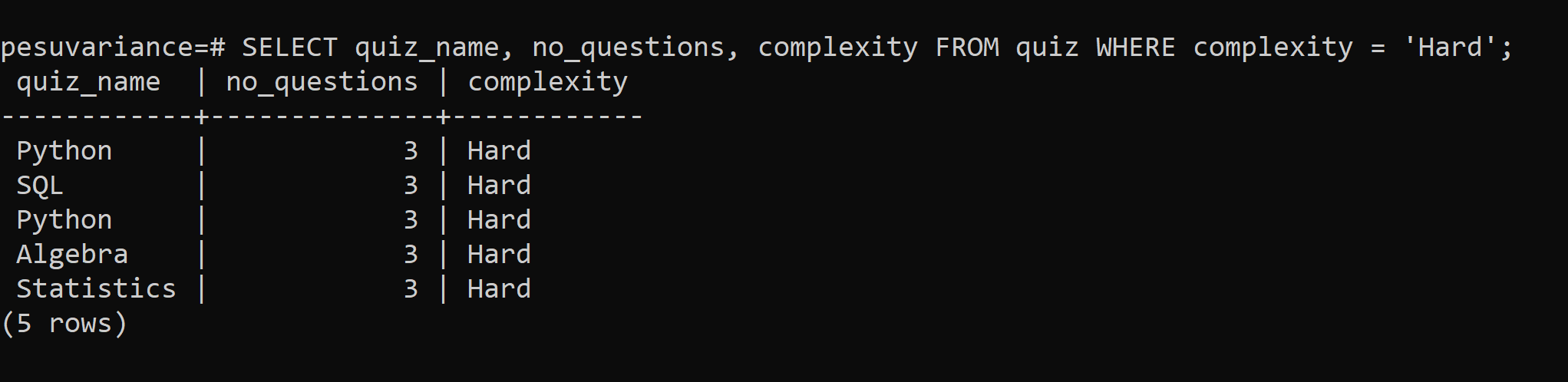
1. **Retrieve the Competition ID and number of entries for each hacking competition. (Competitions)**

SELECT c\_competition\_id, c\_competetion\_name, c\_no\_of\_entries FROM competitions WHERE c\_competetion\_name LIKE '%HACK%';



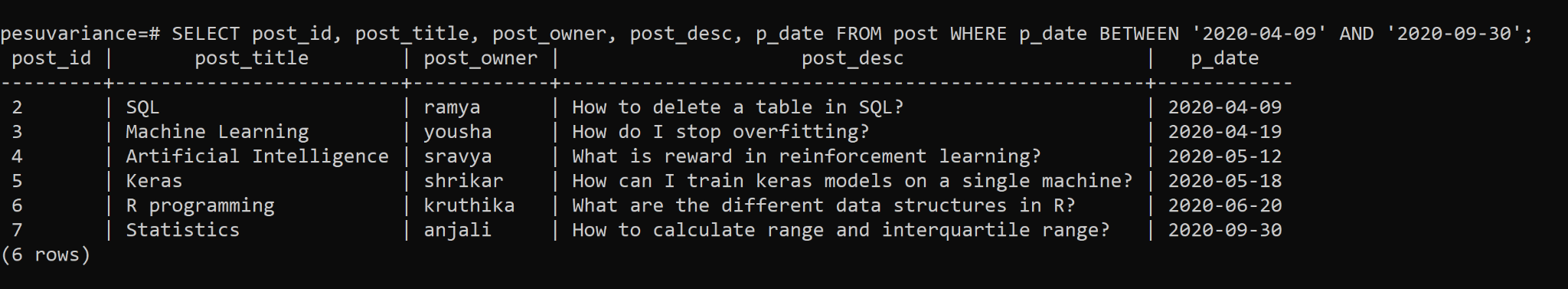
1. **Get the number of questions in a quiz which has its complexity as hard. (Quiz)**

SELECT quiz\_name, no\_questions, complexity FROM quiz WHERE complexity = 'Hard';



1. **Post owners with the posts in between some time. (Post)**

SELECT post\_id, post\_title, post\_owner, post\_desc, p\_date FROM post WHERE p\_date BETWEEN '2020-04-09' AND '2020-09-30';

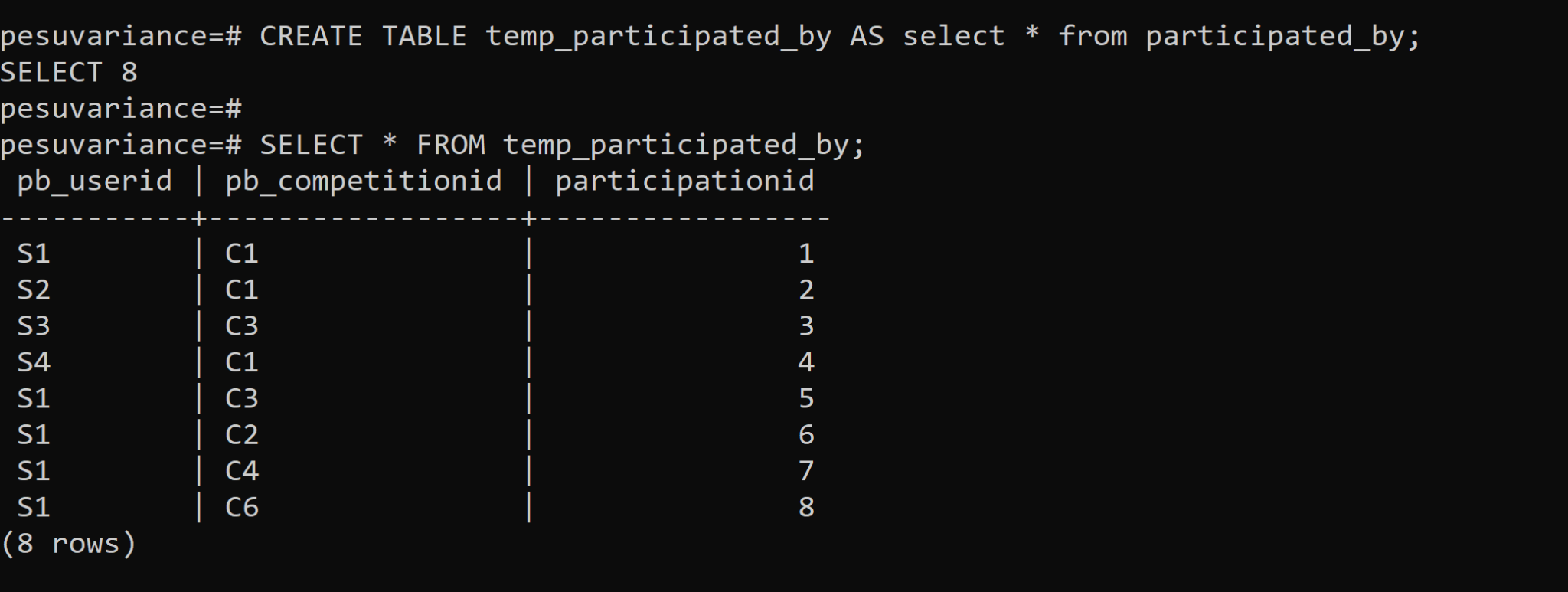


**Complex & Nested Queries:**

1. **Get the names of the competitions being participated in by a particular user (Competitions)**

CREATE TABLE temp\_participated\_by AS select \* from participated\_by;

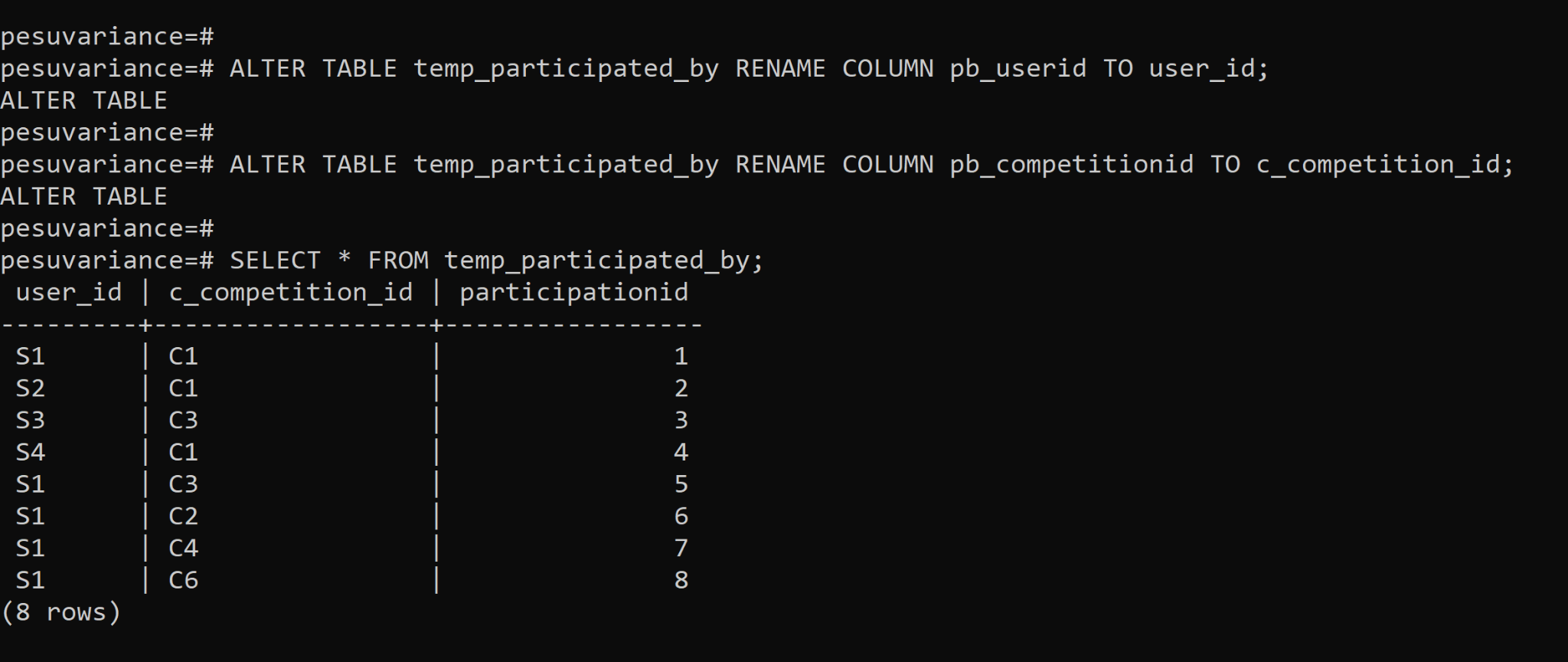
SELECT \* FROM temp\_participated\_by;



ALTER TABLE temp\_participated\_by RENAME COLUMN pb\_userid TO user\_id;

ALTER TABLE temp\_participated\_by RENAME COLUMN pb\_competitionid TO c\_competition\_id;

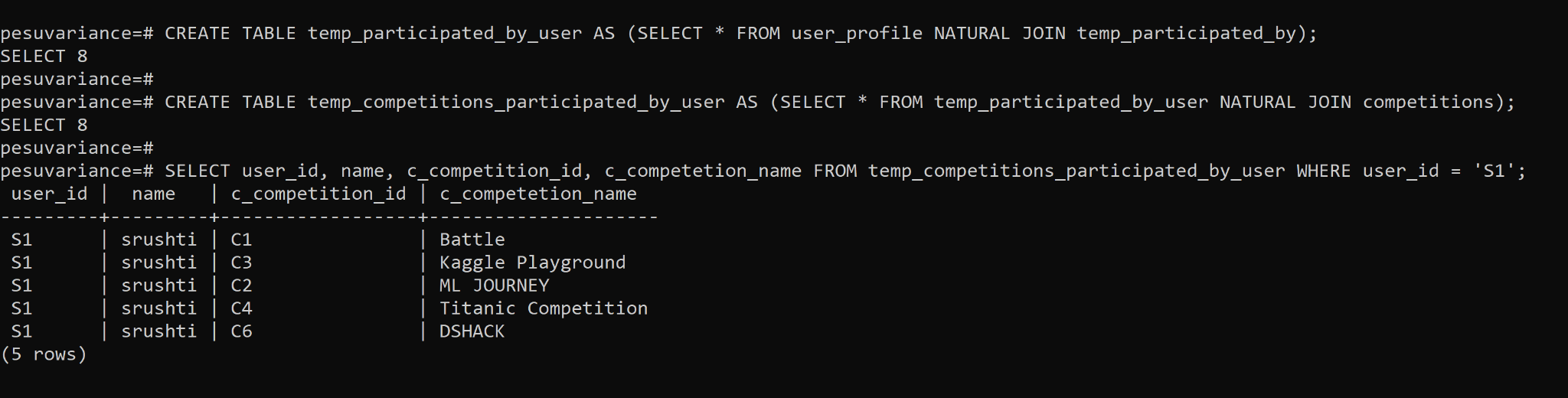
SELECT \* FROM temp\_participated\_by;



CREATE TABLE temp\_participated\_by\_user AS (SELECT \* FROM user\_profile NATURAL JOIN temp\_participated\_by);

CREATE TABLE temp\_competitions\_participated\_by\_user AS (SELECT \* FROM temp\_participated\_by\_user NATURAL JOIN competitions);

SELECT user\_id, name, c\_competition\_id, c\_competetion\_name FROM temp\_competitions\_participated\_by\_user WHERE user\_id = 'S1';



1. **Find the number of downloads and uploads of the datasets by the users who have both uploaded and downloaded.**

CREATE TABLE temp\_downloads AS select \* from downloads;

ALTER TABLE temp\_downloads RENAME COLUMN do\_userid TO user\_id;

CREATE TABLE temp\_user\_downloads AS (SELECT \* FROM user\_profile NATURAL JOIN temp\_downloads);

CREATE TABLE temp\_user\_gb\_downloads AS SELECT user\_id, COUNT(do\_datasetid) AS downloads FROM temp\_user\_downloads GROUP BY user\_id;

select \* from temp\_user\_gb\_downloads;

CREATE TABLE temp\_uploads AS select \* from dataset;

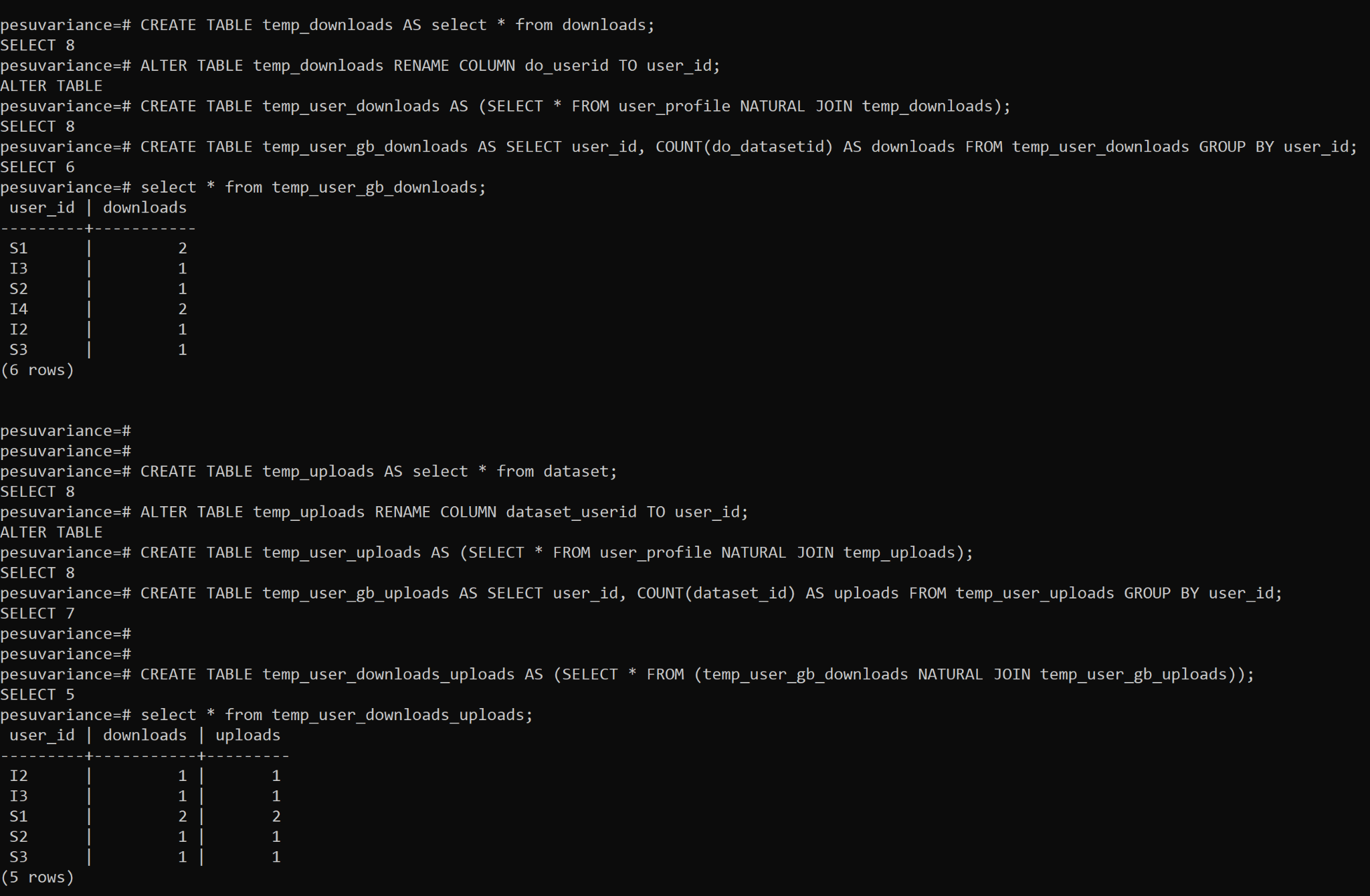
ALTER TABLE temp\_uploads RENAME COLUMN dataset\_userid TO user\_id;

CREATE TABLE temp\_user\_uploads AS (SELECT \* FROM user\_profile NATURAL JOIN temp\_uploads);

CREATE TABLE temp\_user\_gb\_uploads AS SELECT user\_id, COUNT(dataset\_id) AS uploads FROM temp\_user\_uploads GROUP BY user\_id;

CREATE TABLE temp\_user\_downloads\_uploads AS (SELECT \* FROM (temp\_user\_gb\_downloads NATURAL JOIN temp\_user\_gb\_uploads));

select \* from temp\_user\_downloads\_uploads;

****

1. **Obtain the submission IDs of ranks 1,2,3 for a particular competition from the leaderboard.**

CREATE TABLE temp\_leaderboard AS (SELECT \* FROM leaderboard);

ALTER TABLE temp\_leaderboard RENAME COLUMN l\_competitionid TO c\_competition\_id;

CREATE TABLE temp\_leaderboard\_competition AS (SELECT \* FROM (temp\_leaderboard NATURAL JOIN competitions));

SELECT c\_competition\_id, c\_competetion\_name, submission\_id, l\_rank FROM temp\_leaderboard\_competition WHERE c\_competition\_id = 'C1' AND l\_rank IN (1, 2, 3);

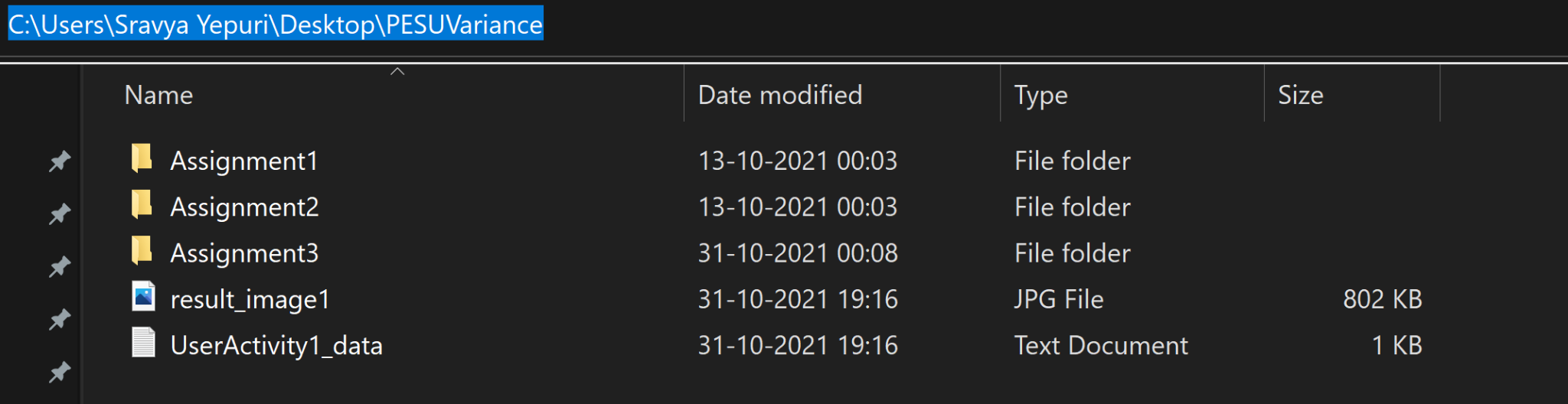


1. **Retrieve the profile picture and user activity of the first user in the database.**

**User\_retr.py**



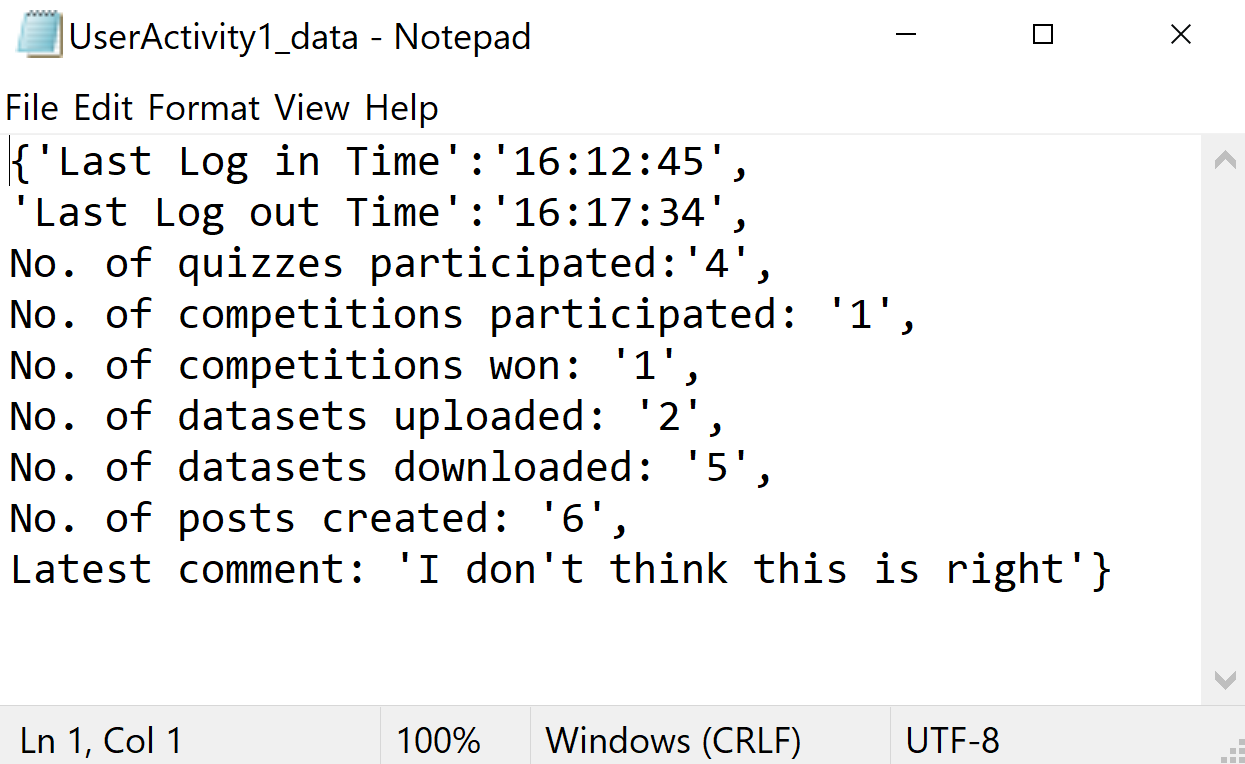
**Files location:**

****

**Result\_image1.jpg**

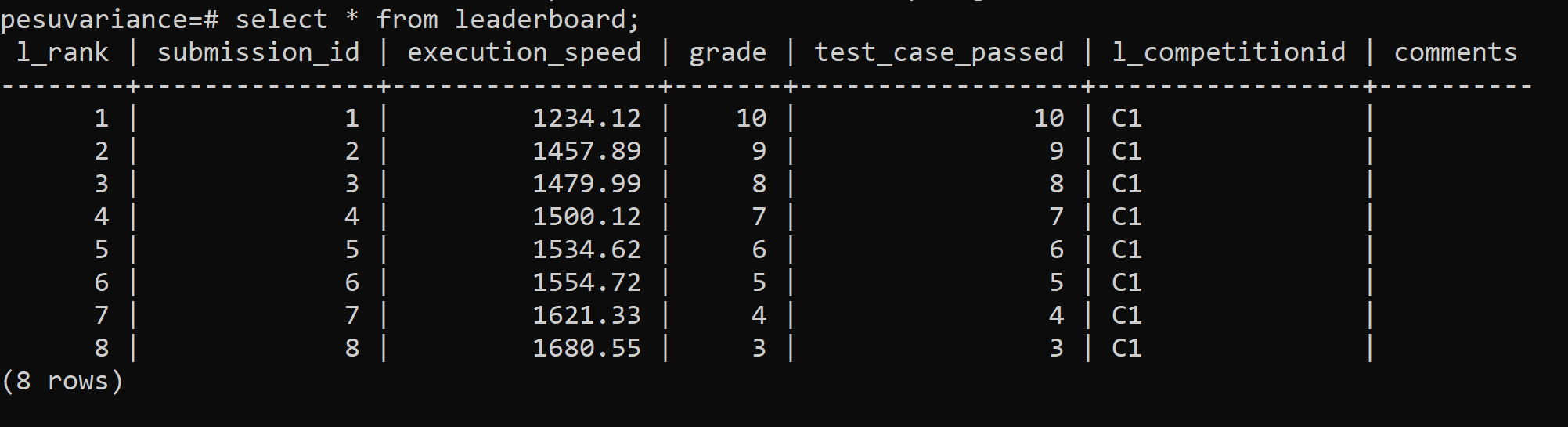
****

**UserActivity1\_data.txt**



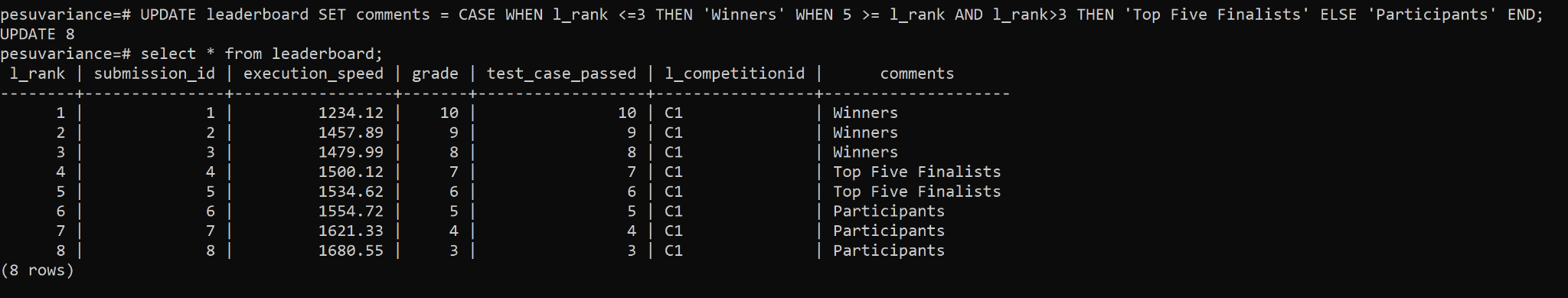
1. **Add comments based on the rank of the teams in the leaderboard using Case..end.**

ALTER TABLE LEADERBOARD ADD COLUMN comments VARCHAR(150);



UPDATE leaderboard SET comments = CASE WHEN l\_rank <=3 THEN 'Winners' WHEN 5 >= l\_rank AND l\_rank>3 THEN 'Top Five Finalists' ELSE 'Participants' END;

select \* from leaderboard;



1. **Display submission ids who have execution time lesser than average.**

CREATE TABLE temp\_avg\_exspeed AS (SELECT AVG(execution\_speed) FROM LEADERBOARD);

SELECT \* FROM temp\_avg\_exspeed;

SELECT submission\_id, execution\_speed FROM LEADERBOARD WHERE execution\_speed <= (SELECT \* FROM temp\_avg\_exspeed);



**Access Privileges for users:**

1. **Student**

CREATE USER Student WITH PASSWORD ‘student’ createdb;

GRANT SELECT,UPDATE,INSERT ON TABLE DATASET to Student;

GRANT SELECT ON TABLE Quiz to Student;

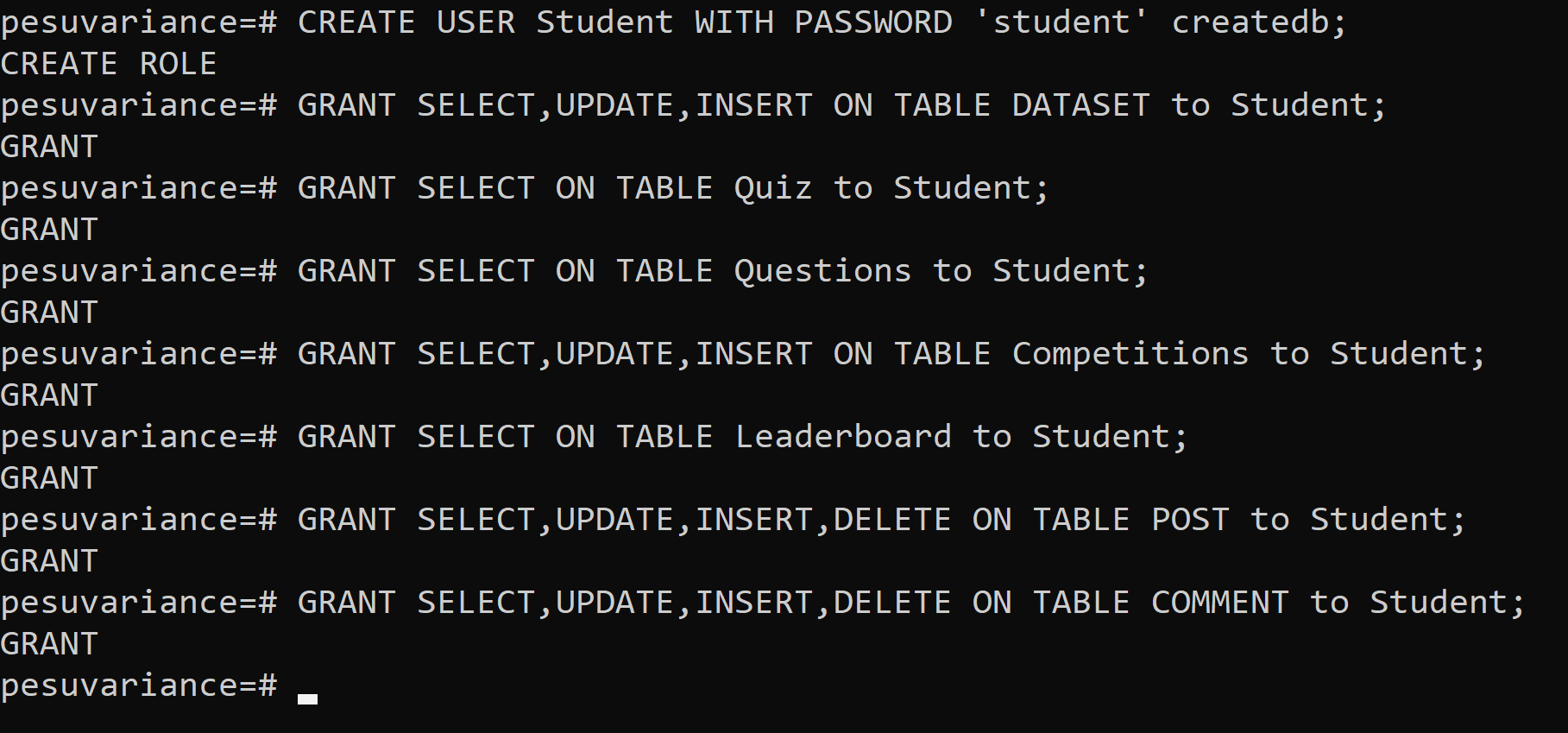
GRANT SELECT ON TABLE Questions to Student;

GRANT SELECT,UPDATE,INSERT ON TABLE Competitions to Student;

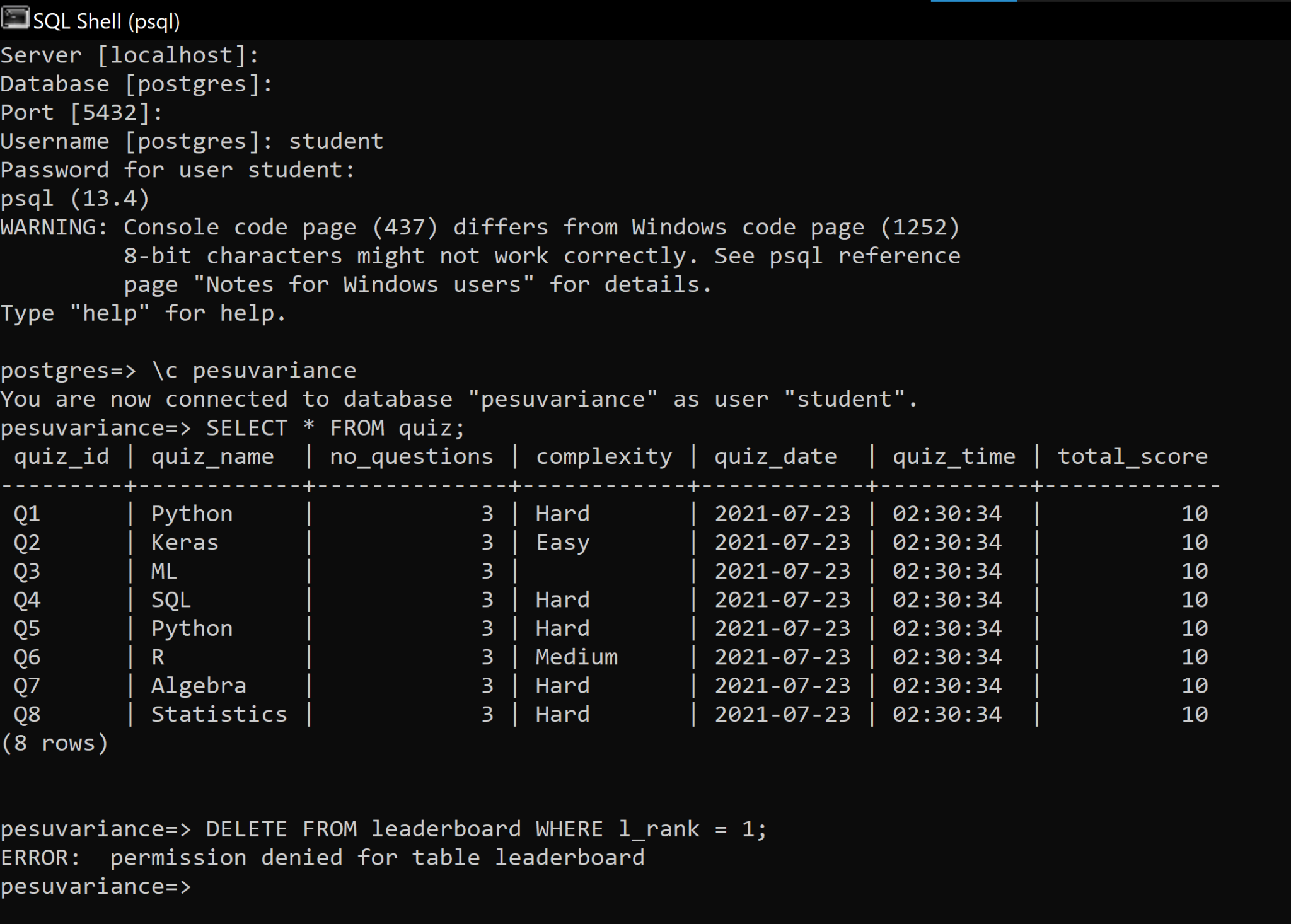
GRANT SELECT ON TABLE Leaderboard to Student;

GRANT SELECT,UPDATE,INSERT,DELETE ON TABLE POST to Student;

GRANT SELECT,UPDATE,INSERT,DELETE ON TABLE COMMENT to Student;



Student Accesses:



1. **Instructor**

CREATE USER Instructor WITH PASSWORD ‘instructor’ createdb;

GRANT SELECT,UPDATE,DELETE,INSERT ON TABLE DATASET to Instructor;

GRANT SELECT,UPDATE,DELETE,INSERT ON TABLE Quiz to Instructor;

GRANT SELECT,UPDATE,DELETE,INSERT ON TABLE Questions to Instructor;

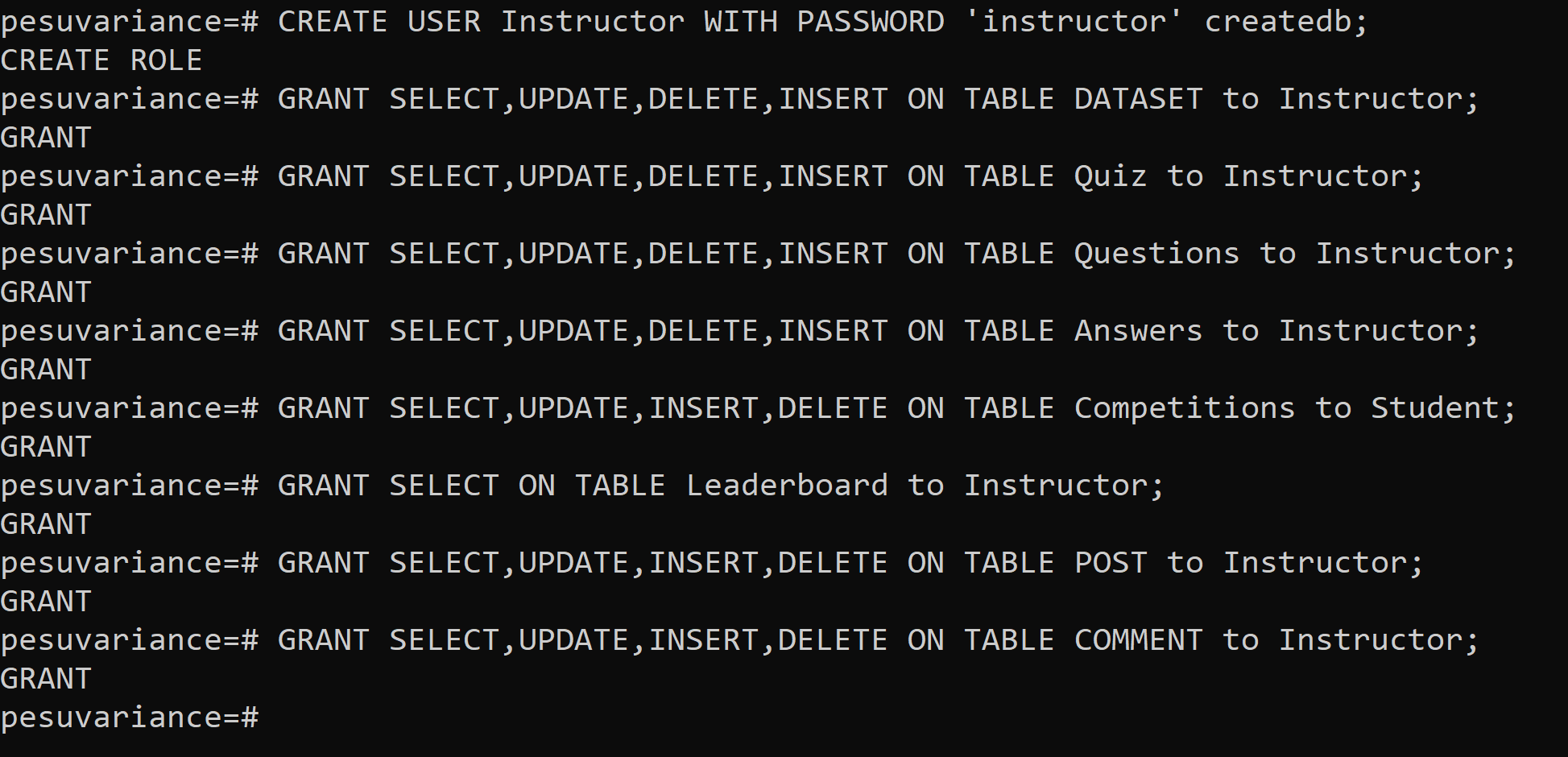
GRANT SELECT,UPDATE,DELETE,INSERT ON TABLE Answers to Instructor;

GRANT SELECT,UPDATE,INSERT,DELETE ON TABLE Competitions to Student;

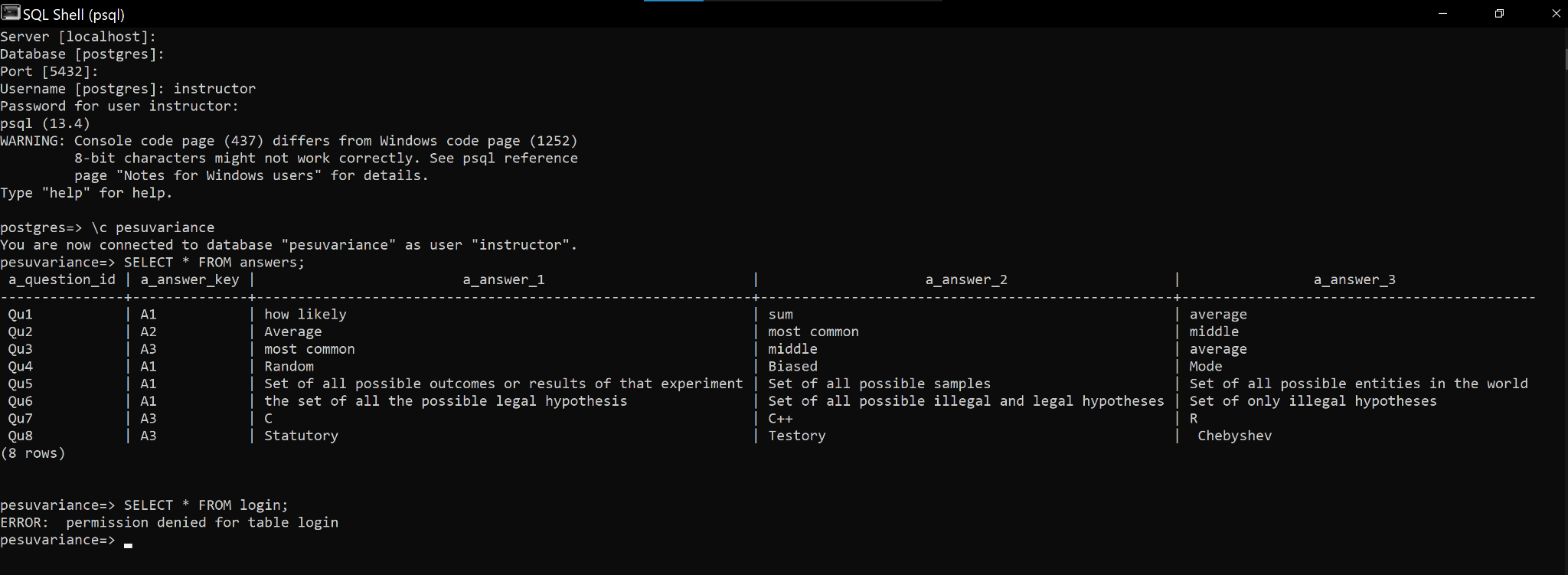
GRANT SELECT ON TABLE Leaderboard to Instructor;

GRANT SELECT,UPDATE,INSERT,DELETE ON TABLE POST to Instructor;

GRANT SELECT,UPDATE,INSERT,DELETE ON TABLE COMMENT to Instructor;



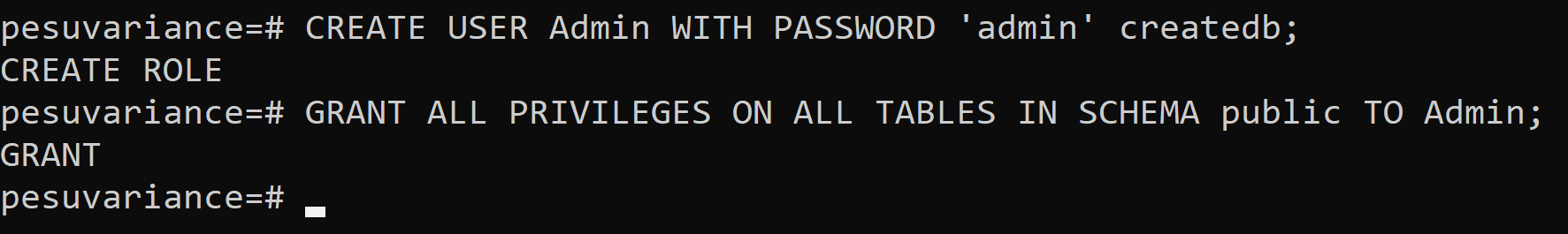
Instructor Accesses:



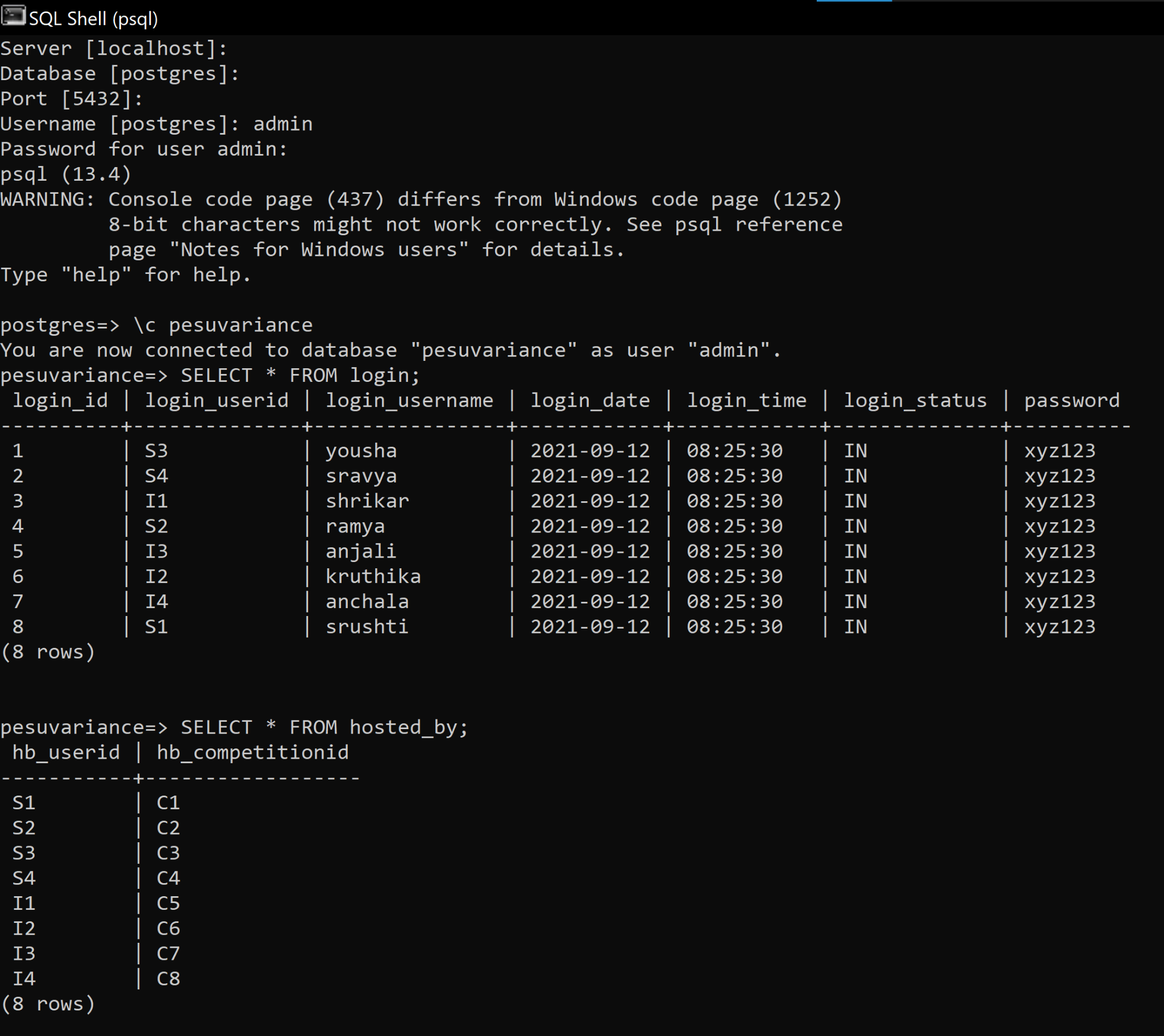
1. **Admin**

CREATE USER Admin WITH PASSWORD ‘admin’ createdb;

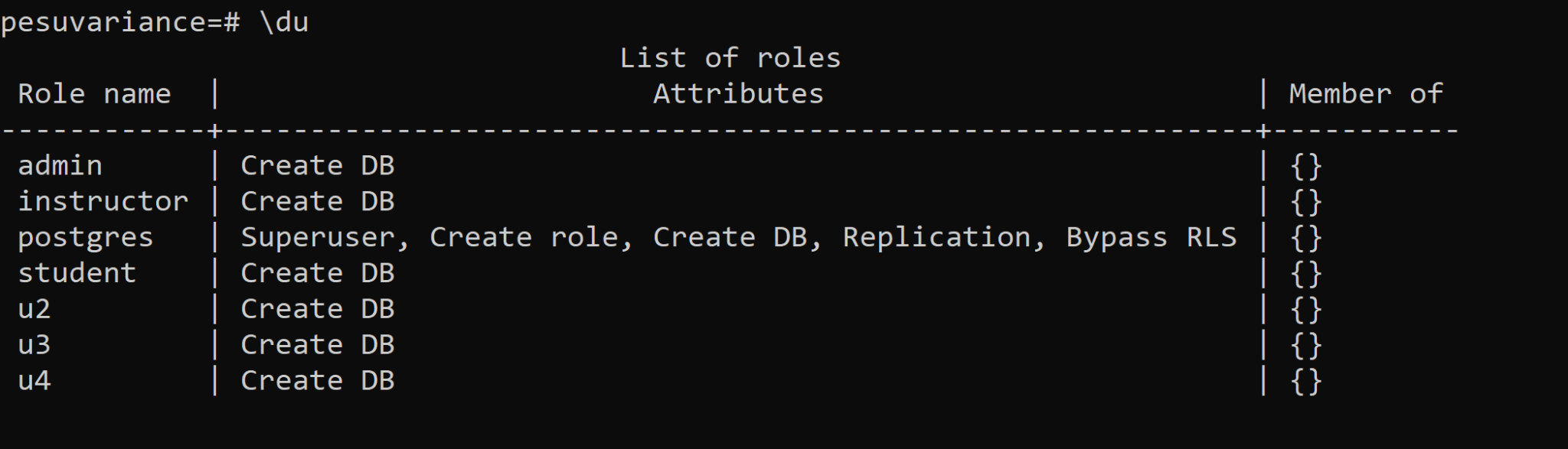
GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO Admin;



Admin accesses:



All users:



**Contribution of each member:**

Shrikar Madhu - Simple Queries, Complex Queries, Creating roles and granting permissions.  
Sravya Yepuri - Simple Queries, Complex Queries, Creating roles and granting permissions.

Sri Ramya Priya Vedula - Simple Queries, Complex Queries, Creating roles and granting permissions.

Yousha Mahamuni - Simple Queries, Complex Queries, Creating roles and granting permissions.

**Time spent to complete assignment 2:**

Simple queries - 2 Hours

Complex queries - 4 Hours

Access privileges for users - 3 Hours

Making final report - 2 Hours

Google Drive Link:

https://drive.google.com/drive/folders/1c4edDO3BsWH6p0nfCj5TaMz7a0MN7vdl