Introduction

You have learned how to design a movie recommendation database. The assignment will give you an opportunity to create such a database from scratch and build applications on top of this database. Assignment 2 has the same background information with Assignment 1. We continue to use the tables you created in Assignment 1.

Requirement

You should assume the data has been loaded into the database. Then you need to implement the following SQL queries. For each query, we provide an example of the schema of the saved query result.

1. Write a SQL query to return the total number of movies for each genre. Your query result should be saved in a table called “query1” which has two attributes: name, moviecount.

	name text	moviecount bigint
1	Romance	1685

2. Write a SQL query to return the average rating per genre. Your query result should be saved in a table called “query2” which has two attributes: name, rating.

	name text	rating numeric
1	Drama	3.8204275534441805

3. Write a SQL query to return the movies have at least 10 ratings. Your query result should be saved in a table called “query3” which has two attributes: title, countofratings.

	title text	countofratings bigint
1	Sleepy Hollow (1999)	11

4. Write a SQL query to return all “Comedy” movies, including movieid and title. Your query result should be saved in a table called “query4” which has two attributes, movieid and title.

	movieid integer	title text
1	33004	Hitchhiker's Guide to the Galaxy, The (2005)

5. Write a SQL query to return the average rating per movie. Your query result should be saved in a table called “query5” which has two attributes, title and average.

	title text	average numeric
1	Where the Heart Is (2000)	4.5000000000000000

6. Write a SQL query to return the average rating for all “Comedy” movies. Your query result should be saved in a table called “query6” which has one attribute, average.

	average numeric
1	3.5797206165703276

7. Write a SQL query to return the average rating for all movies and each of these movies is both “Comedy” and “Romance”. Your query result should be saved in a table called “query7” which has one attribute, average.

	average numeric
1	3.6989528795811518

8. Write a SQL query to return the average rating for all movies and each of these movies is “Romance” but not “Comedy”. Your query result should be saved in a table called “query8” which has one attribute, average.

	average numeric
1	3.7429411764705882

9. Find all movies that are rated by a User such that the userId is equal to v1. The v1 will be an integer parameter passed to the SQL query. Your query result should be saved in a table called “query9” which has two attributes, movieid and rating.

	movieid integer	rating numeric
1	110	5

Above all, your script should be able to generate 9 tables, namely, “query1”, “query2”, …, “query9”.

Assignment Tips!

1. All table names and attribute names must be in lowercase and exactly the same with the specification.

2. The delimiter of all files is "%"

3. You should use the following command to save your query result to a table.

CREATE TABLE query0 AS

YOUR SQL STATEMENT

Hint about Q9:

For instance, select the user from the users table which has userID = v1 and store it in query0 and rename the “username” column to “userfullname”.

psql -f solution.sql -v v1=123

In your SQL script:

CREATE TABLE query0 AS

SELECT username AS userfullname

FROM users

WHERE users.userid = :v1

4. Do not put “create/select/drop database”, or “set system settings or encoding” in your SQL script. This may lead to point deductions. Don’t re-create tables of assignment 1 and don’t load any data.

5. The rows in your query result table **don’t have to be sorted**.

6. You are free to create any other temp/permanent views, temp/permanent tables to help your queries.