

CSP 609 - PG SOFTWARE LAB
Lab Assignment 1

Submission Deadline: Jan 29 2018 8:00pm
Total Weightage of the Assignment: 6%

General Instructions:

- All specifications must be strictly followed. Failure to do so may lead to substantial loss of points.
- It is important to go through the relevant reading material, there may be a viva in the following week on topics related to this lab assignment.
- All submissions must be made on the Moodle site of the course.
- You may use the internet for searching the syntax of the SQL commands.
- **Each submission must have the name and roll number of the student.**
- For each of queries, you are expected to load few tuples (e.g., 4-5) using the insert command into your schema and then run your SQL (select based) query. This would help you in self-evaluating your SQL based on the results obtained.
- **However, care must be taken that your SQL queries are general and conceptually correct so that they give the right answer on any arbitrary dataset.**
- **Substantial points would be deducted if the correctness of your answer (the SQL query) turns out to be limited to your (or just a few) dataset.**
- **Make sure there are no errors in the SQL queries you submit to us for evaluation.**

Important Links:

- SQL tutorial: <https://www.w3schools.com/sql/>
- Spooling the results in PostgreSQL: <https://dzone.com/articles/spooling-queries-with-results-in-psql>

Question 1:

Step 1: Create the following schema in your database:

Movies(title, year, length , genre, studioName, producerCert#)
StarsIn (movieTitle, movieYear, starName)
MovieStar(name, address, gender, YearofBirth)
MovieExec(cert#, name, address, netWorth)
Studio(name, registered_country, PresidentCert#)

1. The underlined attributes are the primary keys of the corresponding relation.
2. producerCert# refers to cert# in the **MovieExec** table. PresidentCert# refers to cert# in the **MovieExec** table.
3. starName attribute in the **StarsIn** relation refers to Name attribute in the **MovieStar** relation.
4. movieTitle and movieYear in the **StarsIn** relation refer to title and year attributes in the **Movies** relations.

5. studioName attribute in **Movies** relation refers to Name attribute in the **Studio** relation.
6. Put some data into the tables. Kindly use your intuition to decide suitable data for each for the following SQL queries. You may prepare (i.e., corresponding insert table commands) data before and load it all at once for convenience.

Step 2:

Write and sql query for each of the following select queries. Use the results obtained to self-evaluate the correctness of the queries. Care must be taken such that your SQL queries are general and conceptually correct so that they give the right answer on any arbitrary dataset.

1. For each MovieStar retrieve the total number of movies he/she have worked in. Your query should retrieve the information of only those MovieStars whose year of birth > 1985.
2. Find all the studios who produced 'action' movies. Name of the studio should appear only once in the result.
3. For each Producer determine the average length of movies he/she has produced. And you should consider only those producers who have produced 2 or more movies.
4. Find MovieExecutives who are worth more than 100000 and have produced a 'comedy' movie with 'MGM' studios. Your result should return both the name of the MovieExecutive and the name of the Movie.
5. Find all the movies which are longer than the 'Titanic'.
6. Find all the MovieExecutives who are worth more than the movie executive 'James Cameron'.
7. Find all the actors who were born before 1980 and acted in 'action' movies.
8. Find all the male actors who worked in films produced by 'James Cameron'.

Things to be submitted:

Text file containing the following:

- a. SQL commands for creating the schema.
- b. SQL commands for inserting the data.
- c. SQL commands the results for each of the select queries mentioned in Step 2.