DDB – Spring 2006

Lab #4 - SQL

You should take max 3 hours to complete this lab.
You can refer to lab#1 for documentation on how to log in to MySQL.

Consider the following database:

people(person_id, name, year-of-birth);
-- year-of-birth: 4 digit numeric
movies(movie_id, name, year)
role(movie_id, person_id, role)
review(movie_id, reviewer_name, review_location, score, review_text)
-- score: number of stars 0 is as bad as it gets, 6 is best

The file movies.sql contains the table creation and sample data. You should formulate the following queries. Note that at the exam you will have to submit queries and an execution session with all queries and the returned result.

1) List all actors who have not directed any movie

2) List all reviewers who have reviewed every movie released in 2005

3) List for each decade, the number of movies released in that decade.
(Note: look at the MySQL mathematical functions. One of them will help define decades.)

4) Create a relation called reviewer_avg(reviewer, avg)
containing for each reviewer, the average score given by the reviewer across all movies.

5) Create a relation called movie_avg_rev(movie_id, avg_rev)
containing the average review score (across all reviewers) of the movie.

6) Create a relation called reviewed_movie_avg(reviewer, avg_of_avg)
which contains for each reviewer, the average, across all movies reviewed by the reviewer, of the avg_rev score from movie_avg_rev.

7) List for each reviewer, a harshness score, computed as the value for that reviewer from Query 4 divided by the value for that reviewer from Query 6.

8) Show all review scores, but scaled by the harshness score for that reviewer, got from Query 7.

9) Create a relation new_people with the same schema as people and containing the same data

10) Suppose several people have the same name, but assume no two people with the same name have the same year of birth.
Update the name of each person in new_people
by concatenating the number of people with the same name, born earlier.

-- e.g. with two people named Bush, the senior one becomes Bush0
-- and the other one becomes Bush1. Don't worry about number
-- formatting, I will accept formats like Bush000001!