CS 61A Summer 2017 Structure and Interpretation of Computer Programs

Quiz 14 Solutions

INSTRUCTIONS

- You have 10 minutes to complete this quiz.
- The exam is closed book, closed notes, closed computer, closed calculator.
- The final score for this quiz will be assigned based on **effort** rather than correctness.
- Mark your answers **on the exam itself**. We will *not* grade answers written on scratch paper.
- For multiple choice questions,
  - ☐ means mark **all options** that apply
  - ◯ means mark a **single choice**

Last name	
First name	
Student ID number	
CalCentral email (`_@berkeley.edu`)	
Teaching Assistant	◯ Alex Stennet ◯ Kelly Chen ◯ Angela Kwon ◯ Michael Gibbes ◯ Ashley Chien ◯ Michelle Hwang ◯ Joyce Luong ◯ Mitas Ray ◯ Karthik Bharathala ◯ Rocky Duan ◯ Kavi Gupta ◯ Samantha Wong
Name of the person to your left	
Name of the person to your right	
All the work on this exam is my own. **(please sign)**	

1. (5 points) Sequels

Assume that the following table of movie `ratings` has been created.

```
create table ratings as
  select "The Matrix" as title,    1999 as year, 9 as rating union
  select "The Matrix Reloaded",    2003,         7           union
  select "The Matrix Revolutions", 2003,         5           union
  select "Toy Story",              1994,         8           union
  select "Toy Story 2",            1999,         8           union
  select "Toy Story 3",            2010,         9           union
  select "Terminator",             1984,         8           union
  select "Judgment Day",           1991,         9           union
  select "Rise of the Machines",   2003,         5;
```

Express the following queries in SQL using only features we've covered in this course.

(a) (1 pt) Select the titles of all movies that have a rating greater than 7 in alphabetical order.

```
select title from ratings where rating > 7 order by title;
```

(b) (2 pt) Select the titles of all movie *triplets* for which there exist 3 movies that have the same rating. Include the titles of all three movies in the result. The results should appear in alphabetical order.

```
Judgement Day|The Matrix|Toy Story 3
Terminator|Toy Story|Toy Story 2
```

```
select r1.title, r2.title, r3.title

    from ratings as r1, ratings as r2, ratings as r3

    where r1.rating = r2.rating and r2.rating = r3.rating and r1.title < r2.title and

          r2.title < r3.title order by r1.title, r2.title, r3.title;
```

(c) (2 pt) For each group of movies containing at least 2 movies with the same rating, select the rating and *year difference (from the group average)* of the first movie released to achieve that rating. For example, `The Matrix Reloaded` was the only movie to receive a rating of 7 so it does not have a group. `Judgement Day` was released in 1991, 9 years before the average year (2000) for movies rated 9.

```
5|0
8|-8.333333333333258
9|-9
```

```
with average as (select rating, avg(year) as year

                 from ratings group by rating having count(*) >= 2)

select ratings.rating, min(ratings.year - average.year)

    from ratings, average

    where ratings.rating = average.rating

    group by ratings.rating;
```

The printed copy of this quiz had a bug in the problem because it asked for the title of the first movie released to achieve each rating. There's a problem of ambiguity when aggregating the rows in a group: once we've reduced all the rows into one row consisting of the lowest year difference, how do we know which title corresponds to that year? Different SQL engines behave differently when trying to answer this question since it is not compliant SQL. Some SQL engines will not even process this query! A nice extenion exercise would be to think of how we could correctly solve the original problem of choosing the title of the first movie to achieve the rating, along with the year difference. A solution most likely requires a little more skeleton than what was provided on the quiz.