SQL Movie-Rating Modification Exercises

Bookmark this page

Exercise due Aug 29, 2021 19:00 +08 Completed

You've started a new movie-rating website, and you've been collecting data on reviewers' ratings of various movies. There's not much data yet, but you can still try out some data modifications. Here's the schema:

Movie (mID, title, year, director)
English: There is a movie with ID number *mID*, a *title*, a release *year*, and a *director*.

Reviewer (rID, name)
English: The reviewer with ID number *rID* has a certain *name*.

Rating (rID, mID, stars, ratingDate)
English: The reviewer *rID* gave the movie *mID* a number of *stars* rating (1-5) on a certain *ratingDate*.

Your modifications will run over a small data set conforming to the schema. View the database. (You can also download the schema and data.)

Instructions: You are to write each of the following data modification commands using SQL. Our back-end runs each modification using SQLite on the original state of the sample database. It then performs a query over the modified database to check whether your command made the correct modification, and restores the database to its original state.

You may perform these exercises as many times as you like, so we strongly encourage you to keep working with them until you complete the exercises with full credit.

Q1
1/1 point (graded)
Add the reviewer Roger Ebert to your database, with an rID of 209.

Note: Your queries are executed using SQLite, so you must conform to the SQL constructs supported by SQLite.

```
insert into Reviewer values(209, 'Roger Ebert');
```

Press ESC then TAB or click outside of the code editor to exit

Submit

Correct

To check your data modification statement for Q1, we ran the following query after your modification: *select * from Reviewer order by rID, name*

Q2
1/1 point (graded)
For all movies that have an average rating of 4 stars or higher, add 25 to the release year. (Update the existing tuples; don't insert new tuples.)

Note: Your queries are executed using SQLite, so you must conform to the SQL constructs supported by SQLite.

```
update Movie
set year = year + 25
where mID in (
    select mID
    from Rating
    group by mID
    having avg(stars) >= 4);
```

Press ESC then TAB or click outside of the code editor to exit

Submit

Correct

To check your data modification statement for Q2, we ran the following query after your modification: *select * from Movie order by mID*

Q3
1/1 point (graded)
Remove all ratings where the movie's year is before 1970 or after 2000, and the rating is fewer than 4 stars.

Note: Your queries are executed using SQLite, so you must conform to the SQL constructs supported by SQLite.

```
delete from Rating
where mID in (
    select mID
    from Movie
    where year < 1970 or year > 2000)
and stars < 4;
```

Press ESC then TAB or click outside of the code editor to exit

Submit

Correct

To check your data modification statement for Q3, we ran the following query after your modification: *select R.rID, R.mID, R.stars, M.title, M.year from Rating R join Movie M on (R.mID = M.mID) order by R.rID, R.mID*

Previous

Next Up: SQL Social-Network Modification Exercises
2 min + 1 activity