Feedback — SQL Movie-Rating Query Exercises (extras)

Help Center

You submitted this quiz on **Mon 27 Oct 2014 11:47 AM PDT**. You got a score of **12.00** out of **12.00**.

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 interesting queries. 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 queries 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 queries using SQL. The "Run Query" command will help you develop and debug your queries by running them using SQLite over the sample database.

Important Notes:

- Your queries are executed using SQLite, so you must conform to the SQL constructs supported by SQLite.
- Unless a specific result ordering is asked for, you can return the result rows in any order.
- *You are to translate the English into a SQL query that computes the desired result over all possible databases.* All we actually check is that your query gets the right answer on the small sample database. Thus, even if your solution is marked as correct, it is possible that your query does not correctly reflect the problem at hand. (For example, if we ask for a complex condition that requires accessing all of the tables, but over our small data set in the end the condition is satisfied only by Star Wars, then the query "select title from Movie where title = 'Star Wars'" will be marked correct even though it doesn't reflect the actual question.) Circumventing the system in this fashion will get you a high score on the exercises, but it

won't help you learn SQL. On the other hand, an incorrect attempt at a general solution is unlikely to produce the right answer, so you shouldn't be led astray by our checking system.

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 a full score.

NOTE: REMEMBER TO CLICK "Submit" WHEN YOU ARE DONE!
Please be patient as it does take time to check all of the exercises.

Question 1

Find the names of all reviewers who rated Gone with the Wind.

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

You entered:

```
select re.name
from Reviewer as re,
  Rating as ra,
  Movie as m
where re.rID = ra.rID
```

Run Query

Your Answer		Score	Explanation

select re.name
from Reviewer as re,
  Rating as ra,
  Movie as m
where re.rID = ra.rID
  and m.mID = ra.mID
  and m.title = "Gone with the Wind"
group by re.name
	✓	1.00	**Correct**
			
			Your Query Result:
			Mike Anderson
			Sarah Martinez
			
			Expected Query Result:
			Mike Anderson
			Sarah Martinez

Total		1.00 / 1.00	

Question Explanation

Note
Even if your solution is marked as correct, it is possible that your query does not correctly reflect the problem at hand. All we check is that your query gets the right answer on the small sample database. For example, if we asked for a complex condition that requires accessing all of the tables, but over our small data set in the end the condition is satisfied only by Star Wars, then the query "select title from Movie where title = 'Star Wars'" will be marked correct even though it doesn't reflect the actual question. Circumventing the system in this fashion will get you a high score on the exercises, but it won't help you learn SQL. On the other hand, an incorrect attempt at a general solution is unlikely to produce the right answer, so you shouldn't be led astray by our checking system.

Question 2

For any rating where the reviewer is the same as the director of the movie, return the reviewer name, movie title, and number of stars.

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

You entered:

```
select re.name, m.title, ra.stars
from Reviewer as re,
  Rating as ra,
  Movie as m
where re.rID = ra.rID
```

Run Query

Your Answer **Score** **Explanation**

```
select re.name, m.title, ra.stars
from Reviewer as re,
  Rating as ra,
  Movie as m
where re.rID = ra.rID
  and m.mID = ra.mID
  and m.director = re.name
group by re.name
```

✓ 1.00

Correct

Your Query Result:

James Cameron	Avatar	5

Expected Query Result:

James Cameron	Avatar	5

Total 1.00 / 1.00

Question Explanation

Note
Even if your solution is marked as correct, it is possible that your query does not correctly reflect the problem at hand. All we check is that your query gets the right answer on the small sample database. For example, if we asked for a complex condition that requires accessing all of the tables, but over our small data set in the end the condition is satisfied only by Star Wars, then the query "select title from Movie where title = 'Star Wars'" will be marked correct even though it doesn't reflect the actual question. Circumventing the system in this fashion will get you a high score on the exercises, but it won't help you learn SQL. On the other hand, an incorrect attempt at a general solution is unlikely to produce the right answer, so you shouldn't be led astray by our checking system.

Question 3

Return all reviewer names and movie names together in a single list, alphabetized. (Sorting by the first name of the reviewer and first word in the title is fine; no need for special processing on last names or removing "The".)

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

You entered:

```
select re1.name
from Reviewer as re1
group by re1.name
union
select m2.title
```

Run Query

Your Answer		Score	Explanation
select re1.name from Reviewer as re1 group by re1.name union select m2.title from Movie as m2 group by m2.title	✓	1.00	**Correct**

Your Query Result:

Ashley White
Avatar
Brittany Harris
Chris Jackson
Daniel Lewis
E.T.

Elizabeth Thomas
Gone with the Wind
James Cameron
Mike Anderson
Raiders of the Lost Ark
Sarah Martinez
Snow White
Star Wars
The Sound of Music
Titanic

Expected Query Result:

Ashley White
Avatar
Brittany Harris
Chris Jackson
Daniel Lewis
E.T.
Elizabeth Thomas
Gone with the Wind
James Cameron
Mike Anderson
Raiders of the Lost Ark
Sarah Martinez
Snow White
Star Wars
The Sound of Music
Titanic

(Order matters)

Total 1.00 / 1.00

Question Explanation

Note

Even if your solution is marked as correct, it is possible that your query does not correctly reflect the problem at hand. All we check is that your query gets the right answer on the small sample database. For example, if we asked for a complex condition that requires accessing all of the tables, but over our small data set in the end the condition is satisfied only by Star Wars, then the query "select title from Movie where title = 'Star Wars'" will be marked correct even though it doesn't reflect the actual question. Circumventing the system in this fashion will get you a high score on the exercises, but it won't help you learn SQL. On the other hand, an incorrect attempt at a general solution is unlikely to produce the right answer, so you shouldn't be led astray by our checking system.

Question 4

Find the titles of all movies not reviewed by Chris Jackson.

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

You entered:

```
select title
from Movie
group by title
having title not in
(select m.title
```

Run Query

Your Answer **Score** **Explanation**

```
select title
from Movie
group by title
having title not in
(select m.title
from Reviewer as re,
  Rating as ra,
  Movie as m
where re.rID = ra.rID
  and m.mID = ra.mID
  and re.name = "Chris Jackson"
group by m.title)
```

✓ 1.00

Correct

Your Query Result:

Avatar
Gone with the Wind
Snow White
Star Wars
Titanic

Expected Query Result:

Avatar
Gone with the Wind
Snow White

Star Wars
Titanic

Total 1.00 / 1.00

Question Explanation

Note
Even if your solution is marked as correct, it is possible that your query does not correctly reflect the problem at hand. All we check is that your query gets the right answer on the small sample database. For example, if we asked for a complex condition that requires accessing all of the tables, but over our small data set in the end the condition is satisfied only by Star Wars, then the query "select title from Movie where title = 'Star Wars'" will be marked correct even though it doesn't reflect the actual question. Circumventing the system in this fashion will get you a high score on the exercises, but it won't help you learn SQL. On the other hand, an incorrect attempt at a general solution is unlikely to produce the right answer, so you shouldn't be led astray by our checking system.

Question 5

For all pairs of reviewers such that both reviewers gave a rating to the same movie, return the names of both reviewers. Eliminate duplicates, don't pair reviewers with themselves, and include each pair only once. For each pair, return the names in the pair in alphabetical order.

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

You entered:

```
select re1.name, re2.name
from Reviewer as re1,
   Reviewer as re2,
   Rating as ra1,
   Rating as ra2
```

Run Query

Your Answer **Score** **Explanation**

```
select re1.name, re2.name
from Reviewer as re1,
   Reviewer as re2,
   Rating as ra1,
   Rating as ra2
where re1.rID = ra1.rID
```

✓ 1.00

Correct

Your Query Result:

Ashley White	Chris Jackson

```
   and re2.rID = ra2.rID
    and ra1.mID = ra2.mID
    and re1.name < re2.name
group by re1.name, re2.name
order by re1.name
```

Brittany Harris	Chris Jackson
Daniel Lewis	Elizabeth Thomas
Elizabeth Thomas	James Cameron
Mike Anderson	Sarah Martinez

Expected Query Result:

Ashley White	Chris Jackson
Brittany Harris	Chris Jackson
Daniel Lewis	Elizabeth Thomas
Elizabeth Thomas	James Cameron
Mike Anderson	Sarah Martinez

Total 1.00 / 1.00

Question Explanation

Note
Even if your solution is marked as correct, it is possible that your query does not correctly reflect the problem at hand. All we check is that your query gets the right answer on the small sample database. For example, if we asked for a complex condition that requires accessing all of the tables, but over our small data set in the end the condition is satisfied only by Star Wars, then the query "select title from Movie where title = 'Star Wars'" will be marked correct even though it doesn't reflect the actual question. Circumventing the system in this fashion will get you a high score on the exercises, but it won't help you learn SQL. On the other hand, an incorrect attempt at a general solution is unlikely to produce the right answer, so you shouldn't be led astray by our checking system.

Question 6

For each rating that is the lowest (fewest stars) currently in the database, return the reviewer name, movie title, and number of stars.

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

You entered:

```
select re.name, m.title, ra.stars
from Movie as m,
  Reviewer as re,
  Rating as ra
where m.mID = ra.mID
```

Run Query

Your Answer **Score** **Explanation**

```
select re.name, m.title, ra.stars
from Movie as m,
  Reviewer as re,
  Rating as ra
where m.mID = ra.mID
  and re.rID = ra.rID
  and ra.stars = (select min(stars) from
Rating)
```

✓ 1.00

Correct

Your Query Result:

Brittany Harris	Raiders of the Lost Ark	2
Brittany Harris	The Sound of Music	2
Chris Jackson	E.T.	2
Sarah Martinez	Gone with the Wind	2

Expected Query Result:

Brittany Harris	Raiders of the Lost Ark	2
Brittany Harris	The Sound of Music	2
Chris Jackson	E.T.	2
Sarah Martinez	Gone with the Wind	2

Total 1.00 / 1.00

Question Explanation

Note
Even if your solution is marked as correct, it is possible that your query does not correctly reflect the problem at hand. All we check is that your query gets the right answer on the small sample database. For example, if we asked for a complex condition that requires accessing all of the tables, but over our small data set in the end the condition is satisfied only by Star Wars, then the query "select title from Movie where title = 'Star Wars'" will be marked correct even though it

doesn't reflect the actual question. Circumventing the system in this fashion will get you a high score on the exercises, but it won't help you learn SQL. On the other hand, an incorrect attempt at a general solution is unlikely to produce the right answer, so you shouldn't be led astray by our checking system.

Question 7

List movie titles and average ratings, from highest-rated to lowest-rated. If two or more movies have the same average rating, list them in alphabetical order.

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

You entered:

```
select m.title, avg(ra.stars) as rating
from Movie as m,
  Rating as ra
where m.mID = ra.mID
group by m.title
```

Run Query

Your Answer		Score	Explanation
select m.title, avg(ra.stars) as rating from Movie as m, Rating as ra where m.mID = ra.mID group by m.title order by rating desc, m.title	✓	1.00	**Correct**

Your Query Result:

Snow White	4.5
Avatar	4.0
Raiders of the Lost Ark	3.33333333333
Gone with the Wind	3.0
E.T.	2.5
The Sound of Music	2.5

Expected Query Result:

Snow White	4.5
Avatar	4.0
Raiders of the Lost Ark	3.33333333333
Gone with the Wind	3.0

E.T.	2.5
The Sound of Music	2.5

(Order matters)

Total 1.00 / 1.00

Question Explanation

Note
Even if your solution is marked as correct, it is possible that your query does not correctly reflect the problem at hand. All we check is that your query gets the right answer on the small sample database. For example, if we asked for a complex condition that requires accessing all of the tables, but over our small data set in the end the condition is satisfied only by Star Wars, then the query "select title from Movie where title = 'Star Wars'" will be marked correct even though it doesn't reflect the actual question. Circumventing the system in this fashion will get you a high score on the exercises, but it won't help you learn SQL. On the other hand, an incorrect attempt at a general solution is unlikely to produce the right answer, so you shouldn't be led astray by our checking system.

Question 8

Find the names of all reviewers who have contributed three or more ratings. (As an extra challenge, try writing the query without HAVING or without COUNT.)

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

You entered:

```
select re.name
from Reviewer as re,
    Rating as ra
where re.rID = ra.rID
group by ra.rID
```

Run Query

Your Answer		Score	Explanation
select re.name from Reviewer as re, Rating as ra where re.rID = ra.rID group by ra.rID having count(*) > 2	✓	1.00	**Correct** Your Query Result:

Brittany Harris

Chris Jackson

Expected Query Result:

Brittany Harris
Chris Jackson

Total	1.00 / 1.00

Question Explanation

Note
Even if your solution is marked as correct, it is possible that your query does not correctly reflect the problem at hand. All we check is that your query gets the right answer on the small sample database. For example, if we asked for a complex condition that requires accessing all of the tables, but over our small data set in the end the condition is satisfied only by Star Wars, then the query "select title from Movie where title = 'Star Wars'" will be marked correct even though it doesn't reflect the actual question. Circumventing the system in this fashion will get you a high score on the exercises, but it won't help you learn SQL. On the other hand, an incorrect attempt at a general solution is unlikely to produce the right answer, so you shouldn't be led astray by our checking system.

Question 9

Some directors directed more than one movie. For all such directors, return the titles of all movies directed by them, along with the director name. Sort by director name, then movie title. (As an extra challenge, try writing the query both with and without COUNT.)

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

You entered:

```
select title, director
from Movie
where director in
    (select director
     from Movie
```

Run Query

Your Answer		Score	Explanation
select title, director from Movie	✓	1.00	**Correct**

```
where director in
   (select director
     from Movie
        group by director
        having count(*) > 1)
order by director, title
```

Your Query Result:

Avatar	James Cameron
Titanic	James Cameron
E.T.	Steven Spielberg
Raiders of the Lost Ark	Steven Spielberg

Expected Query Result:

Avatar	James Cameron
Titanic	James Cameron
E.T.	Steven Spielberg
Raiders of the Lost Ark	Steven Spielberg

(Order matters)

Total 1.00 / 1.00

Question Explanation

Note
Even if your solution is marked as correct, it is possible that your query does not correctly reflect the problem at hand. All we check is that your query gets the right answer on the small sample database. For example, if we asked for a complex condition that requires accessing all of the tables, but over our small data set in the end the condition is satisfied only by Star Wars, then the query "select title from Movie where title = 'Star Wars'" will be marked correct even though it doesn't reflect the actual question. Circumventing the system in this fashion will get you a high score on the exercises, but it won't help you learn SQL. On the other hand, an incorrect attempt at a general solution is unlikely to produce the right answer, so you shouldn't be led astray by our checking system.

Question 10

Find the movie(s) with the highest average rating. Return the movie title(s) and average rating. (Hint: This query is more difficult to write in SQLite than other systems; you might think of it as finding the highest average rating and then choosing the movie(s) with that average rating.)

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

You entered:

```
select m.title, avg(ra.stars) as rate
from Rating as ra,
        Movie as m
where m.mID = ra.mID
group by m.mID
```

Run Query

Your Answer **Score** **Explanation**

```
select m.title, avg(ra.stars) as rate
from Rating as ra,
        Movie as m
where m.mID = ra.mID
group by m.mID
having rate = (select max(rate)
                  from (select mID, avg(stars) as rate
                          from Rating
                          group by mID) )
```

✓ 1.00

Correct

Your Query Result:

Snow White	4.5

Expected Query Result:

Snow White	4.5

Total 1.00 / 1.00

Question Explanation

Note
Even if your solution is marked as correct, it is possible that your query does not correctly reflect the problem at hand. All we check is that your query gets the right answer on the small sample database. For example, if we asked for a complex condition that requires accessing all of the tables, but over our small data set in the end the condition is satisfied only by Star Wars, then the query "select title from Movie where title = 'Star Wars'" will be marked correct even though it doesn't reflect the actual question. Circumventing the system in this fashion will get you a high score on the exercises, but it won't help you learn SQL. On the other hand, an incorrect attempt at a general solution is unlikely to produce the right answer, so you shouldn't be led astray by our checking system.

Question 11

Find the movie(s) with the lowest average rating. Return the movie title(s) and average rating. (Hint: This query may be more difficult to write in SQLite than other systems; you might think of it as finding the highest average rating and then choosing the movie(s) with that average rating.)

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

You entered:

```
select m.title, avg(ra.stars) as rate
from Rating as ra,
        Movie as m
where m.mID = ra.mID
group by m.mID
```

Run Query

Your Answer		Score	Explanation
select m.title, avg(ra.stars) as rate from Rating as ra, Movie as m where m.mID = ra.mID group by m.mID having rate = (select min(rate) from (select mID, avg(stars) as rate from Rating group by mID))	✓	1.00	**Correct**
Total		1.00 / 1.00	

```
select m.title, avg(ra.stars) as rate
from Rating as ra,
        Movie as m
where m.mID = ra.mID
group by m.mID
having rate = (select min(rate)
                        from (select mID, avg(stars) as rate
                                  from Rating
                                  group by mID) )
```

Your Query Result:

E.T.	2.5
The Sound of Music	2.5

Expected Query Result:

E.T.	2.5
The Sound of Music	2.5

Question Explanation

Note
Even if your solution is marked as correct, it is possible that your query does not correctly reflect the problem at hand. All we check is that your query gets the right answer on the small sample database. For example, if we asked for a complex condition that requires accessing all of the tables, but over our small data set in the end the condition is satisfied only by Star Wars, then the query "select title from Movie where title = 'Star Wars'" will be marked correct even though it doesn't reflect the actual question. Circumventing the system in this fashion will get you a high score on the exercises, but it won't help you learn SQL. On the other hand, an incorrect attempt at a general solution is unlikely to produce the right answer, so you shouldn't be led astray by our checking system.

Question 12

For each director, return the director's name together with the title(s) of the movie(s) they directed that received the highest rating among all of their movies, and the value of that rating. Ignore movies whose director is NULL.

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

You entered:

```
select m.director, m.title, ra.stars
from Movie as m,
  Rating as ra,
 (select m.director, max(ra.stars) as maxrate
 from Rating as ra,
```

Run Query

Your Answer | **Score** | **Explanation**

```
select m.director, m.title, ra.stars
from Movie as m,
  Rating as ra,
 (select m.director, max(ra.stars) as
maxrate
 from Rating as ra,
          Movie as m
 where m.mID = ra.mID
        and m.director not null
 group by m.director) as table1
where table1.director = m.director
      and table1.maxrate = ra.stars
      and ra.mID = m.mID
group by m.director
```

✓ 1.00

Correct

Your Query Result:

James Cameron	Avatar	5
Robert Wise	The Sound of Music	3
Steven Spielberg	Raiders of the Lost Ark	4
Victor Fleming	Gone with the Wind	4

Expected Query Result:

James Cameron	Avatar	5
Robert Wise	The Sound of Music	3
Steven Spielberg	Raiders of the Lost Ark	4
Victor Fleming	Gone with the Wind	4

Total 1.00 / 1.00

Question Explanation

Note
Even if your solution is marked as correct, it is possible that your query does not correctly reflect the problem at hand. All we check is that your query gets the right answer on the small sample database. For example, if we asked for a complex condition that requires accessing all of the tables, but over our small data set in the end the condition is satisfied only by Star Wars, then the

query "select title from Movie where title = 'Star Wars'" will be marked correct even though it doesn't reflect the actual question. Circumventing the system in this fashion will get you a high score on the exercises, but it won't help you learn SQL. On the other hand, an incorrect attempt at a general solution is unlikely to produce the right answer, so you shouldn't be led astray by our checking system.