

Feedback — SQL Movie-Rating Query Exercises (core set)

[Help Center](#)

You submitted this quiz on **Mon 27 Oct 2014 9:44 AM PDT**. You got a score of **9.00** out of **9.00**.

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 full credit.

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 titles of all movies directed by Steven Spielberg.

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

You entered:

```
select title
from Movie
where director = "Steven Spielberg"
```

Run Query

Your Answer	Score	Explanation				
<pre>select title from Movie where director = "Steven Spielberg"</pre>	<div>✓</div> 1.00	<div>Correct</div> <div>Your Query Result:</div> <table><tr><td>E.T.</td></tr><tr><td>Raiders of the Lost Ark</td></tr></table> <div>Expected Query Result:</div> <table><tr><td>E.T.</td></tr><tr><td>Raiders of the Lost Ark</td></tr></table>	E.T.	Raiders of the Lost Ark	E.T.	Raiders of the Lost Ark
E.T.						
Raiders of the Lost Ark						
E.T.						
Raiders of the Lost Ark						
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

Find all years that have a movie that received a rating of 4 or 5, and sort them in increasing order.

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

You entered:

```
select m.year
from Movie as m,
     Rating as r
where m.mID = r.mID
and r.stars >3
```

Run Query

Your Answer	Score	Explanation						
<pre>select m.year from Movie as m, Rating as r where m.mID = r.mID and r.stars >3 group by m.year order by m.year</pre>	<div>✓</div> 1.00	<div>Correct</div> <div>Your Query Result:</div> <table><tr><td>1937</td></tr><tr><td>1939</td></tr><tr><td>1981</td></tr><tr><td>2009</td></tr></table> <div>Expected Query Result:</div> <table><tr><td>1937</td></tr><tr><td>1939</td></tr></table>	1937	1939	1981	2009	1937	1939
1937								
1939								
1981								
2009								
1937								
1939								

1981
2009

(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 3

Find the titles of all movies that have no ratings.

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

You entered:

```
select title
from Movie
where mID not in (
select m.mID
from Movie as m,
```

Run Query

Your Answer	Score	Explanation		
<pre>select title from Movie where mID not in (select m.mID from Movie as m, Rating as r where m.mID=r.mID)</pre>	<div>✓</div> 1.00	<div>Correct</div> <div>Your Query Result:</div> <table><tr><td>Star Wars</td></tr><tr><td>Titanic</td></tr></table>	Star Wars	Titanic
Star Wars				
Titanic				

Expected Query Result:

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 4

Some reviewers didn't provide a date with their rating. Find the names of all reviewers who have ratings with a NULL value for the date.

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
and ra.ratingDate is null
```

Run Query

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

```
select re.name
from Reviewer as re,
     Rating as ra
where re.rID = ra.rID
and ra.ratingDate is null
```



1.00

Correct

Your Query Result:

Chris Jackson

Daniel Lewis

Expected Query Result:

Chris Jackson

Daniel Lewis

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

Write a query to return the ratings data in a more readable format: reviewer name, movie title, stars, and ratingDate. Also, sort the data, first by reviewer name, then by movie title, and lastly by 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, ra.ratingDate
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,
ra.ratingDate
```



1.00

Correct

from Movie as m,
 Reviewer as re,
 Rating as ra
 where m.mID=ra.mID
 and re.rID = ra.rID
 order by re.name, m.title, ra.sta
 rs

Your Query Result:

Ashley White	E.T.	3	2011-01-02
Brittany Harris	Raiders of the Lost Ark	2	2011-01-30
Brittany Harris	Raiders of the Lost Ark	4	2011-01-12
Brittany Harris	The Sound of Music	2	2011-01-20
Chris Jackson	E.T.	2	2011-01-22
Chris Jackson	Raiders of the Lost Ark	4	<NULL>
Chris Jackson	The Sound of Music	3	2011-01-27
Daniel Lewis	Snow White	4	<NULL>
Elizabeth Thomas	Avatar	3	2011-01-15
Elizabeth Thomas	Snow White	5	2011-01-19
James Cameron	Avatar	5	2011-01-20
Mike Anderson	Gone with the Wind	3	2011-01-09
Sarah Martinez	Gone with the Wind	2	2011-01-22
Sarah Martinez	Gone with the Wind	4	2011-01-27

Expected Query Result:

Ashley White	E.T.	3	2011-01-02
Brittany Harris	Raiders of the Lost Ark	2	2011-01-30
Brittany Harris	Raiders of the Lost Ark	4	2011-01-12
Brittany Harris	The Sound of Music	2	2011-01-20

Chris Jackson	E.T.	2	2011-01-22
Chris Jackson	Raiders of the Lost Ark	4	<NULL>
Chris Jackson	The Sound of Music	3	2011-01-27
Daniel Lewis	Snow White	4	<NULL>
Elizabeth Thomas	Avatar	3	2011-01-15
Elizabeth Thomas	Snow White	5	2011-01-19
James Cameron	Avatar	5	2011-01-20
Mike Anderson	Gone with the Wind	3	2011-01-09
Sarah Martinez	Gone with the Wind	2	2011-01-22
Sarah Martinez	Gone with the Wind	4	2011-01-27

(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 6

For all cases where the same reviewer rated the same movie twice and gave it a higher rating the second time, return the reviewer's name and the title of the movie.

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
from Movie as m,
  Reviewer as re,
  Rating as ra
where m.mID = ra.mID
```

Run Query

Your Answer	Score	Explanation				
<pre>select re.name, m.title from Movie as m, Reviewer as re, Rating as ra where m.mID = ra.mID and re.rID = ra.rID group by re.name, m.title having count(m.title) = 2 and ra.ratingDate = max(ra.ratingDate) and ra.stars > avg(ra.stars)</pre>	<div>✓1.00</div>	<div>Correct</div> <div>Your Query Result:</div> <table><tr><td>Sarah Martinez</td><td>Gone with the Wind</td></tr></table> <div>Expected Query Result:</div> <table><tr><td>Sarah Martinez</td><td>Gone with the Wind</td></tr></table>	Sarah Martinez	Gone with the Wind	Sarah Martinez	Gone with the Wind
Sarah Martinez	Gone with the Wind					
Sarah Martinez	Gone with the Wind					
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

For each movie that has at least one rating, find the highest number of stars that movie received. Return the movie title and number of stars. Sort by movie title.

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

You entered:

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

Run Query

Your Answer

Score

Explanation

```
select m.title, ra.stars
from Movie as m,
     Rating as ra
where m.mID = ra.mID
group by ra.mID
having count(ra.mID) > 1
     and ra.mID = max(ra.mID)
order by m.title
```



1.00

Correct

Your Query Result:

Avatar	5
E.T.	3
Gone with the Wind	4
Raiders of the Lost Ark	4
Snow White	5
The Sound of Music	3

Expected Query Result:

Avatar	5
E.T.	3
Gone with the Wind	4
Raiders of the Lost Ark	4
Snow White	5
The Sound of Music	3

(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

For each movie, return the title and the 'rating spread', that is, the difference between highest and lowest ratings given to that movie. Sort by rating spread from highest to lowest, then by movie title.

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

You entered:

```
select m.title, max(ra.stars)-min(ra.stars) as spread
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, max(ra.stars)-min(ra.stars) as spread
from Movie as m,
    Rating as ra
where m.mID = ra.mID
group by m.title
order by spread desc, m.title
```

✓ 1.00

Correct

Your Query Result:

Avatar	2
Gone with the Wind	2
Raiders of the Lost Ark	2
E.T.	1
Snow White	1
The Sound of Music	1

Expected Query Result:

Avatar	2
Gone with the Wind	2
Raiders of the Lost Ark	2
E.T.	1
Snow White	1
The Sound of Music	1

(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 9

Find the difference between the average rating of movies released before 1980 and the average rating of movies released after 1980. (Make sure to calculate the average rating for each movie, then the average of those averages for movies before 1980 and movies after. Don't just calculate the overall average rating before and after 1980.)

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

You entered:

```

select avg(temp1.rating)-avg(temp2.rating)
from
(select m.year, m.mID, avg(ra.stars) as rating
from Movie as m,
      Rating as ra

```

Run Query

Your Answer	Score	Explanation
<pre> select avg(temp1.rating)-avg(temp2.rating) from (select m.year, m.mID, avg(ra.stars) as rating from Movie as m, Rating as ra where m.mID = ra.mID group by ra.mID) as temp1, (select m.year, m.mID, avg(ra.stars) as rating from Movie as m, Rating as ra where m.mID = ra.mID group by ra.mID) as temp2 where temp1.year < 1980 and temp2.year >=1980 </pre>	<p>✓ 1.00</p>	<p>Correct</p> <p>Your Query Result:</p> <div style="border: 1px solid black; padding: 2px; width: fit-content;">0.0555555555556</div> <p>Expected Query Result:</p> <div style="border: 1px solid black; padding: 2px; width: fit-content;">0.0555555555556</div>
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.