

Assignment 2 SQL

Student name: Luo YiFeng

Student ID: 20448589

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

Movie (mID, title, year, director)

There is a movie with ID number *mID*, a *title*, a release *year*, and a *director*.

Reviewer (rID, name)

The reviewer with ID number *rID* has a certain *name*.

Rating (rID, mID, stars, ratingDate) [rID, mID are foreign keys. rID refers to Reviewer.rID, mID refers to Movie.mID]

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

In order to check whether my answer is correct, I created a database with python. There are three tables in this database, and some basic data are inserted. The three tables and the inserted data are as below:

The code for how to create the database and insert the data has been uploaded to my github account:

mID	title	year	director
1	Ready Player One	2018	Steven Spielberg
2	The Post	2017	Steven Spielberg
3	The Lost World: Jurassic Park	1997	Steven Spielberg
4	Tenet	2020	Christopher Nolan
5	Dunkirk	2017	Christopher Nolan
6	Inception	2010	Christopher Nolan
7	Interstellar	2014	Christopher Nolan
8	Mulan	2020	Niki Caro
9	Cats	2019	Tom Hooper
10	Jack and Jill	2011	Adam Sandler
11	The Room	2003	Tommy Wiseau
12	dummy1	2000	dummy
13	dummy2	2000	dummy
14	Gone with the Wind	1952	Victor Fleming

Movies table

rID	mID	stars	ratingDate
4	1	5	2019
1	2	3	2018
3	3	4	2000
1	4	4	2020
2	5	4	2019
5	6	5	2018
1	7	5	2011
9	8	3	2020
9	9	1	2020
9	10	2	2012
7	11	1	NULL
6	12	NULL	NULL
6	14	NULL	NULL
7	14	2	2020
8	14	5	1990
9	14	5	2009
12	1	5	2020
12	3	5	2020
12	14	5	2020

Rating table

rID	name
1	Michael
2	Dwight
3	Pam
4	Jim
5	Andy
6	Jason
7	Andrewson
8	Andrew
9	Chris Jackson
10	Nobody
11	NO THIS PERSON
12	Sarah Martinez

Reviewer table

Write SQL statements to answer the following questions:

①(5') Find the titles of all movies directed by Steven Spielberg.

```
SELECT title
FROM Movies
WHERE director = 'Steven Spielberg'
```

1	SELECT title
2	FROM Movies
3	WHERE director = 'Steven Spielberg'

	title
1	Ready Player One
2	The Post
3	The Lost World: Jurassic Park

No.1

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

```
SELECT year
FROM Movies, Rating
WHERE Movies.mID = Rating.mID AND (stars = 4 or stars = 5)
ORDER BY stars
```

1	SELECT year
2	FROM Movies, Rating
3	WHERE Movies.mID = Rating.mID AND (stars = 4 or stars = 5)
4	ORDER BY stars

	year
1	1997
2	2020
3	2017
4	2018
5	2010
6	2014
7	1952
8	1952

③(6') Find the titles of all movies that have no ratings.

```
SELECT title
FROM Movies
  a left join Rating b
on a.mID = b.mID
WHERE stars is null
```

1	SELECT title
2	FROM Movies
3	a left join Rating b
4	on a.mID = b.mID
5	WHERE stars is null

	title
1	dummy1
2	dummy2

No.4

④(6') 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.

```
SELECT name
FROM Reviewer INNER JOIN Rating ON Reviewer.rID = Rating.rID
WHERE ratingDate IS NULL
```

1	SELECT name
2	FROM Reviewer INNER JOIN Rating ON Reviewer.rID = Rating.rID
3	WHERE ratingDate IS NULL

	name
1	Andrewson
2	Jason
3	Andrew

No.5

⑤(7') Find the title(s) of the movie(s) with the highest stars.

```

SELECT DISTINCT M.title
FROM Movies AS M, Rating AS R
WHERE M.mID = R.mID AND R.stars = (
    SELECT max(stars)
    FROM Rating)

```

1	SELECT DISTINCT M.title
2	FROM Movies AS M, Rating AS R
3	WHERE M.mID = R.mID AND R.stars = (
4	SELECT max(stars)
5	FROM Rating)

	title
1	Ready Player One
2	Inception
3	Interstellar
4	Gone with the Wind

NO.5

⑥(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. (hint: use GROUP BY)

```

SELECT title, MAX(stars)
FROM Movies AS M INNER JOIN Rating AS R ON M.mID = R.mID
GROUP BY title

```

1	SELECT title, MAX(stars)
2	FROM Movies AS M INNER JOIN Rating AS R ON M.mID = R.mID
3	GROUP BY title
4	

	title	MAX(stars)
1	Cats	1
2	Dunkirk	4
3	Gone with the Wind	5
4	Inception	5
5	Interstellar	5
6	Jack and Jill	2
7	Mulan	3
8	Ready Player One	5
9	Tenet	4
10	The Lost World: Jurassic Park	4
11	The Post	3
12	The Room	1
13	dummy1	NULL

no.6

- ⑦(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.

```
SELECT title, AVG(stars) AS avger
FROM Movies AS M INNER JOIN Rating AS R ON M.mID = R.mID
GROUP BY title
ORDER BY avger DESC, title
```

1	SELECT title, AVG(stars) AS avger
2	FROM Movies AS M INNER JOIN Rating AS R ON M.mID = R.mID
3	GROUP BY title
4	ORDER BY avger DESC, title

	title	avger
1	Inception	5.0
2	Interstellar	5.0
3	Ready Player One	5.0
4	Dunkirk	4.0
5	Gone with the Wind	4.0
6	Tenet	4.0
7	The Lost World: Jurassic Park	4.0
8	Mulan	3.0
9	The Post	3.0
10	Jack and Jill	2.0
11	Cats	1.0
12	The Room	1.0
13	dummy1	NULL

No.7

- ⑧(7') Find the names of all reviewers who rated Gone with the Wind.

```
SELECT DISTINCT name
FROM Movies
INNER JOIN Rating USING(mID)
INNER JOIN Reviewer USING(rID)
WHERE title = 'Gone with the Wind'
```

1	SELECT DISTINCT name
2	FROM Movies
3	INNER JOIN Rating USING(mID)
4	INNER JOIN Reviewer USING(rID)
5	WHERE title = 'Gone with the Wind'

	name
1	Jason
2	Andrewson
3	Andrew
4	Chris Jackson

No.8

- ⑨(7') 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.

```
SELECT name, title, stars
FROM Movies
INNER JOIN Rating USING(mID)
INNER JOIN Reviewer USING(rID)
WHERE director = name
```

```
1 SELECT name, title, stars
2 FROM Movies
3 INNER JOIN Rating USING(mID)
4 INNER JOIN Reviewer USING(rID)
5 WHERE director = name
```

No.9

- ⑩(7') Return all reviewer names and movie names together in a single column, 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".)

```
SELECT title FROM Movies
UNION
SELECT name FROM Reviewer
ORDER BY name, title
```

1	SELECT title FROM Movies
2	UNION
3	SELECT name FROM Reviewer
4	ORDER BY name, title

	title
1	Andrew
2	Andrewson
3	Andy
4	Cats
5	Chris Jackson
6	Dunkirk
7	Dwight
8	Gone with the Wind
9	Inception
10	Interstellar
11	Jack and Jill
12	Jason
13	Jim
14	Michael
15	Mulan
16	NO THIS PERSON
17	Nobody
18	Pam
19	Ready Player One
20	Tenet
21	The Lost World: Jurassic Park
22	The Post

No.10

⑪(7') Find the titles of all movies not reviewed by Chris Jackson.

```
SELECT title
FROM Movies
WHERE mID NOT IN (
  SELECT mID
  FROM Rating
  INNER JOIN Reviewer USING(rID)
  WHERE name = 'Chris Jackson'
)
```

1	SELECT title
2	FROM Movies
3	WHERE mID NOT IN (
4	SELECT mID
5	FROM Rating
6	INNER JOIN Reviewer USING(rID)
7	WHERE name = 'Chris Jackson'
8)

	title
1	Ready Player One
2	The Post
3	The Lost World: Jurassic Park
4	Tenet
5	Dunkirk
6	Inception
7	Interstellar
8	The Room
9	dummy1
10	dummy2

No.11

⑫(7') Find the movies reviewed by Sarah Martinez and not reviewed by Chris Jackson.

```
SELECT DISTINCT M.title
FROM Movies AS M
WHERE M.mID IN (
  SELECT RA.mID
  FROM Rating AS RA, Reviewer AS RE
  WHERE RE.rID = RA.rID AND re.name = 'Sarah Martinez') AND M.title
NOT IN (
  SELECT DISTINCT M.title
  FROM Movies AS M
  WHERE M.mID IN (
    SELECT RA.mID
    FROM Rating AS RA, Reviewer AS RE
    WHERE RE.rID = RA.rID AND re.name = 'Chris Jackson'))
```

1	SELECT DISTINCT M.title
2	FROM Movies AS M
3	WHERE M.mID IN (
4	SELECT RA.mID
5	FROM Rating AS RA, Reviewer AS RE
6	WHERE RE.rID = RA.rID AND re.name = 'Sarah Martinez') AND M.title NOT IN (
7	SELECT DISTINCT M.title
8	FROM Movies AS M
9	WHERE M.mID IN (
10	SELECT RA.mID
11	FROM Rating AS RA, Reviewer AS RE
12	WHERE RE.rID = RA.rID AND re.name = 'Chris Jackson'))
13	

	title
1	Ready Player One
2	The Lost World: Jurassic Park

No.12

⑬(7') Find the movies that have been reviewed by at least two reviewers. Please write an SQL with only one SELECT and without GROUP BY.

```
SELECT DISTINCT T1.title
FROM (Movies Join Rating USING (mID)) AS T1, (Movies Join Rating USING (mID)) AS
T2
WHERE T1.mID = T2.mID AND T1.rID <> T2.rID
```

1	SELECT DISTINCT T1.title
2	FROM (Movies Join Rating USING (mID)) AS T1, (Movies Join Rating USING (mID)) AS T2
3	WHERE T1.mID = T2.mID AND T1.rID <> T2.rID
4	
5	

	title
1	Ready Player One
2	The Lost World: Jurassic Park
3	Gone with the Wind

No.13

⑭(7') Find the movies that have been reviewed by at least two reviewers. Please write an SQL with subquery.

```
SELECT title
FROM Movies
WHERE mID IN (
    SELECT mID
    FROM Rating R1
    INNER JOIN Rating R2 USING (mID)
    GROUP BY R1.mID
    HAVING COUNT(*) >1)
```


1	SELECT title
2	FROM Movies
3	WHERE mID IN (SELECT mID
4	FROM Rating R1
5	INNER JOIN Rating R2 USING (mID)
6	GROUP BY R1.mID
7	HAVING COUNT(*) > 1)
8	

	title
1	Ready Player One
2	The Lost World: Jurassic Park
3	Gone with the Wind

No.14

⑮(7') Find the reviewers who do not review any movie.

ELECT rID, name

FROM Reviewer

WHERE Reviewer.rID NOT IN (SELECT DISTINCT Reviewer.rID

FROM Reviewer, Rating

WHERE Reviewer.rID = Rating.rID

)

1	SELECT rID, name
2	FROM Reviewer
3	WHERE Reviewer.rID NOT IN (SELECT DISTINCT Reviewer.rID
4	FROM Reviewer, Rating
5	WHERE Reviewer.rID = Rating.rID
6)
7	

	rID	name
1	10	Nobody
2	11	NO THIS PERSON

No.15