<https://lagunita.stanford.edu/courses/DB/SQL/SelfPaced/courseware/ch-sql/seq-exercise-sql_movie_query_core/>

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*.

Q1

Find the titles of all movies directed by Steven Spielberg.

SELECT title

FROM Movie

WHERE director ="Steven Spielberg";

Q2

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

SELECT DISTINCT year

FROM Movie

JOIN Rating

ON Movie.mID = Rating.mID

WHERE stars >=4

ORDER BY year;

Q3

Find the titles of all movies that have no ratings.

SELECT title

FROM Movie

WHERE mID NOT IN

(SELECT mID

FROM Rating);

Q4

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

JOIN Rating ON Reviewer.rID = Rating.rID

WHERE ratingDate IS NULL;

Q5

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.

SELECT re.name, m.title, r.stars, r.ratingDate

FROM Rating AS r

LEFT JOIN Movie AS m

ON r.mID = m.mID

LEFT JOIN Reviewer AS re

ON r.rID = re.rID

ORDER BY re.name, m.title, r.stars;

Q6

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.

select Reviewer.name, Movie.title

from Reviewer, Movie, (

select R1.rID, R1.mID from Rating R1, Rating R2

where R1.rID=R2.rID AND R1.mID=R2.mID AND R2.ratingDate>R1.ratingDate AND R2.stars>R1.stars) as T

where Reviewer.rID=T.rID and Movie.mID=T.mID;

Q7

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.

select Movie.title, MAX(stars) from Rating

JOIN Movie ON Rating.mID = Movie.mID

GROUP BY title;

Q8

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.

SELECT Movie.title, MAX(stars) - MIN(stars) as ratingSpread

FROM rating JOIN movie ON rating.mID = movie.mID

GROUP BY title

ORDER BY ratingSpread DESC;

Q9

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.)

select max(a1)-min(a1) from

(select avg(av1) a1 from

(select avg(stars) av1 from rating r join movie m on r.mid=m.mid where m.year < 1980

group by r.mid)

union

select avg(av2) a1 from

(select avg(stars) av2 from rating r join movie m on r.mid=m.mid where m.year > 1980

group by r.mid))

<https://lagunita.stanford.edu/courses/DB/SQL/SelfPaced/courseware/ch-sql/seq-exercise-sql_movie_query_extra/>

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*.

Q1

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

SELECT DISTINCT re.name

FROM Rating AS r

LEFT JOIN Movie AS m

ON r.mID = m.mID

LEFT JOIN Reviewer AS re

ON r.rID = re.rID

WHERE m.title ="Gone with the Wind";

Q2

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 DISTINCT re.name, m.title, r.stars

FROM Rating AS r

LEFT JOIN Movie AS m

ON r.mID = m.mID

LEFT JOIN Reviewer AS re

ON r.rID = re.rID

WHERE m.director = re.name;