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

select distinct name from reviewer join rating using(rID) join movie using(mID)

where title = 'Gone with the Wind'

**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 name, title, stars from reviewer join rating using(rID) join movie using(mID)

where name = director

**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".)**

select n from (

select name n from reviewer

union

select title n from movie

)

order by n

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

select title from movie where mID not in

(select mID from movie join rating using(mID) join reviewer using(rID)

where name = 'Chris Jackson')

**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.**

select distinct r1.name, r2.name from

(select rID, name, mID from reviewer r1 join rating t1 using(rID)) r1 join

(select rID, name, mID from reviewer r2 join rating t2 using(rID)) r2 using(mID)

where r1.name < r2.name

order by r1.name, r2.name

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

select name, title, stars from movie join rating using(mID) join reviewer using(rID)

where stars = (select min(stars) from rating)

**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) from movie join rating

using(mID) group by title order by avg(stars) desc, title

**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.)**

select name from reviewer where rID in

(select rID from rating group by rID having(count(\*)) >= 3)

**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.)**

select title, director from movie where director in

(select director from movie group by director having count(\*) > 1)

order by director, title

**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.)**

select title, avg(stars) from movie join rating using(mID)

group by title

having avg(stars) =

(select max(a) from

(select avg(stars) a from movie join rating using(mID) group by mID))

select title, avg(stars) from movie join rating using(mID) group by title

having avg(stars) >= all (select avg(stars) from rating group by mID)

**Find the movie(s) with the lowest average rating. Return the movie title(s) and average rating.**

select title, avg(stars) from movie join rating using(mID)

group by title

having avg(stars) =

(select min(a) from

(select avg(stars) a from movie join rating using(mID) group by mID))

select title, avg(stars) from movie join rating using(mID) group by title

having avg(stars) <= all (select avg(stars) from rating group by mID)

**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.**

select distinct director, title, stars from movie m join rating using(mID)

where stars >= (select max(stars) from movie join rating using(mID) where director = m.director)

**Add the reviewer Roger Ebert to your database, with an rID of 209.**

insert into reviewer(rID, name) values(209, 'Roger Ebert')

**Insert 5-star ratings by James Cameron for all movies in the database. Leave the review date as NULL.**

insert into rating(rID, mID, stars)

select (select rID from reviewer where name = 'James Cameron'), mID, 5 from movie

**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.)**

update movie set year = year + 25

where mID in (select mID from movie join rating using(mID) group by mID having avg(stars) >= 4)

**Remove all ratings where the movie's year is before 1970 or after 2000, and the rating is fewer than 4 stars.**

delete from rating where mID in (select mID from movie where year < 1970 or year > 2000)

and stars < 4