

Lab3

Question 1

- List names of reviewers who reviewed all movies reviewed by 'Elizabeth Thomas'.
- Let us do a similar question
 - Q1_helper1 - List names of reviewers who reviewed all movies
 - Let's first write in RA, then SQL

In RA: List names of reviewers who reviewed all movies

- Find all movies
 - $T1 = \pi_{mID}(Movie)$
- Find every reviewer (id) who has reviewed all movies
 - $T2 = \pi_{mID, rID}(Rating) \div T1$
- Find and display names of all reviewers in T2
 - $Answer = \pi_{name}(T2 \bowtie Reviewer)$

In SQL – method 1 – using NOT EXISTS

- In method1, we will replace a “for all” (such as for all movies) by 2 negations (NOT EXISTS).
- So try and rewrite or rethink the question using 2 negations – for example, we can rephrase

List names of reviewers who reviewed all movies

using 2 negations as

List names of reviewers such that there does not exist a movie (in the database) that the reviewer does not review.

In SQL – method 1 – using NOT EXISTS

List names of reviewers who reviewed all movies OR

List names of reviewers such that there does not exist a movie (in the database) that the reviewer does not review.

SELECT name
FROM reviewer

WHERE NOT EXISTS (SELECT *

FROM movie

WHERE NOT EXISTS (SELECT *

FROM rating

WHERE reviewer.rID = rating.rID

AND movie.mID = rating.mID));

Method 2

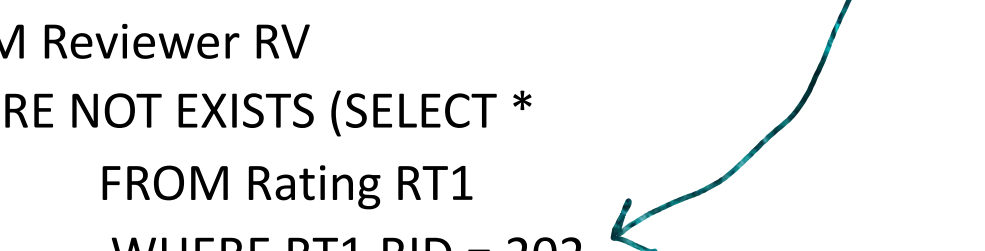
```
select name  
from reviewer  
where rid IN (select rid  
              from Rating  
              group by rid  
              having count(rid) = (select count(*) from movie));
```

Q1 – helper2

List names of reviewers who review all movies that are reviewed by reviewer id 202.

Rephrase using 2 negations:

- List names of reviewers such that there does not exist a movie reviewed by 202 that the reviewer has not reviewed.



```
SELECT NAME
FROM Reviewer RV
WHERE NOT EXISTS (SELECT *
                  FROM Rating RT1
                  WHERE RT1.RID = 202
                  AND NOT EXISTS (SELECT *
                                FROM Rating RT2
                                WHERE RT1.MID = RT2.MID
                                AND RV.RID = RT2.RID));
```

The diagram illustrates the SQL query structure with arrows indicating table references. A blue arrow points from the 'SELECT NAME' line to the 'Reviewer RV' table. Another blue arrow points from the 'RT1.RID = 202' condition to the 'Rating RT1' table. A third blue arrow points from the 'RT1.MID = RT2.MID' condition to the 'Rating RT2' table. A fourth blue arrow points from the 'RV.RID = RT2.RID' condition to the 'Reviewer RV' table.

Method 2 List names of reviewers who review all movies that are reviewed by reviewer id 202.

```
SELECT name
FROM Reviewer
WHERE rid IN (SELECT rid
               FROM Rating
               WHERE mid IN (SELECT mid FROM rating WHERE rid = 202)
               GROUP BY rid
               HAVING COUNT(mid) >=
                   (SELECT COUNT(mid) FROM rating WHERE rid=202));
```

Handwritten annotations: A blue underline is under 'Rating' and 'mid'. To the right of the inner query, '106' and '107' are written in blue, with a blue bracket grouping them. Above these numbers are two blue checkmarks.

Q2 – Q5

- Aggregate functions