**Movielens**

You will need to use the movielens database to solve these problems.

The steps to import databases are at the end of the [databases Windows installation module](https://qa-community.co.uk/~/_/learning/databases/databases--introduction).

**Tasks**

1. List the titles and release dates of movies released between 1983-1993 in reverse chronological order.

SELECT title, release\_date

FROM movies

WHERE release\_date BETWEEN '1983-01-01' AND '1993-12-31'

ORDER BY release\_date DESC;

1. Without using LIMIT, list the titles of the movies with the lowest average rating.

SELECT min\_avg\_ratings.title

FROM (

SELECT DISTINCT m.id, m.title, AVG(r.rating) AS avg\_rating

FROM movies m

JOIN ratings r ON m.id=r.movie\_id

GROUP BY m.id

ORDER BY avg\_rating

) AS min\_avg\_ratings

WHERE avg\_rating = 1;

1. List the unique records for Sci-Fi movies where male 24-year-old students have given 5-star ratings.

SELECT DISTINCT m.title

FROM users u

JOIN occupations o ON u.occupation\_id=o.id

JOIN ratings r ON u.id=r.user\_id

JOIN movies m ON r.movie\_id=m.id

JOIN genres\_movies gm ON gm.movie\_id=m.id

JOIN genres g ON g.id=gm.genre\_id

WHERE gender='M'

AND g.name='Sci-Fi'

AND o.name='Student'

AND rating=5

AND age=24;

1. List the unique titles of each of the movies released on the most popular release day.

SELECT DISTINCT m.title

FROM movies m

WHERE m.release\_date=(

SELECT pop.release\_date

FROM (

SELECT m.release\_date, COUNT(m.release\_date) AS release\_count

FROM movies m

GROUP BY m.release\_date

ORDER BY release\_count DESC

LIMIT 1

) AS pop

);

1. Find the total number of movies in each genre; list the results in ascending numeric order.

SELECT g.name, COUNT(m.title) AS number

FROM genres g

JOIN genres\_movies gm ON (g.id=gm.genre\_id)

JOIN movies m ON (m.id=gm.movie\_id)

GROUP BY g.name

ORDER BY number;