

# correlated\_nested\_queries

*-- Count movie rentals of customer 45*

```
SELECT COUNT(*)  
FROM renting AS r  
WHERE r.customer_id = 45;
```

## Explanation:

- This SQL query counts the number of times customer with ID 45 appears in the renting table. It uses the COUNT(\*) aggregate function to count all rows meeting the condition specified in the WHERE clause (that the customer\_id is 45). The result will be a single number representing the total rental count for that customer.

*-- Select customers with less than 5 movie rentals*

```
SELECT *  
FROM customers as c  
WHERE 5 >  
    (SELECT count(*)  
     FROM renting as r  
     WHERE r.customer_id = c.customer_id);
```

## Explanation:

- This SQL query selects all information from the customers table for customers who have rented fewer than 5 movies. It does this by using a subquery to count the number of rentals for each customer and then filtering the main query based on that count. The subquery counts rentals from the renting table where the customer\_id matches the customer\_id in the main query's customers table.

*-- Calculate the minimum rating of customer with ID 7*

```
SELECT MIN(rating)  
FROM renting  
WHERE customer_id = 7;
```

## Explanation:

- This SQL query finds the lowest rating given by customer with ID 7 in the renting table. It uses the MIN() aggregate function to find the minimum value in the rating column, filtering the results to only include rows where customer\_id is 7.

```
SELECT *  
FROM customers AS c  
WHERE 4 > -- Select all customers with a minimum rating smaller than 4
```

```
(SELECT MIN(rating)
FROM renting AS r
WHERE r.customer_id = c.customer_id);
```

#### Explanation:

- This SQL query selects all information from the customers table where the minimum rating a customer received (from the renting table) is less than 4. It uses a subquery to find the minimum rating for each customer and then filters the main query based on that result.

```
SELECT *
FROM movies AS m
WHERE 5 < -- Select all movies with more than 5 ratings
(SELECT COUNT(rating)
FROM renting AS r
WHERE r.movie_id = m.movie_id);
```

#### Explanation:

- This SQL query retrieves all information from the movies table for movies that have received more than 5 ratings. It does this by using a subquery to count the number of ratings for each movie in the renting table and then filtering the movies table based on this count. The subquery counts ratings associated with each movie ID, and the outer query only selects movies where this count is greater than 5.

```
SELECT *
FROM movies AS m
WHERE 8 < -- Select all movies with an average rating higher than 8
(SELECT AVG(rating)
FROM renting AS r
WHERE r.movie_id = m.movie_id);
```

#### Explanation:

- This SQL query retrieves all information from the movies table for movies that have an average rating greater than 8. It achieves this by using a subquery to calculate the average rating for each movie from the renting table and then filtering the movies table based on this calculated average. The WHERE clause compares 8 with the average rating for each movie.