

queries_with_group_settings

SELECT

nationality, -- *Select nationality of the actors*

gender, -- *Select gender of the actors*

COUNT(*) as actor_count -- *Count the number of actors and give it an alias*

FROM actors

GROUP BY GROUPING SETS ((nationality), (gender), ()); -- *Use the correct GROUPING SETS operation*

Explanation:

- This SQL query uses GROUPING SETS to generate a summary of actors grouped by nationality, gender, and overall total count. GROUPING SETS allows you to get multiple aggregations in a single query, avoiding the need for multiple separate queries. The result will show three sets of counts: one for each nationality, one for each gender, and a final row representing the total count of all actors.

SELECT

country, -- *Select country, gender and rating*

gender,

rating

FROM renting **AS** r

LEFT JOIN customers **AS** c -- *Use the correct join*

ON r.customer_id = c.customer_id;

Explanation:

- This SQL query retrieves information about rentals, joining data from two tables: renting and customers. A LEFT JOIN ensures that all rows from the renting table are included in the result, even if there's no matching entry in the customers table (in which case, country and gender would be NULL). The join condition r.customer_id = c.customer_id links rows based on matching customer IDs. The query selects the country, gender, and rating columns from the joined result.

SELECT

c.country,

c.gender,

AVG(rating) **AS** average_rating -- *Calculate average rating and name it 'average_rating' for clarity*

FROM renting **AS** r

LEFT JOIN customers **AS** c

ON r.customer_id = c.customer_id

GROUP BY c.country, c.gender -- *Group the results by country and gender*

ORDER BY c.country, c.gender;

Explanation:

- This SQL query calculates the average rating for each country and gender combination from a database with two tables: renting and customers. It uses a LEFT JOIN to combine data from both tables based on customer_id, calculates the average rating using AVG(), groups the results using GROUP BY, and orders them using ORDER BY. The added AS average_rating provides a more descriptive column name for the average rating.

```
SELECT
  c.country,
  c.gender,
  AVG(r.rating)
FROM renting AS r
LEFT JOIN customers AS c
ON r.customer_id = c.customer_id
GROUP BY GROUPING SETS ((country, gender)); -- Group by country and gender with GROUPING SETS
```

Explanation:

- This SQL query calculates the average rating for each country and gender combination from two tables: renting and customers. It uses a LEFT JOIN to combine the data, ensuring all customers are included even if they haven't made any rentals. The GROUPING SETS clause allows for aggregation at different levels: specifically, it provides the average rating for each unique (country, gender) pair and also the overall average rating (if a row with NULL values for country and gender appears).

```
SELECT
  c.country,
  c.gender,
  AVG(r.rating)
FROM renting AS r
LEFT JOIN customers AS c
ON r.customer_id = c.customer_id
-- Report all info from a Pivot table for country and gender
GROUP BY GROUPING SETS ((country, gender), (country), (gender), ());
```

Explanation:

- This SQL query calculates the average rating for each combination of country and gender, and also provides the average rating for each country, each gender, and overall. It uses GROUPING SETS to achieve this multi-level aggregation in a single query, avoiding the need for multiple separate queries. The LEFT JOIN ensures that all customers are included, even if they haven't made any rentals (their average rating will be NULL in that case).