**Famous Paintings: SQL Query Analysis**

**1. Paintings Not in Any Museum**

*This query finds all paintings that do not have associated museums.*

-- Select all columns from the 'work' table

-- where the museum\_id is NULL, indicating no museum assignment.

SELECT \*

FROM work

WHERE museum\_id IS NULL;

**2. Museums Without Any Paintings**

*This query identifies museums that have no paintings listed on the 'work' table.*

-- Select all columns from the 'museum' table

-- for which no corresponding record exists on the 'work' table.

-- The NOT EXISTS clause is an efficient way to check for non-matching records.

SELECT \*

FROM museum m

WHERE NOT EXISTS (

SELECT 1

FROM work w

WHERE w.museum\_id = m.museum\_id

);

**3. Paintings with a Sale Price Higher Than Regular Price**

*This query finds products where the asking price is greater than the regular price.*

-- Select all records from 'product\_size'

-- where the sale\_price is higher than the regular\_price.

SELECT \*

FROM product\_size

WHERE sale\_price > regular\_price;

**4. Paintings with a Sale Price Less Than 50% of Regular Price**

*This query identifies paintings being sold for less than half of their regular price.*

-- Select all records from 'product\_size'

-- where the sale\_price is less than 50% of the regular\_price.

SELECT \*

FROM product\_size

WHERE sale\_price < (regular\_price \* 0.5);

**5. The Most Expensive Canvas Size**

*This query finds the label of the canvas size that has the highest sale price.*

-- In a subquery, rank all products by sale\_price in descending order.

-- Then, join with 'canvas\_size' to get the label for the top-ranked product (rnk=1).

-- Note: The cast cs.size\_id:text is specific to PostgreSQL. For MySQL, use CAST(cs.size\_id AS CHAR).

SELECT

cs.label AS canva,

ps.sale\_price

FROM (

SELECT \*,

RANK() OVER (ORDER BY sale\_price DESC) AS rnk

FROM product\_size

) ps

JOIN canvas\_size cs ON CAST(cs.size\_id AS CHAR) = ps.size\_id

WHERE ps.rnk = 1;

**6. Delete Duplicate Records**

*These queries remove duplicate rows from four different tables based on their primary keys.*

-- Note: 'ctid' is a system column specific to PostgreSQL.

-- For MySQL, you would typically use a self-join with the primary key.

-- Example for MySQL 'work' table:

-- DELETE t1 FROM work t1 JOIN work t2

-- WHERE t1.work\_id = t2.work\_id AND t1.id > t2.id;

-- Delete from 'work' where the ctid is not the minimum ctid for that work\_id.

DELETE FROM work

WHERE ctid NOT IN (

SELECT MIN(ctid)

FROM work

GROUP BY work\_id

);

-- Delete duplicates from 'product\_size' based on the composite key.

DELETE FROM product\_size

WHERE ctid NOT IN (

SELECT MIN(ctid)

FROM product\_size

GROUP BY work\_id, size\_id

);

-- Delete duplicates from 'subject' based on the composite key.

DELETE FROM subject

WHERE ctid NOT IN (

SELECT MIN(ctid)

FROM subject

GROUP BY work\_id, subject

);

-- Delete duplicates from 'image\_link'.

DELETE FROM image\_link

WHERE ctid NOT IN (

SELECT MIN(ctid)

FROM image\_link

GROUP BY work\_id

);

**7. Museums with Invalid City Information**

*This query finds museums where the city name starts with a number.*

-- Select all museums where the 'city' field matches a regular expression

-- indicating that it starts with a digit (0-9).

-- Note: '~' is PostgreSQL syntax. In MySQL, use REGEXP: WHERE city REGEXP '^[0-9]'.

SELECT \*

FROM museum

WHERE city ~ '^[0-9]';

**8. Remove Invalid Entry from Museum Hours**

*This query removes duplicate entries from the museum\_hours table based on museum and day.*

-- Deletes rows from 'museum\_hours' keeping only one entry per museum\_id and day.

-- This is useful for cleaning up data where a museum might have multiple

-- open/close times listed for the same day.

DELETE FROM museum\_hours

WHERE ctid NOT IN (

SELECT MIN(ctid)

FROM museum\_hours

GROUP BY museum\_id, day

);

**9. Top 10 Most Famous Painting Subjects**

*This query identifies the 10 subjects that appear most frequently in the dataset.*

-- First, count the number of paintings for each subject and rank them.

-- Then, select the top 10 from the ranked results.

SELECT \*

FROM (

SELECT

s.subject,

COUNT(1) AS no\_of\_paintings,

RANK() OVER (ORDER BY COUNT(1) DESC) AS ranking

FROM work w

JOIN subject s ON s.work\_id = w.work\_id

GROUP BY s.subject

) x

WHERE ranking <= 10;

**10. Museums Open on Both Sunday and Monday**

*This query lists all museums that are open on both Sunday and Monday.*

-- Fetches museums that have an entry for 'Sunday' and also have a

-- corresponding entry for 'Monday' for the same museum\_id.

SELECT DISTINCT

m.name AS museum\_name,

m.city,

m.state,

m.country

FROM museum\_hours mh

JOIN museum m ON m.museum\_id = mh.museum\_id

WHERE mh.day = 'Sunday'

AND EXISTS (

SELECT 1

FROM museum\_hours mh2

WHERE mh2.museum\_id = mh.museum\_id

AND mh2.day = 'Monday'

);

**11. Museums Open Every Day**

*This query counts how many museums are open 7 days a week.*

-- Groups the 'museum\_hours' table by museum\_id and counts the number of days.

-- The outer query then counts how many of these groups have a count of 7.

SELECT COUNT(1)

FROM (

SELECT museum\_id, COUNT(1)

FROM museum\_hours

GROUP BY museum\_id

HAVING COUNT(1) = 7

) x;

**12. Top 5 Most Popular Museums**

*This query finds the 5 museums with the highest number of paintings.*

-- Ranks museums by their painting count in a subquery.

-- The outer query then selects the top 5 and joins to get museum details.

SELECT

m.name AS museum,

m.city,

m.country,

x.no\_of\_painintgs

FROM (

SELECT

m.museum\_id,

COUNT(1) AS no\_of\_painintgs,

RANK() OVER (ORDER BY COUNT(1) DESC) AS rnk

FROM work w

JOIN museum m ON m.museum\_id = w.museum\_id

GROUP BY m.museum\_id

) x

JOIN museum m ON m.museum\_id = x.museum\_id

WHERE x.rnk <= 5;

**13. Top 5 Most Popular Artists**

*This query finds the 5 artists who have created the most paintings in the dataset.*

-- Ranks artists by their painting count in a subquery.

-- The outer query then selects the top 5 and joins to get artist details.

SELECT

a.full\_name AS artist,

a.nationality,

x.no\_of\_painintgs

FROM (

SELECT

a.artist\_id,

COUNT(1) AS no\_of\_painintgs,

RANK() OVER (ORDER BY COUNT(1) DESC) AS rnk

FROM work w

JOIN artist a ON a.artist\_id = w.artist\_id

GROUP BY a.artist\_id

) x

JOIN artist a ON a.artist\_id = x.artist\_id

WHERE x.rnk <= 5;

**14. Three Least Popular Canvas Sizes**

*This query identifies the three canvas sizes used for the fewest number of paintings.*

-- Ranks canvas sizes by their usage count in ascending order.

-- Then selects the bottom 3 from the ranked results.

SELECT

label,

ranking,

no\_of\_paintings

FROM (

SELECT

cs.size\_id,

cs.label,

COUNT(1) AS no\_of\_paintings,

DENSE\_RANK() OVER (ORDER BY COUNT(1)) AS ranking

FROM work w

JOIN product\_size ps ON ps.work\_id = w.work\_id

JOIN canvas\_size cs ON CAST(cs.size\_id AS TEXT) = ps.size\_id

GROUP BY cs.size\_id, cs.label

) x

WHERE x.ranking <= 3;

**15. Museum with the Longest Open Hours in a Single Day**

*This query finds which museum stays open the longest on any given day.*

-- Calculates the duration between open and close times for each museum's daily hours.

-- It then ranks these durations to find the longest one.

-- Note: 'to\_timestamp' is PostgreSQL syntax. MySQL would use STR\_TO\_DATE and TIMEDIFF.

SELECT

museum\_name,

state AS city,

day,

open,

close,

duration

FROM (

SELECT

m.name AS museum\_name,

m.state,

day,

open,

close,

(to\_timestamp(close, 'HH:MI PM') - to\_timestamp(open, 'HH:MI AM')) AS duration,

RANK() OVER (ORDER BY (to\_timestamp(close, 'HH:MI PM') - to\_timestamp(open, 'HH:MI AM')) DESC) AS rnk

FROM museum\_hours mh

JOIN museum m ON m.museum\_id = mh.museum\_id

) x

WHERE x.rnk = 1;

**16. Museum with the Most Paintings in the Most Popular Style**

*This query first finds the most popular painting style and then finds which museum has the most paintings of that style.*

-- A Common Table Expression (CTE) 'pop\_style' first identifies the most popular painting style.

-- A second CTE 'cte' then counts paintings of that style for each museum and ranks them.

-- The final query selects the top-ranked museum from the second CTE.

WITH pop\_style AS (

SELECT style,

RANK() OVER (ORDER BY COUNT(1) DESC) AS rnk

FROM work

GROUP BY style

),

cte AS (

SELECT

w.museum\_id,

m.name AS museum\_name,

ps.style,

COUNT(1) AS no\_of\_paintings,

RANK() OVER (ORDER BY COUNT(1) DESC) AS rnk

FROM work w

JOIN museum m ON m.museum\_id = w.museum\_id

JOIN pop\_style ps ON ps.style = w.style

WHERE w.museum\_id IS NOT NULL

AND ps.rnk = 1

GROUP BY w.museum\_id, m.name, ps.style

)

SELECT museum\_name, style, no\_of\_paintings

FROM cte

WHERE rnk = 1;

**17. Artists with Paintings in Multiple Countries**

*This query identifies artists whose work is displayed in more than one country.*

-- Creates a distinct list of artists and the countries where their work is displayed.

-- Then, it groups by artist and counts the number of countries, filtering for those with more than one.

WITH cte AS (

SELECT DISTINCT

a.full\_name AS artist,

m.country

FROM work w

JOIN artist a ON a.artist\_id = w.artist\_id

JOIN museum m ON m.museum\_id = w.museum\_id

)

SELECT

artist,

COUNT(1) AS no\_of\_countries

FROM cte

GROUP BY artist

HAVING COUNT(1) > 1

ORDER BY 2 DESC;

**18. Country and City with the Most Museums**

*This query finds the country and city with the highest number of museums, combining results with a comma if there are ties.*

-- Uses two CTEs to rank countries and cities by their museum count.

-- The final query uses STRING\_AGG to concatenate the names of any top-ranked

-- countries or cities into a single field.

WITH cte\_country AS (

SELECT country, COUNT(1),

RANK() OVER (ORDER BY COUNT(1) DESC) AS rnk

FROM museum

GROUP BY country

),

cte\_city AS (

SELECT city, COUNT(1),

RANK() OVER (ORDER BY COUNT(1) DESC) AS rnk

FROM museum

GROUP BY city

)

SELECT

STRING\_AGG(DISTINCT country.country, ', '),

STRING\_AGG(city.city, ', ')

FROM cte\_country country

CROSS JOIN cte\_city city

WHERE country.rnk = 1

AND city.rnk = 1;

**19. Most and Least Expensive Paintings**

*This query finds the most expensive and least expensive paintings in the dataset and displays their details.*

-- Ranks all products by sale\_price in both ascending and descending order.

-- Then selects the records where the rank is 1 (for both highest and lowest)

-- and joins with other tables to get the full details.

WITH cte AS (

SELECT \*,

RANK() OVER (ORDER BY sale\_price DESC) AS rnk,

RANK() OVER (ORDER BY sale\_price) AS rnk\_asc

FROM product\_size

)

SELECT

w.name AS painting,

cte.sale\_price,

a.full\_name AS artist,

m.name AS museum,

m.city,

cz.label AS canvas

FROM cte

JOIN work w ON w.work\_id = cte.work\_id

JOIN museum m ON m.museum\_id = w.museum\_id

JOIN artist a ON a.artist\_id = w.artist\_id

JOIN canvas\_size cz ON cz.size\_id = CAST(cte.size\_id AS NUMERIC)

WHERE rnk = 1 OR rnk\_asc = 1;

**20. Country with the 5th Highest Number of Paintings**

*This query identifies which country ranks fifth for the total number of paintings.*

-- Ranks countries by their total painting count.

-- Then filters the results to show only the country with a rank of 5.

WITH cte AS (

SELECT

m.country,

COUNT(1) AS no\_of\_Paintings,

RANK() OVER (ORDER BY COUNT(1) DESC) AS rnk

FROM work w

JOIN museum m ON m.museum\_id = w.museum\_id

GROUP BY m.country

)

SELECT country, no\_of\_Paintings

FROM cte

WHERE rnk = 5;

**21. Three Most and Three Least Popular Painting Styles**

*This query lists the top 3 most popular and bottom 3 least popular painting styles.*

-- Ranks styles by popularity (descending) and gets the total number of unique styles.

-- The final query selects the top 3 ranks and the bottom 3 ranks (by subtracting from the total count).

WITH cte AS (

SELECT

style,

COUNT(1) AS cnt,

RANK() OVER (ORDER BY COUNT(1) DESC) rnk,

COUNT(1) OVER () AS no\_of\_records

FROM work

WHERE style IS NOT NULL

GROUP BY style

)

SELECT

style,

CASE WHEN rnk <= 3 THEN 'Most Popular' ELSE 'Least Popular' END AS remarks

FROM cte

WHERE rnk <= 3

OR rnk > no\_of\_records - 3;

**22. Artist with the Most Portraits Outside the USA**

*This query finds the artist who has painted the most portraits displayed in countries other than the USA.*

-- Filters for works that are 'Portraits' and not in the 'USA'.

-- It then counts and ranks the artists based on this filtered set.

-- The final query selects the top-ranked artist.

SELECT full\_name AS artist\_name, nationality, no\_of\_paintings

FROM (

SELECT

a.full\_name,

a.nationality,

COUNT(1) AS no\_of\_paintings,

RANK() OVER (ORDER BY COUNT(1) DESC) AS rnk

FROM work w

JOIN artist a ON a.artist\_id = w.artist\_id

JOIN subject s ON s.work\_id = w.work\_id

JOIN museum m ON m.museum\_id = w.museum\_id

WHERE s.subject = 'Portraits'

AND m.country != 'USA'

GROUP BY a.full\_name, a.nationality

) x

WHERE rnk = 1;