Workspace Question Set #1

/* Q1. We want to understand more about the movies that families are watching. The following categories are considered family movies: Animation, Children, Classics, Comedy, Family and Music.
Create a query that lists each movie, the film category it is classified in, and the number of times it has been rented out. */

```
  Select f.title, c.name, count(r.rental_id)
FROM film_category fc
JOIN category c
ON c.category_id = fc.category_id
JOIN film f
ON f.film_id = fc.film_id
JOIN inventory i
ON i.film_id = fc.film_id
JOIN rental r
ON r.inventory_id=i.inventory_id
WHERE c.name IN ('Animation','Children','Classics','Comedy','Family','Music')
GROUP BY 1,2
ORDER BY 2,1
```

/ * Q2. Now we need to know how the length of rental duration of these family-friendly movies compares to the duration that all movies are rented for.
Can you provide a table with the movie titles and divide them into 4 levels (first_quarter, second_quarter, third_quarter, and final_quarter) based on the quartiles (25%, 50%, 75%) of the rental duration for movies across all categories? Make sure to also indicate the category that these family-friendly movies fall into. */

```
SELECT f.title, c.name, f.rental_duration, NTILE(4) OVER (ORDER BY
f.rental_duration) AS standard_quartile
FROM film_category fc
JOIN category c
ON c.category_id = fc.category_id
JOIN film f
ON f.film_id = fc.film_id
WHERE c.name IN ('Animation', 'Children', 'Classics', 'Comedy', 'Family', 'Music')
ORDER BY 3
```

2nd way:

```
select x.film_title,
          x.category_name,
          x.rental_duration,
          NTILE(4) OVER (ORDER BY x.rental_duration) standard_Quartile
from (
select a.title film_title,
          c.name   category_name ,
          sum (((TO_DATE(TO_CHAR((return_date) ,'DD-MON-YYYY'),'DD-
MON-YYYY')) -
(TO_DATE(TO_CHAR((RENTAL_date) ,'DD-MON-YYYY'),'DD-MON-
YYYY')))) rental_duration
from      film a,
          film_category b,
          category c,
          inventory d,
          rental e
 where a.film_id = b.film_id
 and     b.category_id = c.category_id
 and     a.film_id = d.film_id
 and     d.inventory_id = e.inventory_id
 group by a.title,c.name
) x
```

/* Q3 Finally, provide a table with the family-friendly film category, each of the quartiles, and the corresponding count of movies within each combination of film category for each corresponding rental duration category.

The resulting table should have three columns:

- **Category**
- **Rental length category**
- **Count**

*/

```
SELECT t1.name, t1.standard_quartile, COUNT(t1.standard_quartile)
FROM
```

```
(SELECT f.title, c.name , f.rental_duration, NTILE(4) OVER (ORDER BY
f.rental_duration) AS standard_quartile
FROM film_category fc
JOIN category c
ON c.category_id = fc.category_id
JOIN film f
ON f.film_id = fc.film_id
WHERE c.name IN ('Animation', 'Children', 'Classics', 'Comedy', 'Family',
'Music')) t1
GROUP BY 1, 2
ORDER BY 1, 2
```

2nd way:

```
select a.category_name,
        a.standard_Quartile,
        count(b.rental_count) rental_count
from
(select x.film_title,
        x.category_name,
        x.rental_duration,
        NTILE(4) OVER (ORDER BY x.rental_duration) standard_Quartile
from (
select a.title film_title,
        c.name   category_name ,
        sum (((TO_DATE(TO_CHAR((return_date) ,'DD-MON-YYYY'),'DD-
MON-YYYY')) -
(TO_DATE(TO_CHAR((RENTAL_date) ,'DD-MON-YYYY'),'DD-MON-
YYYY')))) rental_duration
from     film a,
        film_category b,
        category c,
        inventory d,
        rental e
 where a.film_id = b.film_id
 and     b.category_id = c.category_id
 and     a.film_id = d.film_id
 and     d.inventory_id = e.inventory_id
 group by a.title,c.name
) x
) a ,
(
select x.film_title,
          x.category_name,
          x.rental_count
from (
select a.title film_title,
          c.name   category_name ,
          count(e.rental_id) rental_count
from     film a,
          film_category b,
          category c,
          inventory d,
          rental e
 where a.film_id = b.film_id
 and     b.category_id = c.category_id
 and     a.film_id = d.film_id
 and     d.inventory_id = e.inventory_id
 group by a.title,c.name
) x
) b
where a.film_title = b.film_title
and     a.category_name = b.category_name
group by   a.category_name,
          a.standard_Quartile
order by a.category_name,a.standard_quartile
```

Workspace Question set#2.

/* Q.1 We want to find out how the two stores compare in their count of rental orders during every month for all the years we have data for.
Write a query that returns the store ID for the store, the year and month and the number of rental orders each store has fulfilled for that month.
Your table should include a column for each of the following: year, month, store ID and count of rental orders fulfilled during that month.*/

```
 SELECT DATE_PART('month', r1.rental_date) AS rental_month,
            DATE_PART('year', r1.rental_date) AS rental_year,
            ('Store ' || s1.store_id) AS store,
            COUNT(*)
   FROM store AS s1
            JOIN staff AS s2
              ON s1.store_id = s2.store_id

            JOIN rental r1
              ON s2.staff_id = r1.staff_id
  GROUP BY 1, 2, 3
  ORDER BY 2, 1;
```

/* Q.2 We would like to know who were our top 10 paying customers, how many payments they made on a monthly basis during 2007, and what was the amount of the monthly payments.
Can you write a query to capture the customer name, month and year of payment, and total payment amount for each month by these top 10 paying customers? */

```
SELECT DATE_TRUNC('month', p.payment_date) pay_month, c.first_name || ' ' ||
c.last_name AS full_name, COUNT(p.amount) AS pay_countpermon,
SUM(p.amount) AS pay_amount
FROM customer c
JOIN payment p
ON p.customer_id = c.customer_id
WHERE c.first_name || ' ' || c.last_name IN
(SELECT t1.full_name
FROM
(SELECT c.first_name || ' ' || c.last_name AS full_name, SUM(p.amount) as
amount_total
FROM customer c
JOIN payment p
ON p.customer_id = c.customer_id
GROUP BY 1
ORDER BY 2 DESC
LIMIT 10) t1) AND (p.payment_date BETWEEN '2007-01-01' AND '2008-01-01')
GROUP BY 2, 1
```

```
ORDER BY 2, 1, 3
```

/* Q.3 Finally, for each of these top 10 paying customers, I would like to find out the difference across their monthly payments during 2007. **Please go ahead and write a query to compare the payment amounts in each successive month.**
Repeat this for each of these 10 paying customers. Also, it will be tremendously helpful if you can identify the customer name who paid the most difference in terms of payments. */

```
WITH t1 AS (SELECT (first_name || ' ' || last_name) AS name,
                       c.customer_id,
                       p.amount,
                       p.payment_date
                 FROM customer AS c
                       JOIN payment AS p
                         ON c.customer_id = p.customer_id),

      t2 AS (SELECT t1.customer_id
                 FROM t1
                GROUP BY 1
                ORDER BY SUM(t1.amount) DESC
                LIMIT 10),


t3 AS (SELECT t1.name,
                 DATE_PART('month', t1.payment_date) AS payment_month,
                 DATE_PART('year', t1.payment_date) AS payment_year,
                 COUNT (*),
                 SUM(t1.amount),
                 SUM(t1.amount) AS total,
                 LEAD(SUM(t1.amount)) OVER(PARTITION BY t1.name
ORDER BY DATE_PART('month', t1.payment_date)) AS lead,
                 LEAD(SUM(t1.amount)) OVER(PARTITION BY t1.name
ORDER BY DATE_PART('month', t1.payment_date)) - SUM(t1.amount) AS
lead_dif
           FROM t1
                 JOIN t2
```

```
               ON t1.customer_id = t2.customer_id
        WHERE t1.payment_date BETWEEN '20070101' AND '20080101'
        GROUP BY 1, 2, 3
        ORDER BY 1, 3, 2)

SELECT t3.*,
       CASE
           WHEN t3.lead_dif = (SELECT MAX(t3.lead_dif) FROM t3
ORDER BY 1 DESC LIMIT 1) THEN 'this is the maximum difference'
           ELSE NULL
           END AS is_max
  FROM t3
 ORDER BY 1;
```