Zahin Rahman

1. 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
film.title,
category.name,
COUNT (rental.inventory_id)
FROM category
JOIN film_category
ON category.category_id=film_category.category_id
JOIN film
ON film_category.film_id=film.film_id
JOIN inventory
ON inventory.film_id=film.film_id
JOIN rental
ON rental.inventory_id=inventory.inventory_id
Group By film.title, category.name
having category.name='Animation' or category.name='Children' or category.name='Classics'
or category.name='Comedy' or category.name='Family' or category.name='Music'
Order by category.name
```

2. 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
film.title,
category.name,
film.rental_duration,
ntile(4) OVER (Order by film.rental_duration) as quartile
FROM category
JOIN film_category
ON category.category_id=film_category.category_id
JOIN film
ON film_category.film_id=film.film_id
Group By film.title, category.name, film.rental_duration
having category.name='Animation' or category.name='Children' or category.name='Classics'
or category.name='Comedy' or category.name='Family' or category.name='Music'
```

3. 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 name, quartile,
COUNT(title)
FROM
(
SELECT
film.title as title,
category.name as name,
ntile(4) OVER (Order by film.rental_duration) as quartile
FROM category
JOIN film_category
ON category.category_id=film_category.category_id
JOIN film
ON film_category.film_id=film.film_id
Group By category.name, film.title, film.rental_duration
having category.name='Animation' or category.name='Children' or category.name='Classics' or category.name='Comedy' or category.name='Family' or category.name='Music'
) as t1
GROUP BY 1, 2
Order by 1, 2
```

4. We want to find out how the two stores compare in their count of rental orders during every month for all 2005/2006 data. 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', rental.rental_date) AS Rental_month,
DATE_PART('year', rental.rental_date) AS Rental_year,
store.store_id,
COUNT(store.store_id) as count_rentals
FROM store
JOIN staff
ON store.store_id=staff.store_id
JOIN rental
ON staff.staff_id=rental.staff_id
GROUP BY 1, 2, 3
ORDER BY 2, 1
```

5. We would like to know, for all/each 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
payment_month,
payment_year,
customer_name,
SUM(amount),
count(customer_name)
from
(
SELECT
payment.payment_date as date,
CONCAT (customer.first_name, ' ', customer.last_name) as customer_name,
DATE_PART('month', payment.Payment_date) AS payment_month,
DATE_PART('year', payment.Payment_date) AS payment_year,
payment.amount as amount
FROM payment
JOIN customer
ON payment.customer_id=customer.customer_id
Group by payment.payment_date, customer.first_name, customer.last_name,
customer.customer_id, amount
Order by customer_name
) as t1
Group by payment_month,
payment_year, customer_name
Order by customer_name, payment_month, sum
```