Question 1:

Create a query that lists each movie, the film category it is classified in, and the number of times it has been rented out?

SQL Query:

```
WITH main AS (
        SELECT f.title AS film_title,
            c.name AS category_name,
            r.rental_date
        FROM film f
        JOIN film_category fc ON f.film_id = fc.film_id
        JOIN category c ON c.category_id = fc.category_id
        AND c.name IN ('Animation','Children','Classics','Comedy','Family','Music')
        JOIN inventory i ON i.film_id = f.film_id
        JOIN rental r ON i.inventory_id = r.inventory_id
       )
SELECT DISTINCT(film_title),
    category_name,
    COUNT(rental_date) OVER(PARTITION BY film_title) AS rental_count
FROM main
ORDER BY 2
```

Sample Output: (Refer Csv File for complete output of the Query)

film_title	category_name	rental_count
Alter Victory	Animation	22
Anaconda Confessions	Animation	21
Bikini Borrowers	Animation	17
Blackout Private	Animation	27
Borrowers Bedazzled	Animation	22
Canyon Stock	Animation	19
Carol Texas	Animation	18
Champion Flatliners	Animation	13
Clash Freddy	Animation	25
Club Graffiti	Animation	19
Crossroads Casualties	Animation	21
Dares Pluto	Animation	9
Desire Alien	Animation	8
Dogma Family	Animation	30
Donnie Alley	Animation	14
Doors President	Animation	19
Double Wrath	Animation	25
Duck Racer	Animation	6
Early Home	Animation	7
Falcon Volume	Animation	22
Fight Jawbreaker	Animation	22

Question 2:

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?

SQL Query:

```
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
```

Sample Output: (Refer Csv File for complete output of the Query)

title	name	rental_duration	standard_quartile
Sweethearts Suspects	Children	3	1
Go Purple	Music	3	1
Bilko Anonymous	Family	3	1
Wait Cider	Animation	3	1
Daughter Madigan	Children	3	1
Turn Star	Animation	3	1
Rush Goodfellas	Family	3	1
King Evolution	Family	3	1
Tracy Cider	Animation	3	1
Wisdom Worker	Comedy	3	1
Telegraph Voyage	Music	3	1
Miracle Virtual	Animation	3	1
Jerk Paycheck	Classics	3	1
Doors President	Animation	3	1

Question 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:

a. Category

b. Rental length category

c. Count

SQL Query:

```
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
```

Sample Output: (Refer Csv File for complete output of the Query)

name	standard_quartile	count
Animation	1	22
Animation	2	12
Animation	3	15
Animation	4	17
Children	1	14
Children	2	18
Children	3	14
Children	4	14
Classics	1	14
Classics	2	14
Classics	3	13

Question Set 2:

Question 1:

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.

SQL Query:

```
SELECT DATE_PART('month',r.rental_date) rental_month,
    DATE_PART('year',r.rental_date) rental_year,
    s.store_id,
    COUNT(r.rental_id) rental_count
FROM rental AS r
JOIN staff AS st ON r.staff_id = st.staff_id
JOIN store AS s ON s.store_id = st.store_id
GROUP BY 1,2,3
ORDER BY 4 DESC
```

Sample Output: (Refer Csv File for complete output of the Query)

rental_month	rental_year	store_id	rental_count
7	2005	2	3367
7	2005	1	3342
8	2005	1	2892
8	2005	2	2794
6	2005	1	1163
6	2005	2	1148
5	2005	2	598

Question 2:

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?

SQL Query:

```
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
```

Sample Output: (Refer Csv File for complete output of the Query)

pay_month	full_name	pay_countpermon	pay_amount
2/1/2007 0:00	Ana Bradley	4	19.96
3/1/2007 0:00	Ana Bradley	16	71.84
4/1/2007 0:00	Ana Bradley	12	72.88
5/1/2007 0:00	Ana Bradley	1	2.99
2/1/2007 0:00	Clara Shaw	6	22.94
3/1/2007 0:00	Clara Shaw	16	72.84
4/1/2007 0:00	Clara Shaw	18	93.82
2/1/2007 0:00	Curtis Irby	6	22.94
3/1/2007 0:00	Curtis Irby	17	86.83
4/1/2007 0:00	Curtis Irby	14	54.86

Question 3:

write a query to compare the payment amounts in each successive month.

SQL Query:

```
WITH t1 AS (
        SELECT c.first_name || ' ' || c.last_name AS customer,
                                                c.customer_id,
                                                SUM(p.amount) AS total
                                            FROM customer c
                                            JOIN payment p ON c.customer_id = p.customer_id
                                            GROUP BY 1,2
                                            ORDER BY 2 DESC
                                            LIMIT 10
      ),
   t2 AS (
        SELECT date_trunc('month',payment_date) AS pay_month,
            customer,
            COUNT(p.*) pay_ct_per_month,
            SUM(p.amount) pay_amount
            FROM t1
            JOIN payment p ON p.customer_id = t1.customer_id
            GROUP BY 1,2
            ORDER BY 2                        )
SELECT *,
            COALESCE(LEAD(pay_amount) OVER (PARTITION BY customer ORDER BY pay_month) - pay_amount,0) AS difference
FROM t2
```

Sample Output: (Refer Csv File for complete output of the Query)

pay_month	customer	pay_ct_per_month	pay_amount	difference
2/1/2007 0:00	Austin Cintron	4	20.96	6.97
3/1/2007 0:00	Austin Cintron	7	27.93	2
4/1/2007 0:00	Austin Cintron	7	29.93	0
2/1/2007 0:00	Eduardo Hiatt	6	23.94	-4.97
3/1/2007 0:00	Eduardo Hiatt	3	18.97	46.89
4/1/2007 0:00	Eduardo Hiatt	14	65.86	0