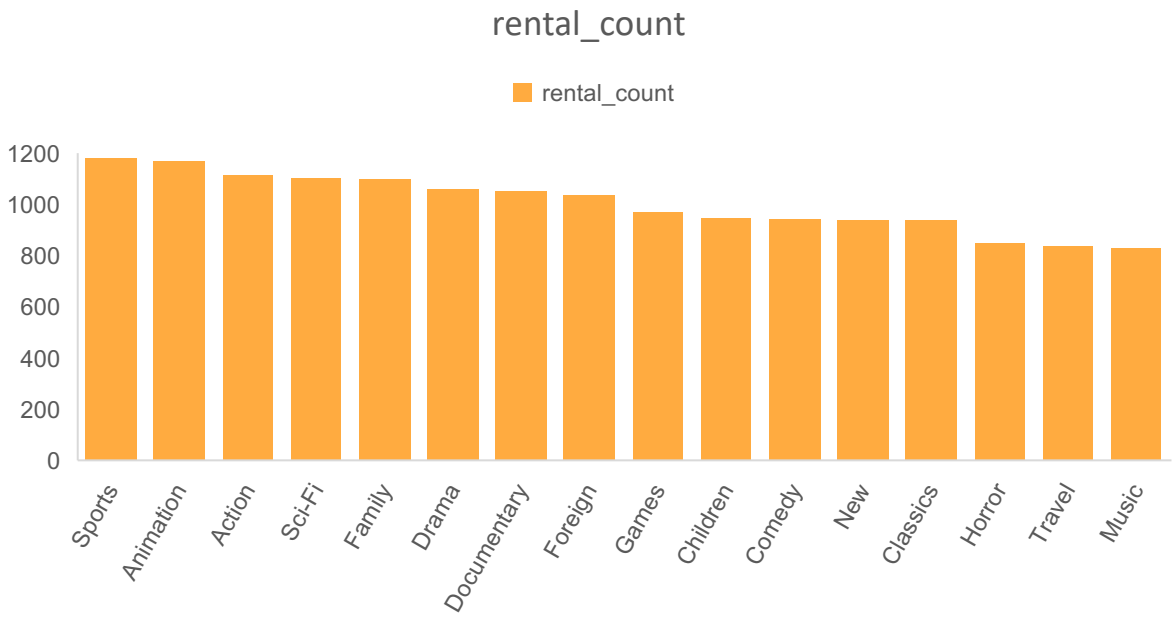
Create a query that lists each movie, the film category it is classified in, and the number of times it has been rented out.

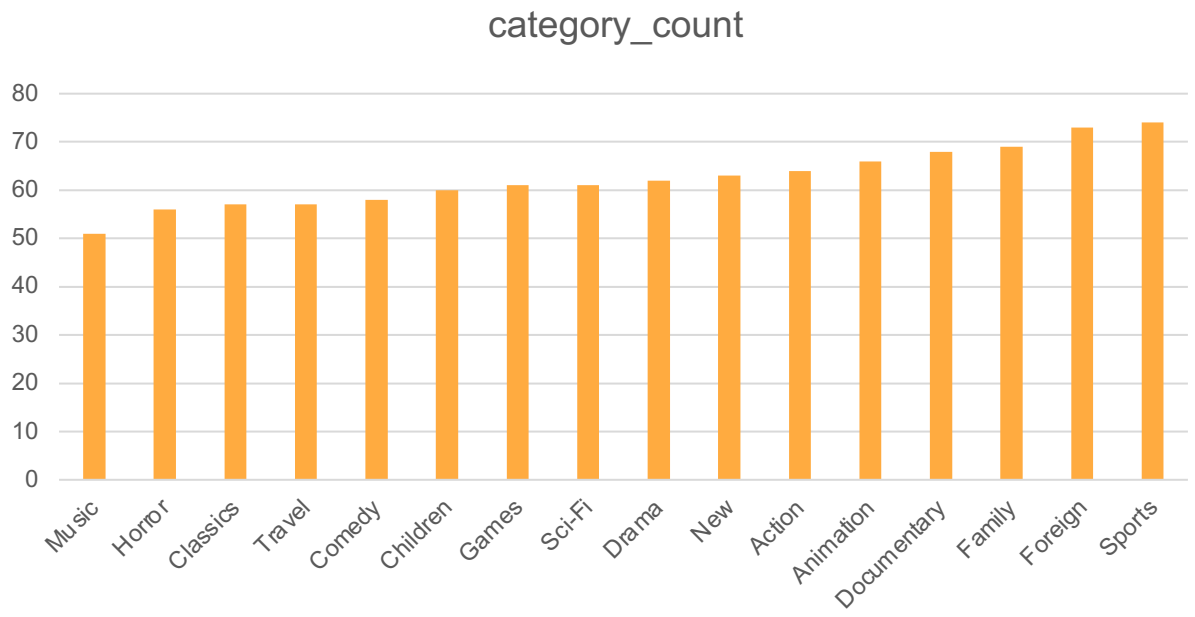


This query list each movie, category and the number of times It has been rented.

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 DISTINCT film_title, category_name,
COUNT(rental_id) over (PARTITION BY film_title) as rental_count
FROM
(
SELECT flm.title film_title, cat.name category_name, ren.rental_id rental_id
FROM film flm
JOIN film_category flmcat ON flmcat.film_id = flm.film_id
 JOIN category cat ON cat.category_id = flmcat.category_id
JOIN inventory i ON i.film_id = flm.film_id
JOIN rental ren ON ren.inventory_id = i.inventory_id
 ) table1
ORDER BY category_name, film_title;
```

Count film based on category.

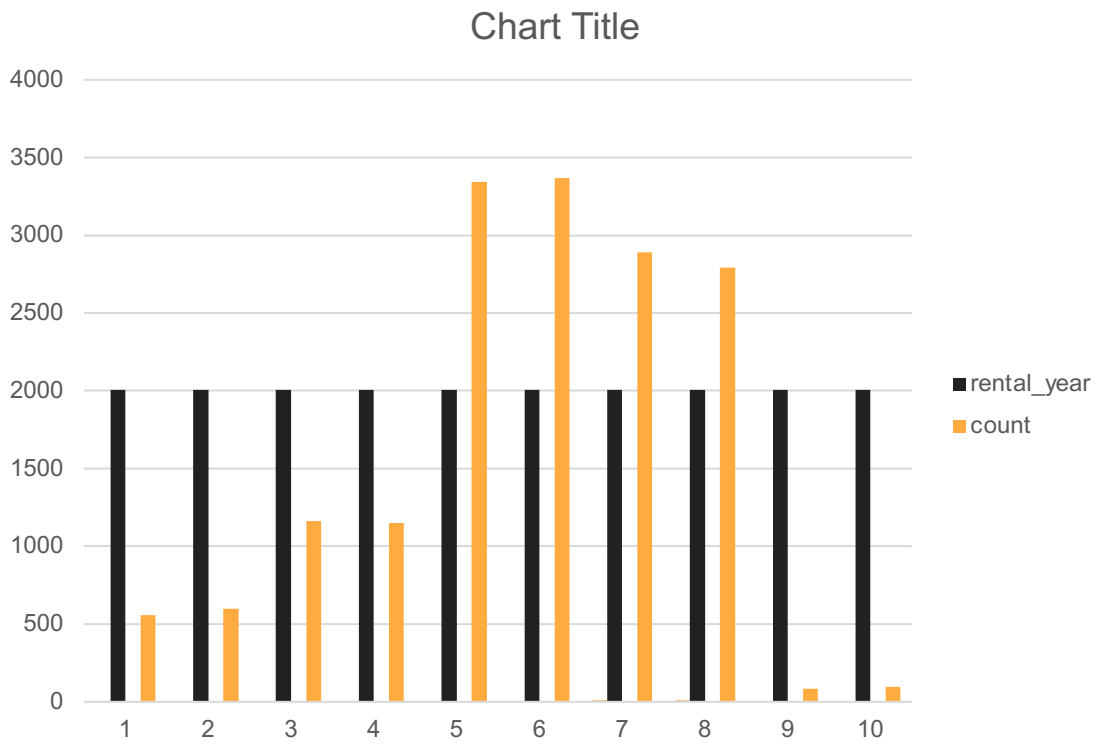


This query list each movie category and the number of movie in that category

Count film based on category.

```
SELECT
DISTINCT category_name,
COUNT(film_title) OVER(
PARTITION BY category_name) AS category_count
FROM (
SELECT f.title film_title, c.name category_name
FROM film f
JOIN film_category fc ON fc.film_id = f.film_id
JOIN category c ON c.category_id = fc.category_id )tab
ORDER BY category_count;
```

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.

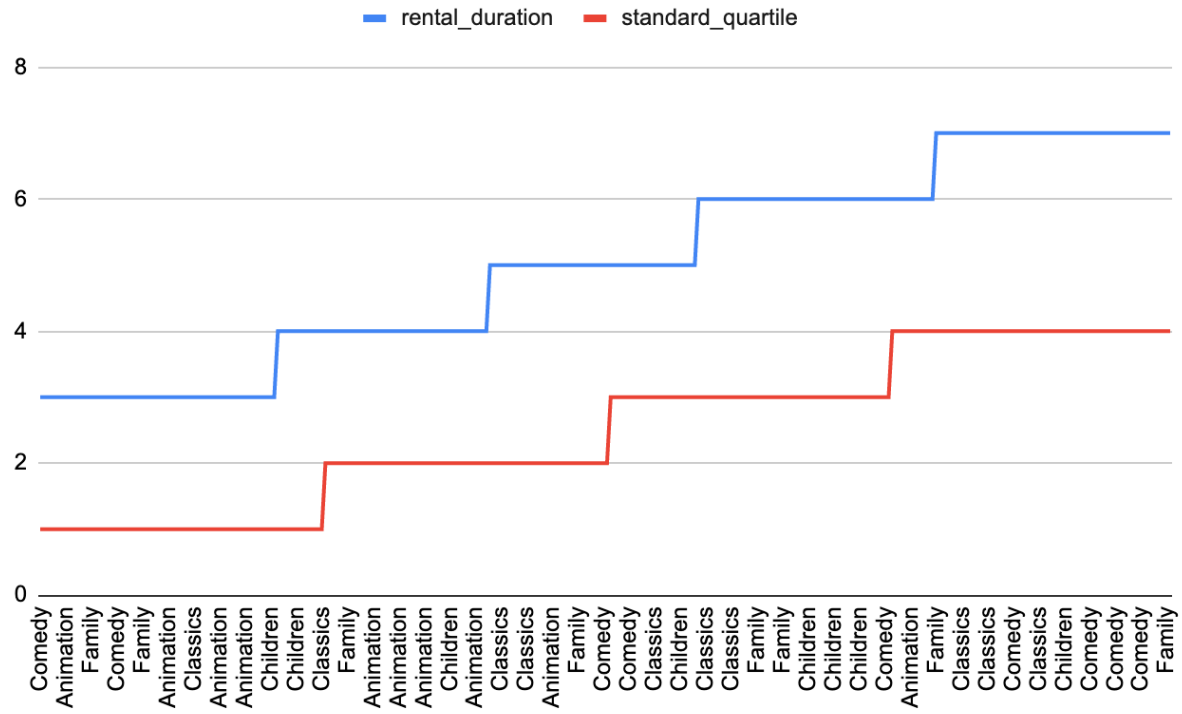


This query list the count number of orders in each rental year.

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.

```
SELECT
DATE_PART('month', renm.rental_date) as rental_month,
DATE_PART('year', renm.rental_date) as rental_year,
 ( '' || store.store_id ) as store_id,
COUNT(*)
FROM store as store
JOIN staff as staff ON store.store_id = staff.store_id
JOIN rental renm ON staff.staff_id = renm.staff_id
GROUP BY
1,
2,
3
ORDER BY 2,
 1;
```

movie titles and divide them into 4 levels (first_quarter, second_quarter, third_quarter, and final_quarter)



This query divide movies into 4 quarter and the rental of movies in all category

movie titles and divide them into 4 levels (first_quarter, second_quarter, third_quarter, and final_quarter)

```
SELECT flm.title, cat.name, flm.rental_duration,
NTILE(4) OVER
(
ORDER BY flm.rental_duration) as standard_quartile
FROM film_category flmcat
JOIN category cat ON cat.category_id = flmcat.category_id
JOIN film flm ON flm.film_id = flmcat.film_id
WHERE cat.name IN
(
'Animation’,
'Children’,
'Classics’,
 'Comedy’,
'Family’
)
ORDER BY 3
```