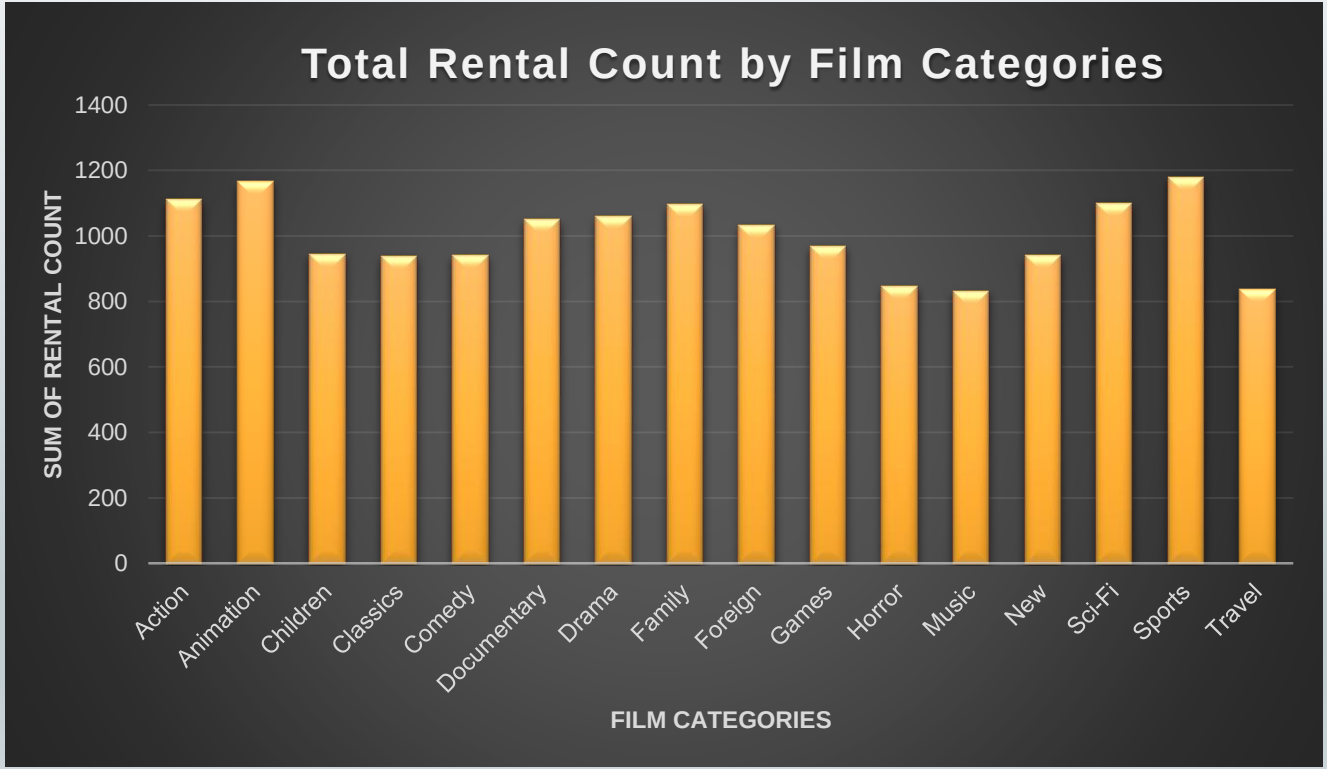
# Rental Count of Movie classified by Category Name




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

(Concept used : subquery, join, window function, visualisation)

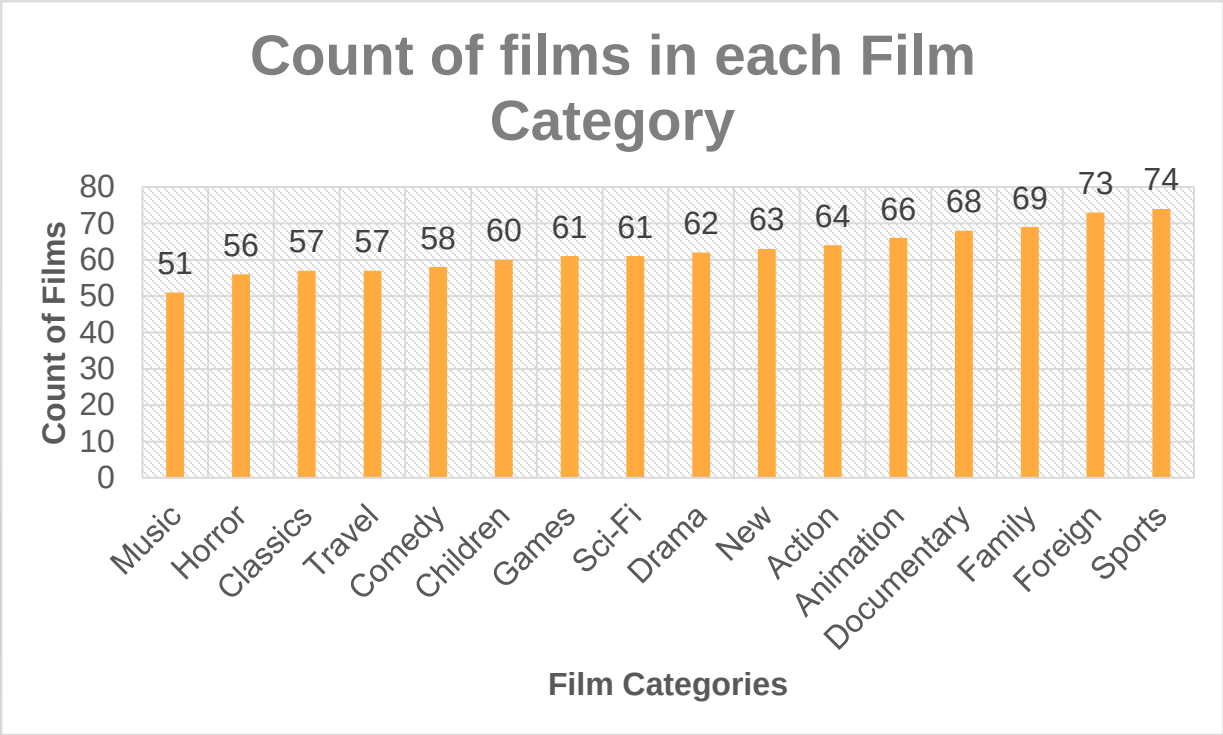
# Rental Count of Movie classified by Category Name

```
SELECT DISTINCT film_title, category_name,
                COUNT(rental_id) OVER(PARTITION BY  film_title) AS rental_count

FROM
                (SELECT f.title film_title, c.name category_name, r.rental_id rental_id
                FROM film f
                JOIN film_category fc ON fc.film_id = f.film_id
                JOIN category c ON c.category_id = fc.category_id
                JOIN inventory i ON i.film_id = f.film_id
                JOIN rental r ON r.inventory_id = i.inventory_id) t1

ORDER BY category_name,film_title;
```

# Categorize and count Films as per film category




This query lists each film category and the number of films made in that category.

(Concepts used : subquery, join, window function, visualisation)

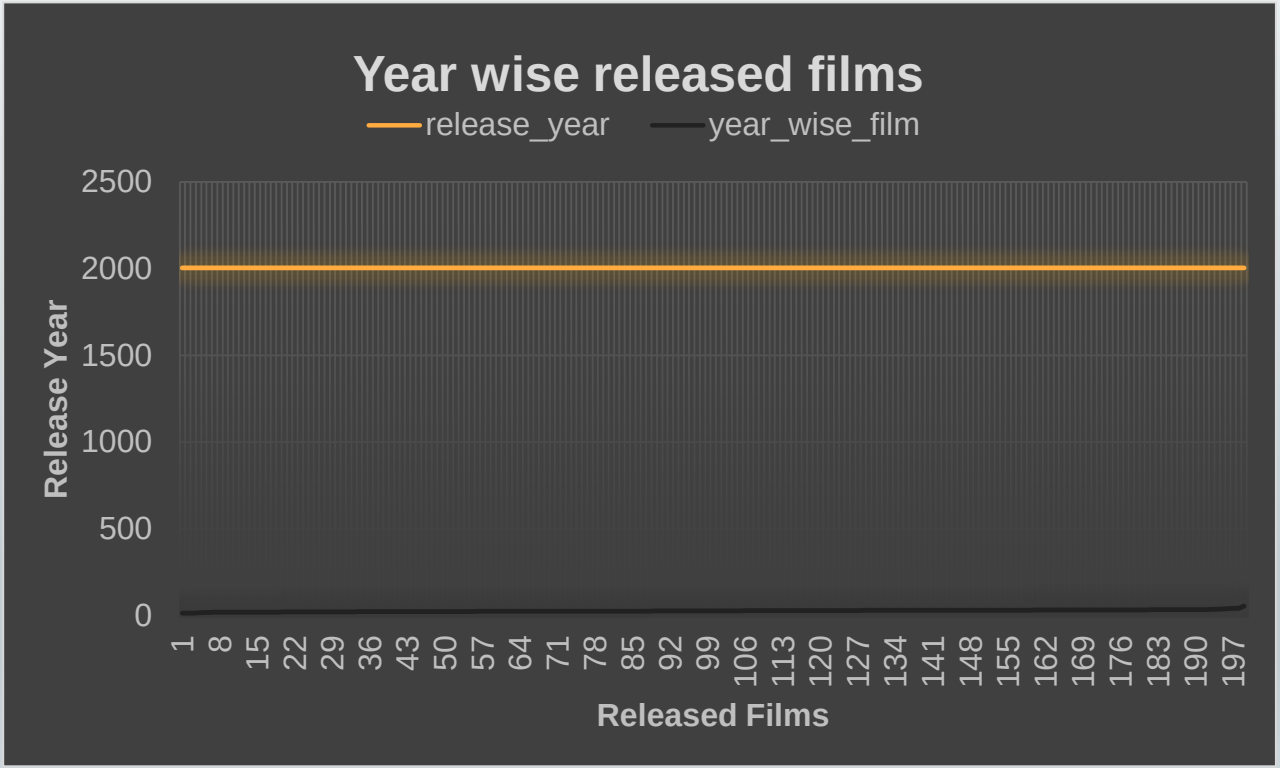
# Categorize and count Films as per film category

```
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) t1

ORDER BY category_count;
```

# Count year wise released films as per actor



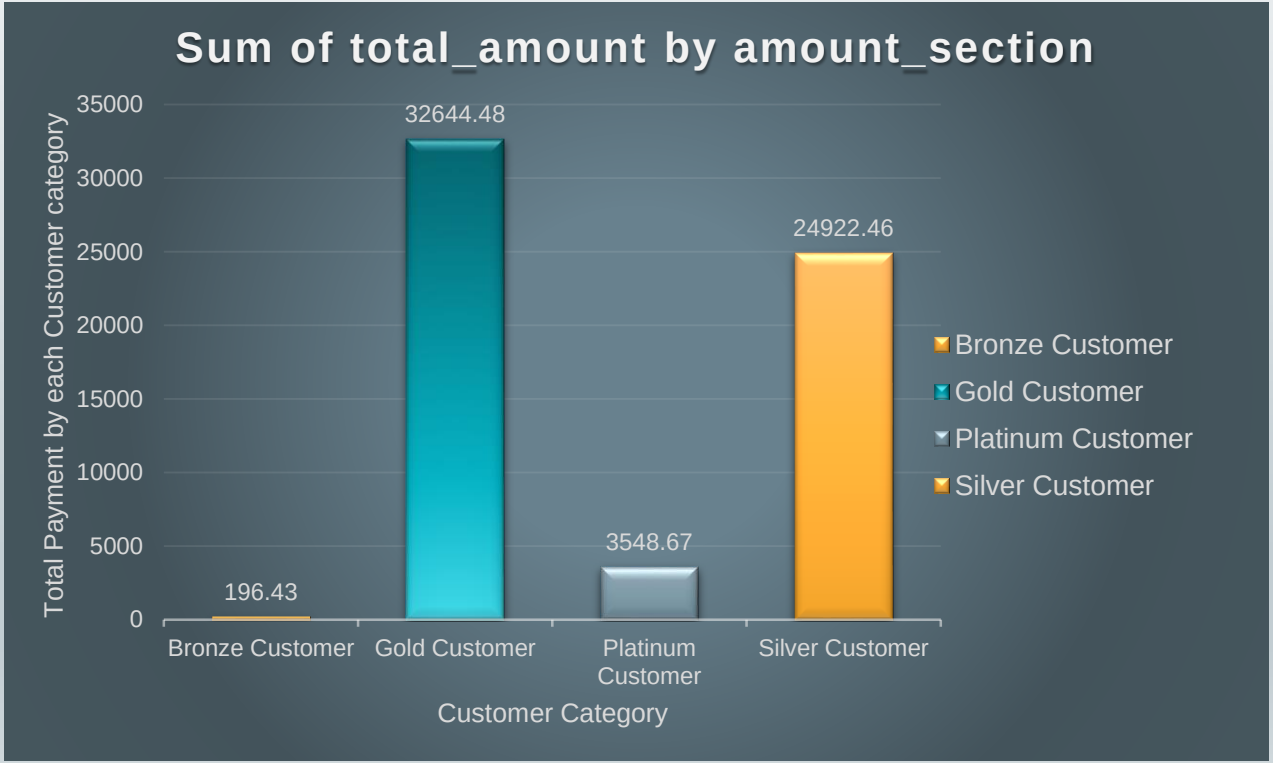

This query lists each actor's full name, film release year and number of year wise released film.

(Concept used: join, aggregrate function, visualisation)

# Count year wise released films as per actor

```
SELECT CONCAT(a.first_name,' ',a.last_name) actor_name,
                COUNT(f.release_year) year_wise_film
FROM actor a
JOIN film_actor fa ON fa.actor_id = a.actor_id
JOIN film f ON f.film_id = fa.film_id
GROUP BY actor_name
ORDER BY COUNT(f.release_year);
```

# Categorize Customers as per payment




This query lists each customer's full name, total payment he has done and category of customer he has been as per payment

(**Bronze, Silver, Gold and Platinum**)

(Concepts used: join, aggregrate function, case, visualisation)

# Categorize Customers as per payment

```
SELECT CONCAT(c.first_name,' ',c.last_name) customer_full_name,
		SUM(p.amount) as total_amount,
		CASE
				WHEN SUM(p.amount)<50 THEN 'Bronze Customer'
				WHEN SUM(p.amount)>50 and SUM(p.amount)<=100 THEN 'Silver Customer'
				WHEN SUM(p.amount)>100 and SUM(p.amount)<=150  THEN 'Gold Customer'
				ELSE 'Platinum Customer'
		END as amount_section
FROM payment p
JOIN customer c ON c.customer_id = p.customer_id
GROUP BY 1
ORDER BY SUM(p.amount);
```