

# Advanced Data Analysis Project

## Using Postgres and DVD-Rental datasets

By: Rana Ghazzi

### Requirements:

- 1- List of the most rented films in each category.
- 2- Calculates the amount of payments were made on a given day & a cumulative sum of payments (Running total by month).
- 3- Find the disks that spent the most time in the rental shop on shelves.

### Solutions:

- 1: List of the most rented films in each category.

```
WITH rental_count AS (  
  SELECT  
    film.*,  
    COUNT(rental.*) AS rental_count  
  FROM  
    film  
    JOIN inventory USING (film_id)  
    JOIN rental USING (inventory_id)  
  GROUP BY  
    film.film_id  
)  
category_rankings AS (  
  SELECT  
    category.name AS category_name,  
    ROW_NUMBER() OVER (  
      PARTITION BY category.category_id ORDER BY  
rental_count.rental_count desc  
    ) AS category_rank,  
    rental_count.title AS film_title,  
    rental_count.rental_count AS rental_count  
  FROM  
    rental_count  
    JOIN film_category USING (film_id)  
    JOIN category USING (category_id)
```

```

)
SELECT
*
FROM
category_rankings
WHERE
category_rank <= 3
ORDER BY
category_name;

```

Q2: Calculates the amount of payments were made on a given day and a cumulative sum of payments (Running total by month).

```

SELECT
Sum(payment.amount),
EXTRACT(
'Day'
From
payment_date
),
EXTRACT(
'Month'
From
payment_date
)
FROM
payment
WHERE
payment_date Between '2007-02-14'
And ' 2007-03-23'
GROUP BY
EXTRACT(
'Month'
From
payment_date
),
EXTRACT(
'Day'
From
payment_date

```

```
)  
ORDER BY  
3,  
2
```

Q3 : what films spent the most time on shelves

```
SELECT  
  RENTAL_DATE - LAG(RETURN_DATE) OVER(  
    PARTITION BY INVENTORY_ID  
    ORDER BY  
      RENTAL_DATE  
  ),  
  title as "Film Name",  
  INVENTORY_ID  
FROM  
  RENTAL  
  join inventory USING(inventory_id)  
  join film USING(film_id)  
WHERE  
  RETURN_DATE IS NOT NULL  
ORDER BY  
  1 DESC NULLS LAST
```