**Guided Project – 1**

Use **[mavenmovies](https://s3.amazonaws.com/grey_campus/production/system/OdinSchool/2023/Edmingle/DataSet/create_mavenmovies+(1).sql" \t "_blank)**database to solve the following:

1. We will need a list of all staff members, including their first and last names,  email addresses, and the store identification number where they work.
2. We will need separate counts of inventory items held at your two stores.
3. We will need a count of active customers for each of your stores. Separately, please.
4. In order to assess the liability of a data breach, we will need you to provide a count of all customer email addresses stored in the database.
5. We are interested in how diverse your film offering is as a means of understanding how likely you are to keep customers engaged in the future. Please provide a count of unique film titles you have in inventory at each store and then provide a count of the unique categories of films you provide.
6. We would like to understand the replacement cost of your films. Please provide the replacement cost for the film that is least expensive to replace, the most expensive to replace, and the average of all films you carry.
7. We are interested in having you put payment monitoring systems and maximum payment processing restrictions in place in order to minimize the future risk of fraud by your staff. Please provide the average payment you processed, as well as the maximum payment you have processed.
8. We would like to better understand what your customer base looks like. Please provide a list of all customer identification values, with a count of rentals they have made at all-time, with your highest volume customers at the top of the list.

-- 1) We will need a list of all staff members, including their first and last names,

-- email addresses, and the store identification number where they work.

select first\_name, last\_name, email,store\_id from staff;

-- 2) We will need separate counts of inventory items held at your two stores.

select store\_id, count(inventory\_id) over(partition by store\_id) from inventory;

select store\_id, count(inventory\_id)

from inventory

group by store\_id;

-- 3) We will need a count of active customers for each of your stores. Separately, please.

select store\_id, count(customer\_id) over(partition by store\_id)

from customer

where active = 1;

select store\_id, count(customer\_id)

from customer

where active =1

group by store\_id;

-- 4) In order to assess the liability of a data breach,

-- we will need you to provide a count of all customer email addresses stored in the database.

select count(email) from customer;

-- 5) We are interested in how diverse your film offering is as a means of understanding

-- how likely you are to keep customers engaged in the future.

-- Please provide a count of unique film titles you have in inventory at each store

-- and then provide a count of the unique categories of films you provide.

select store\_id, count(distinct i.film\_id), count(distinct fc.category\_id)

from inventory i join film\_category fc on i. film\_id = fc.film\_id

group by store\_id;

-- 6) We would like to understand the replacement cost of your films.

-- Please provide the replacement cost for the film that is least expensive to replace,

-- the most expensive to replace, and the average of all films you carry.

select min(replacement\_cost), max(replacement\_cost),avg(replacement\_cost)

from film;

-- 7) We are interested in having you put payment monitoring systems and

-- maximum payment processing restrictions in place in order to minimize the future risk of fraud by your staff.

-- Please provide the average payment you processed, as well as the maximum payment you have processed.

select avg(amount), max(amount) from payment;

-- 8) We would like to better understand what your customer base looks like.

-- Please provide a list of all customer identification values, with a count of rentals they have made at all-time,

-- with your highest volume customers at the top of the list.

select c.customer\_id, count(r.rental\_id) as bc

from customer c join rental r on c.customer\_id=r.customer\_id

group by c.customer\_id

order by bc desc;