**Mini Project – SQL**

Using the dvdrental tables, answer the following questions with a SQL query.

1. Write a select statement to return all columns and rows from the customer table.

SELECT \* FROM customer

Table

Description automatically generated

1. Write a query to return unique rows from the release\_year column in the film table.

SELECT DISTINCT release\_year

FROM film

Graphical user interface, text, application, email

Description automatically generated

1. We’re trying to find a customer located at a certain address ‘259 Ipoh Drive’ – can you find their phone number?

SELECT phone FROM address

WHERE address LIKE ‘259 Ipoh Drive’

Graphical user interface, text, application

Description automatically generated

1. Write a query from the payment table where the amount is either 4.99 or 1.99.

SELECT \* FROM payment

WHERE amount = 4.99 or amount = 1.99

Table

Description automatically generated

1. Display the total amount payed by all customers in the payment table.

SELECT sum(amount) FROM payment

Graphical user interface, text, application

Description automatically generated

1. For each store, display the number of customers.

SELECT store\_id, count(customer\_id) FROM customer

GROUP BY store\_id

Graphical user interface, text, application

Description automatically generated

1. Show the number of movies each actor acted in

SELECT actor\_id, count(film\_id) as Total\_Movies FROM film\_actor

GROUP BY actor\_id

ORDER BY actor\_id

Table

Description automatically generated with medium confidence

1. Count the number of actors who’s first\_names don’t start with an ‘A’

SELECT count(first\_name) from actor

WHERE lower(first\_name) NOT LIKE ‘a%’

Graphical user interface, text, application, email

Description automatically generated

1. Show the number of movies each actor acted in

----SAME AS # 7

1. Return the total amount spent by each customer. Only return those customers who spent more than $200

SELECT customer\_id, sum(amount)

FROM payment

GROUP BY customer\_id

HAVING sum(amount) > 200

Graphical user interface, application

Description automatically generated

1. Return the top 10 customers by payment amount

SELECT customer\_id, sum(amount)

FROM payment

GROUP BY customer\_id

ORDER BY sum(amount) DESC

LIMIT 10

Graphical user interface, text, application

Description automatically generated with medium confidence

1. Return the name of the actor who was in the most movies

SELECT film\_actor.actor\_id, count(film\_id), actor.first\_name, actor.last\_name

FROM film\_actor

JOIN actor

ON film\_actor.actor\_id = actor.actor\_id

GROUP BY film\_actor.actor\_id, actor.actor\_id

ORDER BY count(film\_id) DESC

LIMIT 1

Graphical user interface, text, application

Description automatically generated

1. Edit the SQL Query from 12 to include the name of the movies they acted in

***Used a subquery to filter out the actor with highest amount of movies first, then joined tables to get the list of movies for the selected actor id***

SELECT total, q1.first\_name, q1.last\_name, film.title

FROM (SELECT count(film\_actor.actor\_id) total, film\_actor.actor\_id,

actor.first\_name, actor.last\_name

FROM actor

JOIN film\_actor ON actor.actor\_id = film\_actor.actor\_id

GROUP BY actor.first\_name, actor.last\_name, film\_actor.actor\_id

ORDER BY total DESC

LIMIT 1

)q1

JOIN film\_actor ON q1.actor\_id = film\_actor.actor\_id

JOIN film ON film\_actor.film\_id = film.film\_id

GROUP BY total, q1.first\_name, q1.last\_name, film.title

Graphical user interface, text

Description automatically generated

1. For the customer with the highest payment amount, return their name, payment amount, address, address 2, city name, and country

***Same as #13, used a subquery to filter out the highest then join tables to select name etc.***

SELECT customer.first\_name, customer.last\_name, q1.total,

address.address, address.address2,

city.city, country.country

FROM (

SELECT sum(amount) total, payment.customer\_id, customer.address\_id

FROM payment

JOIN customer ON payment.customer\_id = customer.customer\_id

GROUP BY payment.customer\_id, customer.address\_id

ORDER BY total DESC

LIMIT 1

) q1

JOIN customer ON q1.customer\_id = customer.customer\_id

JOIN address ON q1.address\_id = address.address\_id

JOIN city ON address.city\_id = city.city\_id

JOIN country ON city.country\_id = country.country\_id

GROUP BY customer.first\_name, customer.last\_name, q1.total,

address.address, address.address2,

city.city, country.country

Text

Description automatically generated with medium confidence