1.Write a query to display the customer's first name, last name, email, and city they live in.

SELECT c.first\_name, c.last\_name, c.email, a.city FROM customer c JOIN address a ON c.address\_id = a.address\_id;

2. Retrieve the film title, description, and release year for the film that has the longest duration.

SELECT title, description, release\_year FROM film WHERE length = (SELECT MAX(length) FROM film);

3.List the customer name, rental date, and film title for each rental made. Include customers

who have never rented a film.

SELECT c.first\_name, c.last\_name, r.rental\_date, f.title

FROM customer c

LEFT JOIN rental r ON c.customer\_id = r.customer\_id

LEFT JOIN inventory i ON r.inventory\_id = i.inventory\_id

LEFT JOIN film f ON i.film\_id = f.film\_id;

4.Find the number of actors for each film. Display the film title and the number of actors for each

Film.

SELECT f.title, COUNT(actor\_id) AS actor\_count

FROM film f

LEFT JOIN film\_actor fa ON f.film\_id = fa.film\_id

GROUP BY f.title;

5.Display the first name, last name, and email of customers along with the rental date, film title,

and rental return date.

SELECT c.first\_name, c.last\_name, c.email, r.rental\_date, f.title, r.return\_date

FROM customer c

JOIN rental r ON c.customer\_id = r.customer\_id

JOIN inventory i ON r.inventory\_id = i.inventory\_id

JOIN film f ON i.film\_id = f.film\_id;

6.Retrieve the film titles that are rented by customers whose email domain ends with '.net'.

SELECT title

FROM film

WHERE film\_id IN (

SELECT film\_id

FROM inventory

WHERE store\_id IN (

SELECT store\_id

FROM customer

WHERE email LIKE '%.net'

)

);

7. Show the total number of rentals made by each customer, along with their first and last

names.

SELECT c.first\_name, c.last\_name, COUNT(r.rental\_id) AS total\_rentals

FROM customer c

LEFT JOIN rental r ON c.customer\_id = r.customer\_id

GROUP BY c.customer\_id

ORDER BY total\_rentals DESC;

8.List the customers who have made more rentals than the average number of rentals made by

all customers.

SELECT first\_name, last\_name

FROM customer

WHERE customer\_id IN (

SELECT customer\_id

FROM rental

GROUP BY customer\_id

HAVING COUNT(rental\_id) > (

SELECT AVG(rental\_count)

FROM (

SELECT COUNT(rental\_id) AS rental\_count

FROM rental

GROUP BY customer\_id

) AS avg\_rentals

)

);

9. Display the customer first name, last name, and email along with the names of other

customers living in the same city.

SELECT c1.first\_name, c1.last\_name, c1.email, c2.first\_name AS other\_first\_name,

c2.last\_name AS other\_last\_name

FROM customer c1

JOIN address a1 ON c1.address\_id = a1.address\_id

JOIN address a2 ON a1.city = a2.city AND a1.address\_id != a2.address\_id

JOIN customer c2 ON a2.address\_id = c2.address\_id;

10.Retrieve the film titles with a rental rate higher than the average rental rate of films in the

same category.

SELECT title, rental\_rate

FROM film f

WHERE rental\_rate > (

SELECT AVG(rental\_rate)

FROM film

WHERE category\_id = f.category\_id

);

11.Retrieve the film titles along with their descriptions and lengths that have a rental rate greater

than the average rental rate of films released in the same year.

SELECT title, description, length

FROM film

WHERE rental\_rate > (

SELECT AVG(rental\_rate)

FROM film

WHERE release\_year = film.release\_year

);

12.List the first name, last name, and email of customers who have rented at least one film in the

'Documentary' category.

SELECT first\_name, last\_name, email

FROM customer

WHERE customer\_id IN (

SELECT DISTINCT c.customer\_id

FROM customer c

JOIN rental r ON c.customer\_id = r.customer\_id

JOIN inventory i ON r.inventory\_id = i.inventory\_id

JOIN film\_category fc ON i.film\_id = fc.film\_id

JOIN category cat ON fc.category\_id = cat.category\_id

WHERE cat.name = 'Documentary'

);

13.Show the title, rental rate, and difference from the average rental rate for each film.

SELECT title, rental\_rate,

rental\_rate - (SELECT AVG(rental\_rate) FROM film) AS rate\_difference

FROM film;

14.Retrieve the titles of films that have never been rented.

SELECT title

FROM film

WHERE film\_id NOT IN (

SELECT DISTINCT film\_id

FROM inventory

WHERE film\_id IS NOT NULL

);

15. List the titles of films whose rental rate is higher than the average rental rate of films released

in the same year and belong to the 'Sci-Fi' category.

SELECT title

FROM film f

WHERE rental\_rate > (

SELECT AVG(rental\_rate)

FROM film

WHERE release\_year = f.release\_year)

AND film\_id IN (

SELECT fc.film\_id

FROM film\_category fc

JOIN category c ON fc.category\_id = c.category\_id

WHERE c.name = 'Sci-Fi'

);

16.Find the number of films rented by each customer, excluding customers who have rented

fewer than five films.

SELECT customer\_id, COUNT(rental\_id) AS film\_count

FROM rental

GROUP BY customer\_id

HAVING COUNT(rental\_id) >= 5;