

1. SHOW TABLES;
2. SELECT title
FROM film
WHERE length > 120;
3. SELECT title
FROM film
WHERE rating = 'G'
ORDER BY length, title
LIMIT 4;
4. SELECT film.title, language.name
FROM film INNER JOIN language ON film.language_id = language.language_id
WHERE film.description LIKE '%Drama%';
5. SELECT film.title
FROM film INNER JOIN film_category USING(film_id) INNER JOIN category
USING(category_id)
WHERE category.name = 'Family' AND film.description LIKE '%Documentary%';
6. SELECT film.title
FROM film INNER JOIN film_category USING(film_id) INNER JOIN category
USING(category_id) WHERE category.name = 'Children' AND NOT film.rating = 'G';
7. SELECT rating, COUNT(*)
FROM film
GROUP BY rating;
8. SELECT DISTINCT film.title
FROM film INNER JOIN inventory USING(film_id) INNER JOIN rental
USING(inventory_id)
WHERE DATE_FORMAT(rental.rental_date, '%Y-%m-%d') BETWEEN '2005-05-31' AND
'2005-06-15' ORDER BY film.title;
9. SELECT DISTINCT actor.first_name, actor.last_name
FROM actor INNER JOIN film_actor USING(actor_id) INNER JOIN film USING(film_id)
WHERE film.special_features LIKE '%Deleted Scenes%'
ORDER BY actor.last_name;
10. SELECT first_name, last_name
FROM customer
WHERE customer_id IN (
SELECT DISTINCT rental.customer_id
FROM rental INNER JOIN payment USING(rental_id)
WHERE rental.staff_id <> payment.staff_id
);
11. SELECT first_name, last_name
FROM customer INNER JOIN rental USING(customer_id)
GROUP BY customer_id HAVING COUNT(rental_id) > (
SELECT COUNT(rental_id)
FROM customer INNER JOIN rental USING(customer_id)
WHERE customer.email = 'MARY.SMITH@sakilacustomer.org'
);
12. SELECT CONCAT(a1.first_name, ' ', a1.last_name) AS first_actor, CONCAT(a2.first_name, '
' , a2.last_name) AS second_actor
FROM (
SELECT DISTINCT actor1.actor_id AS id1, actor2.actor_id AS id2

```

FROM film_actor AS actor1 INNER JOIN film_actor AS actor2 USING(film_id)
WHERE actor1.actor_id < actor2.actor_id
GROUP BY id1, id2
HAVING COUNT(*) > 1
) AS pairs INNER JOIN actor AS a1 ON pairs.id1 = a1.actor_id INNER JOIN actor AS a2 ON
pairs.id2 = a2.actor_id
ORDER BY first_actor;
13. SELECT last_name
FROM actor
WHERE actor_id NOT IN (
SELECT DISTINCT actor_id
FROM film_actor
WHERE film_id IN (
SELECT film_id
FROM film
WHERE title LIKE 'C%'
)
);
14. CREATE VIEW actor_category AS (
SELECT film_id, name AS category_name, actor_id
FROM film_category USING(category_id) INNER JOIN film_actor USING(film_id)
WHERE category.name IN('Action', 'Horror')
);
SELECT DISTINCT last_name
FROM actor
WHERE actor_id IN (
SELECT actor_id
FROM (
SELECT actor_id, COUNT(film_id) AS no_action
FROM actor_category
WHERE category_name = 'Action'
GROUP BY actor_id
) AS action LEFT JOIN (
SELECT actor_id, COUNT(film_id) AS no_horror
FROM actor_category
WHERE category_name = 'Horror'
GROUP BY actor_id
) AS horror USING(actor_id)
WHERE no_action > no_horror OR no_horror IS NULL
);
15. SELECT customer_id, first_name, last_name
FROM customer
WHERE customer_id IN (
SELECT customer_id
FROM payment
GROUP BY customer_id
HAVING AVG(amount) < (
SELECT AVG(amount)
FROM payment

```

```

        WHERE DATE_FORMAT(payment_date, '%Y-%m-%d') = '2005-07-30'
    )
);
16. UPDATE film
    SET language_id = (
        SELECT language_id
        FROM language
        WHERE name = 'Italian'
    ) WHERE title = 'YOUNG LANGUAGE';
17. INSERT INTO language VALUES(7, 'Spanish', CURDATE());
    UPDATE film
    SET language_id = (
        SELECT language_id
        FROM language
        WHERE name = 'Spanish'
    ) WHERE film_id IN (
        SELECT film_id
        FROM film INNER JOIN film_actor USING(film_id) INNER JOIN actor USING(actor_id)
        WHERE actor.first_name = 'ED' AND actor.last_name = 'CHASE'
    );
18. ALTER TABLE language
    ADD films_no int
    AFTER last_update;
    UPDATE language
    SET films_no = (
        SELECT COUNT(language_id)
        FROM film
        WHERE film.language_id = language.language_id
    );
19. ALTER TABLE film
    DROP COLUMN release_year;

```