

1. Wypisz wszystkie znajdujące się w bazie tabele.

```
SHOW TABLES;
```

2. Wypisz tytuły filmów o długości większej niż 2 godziny.

```
SELECT title
FROM film
WHERE length > 120;
```

3. Wypisz tytuły 4 najkrótszych filmów o kategorii wiekowej PG-13.

```
SELECT title
FROM film
WHERE rating='PG-13'
ORDER BY length ASC
LIMIT 4;
```

4. Wypisz tytuły filmów oraz ich język, dla wszystkich filmów, w których opisie występuje słowo Drama.

```
SELECT film.title, language.name
FROM film JOIN language
ON film.language_id = language.language_id
WHERE film.description LIKE "%Drama%";
```

5. Wypisz tytuły filmów z kategorii Family, które w swoim opisie zawierają słowo Documentary.

```
SELECT film.title FROM film
JOIN film_category ON film.film_id = film_category.film_id
JOIN category ON film_category.category_id = category.category_id
WHERE category.name = 'Family'
AND film.description LIKE "%Documentary%";
```

6. Wypisz tytuły filmów z kategorii Children, które nie należą do kategorii wiekowej PG-13.

```
SELECT film.title FROM film
JOIN film_category ON film.film_id = film_category.film_id
JOIN category ON film_category.category_id = category.category_id
WHERE category.name = 'Children'
```

```
AND film.rating != 'PG-13';
```

7. Dla każdej kategorii wiekowej filmów (G, PG-13, PG, NC-17, R) wypisz liczbę filmów do niej należących.

```
SELECT rating, COUNT(film_id)
FROM film
GROUP BY rating;
```

8. Wypisz tytuły filmów wypożyczonych pomiędzy 31 maja a 30 czerwca 2005. Wyniki posortuj w odwrotnej kolejności alfabetycznej.

```
SELECT film.title FROM film
JOIN inventory ON film.film_id = inventory.film_id
JOIN rental ON inventory.inventory_id = rental.inventory_id
WHERE DATE(rental.rental_date) >= '2005-05-31' AND DATE(rental.rental_date)
<= '2005-06-30'
GROUP BY film.title
ORDER BY film.title DESC;
```

9. Wypisz imiona i nazwiska wszystkich aktorów, którzy wystąpili w filmach zawierających usunięte sceny.

```
SELECT actor.first_name, actor.last_name FROM actor
JOIN film_actor ON actor.actor_id = film_actor.actor_id
JOIN film ON film_actor.film_id = film.film_id
WHERE film.special_features LIKE '%Deleted scenes%'
GROUP BY actor.actor_id;
```

10. Wypisz imiona oraz nazwiska wszystkich klientów, których wypożyczenie i odpowiadająca mu płatność były obsługane przez 2 różnych pracowników.

```
SELECT customer.first_name, customer.last_name FROM customer
JOIN rental ON customer.customer_id = rental.customer_id
JOIN payment ON rental.rental_id = payment.rental_id
WHERE rental.staff_id != payment.staff_id
GROUP BY customer.customer_id;
```

11. Wypisz imiona i nazwiska wszystkich klientów, którzy wypożyczyli więcej filmów niż klient o e-mailu MARY.SMITH@sakilacustomer.org.

```
SELECT customer.first_name, customer.last_name FROM customer
JOIN rental ON customer.customer_id = rental.customer_id
GROUP BY customer.customer_id
HAVING COUNT(rental.rental_id) > (
    SELECT COUNT(rental.rental_id) FROM rental
    JOIN customer ON customer.customer_id = rental.customer_id
    WHERE customer.email = 'MARY.SMITH@sakilacustomer.org'
);
```

12. Wypisz wszystkie pary aktorów, którzy wystąpili razem w więcej niż jednym filmie. Każda para powinna występować co najwyżej raz. Jeśli występuje para (X, Y), to nie wypisuj pary (Y, X).

```
SELECT a1.first_name, a1.last_name, a2.first_name, a2.last_name FROM
film_actor AS f1
JOIN actor AS a1 ON a1.actor_id = f1.actor_id
JOIN film_actor AS f2 ON f1.film_id = f2.film_id
JOIN actor AS a2 ON a2.actor_id = f2.actor_id
WHERE a1.actor_id < a2.actor_id
GROUP BY a1.actor_id, a2.actor_id
HAVING COUNT(f1.film_id) > 1;
```

13. Wypisz nazwiska aktorów, którzy nie wystąpili w żadnym filmie, którego tytuł zaczyna się literą C.

```
SELECT actor.last_name FROM actor
WHERE actor.actor_id NOT IN (
    SELECT actor.actor_id FROM actor
    JOIN film_actor ON actor.actor_id = film_actor.actor_id
    JOIN film ON film_actor.film_id = film.film_id
    WHERE film.title LIKE "C%"
);
```

14. Wypisz nazwiska aktorów, którzy zagraли w większej liczbie horrorów niż filmów akcji.

```
SELECT actor.last_name FROM actor
JOIN film_actor ON actor.actor_id = film_actor.actor_id
JOIN film_category ON film_actor.film_id = film_category.film_id
JOIN category ON film_category.category_id = category.category_id
WHERE category.name = "Horror"
GROUP BY actor.actor_id
HAVING COUNT(film_actor.film_id) > (
    SELECT COUNT(film_actor.film_id) FROM film_actor
    JOIN film_category ON film_actor.film_id = film_category.film_id
    JOIN category ON film_category.category_id = category.category_id
    WHERE category.name = "Action" AND film_actor.actor_id = actor.actor_id
);
```

Lepsza wersja:

```
SELECT actor.last_name FROM actor
JOIN film_actor ON actor.actor_id = film_actor.actor_id
JOIN film_category ON film_actor.film_id = film_category.film_id
JOIN category ON film_category.category_id = category.category_id
GROUP BY actor.actor_id
HAVING SUM(CASE WHEN category.name = "Horror" THEN 1 ELSE 0 END) > SUM(CASE
WHEN category.name = "Action" THEN 1 ELSE 0 END);
```

15. Wypisz wszystkich klientów, których średnia opłata za wypożyczony film jest niższa niż średnia opłata dokonana 30 lipca 2005.

```
SELECT customer.first_name, customer.last_name FROM customer
JOIN payment ON customer.customer_id = payment.customer_id
GROUP BY customer.customer_id
HAVING AVG(payment.amount) < (
    SELECT AVG(payment.amount) FROM payment
    WHERE DATE(payment.payment_date) = '2005-07-30'
);
```

16. Zmień język filmu YOUNG LANGUAGE na włoski.

```
UPDATE film
SET language_id = (SELECT language_id FROM language WHERE name = 'Italian')
WHERE title = 'YOUNG LANGUAGE';
```

17. Dodaj do tabeli language język hiszpański i zmień język wszystkich filmów, w których występuje ED CHASE na hiszpański.

```
-- Add language
INSERT INTO language (name, last_update)
VALUES ('Spanish', CURRENT_TIMESTAMP());

-- Change language
UPDATE film
SET language_id = (SELECT language_id FROM language WHERE name = 'Spanish')
WHERE film_id IN (
    SELECT film_actor.film_id FROM film_actor
    JOIN actor ON film_actor.actor_id = actor.actor_id
    WHERE actor.first_name = "ED" AND actor.last_name = "CHASE"
);
```

18. Do tabeli language dodaj kolumnę films_no i uzupełnij ją liczbą filmów w danym języku.

```
-- Add column
ALTER TABLE language
ADD films_no INT UNSIGNED;

-- Add numbers
UPDATE language
SET language.films_no = (SELECT COUNT(film_id) FROM film WHERE
film.language_id = language.language_id);
```

19. Usuń kolumnę release_year z tabeli film.

```
ALTER TABLE film
DROP COLUMN release_year;
```