

1. Wypisz wszystkie znajdujące się w bazie tabele.

```
01 | SHOW TABLES;
```

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

```
01 | SELECT title
02 | FROM film
03 | WHERE length > 120;
```

3. Wypisz tytuły 4 najkrótszych filmów o kategorii wiekowej G.

```
01 | SELECT title
02 | FROM film
03 | WHERE rating = 'G'
04 | ORDER BY length
05 | LIMIT 4;
```

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

```
01 | SELECT film.title, language.name
02 | FROM film LEFT JOIN language ON film.language_id = language.language_id
03 | WHERE film.description LIKE '%Drama%';
```

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

```
01 | SELECT film.title
02 | FROM film
03 | LEFT JOIN film_category ON film.film_id = film_category.film_id
04 | LEFT JOIN category ON film_category.category_id = category.category_id
05 | WHERE film.description LIKE '%Documentary%' AND category.name = 'Family';
```

6. Wypisz tytuły filmów z kategorii Children, które nie należą do kategorii wiekowej G.

```
01 | SELECT film.title
02 | FROM film
03 | LEFT JOIN film_category ON film.film_id = film_category.film_id
04 | LEFT JOIN category ON film_category.category_id = category.category_id
05 | WHERE film.rating <> 'G' AND category.name = 'Children';
```

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

```
01 | SELECT rating, COUNT(film_id) AS liczba
02 | FROM film
03 | GROUP BY rating;
```

8. Wypisz tytuły filmów wypożyczonych pomiędzy 31 maja a 15 czerwca 2005. Wyniki posortuj alfabetycznie.

```
01 | SELECT DISTINCT film.title
02 | FROM rental
03 | LEFT JOIN inventory ON rental.inventory_id = inventory.inventory_id
04 | JOIN film ON inventory.film_id = film.film_id
05 | WHERE rental.rental_date BETWEEN '2005-05-31' AND '2005-06-15'
06 | ORDER BY film.title;
```

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

```
01 | SELECT DISTINCT actor.first_name, actor.last_name
02 | FROM film
03 | RIGHT JOIN film_actor ON film.film_id = film_actor.film_id
04 | LEFT JOIN actor ON film_actor.actor_id = actor.actor_id
05 | WHERE film.special_features LIKE '%Deleted Scenes%';
```

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

```
01 | SELECT DISTINCT customer.first_name, customer.last_name
02 | FROM rental
03 | LEFT JOIN payment ON rental.rental_id = payment.rental_id
04 | LEFT JOIN customer ON rental.customer_id = customer.customer_id
05 | WHERE rental.staff_id <> payment.staff_id;
```

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

```
01 | CREATE VIEW t AS
02 | SELECT customer.customer_id, COUNT(rental.rental_id) AS num, customer.
   | first_name, customer.last_name, customer.email
03 | FROM rental
04 | LEFT JOIN customer ON rental.customer_id = customer.customer_id
05 | GROUP BY customer.customer_id;
06 |
07 | SELECT first_name, last_name
08 | FROM t
09 | WHERE num > (SELECT num
   | FROM t
   | WHERE email = 'MARY.SMITH@sakilacustomer.org');
12 |
13 | DROP VIEW t;
```

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).

```
01 | CREATE VIEW t AS
02 | SELECT t1.actor_id AS actor1_id, t2.actor_id AS actor2_id, t1.film_id
03 | FROM film_actor AS t1
04 | JOIN film_actor AS t2 ON t1.film_id = t2.film_id AND t1.actor_id < t2.actor_id;
05 |
06 | CREATE VIEW u AS
07 | SELECT DISTINCT t1.actor1_id, t1.actor2_id
08 | FROM t AS t1
09 | JOIN t AS t2 ON t1.actor1_id = t2.actor1_id AND t1.actor2_id = t2.actor2_id
   | AND t1.film_id < t2.film_id;
10 |
11 | SELECT t1.first_name, t1.last_name, t2.first_name, t2.last_name
12 | FROM u
13 | LEFT JOIN actor AS t1 on u.actor1_id = t1.actor_id
14 | LEFT JOIN actor AS t2 on u.actor2_id = t2.actor_id;
15 |
16 | DROP VIEW t, u;
```

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

```
01 | CREATE VIEW t AS
02 | SELECT DISTINCT actor.actor_id
03 | FROM film_actor
04 | LEFT JOIN film ON film_actor.film_id = film.film_id
05 | LEFT JOIN actor ON film_actor.actor_id = actor.actor_id
06 | WHERE film.title LIKE 'C%';
07 |
08 | SELECT DISTINCT last_name
09 | FROM actor
10 | WHERE actor_id NOT IN (SELECT * FROM t);
11 |
12 | DROP VIEW t;
```

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

```
01 | CREATE VIEW t AS
02 | SELECT actor.actor_id, film.film_id, category.name
03 | FROM film_actor
04 | LEFT JOIN film ON film_actor.film_id = film.film_id
05 | LEFT JOIN actor ON film_actor.actor_id = actor.actor_id
06 | LEFT JOIN film_category ON film.film_id = film_category.film_id
07 | LEFT JOIN category ON film_category.category_id = category.category_id;
08 |
09 | CREATE VIEW t_a AS
10 | SELECT actor_id, COUNT(film_id) as a_num
11 | FROM t
12 | WHERE name = 'Action'
13 | GROUP BY actor_id;
14 |
15 | CREATE VIEW t_h AS
16 | SELECT actor_id, COUNT(film_id) as h_num
17 | FROM t WHERE name = 'Horror'
18 | GROUP BY actor_id;
19 |
20 | SELECT DISTINCT actor.last_name
21 | FROM actor
22 | LEFT JOIN t_a ON actor.actor_id = t_a.actor_id
23 | LEFT JOIN t_h ON actor.actor_id = t_h.actor_id
24 | WHERE IFNULL(t_a.a_num, 0) > IFNULL(t_h.h_num, 0);
25 |
26 | DROP VIEW t, t_a, t_h;
```

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

```
01 | SELECT customer.customer_id, customer.first_name, customer.last_name
02 | FROM customer
03 | RIGHT JOIN payment ON customer.customer_id = payment.customer_id
04 | GROUP BY customer.customer_id
05 | HAVING AVG(payment.amount) < (SELECT AVG(amount)
06 | FROM payment
07 | WHERE payment_date LIKE '2005-07-30%');
```

16. Zmien język filmu YOUNG LANGUAGE na włoski.

```
01 | UPDATE film
02 | SET language_id = (SELECT language_id
03 |                     FROM language
04 |                     WHERE name = 'Italian')
05 | WHERE title = 'YOUNG LANGUAGE';
```

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

```
01 | INSERT INTO language (name)
02 | VALUES('Spanish');
03 |
04 | UPDATE film
05 | SET language_id = (SELECT language_id
06 |                     FROM language
07 |                     WHERE name = 'Spanish')
08 | WHERE film_id IN (SELECT film_id
09 |                   FROM film_actor
10 |                   WHERE actor_id = (SELECT actor_id
11 |                                     FROM actor
12 |                                     WHERE first_name = 'ED'
13 |                                     AND last_name = 'CHASE'));
```

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

```
01 | ALTER TABLE language
02 | ADD COLUMN films_no INT AFTER last_update;
03 |
04 | UPDATE language
05 | SET films_no = (SELECT COUNT(film.film_id)
06 |                 FROM film
07 |                 WHERE film.language_id = language_id);
```

19. Usuń kolumnę release_year z tabeli film.

```
01 | ALTER TABLE film
02 | DROP release_year;
```