

1. Wypisz wszystkie znajdujące się w bazie tabele.

```
SHOW tables FROM sakila;
```

2. Wypisz tytuły filmów o długości większej niż 120 minut.

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

3. Wypisz tytuły filmów oraz ich język, dla wszystkich filmów, w których opisie występuje słowo Documentary.

```
SELECT title, name FROM film a JOIN language b ON a.language_id=b.language_id WHERE description LIKE '%Documentary%';
```

4. Wypisz tytuły filmów z kategorii Documentary, które w swoim opisie nie zawierają słowa Documentary.

```
SELECT title FROM (SELECT title, category_id, description FROM film a JOIN film_category b ON a.film_id=b.film_id) c JOIN category ON c.category_id=category.category_id WHERE name LIKE 'Documentary' and description NOT LIKE '%Documentary%';
```

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

```
SELECT DISTINCT first_name, last_name FROM (SELECT actor_id FROM (SELECT film_id FROM film WHERE special_features LIKE '%Deleted Scenes%') a JOIN film_actor ON a.film_id=film_actor.film_id) b JOIN actor on b.actor_id=actor.actor_id;
```

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

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

7. Wypisz tytuły filmów wypożyczonych pomiędzy 25 a 30 maja 2005. Wyniki posortuj alfabetycznie

```
SELECT DISTINCT title FROM (SELECT film_id FROM (SELECT inventory_id FROM rental WHERE rental_date BETWEEN '2005-05-25 00:00:00' AND '2005-05-30 23:59:59') a JOIN inventory ON a.inventory_id=inventory.inventory_id) b JOIN film ON b.film_id=film.film_id ORDER BY title;
```

8. Wypisz tytuły 5 najdłuższych filmów o kategorii R.

```
SELECT title FROM film WHERE rating='R' ORDER BY length DESC LIMIT 5;
```

9. Wypisz imiona oraz nazwiska wszystkich klientów, którzy byli obsłużeni przez 2 różnych pracowników. (Uwaga: Nie używaj grupowania.)

```
SELECT first_name, last_name FROM (SELECT DISTINCT a.customer_id FROM rental a JOIN rental b ON a.staff_id>b.staff_id AND a.customer_id=b.customer_id) c JOIN customer d ON c.customer_id=d.customer_id;
```

10. Wypisz nazwy krajów o liczbie miast w bazie nie mniejszej niż liczba miast kanadyjskich w bazie.

```
SELECT country FROM (SELECT country_id, COUNT(*) AS liczba FROM city GROUP BY country_id) a JOIN country b ON a.country_id=b.country_id WHERE liczba>=(SELECT COUNT(*) AS LiczbaMiastKanadyjskich FROM city WHERE country_id=(SELECT country_id FROM country WHERE country LIKE 'Canada'));
```

11. Wybierz wszystkich klientów, którzy wypożyczyli więcej filmów niż klient o emailu PETER.MENARD@sakilacustomer.org.

```
SELECT first_name, last_name FROM (SELECT customer_id FROM (SELECT COUNT(*) AS liczba1 FROM (SELECT customer_id FROM customer where email LIKE 'PETER.MENARD@sakilacustomer.org') a JOIN rental ON a.customer_id=rental.customer_id b JOIN (SELECT customer_id, COUNT(*) AS liczba2 FROM rental GROUP BY customer_id) c ON liczba1<liczba2) d JOIN customer ON d.customer_id=customer.customer_id;
```

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

```
SELECT e.first_name, e.last_name, actor.first_name, actor.last_name FROM (SELECT * FROM (SELECT aactor, bactor FROM (SELECT a.actor_id AS aactor, b.actor_id AS bactor, COUNT(*) AS liczba FROM film_actor a JOIN film_actor b ON a.film_id=b.film_id AND a.actor_id>b.actor_id GROUP BY a.actor_id, b.actor_id) c WHERE liczba>1) d JOIN actor ON aactor=actor.actor_id) e JOIN actor ON bactor=actor.actor_id;
```

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

```
SELECT last_name FROM (SELECT d.actor_id FROM (SELECT actor_id FROM film_actor a JOIN film b ON a.film_id=b.film_id WHERE title LIKE 'B%' GROUP BY actor_id) c RIGHT JOIN actor d ON c.actor_id=d.actor_id WHERE c.actor_id IS NULL) e JOIN actor ON e.actor_id=actor.actor_id;
```

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

```
CREATE VIEW horrorfilm AS SELECT actor_id, COUNT(*) AS horrorfilm FROM (SELECT film_id FROM film_category a JOIN category b ON a.category_id=b.category_id WHERE name LIKE 'Horror') c JOIN film_actor ON c.film_id=film_actor.film_id GROUP BY actor_id;
```

```
CREATE VIEW actionfilm AS SELECT actor_id, COUNT(*) AS actionfilm FROM (SELECT film_id FROM film_category a JOIN category b ON a.category_id=b.category_id WHERE name LIKE 'Action') c JOIN film_actor ON c.film_id=film_actor.film_id GROUP BY actor_id;
```

```
SELECT last_name FROM (SELECT a.actor_id FROM actionfilm a JOIN horrorfilm b ON a.actor_id=b.actor_id WHERE horrorfilm>actionfilm) c JOIN actor ON c.actor_id=actor.actor_id;
```

15. Wypisz wszystkich klientów, których średnia opłata za wypożyczony film jest wyższa niż średnia opłata dokonana 7 lipca 2005.

```
SELECT first_name, last_name FROM (SELECT customer_id FROM (SELECT customer_id, AVG(amount) sredniaklienta FROM payment GROUP BY customer_id) a WHERE sredniaklienta>(SELECT AVG(amount) AS srednia FROM payment WHERE payment_date LIKE '2005-07-07%')) b JOIN customer c ON b.customer_id=c.customer_id;
```

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

```
ALTER TABLE language ADD films_no int AFTER name;
```

```
UPDATE language SET film_no=(SELECT liczba FROM (SELECT language_id, COUNT(*) AS liczba FROM film GROUP BY language_id) a WHERE language.language_id=language_id);
```

```
UPDATE language SET film_no=IF(film_no IS NULL, 0, film_no);
```

17. Zmień język filmu WON DARES na mandaryński oraz wszystkich filmów, w których występuje NICK WAHLBERG na niemiecki. Zaktualizuj dane w tabeli language.

a.

```
UPDATE film SET language_id=(SELECT language_id FROM language WHERE name LIKE 'Mandarin') WHERE title LIKE 'WON DARES';
```

b.

```
UPDATE film SET language_id=(SELECT language_id FROM language WHERE name LIKE 'German') WHERE film_id IN (SELECT film_id FROM (SELECT actor_id FROM actor WHERE first_name LIKE 'NICK' and last_name LIKE 'WAHLBERG') a JOIN film_actor b ON a.actor_id=b.actor_id);
```

c.

```
UPDATE language SET film_no=(SELECT liczba FROM (SELECT language_id, COUNT(*) AS liczba FROM film GROUP BY language_id) a WHERE language.language_id=language_id);
```

```
UPDATE language SET film_no=IF(film_no IS NULL, 0, film_no);
```

18. W tabeli film wszystkie wiersze mają identyczną wartość w kolumnie release year przez co jej wartość informacyjna jest niewielka. Usuń kolumnę z tabeli.

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