

```

/*    Užduotis 1;
    Naudoti: sakila;
    Pateikite filmo pavadinimus, nuomos trukmes, nuomos kainas, kai
    filmo nuomos kaina yra ne mažiau nei 4.99 o nuomos trukmė nėra lygu šešiem;
*/
USE sakila;
SELECT
    title AS 'Filmo pavadinimas',
    rental_duration AS 'Nuomos trukmė',
    rental_rate AS 'Nuomos kaina'
FROM film
    WHERE rental_rate >= 4.99 AND rental_duration <> 6;

/*    Užduotis 2;
    Naudoti: sakila;
    Paskaičiuokite, kiek nuomai išleido klientas, kurio ID yra 15.
    Nuomai išleistą sumą pateikite stulpelyje „Išleido“. */
USE sakila;

SELECT
    customer_id AS 'Kliento ID',
    SUM(amount) AS 'Išleista suma ($)'
FROM payment
    WHERE customer_id = 15;

/*    Užduotis 3;
    Naudoti: sql_hr;

    Parašykite SQL užklausą, kuri suskaičiuotų visų vadovų (skyrius nesvarbu)
    algų sumą
    Stulpelį pavadinkite `sum_salary`; */

USE sql_hr;

SELECT
    SUM(salary) AS sum_salary
FROM employees
    WHERE job_title LIKE '%vadovas%' OR job_title = 'Direktorius';

/*    Užduotis 4;
    Naudoti: sql_store;
    Parašykite SQL užklausą, kuri ištrauktų visus klientus, kurių miestas yra
    Vilnius, Klaipėda ir Alytus,
    o lojalumo taškų turi surinkęs mažiau nei 1000.
    Išrikiuoti rezultatus pagal lojalumo taškus didėjančia tvarka; */
USE sql_store;

SELECT
    first_name AS Vardas,
    last_name AS Pavardė,
    city AS Miestas,
    points AS 'Lojalumo taškai'
FROM customers
    WHERE (city = 'Vilnius' or city = 'Klaipėda' or city = 'Alytus') AND points <
1000
ORDER BY points;

/*    Užduotis 5;
    Parašykite SELECT užklausą, kuri atvaizduotų:

```

Jūsų vardą, kaip reikšmę, stulpelyje pavadinimu 'vardas'  
Jūsų pavardę, kaip reikšmę, stulpelyje pavadinimu 'pavarde'  
stulpelį 'Surinkau taškų' su taškų skaičiumi, kurį manote jog surinkote  
spręsdami šį testą; \*/

```
SELECT
    "Mantas" as Vardas,
    "Kreivėnas" as Pavardė,
    '100 :D' AS 'Surinkau taškų';

/*    Užduotis 6;
    Naudoti: sakila
    Pateikti kiekvienos parduojuvės TOP5 žanrus atnešančius didžiausias pajamas;
*/
```

USE sakila;

```
(SELECT
    inventory.store_id as 'Parduojuvės ID',
    category.name AS 'Žanras',
    SUM(payment.amount) AS 'Pajamos'
FROM category
    JOIN film_category
    ON film_category.category_id=category.category_id
    JOIN film
    ON film_category.film_id = film.film_id
    JOIN inventory
    ON film.film_id = inventory.film_id
    JOIN rental
    ON rental.inventory_id=inventory.inventory_id
    RIGHT JOIN payment
    ON payment.rental_id=rental.rental_id
    WHERE inventory.store_id =1
GROUP BY category.name
ORDER BY SUM(payment.amount) DESC
LIMIT 5)
UNION
(SELECT
    inventory.store_id as 'Parduojuvės ID',
    category.name AS 'Žanras',
    SUM(payment.amount) AS 'Pajamos'
FROM category
    JOIN film_category
    ON film_category.category_id=category.category_id
    JOIN film
    ON film_category.film_id = film.film_id
    JOIN inventory
    ON film.film_id = inventory.film_id
    JOIN rental
    ON rental.inventory_id=inventory.inventory_id
    RIGHT JOIN payment
    ON payment.rental_id=rental.rental_id
    WHERE inventory.store_id =2
GROUP BY category.name
ORDER BY SUM(payment.amount) DESC
LIMIT 5
);
```

```
/*    Užduotis 7;
```

Naudoti: sakila  
Kurio žanro filmai yra dažniausiai išnuomojami. Pateikti žanro pavadinimą ir išnuomavimo kiekį; \*/

USE sakila;

```
SELECT
    category.name AS 'Filmo žanras',
    COUNT(payment.payment_id) AS 'Nuomų skaičius'
FROM category
    JOIN film_category
    ON film_category.category_id=category.category_id
    JOIN film
    ON film_category.film_id = film.film_id
    JOIN inventory
    ON film.film_id = inventory.film_id
    JOIN rental
    ON rental.inventory_id=inventory.inventory_id
    RIGHT JOIN payment
    ON payment.rental_id=rental.rental_id
GROUP BY category.name
HAVING COUNT(payment.payment_id)=
    (SELECT MAX(ggg)
    FROM (
        SELECT category.name,
        COUNT(payment.payment_id) AS ggg
        FROM category
            JOIN film_category
            ON film_category.category_id=category.category_id
            JOIN film
            ON film_category.film_id = film.film_id
            JOIN inventory
            ON film.film_id = inventory.film_id
            JOIN rental
            ON rental.inventory_id=inventory.inventory_id
            RIGHT JOIN payment
            ON payment.rental_id=rental.rental_id
        GROUP BY category.name)x);
```

```
/*SELECT
    category.name AS 'Filmo žanras',
    COUNT(payment.payment_id) AS 'Nuomų kiekis'
FROM category
    JOIN film_category
    ON film_category.category_id=category.category_id
    JOIN film
    ON film_category.film_id = film.film_id
    JOIN inventory
    ON film.film_id = inventory.film_id
    JOIN rental
    ON rental.inventory_id=inventory.inventory_id
    RIGHT JOIN payment
    ON payment.rental_id=rental.rental_id
GROUP BY category.name
ORDER BY COUNT(payment.payment_id) DESC
LIMIT 1*/
```

/\* Užduotis 8;  
Naudoti: sakila,  
Pateikite klientų vardus ir pavardes, kurie išsinuomavo filmą 'AGENT TRUMAN'.

```
Išrikiuokite rezultatą pagal kliento vardą abėcėlės tvarka;*/  
USE sakila;
```

```
SELECT  
    customer.first_name AS 'Vardas',  
    customer.last_name AS 'Pavardė',  
    film.title AS 'Filmas'  
FROM customer  
    JOIN rental  
    ON customer.customer_id=rental.customer_id  
    JOIN inventory  
    ON rental.inventory_id=inventory.inventory_id  
    JOIN film  
    ON inventory.film_id=film.film_id  
    WHERE film.title = 'AGENT TRUMAN'  
ORDER BY first_name;
```

```
/*    Uždutis 9;  
    Naudoti: sakila,  
    Kokie aktoriai turi tokius pat vardus, kaip ir klientai? Pateikti aktorių  
vardus ir pavardes.*/
```

```
USE sakila;
```

```
SELECT  
    actor.first_name AS 'Aktoriaus Vardas',  
    actor.last_name AS 'Aktoriaus Pavardė'  
FROM customer, actor  
    WHERE customer.first_name=actor.first_name  
ORDER BY actor.first_name;
```

```
/*    Uždutis 10;  
    Naudoti: sakila,  
    Sukurti klientų sąrašą, kurie vėluoja grąžinti filmus.  
    Parodyti asmenis, filmus ir datą, kurią turėjo grąžinti;  
    Pagalba #1: naudoti 'INTERVAL' funkcija skaičiuojant datą, kurią turėjo  
sugrąžinti;  
    Pagalba #2: filmas negražintas, kai nuomos įrašė return_date reikšmės nėra;*/
```

```
USE sakila;
```

```
SELECT  
    customer.first_name AS 'Vardas',  
    customer.last_name AS 'Pavardė',  
    film.title AS 'Negražinto filmo pavadinimas',  
    rental_date AS 'Nuomos data',  
    rental.rental_date + INTERVAL film.rental_duration DAY AS 'Pradelsto  
grąžinimo terminas'  
FROM rental  
    JOIN customer  
    ON customer.customer_id=rental.customer_id  
    JOIN inventory  
    ON rental.inventory_id=inventory.inventory_id  
    JOIN film  
    ON inventory.film_id=film.film_id  
    JOIN payment  
    ON payment.rental_id=rental.rental_id  
    WHERE rental.return_date IS NULL  
    ORDER BY customer.first_name;
```

