

SQL Baigiamojo darbo užduotys

1) Parašykite SQL užklausą, kuri pateiktų filmų pavadinimus, jų nuomos kainą bei nuomos laikotarpio trukmę, kai nuomos kaina yra 4.99 ir nuomos trukmė yra 6. Naudokite lentelę "film".

Kodas:

```
select title, rental_rate, rental_duration from film  
where rental_rate = 4.99 and rental_duration = 6;
```

2) Parašykite SQL užklausą, kuri pateikia kiekvieno kliento_id ir jo mažiausią mokėjimą stulpelyje "Minimaliausias mokėjimas" didėjančia tvarka. Naudokite lentelę "payment".

Kodas:

```
select customer_id, min(amount) as `Minimaliausias mokėjimas` from payment  
group by customer_id  
order by `Minimaliausias mokėjimas` asc;
```

3) Pateikite klientų vardus ir pavardes, gyvenančius M raide prasidedančiuose miestuose. Naudokite lenteles "customer", "address", "city".

Kodas:

```
select first_name as Vardas, last_name as Pavarde, city as Miestas from customer as T1  
join address as T2 using (address_id)  
join city as T3 using (city_id)  
where city like 'M%';
```

4) Parašykite SQL užklausą, pateikiančią klientų ID, mokėjimo datą ir mažiausią kiekvieno kliento mokėjimą, bet tik tų klientų, kurių mažiausias mokėjimas per dieną yra 6.99. Mokėjimo datą pateikite formatu YYYY-MM-DD stulpelyje "Data", o mažiausią mokėjimą – stulpelyje "Minimalus mokestis". Naudokite lentelę "payment".

Kodas:

```
select customer_id, date(payment_date) as `Data`, min(amount) as `Minimalius mokestis` from payment  
group by customer_id, `Data`  
having `Minimalius mokestis` in (6.99);
```

5) Parašykite SQL užklausą, pateikiančią ilgiausiai trunkančių filmų pavadinimus. Naudokite lentelę „Film“.

Kodas:

```
select title, length from film  
where length = (select max(length) from film);
```

6) Parašykite SQL užklausą, pateikiančią klientų vardus, pavardes, jų iš viso nuomai išleidžiamą sumą (stulpelyje „Iš viso“), o stulpelyje „Rėžiai“ pateikite suskirstytus klientus tokiu būdu: klientus, kurie iš viso nuomai išleidžia 100 ir daugiau, pažymėkite kaip „Virš 100“, o išleidžiančius iki 100 pažymėkite „Iki 100“. Naudokite lenteles „customer“ ir „payment“.

Kodas:

```
select first_name, last_name, sum(amount) as 'Iš viso',  
CASE  
WHEN sum(amount) >= 100 then 'Virš 100'  
WHEN sum(amount) < 100 then 'Iki 100'  
else ''  
end 'Rėžiai'  
from customer  
join payment using (customer_id)  
group by first_name, last_name;
```

7) Parašykite SQL užklausą, pateikiančią klientų vardus, pavardes ir vidutinius mokėjimus (Stulpelyje „Average“), bet tik tų klientų, kurių vidutiniškai atliktų mokėjimų vertė yra tarp 3 ir 5. Rezultatą pateikite naujoje lentelėje, pavadinimu „Average_payment“. Naudokite lentelę „payment“ ir „customer“.

Kodas:

```
select first_name, last_name, avg(amount) as 'Average' from customer
join payment using (customer_id)
group by first_name, last_name, 'Average'
having avg(amount) >= 3 and avg(amount) <= 5;

-- susikuriu lentele ir joje pateikiu rezultata --

create table Average_payment

select first_name, last_name, avg(amount) as 'Average' from customer
join payment using (customer_id)
group by first_name, last_name, 'Average'
having avg(amount) >= 3 and avg(amount) <= 5;

-- turinio patikrinimui sukurtos lentelės--:

select * from Average_payment;
```

8) Parašykite SQL užklausą, pakeičiančią lentelės „Average_payment“ stulpelį „Average“ į „Vidutinis_mokėjimas“, o šio stulpelio reikšmes atvaizduokite dviem skaičiais po kablelio.

Naudokite duomenų tipą DECIMAL(n, d), kur n=5, d=2.

Kodas:

```
select first_name, last_name, cast(avg(Average) as decimal(5,2)) as 'Vidutinis_mokėjimas' from
Average_payment

group by first_name, last_name, 'Vidutinis_mokėjimas'

having Vidutinis_mokėjimas >= 3 and Vidutinis_mokėjimas <= 5;
```

9) Parašykite SQL užklausą, pateikiančią aktorių vardus, pavardes ir filmų pavadinimus, kuriuose jie vaidino. Rezultatą pateikite atskiriant brūkšniu viename stulpelyje, pavadinimu „Aktoriai ir filmai“.

Naudokite lenteles „actor“, „film_actor“, „film“.

Kodas:

```
select concat(first_name, '-', last_name, '-', title) as `Aktoriai ir filmai` from actor as L1  
join film_actor as L2 using (actor_id)  
join film as L3 using (film_id);
```

10) Parašykite SQL užklausą, pateikiančią klientų vardus ir pavardes, jų atliktus mokėjimus, kuriuos suskirstykite stulpelyje „Mokesčio tipas“ į tokias grupes:

-- Jei mokėjimas minimalus tai priskirkite grupei „Minimalus“

-- Jei mokėjimas yra maksimalus tai grupė – „Maksimalus“

-- Visus kitus mokėjimus priskirkite grupei „Kita“.

-- Rezultatą surūšiuokite pagal mokėjimus didėjimo tvarka.

-- Naudokite lenteles „customer“, „payment“.

Kodas:

```
select first_name, last_name, amount,  
CASE  
WHEN amount = 0.00 then 'Minimalus'  
WHEN amount = 11.99 then 'Maksimalus'  
else 'Kita'  
end 'Mokesčio tipas'  
from customer  
join payment using (customer_id)  
group by first_name, last_name, amount, 'Mokesčio tipas'  
order by amount asc;
```

11) Funkcija length() grąžina elementų skaičių įrašė. Tai yra length('Labas') grąžins 5, nes žodyje 'Labas' yra 5ki simboliai.

-- Išveskite visus aktorių vardus, kurie yra trumpesni nei 5ki simboliai.

Kodas:

```
select first_name as Vardai from actor
```

```
where length(first_name) < 5;
```

-- Išveskite aktorių vardus ir naują stulpelį, kuriame būtų toks tekstas:

-- • "5ki ir daugiau", jei vardas yra iš 5kių ir daugiau simbolių

-- • "Mažiau 5kių simbolių", jei vardas turi mažiau 5kių simbolių.

Kodas:

```
select first_name as Vardai,
```

```
case
```

```
when length(first_name) >= 5 then '5ki ir daugiau'
```

```
else 'Mažiau 5kių simbolių'
```

```
end as Simboliu_kiekis_vnt
```

```
from actor;
```

-- Suskaičiuokite, kiek vardų yra iš 5kių ir daugiau simbolių ir kiek vardų turi

-- mažiau, nei 5kis simbolius.

Kodas:

```
select count(first_name) as Vardu_kiekis_vnt,
```

```
case
```

```
when length(first_name) >= 5 then '5ki ir daugiau'
```

```
else 'Mažiau 5kių simbolių'
```

```
end as Simboliu_kiekis_vnt
```

```
from actor
```

```
group by Simboliu_kiekis_vnt;
```

12)

-- A.1. Suraskite, kiek vidutiniškai trukdavo filmui, priklausomai nuo reitingo. (lenta `film`)

Kodas:

```
select avg(length) as vid_trukmes_laikas, rating as Reitingas from film  
group by rating;
```

-- A.2. Suraskite vidutinį nuomos laiką filmams pagal reitingą (rating).

Kodas:

```
select avg(rental_duration) as vid_nuomos_laikas, rating as Reitingas from film  
group by rating;
```

-- A.3. Suraskite vidutinę nuomos kainą filmams pagal reitingą.

Kodas:

```
select avg(rental_rate) as vid_nuomos_kaina, rating Reitingas from film  
group by rating;
```

-- B. Išvesti aktorių vardus, pavardes viename stulpelyje. Vardai turi būti mažosiomis raidėmis, pavardės didžiosiomis.

Kodas:

```
select concat(lower(first_name), " ", upper(last_name)) from actor;
```

-- C. Išvesti aktorių vardus, prasidedančius raidėmis 'A', 'B', 'E'.

```
select first_name as Vardas from actor  
where first_name like 'A%' or first_name like 'B%' or first_name like 'E%';
```

-- Suskaičiuoti, kiek vardų prasideda raidėmis 'D', 'E', 'W'. Čia turima omeny, jog jei turime

-- Dan, Ed, Wolf, tai atsakymas bus 3.

Kodas:

```
select count(first_name) as Viso_vardu from actor  
where first_name like 'D%' or first_name like 'E%' or first_name like 'W%';
```