Darbas su duomenimis

3. Užklausos

Select užklausa

```
select
    column_name,
    aggregration_function(column_name) as new_column
from
   table_name
where
   column_name = condition_1
group by
   column_name
having
   new_column = condition_2
order by
   new_column asc
```

Kas sudaro select užklausą?

- Select – aprašo, kurie stulpeliai bus įtraukti į grąžintą lentelę.
- From – nusako kurių lentelių duomenis naudosime ir kaip tas lenteles sujungti.
- Where – išfiltruoja nereikalingus duomenis.
- Group by – naudojama sugrupuoti eilutes naudojantis bendra stulpelio reikšme.
- Having – išfiltruoja nereikalingas grupes.
- Order by – surikiuoja gražintos lentelės eilutes pagal vieno ar kelių stulpelių reikšmes.

Select užklausa

Select užklausa vykdoma paskutinė – nes norint pateikti stulpelius, reikia žinoti iš kokių stulpelių galima rinktis.

Select užklausos stulpeliuose galima naudoti

1. Konstantas
2. Standartines aritmetines operacijas – stulpelio_vardas * -1
3. Duomenų bazės užklausas – ROUND(stulpelio_vardas ,2)
4. Vartotojo apibrėžtas funkcijas

```
SELECT language_id,
'COMMON' language_usage,
language_id * 3.1415927 lang_pi_value,
upper(name) language_name
FROM language;
```

Select užklausa

- Select užklausoje galima nurodyti naujus stulpeliu vardus (aliases).

```
SELECT language_id,
'COMMON' language_usage,
language_id * 3.1415927  as lang_pi_value,
upper(name) language_name
FROM language;
```

- Duplikatų panaikinimas

```
SELECT DISTINCT actor_id FROM film_actor ORDER BY actor_id;
```

Norėdamas grąžinti distinct eilutes, serveris jas turi surikiuoti (sort), kas gali būti brangi operacija, jeigu turite daug eilučių. Sort algoritmo sudėtingumo vidurkis yra O(n*log(n)).

Komanda From

From – nusako kurių lentelių duomenis naudosime ir kaip tas lenteles sujungti.

Kokios gali būti lentelės:

1. Permanent. Pastovios lentelės, jas sukuriame su create table comanda.
2. Derived. Lentelės, kurias gražina (subquery) užklausa užklausoje.
3. Temporary. Laikinos lentelės, kurios saugomos atmintyje.
4. Views. Virtualios lentelės, kurios sukuriamos naudojantis užklausa create view.

Derived

```
 SELECT concat(cust.last_name, ', ', cust.first_name) full_name
FROM
(SELECT first_name, last_name, email
FROM customer
WHERE first_name = 'JESSIE'
) cust;
```

Mėlyna spalva yra pažymėta subquery.

Temporary. Laikinos lentelės

```
CREATE TEMPORARY TABLE actors_j
(actor_id smallint(5),
first_name varchar(45),
last_name varchar(45)
);
INSERT INTO actors_j
SELECT actor_id, first_name, last_name
FROM actor
WHERE last_name LIKE 'J%';
```

Laikinos lentelės egzistuoja tol, kol trunka sesija.

Views

```
CREATE VIEW cust_vw AS
    SELECT customer_id, first_name, last_name, active
    FROM customer;
```

Views padeda sukurti dinaminę ataskaitą arba paslėpti stulpelius, kurių negalima rodyti vartotojams.

Lentelių sujungimas

```
SELECT customer.first_name, customer.last_name,
    time(rental.rental_date) rental_time
    FROM customer
     INNER JOIN rental
     ON customer.customer_id = rental.customer_id
    WHERE date(rental.rental_date) = '2005-06-14';
```

Lentelių nuorodos/vardai - aliases

```
SELECT c.first_name, c.last_name,
  time(r.rental_date) rental_time
FROM customer c
  INNER JOIN rental AS r
  ON c.customer_id = r.customer_id
WHERE date(r.rental_date) = '2005-06-14';
```

Komanda Where

Komanda where padeda išfiltruoti nereikalingas eilutes.

```
SELECT title
    FROM film
    WHERE rating = 'G' AND rental_duration >= 7;
```

Sąlygos yra boolean tipas ir jos turi būti True arba False.

Sąlygų jungimui galioja logikos operacijos AND ir OR.

Komanda Where

```
SELECT title, rating, rental_duration
    FROM film
    WHERE (rating = 'G' AND rental_duration >= 7)
      OR (rating = 'PG-13' AND rental_duration < 4);
```

Group By ir Having

```
SELECT c.first_name, c.last_name, count(*)
    FROM customer c
     INNER JOIN rental r
     ON c.customer_id = r.customer_id
    GROUP BY c.first_name, c.last_name
    HAVING count(*) >= 40;
```

Order by

```
SELECT c.first_name, c.last_name,
    time(r.rental_date) rental_time
    FROM customer c
    INNER JOIN rental r
    ON c.customer_id = r.customer_id
    WHERE date(r.rental_date) = '2005-06-14'
    ORDER BY c.last_name;
```

Rikavimo kontroliavimui galime naudoti komandas asc ir desc. Ascending rikiavimas yra numatytasis.

Order by

```
SELECT c.first_name, c.last_name,
     time(r.rental_date) rental_time
   FROM customer c
    INNER JOIN rental r
    ON c.customer_id = r.customer_id
   WHERE date(r.rental_date) = '2005-06-14'
   ORDER BY 3 desc;
```

Galima rikuoti nurodants stulpelio eilę select užklausoje. Reikia atkreipti dėmesį įtraukiant papildomus stulpelius į select užklausą, nes gauti rezultatai gali nustebinti dėl pasikeitusios stulpelių numeracijos.

Uždaviniai

1. Gražinkite lentelę su aktoriaus id, aktoriaus vardu, aktoriaus pavarde. Surikiuokite lentelę pagal aktoriaus vardą.
2. Gražinkite lentelę su aktoriaus id, aktoriaus vardu, aktoriaus pavarde, kur aktorių pavrdės yra lygios WILLIAMS ir DAVIS.
3. Gražinkite lentelę su klientų Ids, kurie išsinuomojo filmą 2005-07-05. Padarykite, kad klientų Id nesikartotų.

Uždaviniai.

Užpildykite <?> vietas, kad gautumėte tokią pačią lentelę kaip atsakyme.

```
mysql> SELECT c.email, r.return_date
    -> FROM customer c
    ->   INNER JOIN rental <1>
    ->   ON c.customer_id = <2>
    -> WHERE date(r.rental_date) = '2005-06-14'
    -> ORDER BY <3> <4>;
+--------------------------------------+---------------------+
| email                                | return_date         |
+--------------------------------------+---------------------+
| DANIEL.CABRAL@sakilacustomer.org     | 2005-06-23 22:00:38 |
| TERRANCE.ROUSH@sakilacustomer.org    | 2005-06-23 21:53:46 |
| MIRIAM.MCKINNEY@sakilacustomer.org   | 2005-06-21 17:12:08 |
| GWENDOLYN.MAY@sakilacustomer.org     | 2005-06-20 02:40:27 |
| JEANETTE.GREENE@sakilacustomer.org   | 2005-06-19 23:26:46 |
| HERMAN.DEVORE@sakilacustomer.org     | 2005-06-19 03:20:09 |
| JEFFERY.PINSON@sakilacustomer.org    | 2005-06-18 21:37:33 |
| MATTHEW.MAHAN@sakilacustomer.org     | 2005-06-18 05:18:58 |
| MINNIE.ROMERO@sakilacustomer.org     | 2005-06-18 01:58:34 |
| SONIA.GREGORY@sakilacustomer.org     | 2005-06-17 21:44:11 |
| TERRENCE.GUNDERSON@sakilacustomer.org | 2005-06-17 05:28:35 |
| ELMER.NOE@sakilacustomer.org         | 2005-06-17 02:11:13 |
| JOYCE.EDWARDS@sakilacustomer.org     | 2005-06-16 21:00:26 |
| AMBER.DIXON@sakilacustomer.org       | 2005-06-16 04:02:56 |
| CHARLES.KOWALSKI@sakilacustomer.org  | 2005-06-16 02:26:34 |
| CATHERINE.CAMPBELL@sakilacustomer.org | 2005-06-15 20:43:03 |
+--------------------------------------+---------------------+
16 rows in set (0.03 sec)
```