

Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний
інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки Кафедра інформатики
та програмної інженерії

Звіт

з лабораторної роботи № 4 з дисципліни
«Бази даних»

«Створення запитів на групування, сортування, використання
вбудованих функцій. Створення та керування представленнями»
Варіант 8

Виконав студент ІП-13 Дойчев Костянтин Миколайович
(шифр, прізвище, ім'я, по батькові)

Перевірила Марченко Олена Іванівна
(прізвище, ім'я, по батькові)

Лабараторна робота №4

Створення запитів на групування, сортування, використання вбудованих функцій. Створення та керування представленнями.

Постановка задачі лабораторної роботи № 4

При виконанні лабораторної роботи необхідно виконати наступні дії:

1) Створити наступні запити:

- a. запит з використанням функції COUNT;
- b. запит з використанням функції SUM;
- c. запит з використанням функцій UPPER, LOWER;
- d. запит з використанням функцій для роботи з датами;
- e. запит з використанням групування по декільком стовпцям;
- f. запит з використанням умови відбору груп HAVING;
- g. запит з використанням HAVING без GROUP BY;
- h. запит з використанням функцій row_number() over;
- i. запит з використанням сортування по декільком стовпцям.

2) Робота з представленнями (view):

- a. створити представлення, котре містить дані з декількох таблиць;
- b. створити представлення, котре містить дані з декількох таблиць та використовує представлення, котре створене в п.а;
- c. модифікувати представлення з використанням команди ALTER VIEW;
- d. отримати довідникову інформацію про ці представлення з використанням вбудованих процедур (наприклад в MySQL sp_help, sp_helptext та sp_depends).

Індивідуальне завдання:

Програмне забезпечення «Школа». Загальноосвітня школа, в якій навчаються учні, має номер, назву, адресу, ПІБ директора. У школах є певна кількість класів, котрі мають назву, класного керівника, список учнів, певний перелік предметів. Предмети викладаються вчителями, причому один вчитель може викладати декілька предметів, а однакові предмети можуть викладати різні вчителі. Предмети викладаються згідно з розкладом у кабінетах, котрі мають номер, назву, відповідне обладнання та розкладом

класів. Предмети мають назву, кількість годин вивчення, список навчальних посібників.

SQL скрипти

Усі скрипти можна знайти у [репозиторії](#) на платформі GitHub.

```
-- Task 1

-- a
select count(id) as "Number of books"
from "book"

-- b
select sum(credits) as "Total credits"
from "subject"

-- c upper
select upper(first_name) as "Uppercase name"
from "user"

-- c lower
select lower(first_name) as "Lowercase name"
from "user"

-- d
select name, (extract('year' from now()::date) - published_at)
as "age"
from book;

-- e
select "class"."name", count("role".id)
from "role"
inner join "class" on "class".id = "role".class_id
group by "class"."name"
order by "class"."name";
```

```

-- f
select "class"."name", count("role".id)
from "role"
inner join "class" on "class".id = "role".class_id
group by "class"."name"
having count("role".id) > 5
order by "class"."name";

-- g
select sum(credits)
from subject
having sum(credits) > 10

-- h
-- order students by their full names in a class
select "class"."name",
       concat("u".first_name, ' ', "u".last_name) as
"full_name",
       row_number() over (
           partition by "class".name order by
concat("u".first_name, ' ', "u".last_name)
       ) as "student position"
from "role"
inner join "class" on "class".id = "role".class_id
inner join "user" u on u.id = role.user_id;

-- i
select "book".name, "book".published_at,
concat("author"."first_name", ' ', "author"."last_name") as
author_name
from "book"
inner join "author" on book.author_id = author.id
order by "book".published_at desc

```

```
-- Task 2
```

```
-- a
```

```
create or replace view role_with_teacher as
select "role".id, "user".first_name, "user".last_name,
"teacher".id as "class_teacher_id"
from "role"
    inner join "user" on "user".id = "role".user_id
    left join "role" as "teacher"
        on ("teacher"."class_id" = "role"."class_id"
and "teacher"."type" = 'teacher');
```

```
select concat("roles".first_name, ' ', roles.last_name) as
"Role", concat(r.first_name, ' ', r.last_name) as "Teacher"
from role_with_teacher as "roles"
```

```
inner join role_with_teacher as "r"
```

```
on "roles".class_teacher_id = "r".id;
```

```
-- b
```

```
create or replace view class_teacher as
select "class".name, concat("user".first_name, ' ',
"user".last_name) as "Teacher"
from "class"
    inner join "role" on "role".class_id = "class".id
    inner join "role_with_teacher" on "role".id =
"role_with_teacher".class_teacher_id
    inner join "user" on "user".id = "role".user_id
where "role".type = 'teacher'
order by name;
```

```
-- c
```

```
create or replace view students as
select u.first_name, u.last_name
from "user" as "u"
```

```
        inner join "role" as "r" on "u".id = "r".user_id
where r.type = 'student';

-- ALTER VIEW changes various auxiliary properties of a view.
-- (If you want to modify the view's defining query, use CREATE OR
-- REPLACE VIEW.)
-- https://www.postgresql.org/docs/current/sql-alterview.html
alter view students rename to all_students;

-- d
select * from pg_views where schemaname = 'public';
```