

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 № 3 з дисципліни
«Бази даних»

«Побудова простих запитів»
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 №3

Побудова простих запитів

Постановка задачі лабораторної роботи № 3

При виконанні комп'ютерного практикуму необхідно виконати наступні дії:

1) Створити запити для вибірки даних з використанням (разом 12 запитів):

- a. Найпростіших умов
- b. Операторів порівняння
- c. Умов з використанням логічних операторів AND, OR та NOT.
- d. Умов з використанням комбінацій логічних операторів
- e. З використанням виразів над стовпцями, як в якості новостворених стовпців, так і умовах
- f. Використання операторів:
 - i. Приналежності множині
 - ii. Приналежності діапазону
 - iii. Відповідності шаблону
 - iv. Перевірка на невизначене значення

2) Створити запити з використанням підзапитів та з'єднань (разом 11 запитів) (в запитах повинні використовуватись 2 та більше таблиць):

- a. Використання підзапитів в рядку вибірки полів (у секції select) та вибірки з таблиць (у секції from)
- b. Використання підзапитів в умовах з конструкціями EXISTS, IN
- c. Декартовий добуток
- d. З'єднання декількох таблиць за рівністю
- e. З'єднання декількох таблиць за рівністю та умовою відбору
- f. Внутрішнього з'єднання
- g. Лівого зовнішнього з'єднання
- h. Правого зовнішнього з'єднання
- i. Об'єднання запитів

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 concat(first_name, ' ', last_name) as "Full name"
from "user"
where first_name = 'Barbette';

-- b
select name, credits
from subject
where credits > 60;

-- c AND
select *
from address
where country = 'China' and house_number = '999';

-- c OR
select *
from address
where country = 'Morocco' or country = 'Brazil';

-- c NOT
select *
```

```
from address
where not country = 'China';

-- d
select *
from address
where (country = 'China' or country = 'Germany') and street =
'Trailsway';

-- e average
select avg(credits) as "Average credits"
from subject;

-- e modify column
select concat(first_name, ' ', last_name) as "Students"
from "user"
left join role on role.user_id = "user".id
where role.type = 'student';

-- f IN ()
select * from "role"
where class_id in ('96a42758-13b9-42fc-a307-f4e4a1b90a93',
'b643bfc9-a0f3-405b-ad67-3c9045d65f2a');

-- f between
select * from subject
where credits between 40 and 80;

-- f like/ilike
select *
from "user"
where first_name ilike '_arb%';

-- f in not null
select *
```

```

from address
where "zip_code" is not null;

-- Task 2

-- a column
select concat(first_name, ' ', last_name) as "Full name",
(select type from "role" where "role".user_id = "user".id ) as
"Roles"
from "user"
where "user".first_name ilike 'b%';

-- a from
select detailed_books.name, detailed_books.author_name
from (
    select book.name, concat(author.first_name, ' ',
author.last_name) as author_name
    from author, book
    where book.author_id = author.id
) as detailed_books
order by detailed_books.name;

-- b exists
-- select users that have a classroom attached to their class

select "user".first_name, "user".last_name
from "role"
inner join "user" on "user".id = "role".user_id
inner join "class" on "class".id = "role".class_id
where exists(
    select *
    from "classroom"
    where "class".classroom_id = "classroom".id

```

```

);

-- b in
-- users that belong to school with a specific id
select concat("user".first_name, ' ', "user".last_name) as "Full
name"
from "role"
inner join "user" on "user".id = "role".user_id
where class_id in (
    select id from "class"
    where "class".school_id =
'386852c4-2c2e-4097-ba86-4f24f03726c7'
);

-- c
-- doesn't really make sense, but it's required by the task
select concat("a"."first_name", ' ', "a"."last_name") as
"Author",
    "b"."name" as "Book"
from "author" "a"
cross join "book" "b"
order by "Author"

-- d
select concat("user"."first_name", ' ', "user"."last_name") as
"Full name",
    "role"."type" as "Role"
from "user"
join "role" on "role"."user_id" = "user"."id"
order by "Full name"
limit 10

-- e
select concat("user"."first_name", ' ', "user"."last_name") as
"Teachers",
    "role"."type" as "Role"

```

```

from "user"
join "role" on "role"."user_id" = "user"."id"
where "role".type = 'teacher'
order by "Teachers"
limit 10

-- f
-- get a user and its teacher

select concat("roles".first_name, ' ', roles.last_name) as
"Role", concat(r.first_name, ' ', r.last_name) as "Teacher"
-- create a virtual table with role id, name, and supervisor
(teacher) id
from (
    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')
) as "roles"

inner join (
    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')
) as "r"

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

```

```
-- g
select "role".id, coalesce("class".name, 'No Class') as "Class"
from "role"
left join "class" on "role"."class_id" = "class".id;

-- h
select "role".id as "role", "class".name
from "role"
right join "class" on "role"."class_id" = "class".id;

-- i
select "u".first_name, "u".last_name
from "user" "u"
union
select "a".first_name, "a".last_name
from "author" "a";
```