

*Додаток 1*

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 1

Виконав: ІП-13 Ал Хадам Мурат Резгович  
(шифр, прізвище, ім'я, по батькові)

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

## **Лабораторна робота 3** **Побудова простих запитів**

Мета:

- Вивчити оператор, котрий використовується в реляційних СУБД, для вибірки даних з таблиць.
- Вивчити команди SQL для створення запитів з використанням підзапитів та з'єднань.

### **Варіант 1**

#### **Постановка задачі.**

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

- 1) **Поліклініка.** У поліклініці працюють лікарі різних спеціальностей, що мають різну кваліфікацію. Щодня в поліклініку звертаються хворі. Усі вони проходять обов'язкову реєстрацію, при якій у базу даних заносяться стандартні анкетні дані (прізвище, ім'я, по батькові, рік народження). Кожен хворий може звертатися в поліклініку кілька разів, потребуючи різної медичної допомоги. Усі звернення хворих фіксуються, при цьому встановлюється діагноз, визначається вартість лікування, запам'ятовується дата звернення. Загальна вартість лікування залежить від вартості консультацій і процедур, які призначені пацієнтові. Треба врахувати, що за один візит пацієнт може відвідати різних лікарів. Крім того, для певних категорій громадян передбачені знижки.

### TASK1.SQL

```
use `db-labs`;

SELECT *
FROM patients;

SELECT *
FROM discounts
WHERE percentage > 10 ;

SELECT *
FROM patients
WHERE gender = 'чоловік' or is_beneficiary;

SELECT *
FROM patients
WHERE not gender = 'жінка';

SELECT *
FROM patients
WHERE gender = 'чоловік' and is_beneficiary;

SELECT *
FROM patients
WHERE gender = 'чоловік' AND NOT is_beneficiary
OR
gender = 'жінка' AND is_beneficiary;

SELECT medical_group
FROM cards
WHERE height > 170;

SELECT *
FROM cards
WHERE weight % 2 = 0;

SELECT height, weight
FROM cards
WHERE medical_group IN ('основна', 'спеціальна');

SELECT *
FROM patients
WHERE birthday BETWEEN DATE('2000-01-01') AND DATE('2009-12-31');

SELECT *
FROM patients
WHERE first_name LIKE 'М%';
```

```
SELECT *
FROM applications
WHERE possible_diagnosis IS NOT NULL;
```

## TASK2.SQL

```
use `db-labs`;

SELECT p.first_name,
(
    SELECT d.percentage
    FROM discounts d
    WHERE p.patient_id = d.patient_id
) AS discount_percentage
FROM patients AS p;

SELECT concat(cc.first_name, ' ', cc.last_name) AS fullname, cc.height
FROM (
    SELECT p.first_name, p.last_name, c.height
    FROM cards AS c,
    patients AS p
    WHERE p.patient_id = c.patient_id
) AS cc
ORDER BY cc.height DESC;

# patients with any beneficiaries
SELECT *
FROM patients AS p
WHERE EXISTS (
    SELECT *
    FROM discounts AS d
    WHERE d.patient_id = p.patient_id
);

# names with diagnosis
SELECT p.first_name
FROM patients AS p
WHERE p.patient_id IN (
    SELECT a.patient_id
    FROM conclusions AS c,
    applications AS a
    WHERE a.application_id = c.application_id
);

# possible combinations of possible and established diagnosis
SELECT possible_diagnosis, established_diagnosis
FROM applications
CROSS JOIN conclusions;

SELECT concat(p.first_name, ' ', p.last_name) AS fullname,
d.reason AS preferential_group
FROM patients AS p
JOIN discounts AS d ON p.patient_id = d.patient_id;

# patients with the not normal Body Mass Index
SELECT *
FROM (
    SELECT concat(p.first_name, ' ', p.last_name) AS fullname,
    c.height, c.weight,
    ROUND(c.weight / POW((c.height / 100), 2), 2) AS bmi
    FROM patients AS p
    JOIN cards AS c ON p.patient_id = c.patient_id
) AS bmi_stats
WHERE NOT bmi_stats.bmi BETWEEN 18.5 AND 24.9;
```

```
SELECT concat(a.first_name, ' ', a.last_name) AS admin_fullname,
        COUNT(app.service_type) AS processed_applications
FROM admins AS a
JOIN applications AS app ON a.admin_id = app.admin_id
GROUP BY admin_fullname;

SELECT concat(p.first_name, ' ', p.last_name) AS patient_fullname,
        concat(d.first_name, ' ', d.last_name) AS doctor_fullname,
        d.qualification, d.specialization, a.possible_diagnosis
FROM doctors AS d
LEFT OUTER JOIN applications AS a
    ON d.doctor_id = a.doctor_id
LEFT OUTER JOIN patients AS p
    ON p.patient_id = a.patient_id

SELECT concat(p.first_name, ' ', p.last_name) AS patient_fullname,
        c.established_diagnosis, c.treatment_method, c.sum_to_pay
FROM patients AS p
RIGHT OUTER JOIN applications AS a
    ON p.patient_id = a.patient_id
RIGHT OUTER JOIN conclusions AS c
    ON a.application_id = c.application_id;

SELECT concat(d.first_name, ' ', d.last_name) AS worker_fullname
FROM doctors AS d
UNION
SELECT concat(a.first_name, ' ', a.last_name)
FROM admins AS a
```

### Висновок

Отже в ході даної лабораторної роботи було вивчено оператор, який використовується в реляційних СУБД, для вибірки даних з таблиць. Було вивчено команди SQL для створення запитів з використанням підзапитів та різного типу з'єднань.