

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 10

Виконав студент ІП-13, Замковий Дмитро Володимирович
(шифр, прізвище, ім'я, по батькові)

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

Лабораторна робота 3

Побудова простих запитів

Мета заняття: Вивчити оператор, котрий використовується в реляційних СУБД, для вибірки даних з таблиць; Вивчити команди SQL для створення запитів з використанням підзапитів та з'єднань

Завдання:

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

Скрипт:

```
SELECT * FROM employee;

SELECT * FROM employee WHERE employee.tariff >= 450;

SELECT * FROM treaty WHERE treaty.date_conclusion < '2018-01-01' AND
treaty.type_insurance = 'Повне';

SELECT * FROM treaty WHERE NOT treaty.type_insurance = 'Повне';

SELECT * FROM treaty WHERE treaty.type_insurance = 'Повне' OR treaty.type_insurance
= 'Звичайне'

SELECT * FROM treaty WHERE (treaty.type_insurance = 'Повне' AND
treaty.date_conclusion > '2015-01-01') OR (NOT treaty.type_insurance = 'Звичайне'
AND treaty.date_conclusion < '2015-01-01');

SELECT employee.tariff/100 FROM employee WHERE employee.tariff > 200;

SELECT * FROM employee WHERE employee.tariff/100 > 2;

SELECT * FROM employee WHERE employee.tariff IN (100, 200, 300);

SELECT * FROM employee WHERE employee.tariff BETWEEN 200 AND 500;

SELECT * FROM employee WHERE employee.first_name LIKE 'Ж%';

SELECT * FROM employee WHERE employee.fathers_name IS NOT NULL;

SELECT e.first_name, e.tariff, t.type_insurance FROM employee e JOIN treaty t ON
e."id" = t.id_employee;

SELECT A.first_name, A.tariff, A.type_insurance FROM (SELECT e.first_name,
e.tariff, t.type_insurance, t.date_conclusion FROM employee e JOIN treaty t ON
e."id" = t.id_employee) AS A WHERE A.tariff > (SELECT "avg"(tariff) FROM employee);
```

```
SELECT * FROM treaty WHERE EXISTS(SELECT * FROM employee);

SELECT * FROM insurer WHERE insurer.id IN (SELECT treaty.id_insurer FROM treaty);

SELECT * FROM insurer CROSS JOIN employee;

SELECT i.first_name, t.id_object FROM insurer i JOIN treaty t ON i.id =
t.id_insurer;

SELECT i.first_name, i.phone_number, t.date_conclusion, t.type_insurance FROM
insurer i JOIN treaty t ON i."id" = t.id_insurer AND i.phone_number > (SELECT
"avg"(i.phone_number) FROM insurer i);

SELECT * FROM insurer i INNER JOIN treaty t ON i."id" = t.id_insurer;

SELECT * FROM insurer i LEFT JOIN treaty t ON i."id" = t.id_insurer;

SELECT * FROM insurer i RIGHT JOIN treaty t ON i."id" = t.id_insurer;

(SELECT * FROM insurer i LEFT JOIN treaty t ON i."id" = t.id_insurer) UNION (SELECT
* FROM insurer i RIGHT JOIN treaty t ON i."id" = t.id_insurer);
```

Висновок:

В ході даної лабораторної роботи я вивчити оператор, котрий використовується в реляційних СУБД, для вибірки даних з таблиць, також вивчив команди SQL для створення запитів з використанням підзапитів та з'єднань та виконав поставлене мені завдання.