МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”
ІНСТИТУТ КОМП’ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра систем штучного інтелекту

Лабораторна робота №7
із дисципліни
Бази даних

Виконав:
Ст. групи КН-207
Ометюх Р.Б.
Прийняв:
Мельникова Н.І.

Львів – 2018 р.

Мета роботи: Розробити SQL запити відбору даних з одиничних та з'єднаних таблиць, в тому числі з використанням підзапитів, натурального, умовного та лівого з'єднання, із застосуванням у критеріях вибірки функцій та операторів, в т. ч. LIKE, BETWEEN, IS NULL, IS NOT NULL, IN (…), NOT IN (…), ALL, SOME, ANY, EXISTS.

Короткі теоретичні відомості.

Для вибирання даних з таблиць використовується директива SELECT, яка може містити інші директиви SELECT (підзапити, або вкладені запити) та директиви з'єднання таблиць.

```
SELECT
        [ALL | DISTINCT | DISTINCTROW ]
        [STRAIGHT_JOIN]
        [SQL_CACHE | SQL_NO_CACHE] [SQL_CALC_FOUND_ROWS]
    елемент_вибірки [, елемент_вибірки ...]
    [FROM перелік_таблиць]
    [WHERE умова_відбору]
    [GROUP BY {ім'я_поля | синонім | позиція_поля}
        [ASC | DESC], ...]
    [HAVING умова_відбору]
    [ORDER BY {ім'я_поля | синонім | позиція_поля}
        [ASC | DESC], ...]
    [LIMIT {к-сть_рядків [OFFSET зміщення]}
    [PROCEDURE ім'я_процедури(аргументи)]
    [INTO OUTFILE 'ім'я_файлу' опції_експорту
        | INTO DUMPFILE 'ім'я_файлу'
        | INTO змінна [, змінна]]
```

У таблиці нижче описано основні функції порівняння, які можна використовувати при формуванні складних критеріїв вибору.

Функція	Опис
STRCMP(*рядок1*, *рядок2*)	Порівнює два рядки. Повертає значення 0 (False) якщо рядки однакові, -1 якщо перший рядок менший за другий, і 1 (True) в усіх інших випадках.
LIKE *рядок*	Порівняння з рядком-шаблоном. В шаблоні можна використовувати знаки % (довільні символи) і _ (довільний символ).
REGEXP *рядок*	Порівняння з рядком з використанням регулярних виразів. Функція-синонім – RLIKE.
MATCH (*поля*) AGAINST (*рядок*)	Здійснює пошук рядка у вказаних текстових полях таблиці. (Тільки для MyISAM-таблиць.)
BETWEEN ... AND ...	Повертає 1, якщо значення належить даному діапазону.
NOT BETWEEN ... AND ...	Повертає 1, якщо значення не належить діапазону.
IN(*арг1*, *арг2*, ...)	Перевірка належності множині. Повертає 1, якщо значення співпадає хоча б із одним аргументом, і 0 – у протилежному випадку. Повертає NULL, якщо значення є NULL, або якщо співпадіння не знайдено, а один із аргументів є NULL.
NOT IN(*арг1*, *арг2*, ...)	Повертає 1, якщо значення не міститься у множині аргументів, і 0 – у протилежному випадку. Повертає NULL аналогічно до функції IN().
IS NULL, IS NOT NULL	Перевірка визначеності значення.
LEAST(*арг1*, *арг2*, ...)	Повертає мінімальне значення серед аргументів. Повертає NULL, якщо хоча б один із аргументів є NULL.
GREATEST(*арг1*, *арг2*, ...)	Повертає максимальне значення серед аргументів. Повертає NULL, якщо хоча б один із аргументів є NULL.

Хід роботи.

1. Виконаємо ліве з’єднання.

```
select worker.idWorker, worker.Name, worker.Surname, worker.email,
office.Address, office.work_start, office.work_end
from worker left join office on
worker.office_idOffice = office.idOffice;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

idWorker	Name	Surname	email	Address	work_start	work_end
1	Stas	Romanko	stasikIviv@gmail.com	Puluj 21	09:00:00	18:00:00
2	Nastia	Zubritska	nastia.zubritska@gmail.com	Puluj 21	09:00:00	18:00:00
3	Bogdan	Gumenchuk	bogdan.gumenchuk@gmail.com	Kniazia Romana 3	12:00:00	20:00:00

2. Виконаємо умовне з’єднання двох таблиць.

```
select worker.Name, worker.surname, appointment.Appointment
from worker inner join appointment on worker.Appointment_idAppointment = Appointment.idAppointment
where Appointment.appointment = "Manager";
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Name	surname	Appointment
Nastia	Zubritska	Manager
Bogdan	Gumenchuk	Manager

3. Виконаємо умовне з’єднання трьох таблиць.

```
select worker.Name, worker.surname, appointment.Appointment, office.Address, office.work_start
from (worker inner join appointment) inner join office
on worker.Appointment_idAppointment = Appointment.idAppointment
and worker.office_idoffice = office.idoffice
where Appointment.appointment in ("Manager", "Admin");
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Name	surname	Appointment	Address	work_start
Stas	Romanko	Admin	Puluj 21	09:00:00
Nastia	Zubritska	Manager	Puluj 21	09:00:00
Bogdan	Gumenchuk	Manager	Kniazia Romana 3	12:00:00

4. Виконаємо під запит в умові відбору.

```
select worker.Name, worker.surname, office.Address
from worker inner join office
on worker.office_idoffice = office.idoffice
where worker.office_idOffice in (select office.idOffice from office
where office.Address in ("Puluj 21"))
order by worker.surname desc limit 2;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

Name	surname	Address
Nastia	Zubritska	Puluj 21
Stas	Romanko	Puluj 21

Висновок: на цій лабораторній роботі було розглянуто операції реляційної алгебри та їх реалізація в MySql.