# 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 Романко С.А. Прийняв: Мельникова Н.І. **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 \{im's\_nons \mid cuhohim \mid nosuqis\_nons\}
   [ASC | DESC], ...]
[HAVING умова відбору]
[ORDER BY \{im'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 'im'я файлу'
   | INTO змінна [, змінна]]
```

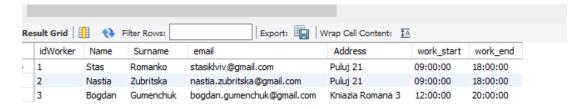
У таблиці нижче описано основні функції порівняння, які можна використовувати при формуванні складних критеріїв вибору.

| 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(apɛ1, apɛ2,)                      | Перевірка належності множині. Повертає 1, якщо значення співпадає хоча б із одним аргументом, і $0-y$ протилежному випадку. Повертає NULL, якщо значення є NULL, або якщо співпадіння не знайдено, а один із аргументів є NULL. |
| NOT IN( <i>apɛ1</i> , <i>apɛ2</i> ,) | 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(apz1, apz2,)                   | Повертає мінімальне значення серед аргументів. Повертає NULL, якщо хоча б один із аргументів є NULL.  |
| GREATEST(aprl, apr2,)                | Повертає максимальне значення серед аргументів. Повертає NULL, якщо хоча б один із аргументів є NULL.   |

### Хід роботи.

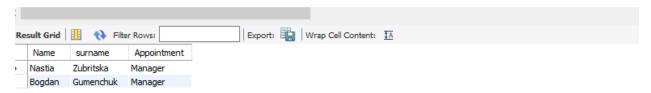
# 1. Виконаємо ліве з'єднання.

- 3 select worker.idWorker, worker.Name, worker.Surname, worker.email,
- 4 office.Address, office.work\_start, office.work\_end
- 5 from worker left join office on
- 6 worker.office\_idOffice = office.idOffice;



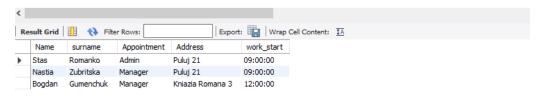
### 2. Виконаємо умовне з'єднання двох таблиць.

- 5 select worker.Name, worker.surname, appointment.Appointment
- 6 from worker inner join appointment on worker.Appointment\_idAppointment = Appointment.idAppointment
  - where Appointment.appointment = "Manager";



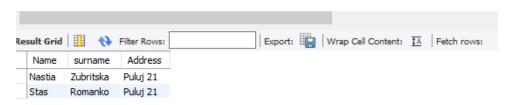
## 3. Виконаємо умовне з'єднання трьох таблиць.

- 5 select worker.Name, worker.surname, appointment.Appointment, office.Address, office.work\_start
- 6 from (worker inner join appointment) inner join office
- 7 on worker.Appointment\_idAppointment = Appointment.idAppointment
- 8 and worker.office\_idoffice = office.idoffice
- where Appointment.appointment in ("Manager", "Admin");



#### 4. Виконаємо під запит в умові відбору.

- 5 select worker.Name, worker.surname, office.Address
- 6 from worker inner join office
- 7 on worker.office\_idoffice = office.idoffice
- 8 ⊝ where worker.office idOffice in (select office.idOffice from office
- 9 where office.Address in ("Puluj 21"))
- 10 order by worker.surname desc limit 2;



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