

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ**  
**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**  
**Інститут комп'ютерних наук та інформаційних технологій**  
**Кафедра систем штучного інтелекту**



**Звіт до практичної роботи №8**  
**з дисципліни “ ОБДЗ ”**

**На тему: «Запити на вибір даних із впорядкуванням та групуванням результатів»**

**Виконав:**  
ст. гр. КН-211  
Качмарик Віктор  
**Викладач:**  
Якимишин Х. М.

**Мета роботи:** Розробити SQL-запити для вибору записів з однієї чи кількох таблиць в т. ч. із застосуванням: впорядкування результатів за значеннями одного і декількох полів, впорядкування результатів за зростанням і спаданням, групування результатів за значеннями одного чи декількох полів.

### **Короткі теоретичні відомості**

Для впорядкування та групування результатів разом з командою SELECT використовують наступні директиви:

**ORDER BY** {ім'я\_поля | синонім | позиція\_поля} [ASC | DESC] [, ...]

Задає порядок сортування значень у результатах запити за вказаним полем. ASC – за зростанням, DESC – за спаданням. За замовчуванням сортування відбувається за зростанням значень. Поля можна вказувати за назвою, псевдонімом або номером позиції поля у таблиці.

**GROUP BY** {ім'я\_поля | синонім | позиція\_поля} [ASC | DESC] [HAVING умова] [WITH ROLLUP] [, ...]

Групує (і одночасно сортує) рядки за вказаними полями. Поля можна вказувати за іменами, синонімами або порядковими номерами в таблиці.

**HAVING** вказує умову відбору для групи з використанням як агрегатних так і не агрегатних виразів. Дає можливість застосування до значень полів агрегатних функцій (COUNT, AVG, MIN, MAX тощо) при відборі чи групуванні рядків. Після слова **WHERE** ці функції не працюють, однак у всіх інших випадках слід використовувати саме **WHERE**.

**WITH ROLLUP** додає до результатів запити рядок із підсумковим значенням для всіх груп.

## Хід роботи

1. Запит вибору записів з таблиці customer із використанням сортування:

```
SELECT customer_id, login
FROM customer
WHERE login LIKE '%user%' ORDER BY customer_id ASC;
```

### Результат виконання:

customer_id	login
10	superuser_yuriy
9	user001_sasha
7	25user_roman
6	userq_lavrino
3	user05_vlad
1	user_vitya

2. Запит вибору замовлень з таблиці order з використанням зворотнього сортування:

```
SELECT order_id, order_date
FROM `order`
ORDER BY order_date DESC LIMIT 4;
```

### Результат виконання:

order_id	order_date
4	2020-03-15 00:00:00
6	2020-03-11 00:00:00
3	2020-03-10 00:00:00
2	2020-03-09 00:00:00

### 3. Запит вибору повідомлень із використанням групування за категоріями.

Для кожної категорії буде відображатись останнє повідомлення у ній.

```
SELECT pharmacy.pharmacy_name, medicine.medicine_name,  
medicine_category.medicine_category, medicine.quantity  
FROM ((medicine INNER JOIN all_medicine) INNER JOIN medicine_category) INNER JOIN pharmacy  
ON medicine.medicine_name = all_medicine.medicine_name  
AND all_medicine.medicine_category_id = medicine_category.medicine_category_id  
AND medicine.pharmacy_id = pharmacy.pharmacy_id  
GROUP BY medicine_category  
ORDER BY medicine_category;
```

Результат виконання:

pharmacy_name	medicine_name	medicine_category	quantity
Phar10	Adhesive plaster	Dosage form	30
Phar3	AntiCorona	Drops	4
Phar1	Mezym	Tablets	7

**Висновок:** на цій лабораторній роботі було розглянуто методи вибору даних з таблиць БД з використанням сортування та групування. Було проведено вибір даних із таблиць Author, Message та Category.