

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



**Лабораторна робота №9**  
**з дисципліни “ ОБДЗ ”**  
**На тему: «Аналітичні та підсумкові запити»**

**Виконав:**  
ст. гр. КН-211  
Ільків Андрій  
**Викладач:**  
Якимишин Х. М.

**Мета роботи:** Розробити SQL запити для вибору записів з однієї чи кількох таблиць із застосуванням агрегатних функцій для отримання підсумкових значень полів.

### Короткі теоретичні відомості

Для побудови аналітичних та підсумкових запитів на SQL використовують директиву GROUP BY, а також агрегатні функції. Основні агрегатні функції подані в таблиці. Аргументами функцій можуть бути як задані множини значень, так і результати підзапиту.

Функція (оператор)	Опис
MAX(), MIN()	Знаходить максимальне, або мінімальне значення для заданих аргументів.
AVG()	Знаходить середнє значення для заданих аргументів.
AVG(DISTINCT ...)	Знаходить середнє значення не враховуючи повтори.
SUM()	Обчислює суму значень.
SUM(DISTINCT ...)	Обчислює суму різних значень.
COUNT()	Рахує кількість рядків, які повертає запит.
COUNT(DISTINCT ...)	Рахує кількість різних значень.
BIT_AND(), BIT_OR()	Повертає побітове "і", "або" для аргументів.
STD(), STDDEV_POP()	Обчислює значення стандартного відхилення для аргументів.
VAR_POP()	Обчислює значення дисперсії для аргументів.

Для застосування агрегатних функцій SUM або AVG з часовими типами даних потрібно проводити двосторонню конвертацію типів за допомогою спеціальних функцій, наведених нижче.

TO\_DAYS() – перевести дату у число, що означає кількість днів починаючи з 0-го року.

FROM\_DAYS() – перевести кількість днів у дату.

TIME\_TO\_SEC() – перевести значення часу у кількість секунд.

SEC\_TO\_TIME() – перевести кількість секунд у час.

Наприклад,

SELECT FROM\_DAYS(SUM(TO\_DAYS(дата))) FROM таблиця;

## Хід роботи:

1. Визначити кількість студентів у кожній групі.

```
SELECT
    student.group_id, COUNT(student.group_id) AS numofstud
FROM
    student
    INNER JOIN
    studentsgroup ON student.group_id = studentsgroup.group_id
GROUP BY group_id;
```

Результат виконання:

	group_id	numofstud
▶	1	2
	2	2
	3	1
	4	1

2. Вивести кількість зареєстрованих по кожному місяцю, вивести підсумок за кожен місяць і за весь час загалом.

```
SELECT
    YEAR(studentinfo.created) AS year,
    MONTHNAME(studentinfo.created) AS month,
    COUNT(student_id) AS numofstud
FROM
    studentinfo
GROUP BY year , month WITH ROLLUP;
```

Результат виконання:

year	month	numofstud
2015	April	1
2015	NULL	1
2018	March	1
2018	NULL	1
2020	November	1
2020	NULL	1
2021	June	1
2021	NULL	1
NULL	NULL	4

3. Визначити користувача з найбільшою кількістю записів у медичній книжці.

```
SELECT
    student.last_name AS student,
    AVG(CHAR_LENGTH(studentinfo.medical_info)) AS avgmedinfo
FROM
    student
    INNER JOIN
    studentinfo ON student.student_id = studentinfo.student_id
GROUP BY student.last_name
ORDER BY avgmedinfo DESC
LIMIT 1;
```

Результат виконання:

	student	avgmedinfo
►	Perec	22.0000

**Висновок:** під час виконання даної лабораторної роботи, було розглянуто методи застосування агрегатних функцій до результатів вибору даних з таблиць БД.