

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

Кафедра систем штучного інтелекту



ЗВІТ № 9  
з курсу “ОБДЗ”  
на тему:  
«Аналітичні та підсумкові запити»

**Виконала:**

студентка групи КН-211

Лаврик Юліана

**Викладач:**

Якимишин Х.М.

## Лабораторна робота № 9

**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

Для досягнення мети роботи, реалізуємо 4 запити до бази даних :

1. Визначити кількість страв, які містять відповідний інгредієнт.
2. Вивести кількість замовлень за кожен місяць.
3. Визначити середню ціну інгредієнтів для кожного постачальника.
4. Визначити найактивнішого в кожному місяці клієнта.

1. Визначити кількість страв, які містять відповідний інгредієнт.

*Запит :*

```
SELECT ingredient.name AS ingredient, COUNT(dish.id) AS dish FROM (dish INNER  
JOIN ingredient_dish)  
INNER JOIN ingredient  
ON dish.id=ingredient_dish.dish_id1  
AND ingredient_dish.ingredient_id=ingredient.id GROUP BY ingredient.name;
```

*Результат :*

ingredient	dish
Flour	5
Milk	3
Butter	4
Eggs	5
Cacao	2

2. Вивести кількість замовлень за кожен місяць.

Для цього використано групування за полями year і month з опцією підведення підсумків **WITH ROLLUP**.

*Запит :*

```
SELECT YEAR(date) AS year, MONTHNAME(date) AS month,  
COUNT(confectionary.order.id) AS orders FROM confectionary.order  
GROUP BY year, month WITH ROLLUP;
```

*Результат :*

year	month	orders
2019	December	1
2019	NULL	1
2020	February	3
2020	January	2
2020	March	5
2020	NULL	10
NULL	NULL	11

3. Визначити середню ціну інгредієнтів для кожного постачальника.

*Запит :*

```
SELECT supplier.name AS supplier, AVG(CHAR_LENGTH(ingredient.name)) AS avgprice  
FROM supplier INNER JOIN ingredient  
ON supplier.id=ingredient.supplier_id GROUP BY supplier;
```

*Результат :*

supplier	avgprice
Hutorok	32.400000
Molokiya	125.970000
Kvochka	1.740000
Roshen	32.000000

\*Для визначення середньої довжини слова чи набору слів ,що відповідають певному полю в таблиці, потрібно використати оператор **AVG(CHAR\_LENGTH(назва\_поля))**

4. Визначити найактивнішого в кожному місяці клієнта.

Рейтинг активності буде визначатись за кількістю замовлень. Умова відбору **WHERE** буде відбирати тільки ті замовлення, які були зроблені за попередній місяць (**MONTH(CURRENT\_DATE)-1**). Групування за іменем клієнта потрібне для агрегатного обчислення рейтингу кожного клієнта.

*Запит :*

```
SELECT first_name AS bestcustomer, COUNT(DISTINCT confectionary.order.id) AS rating  
FROM customer  
INNER JOIN confectionary.order  
ON customer.id = confectionary.order.customer_id  
WHERE MONTH(confectionary.order.date)=(MONTH(CURRENT_DATE)-1)  
GROUP BY first_name ORDER BY rating DESC LIMIT 1;
```

*Результат :*

bestcustomer	rating
Olena	4

**Висновок :** під час виконання даної лабораторної роботи я здобула навички створення SQL запитів для вибору записів з однієї чи кількох таблиць БД із застосуванням агрегатних функцій для отримання підсумкових значень полів.