

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



Звіт до практичної роботи №9
з дисципліни “ ОБДЗ ”
На тему: «Аналітичні та підсумкові запити»

Виконав:
ст. гр. КН-211
Качмарик Віктор
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 ліків у кожній категорії.

```
SELECT medicine_category.medicine_category, COUNT(all_medicine.medicine_name) AS amount
FROM all_medicine INNER JOIN medicine_category
ON all_medicine.medicine_category_id = medicine_category.medicine_category_id
GROUP BY medicine_category
ORDER BY medicine_category;
```

Результат виконання:

medicine_category	amount
Dosage form	2
Drops	4
Tablets	3

2. Вивести кількість нових замовлень за кожен місяць. Для кожного місяця і року буде окремо обчислено кількість замовлень. Для цього використано групування за полями year і month з опцією підведення підсумків WITH ROLLUP.

```
SELECT YEAR(order_date) AS year,
MONTHNAME(order_date) AS month,
COUNT(order_id) AS orders
FROM `order` GROUP BY year, month WITH ROLLUP;
```

Результат виконання:

year	month	orders
2018	April	2
2018	March	1
2018	NULL	3
2019	July	2
2019	May	3
2019	NULL	5
2020	February	2
2020	January	1
2020	March	4
2020	NULL	7
NULL	NULL	15

3. Визначити середні витрати для кожного користувача.

```
SELECT customer.first_name AS customer, customer.login,
AVG(order_item.quantity*medicine.price) AS AverCosts
FROM ((customer INNER JOIN `order`) INNER JOIN order_item) INNER JOIN medicine
ON customer.customer_id = `order`.customer_id
AND `order`.order_id = order_item.order_id
AND order_item.medicine_id = medicine.medicine_id
GROUP BY customer;
```

Результат виконання:

customer	login	AverCosts
Viktor	user_vitya	160.0000
Andrew	andreeew	102.0000
Vlad	user05_vlad	168.6667
Max	max_kachmaryk	231.2500
Olesya	olesia.marko	260.0000
Yuliana	userq_lavrino	300.0000
Petro	klloiak	255.0000
Sasha	user001_sasha	40.0000
Yuriy	superuser_yuriy	442.2500

4. Визначити найактивнішого за останні 2 місяці покупця.

Рейтинг активності буде визначатись за формулою: вартість замовлення * кількість замовлень. Умова відбору WHERE буде відбирати тільки ті замовлення, які були зроблені за останні 2 місяці (MONTH(CURRENT_DATE)-1).

```
SELECT customer.first_name AS bestCustomer, customer.login,
order_item.quantity*medicine.price AS rating
FROM ((customer INNER JOIN `order`) INNER JOIN order_item) INNER JOIN medicine
ON customer.customer_id = `order`.customer_id
AND `order`.order_id = order_item.order_id
AND order_item.medicine_id = medicine.medicine_id
WHERE MONTH(`order`.order_date) < MONTH(CURRENT_DATE)
AND MONTH(`order`.order_date) > (MONTH(CURRENT_DATE) - 2)
GROUP BY customer.first_name
ORDER BY rating DESC LIMIT 1;
```

Результат виконання:

bestCustomer	login	rating
Petro	klloiak	440

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