

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"
ІНСТИТУТ ПІСЛЯДИПЛОМНОЇ ОСВІТИ
КАФЕДРА СИСТЕМ ШТУЧНОГО ІНТЕЛЕКТУ



ЗВІТ ДО ЛАБОРАТОРНОЇ РОБОТИ №9

Підготувала:

Студентка групи КН-209

Кульчицька Олена

Викладач:

Мельникова Н.І.

Лабораторна робота №9
на тему:
“Аналітичні та підсумкові запити”

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 замовлень, які зробив кожен клієнт.
2. Вивести кількість відвідувачів за кожен місяць.
3. Визначити середню ціну страв в кондитерській.
4. Визначити найактивнішого клієнта минулого місяця.

1. Визначимо кількість замовлень, які зробив кожен клієнт.
Подивимось всі замовлення, які зробили клієнти.

Запит:

```
SELECT customer.first_name, order.date
FROM confectionary.order INNER JOIN confectionary.customer
ON customer_id = customer.id;
```

Результати запиту:

first_name	date
Olena	2020-03-20
Lina	2020-03-21
Olena	2020-03-23
Lina	2020-03-14
Jan	2020-02-25
Rob	2020-03-07
Olena	2020-04-14
Olena	2020-04-10
Lina	2020-04-01

Запит:

```
SELECT customer.first_name, COUNT(order.id) AS quantity
FROM confectionary.order INNER JOIN confectionary.customer
ON customer_id = customer.id
GROUP BY customer_id
ORDER BY quantity DESC;
```

Результат запиту:

first_name	quantity
Olena	4
Lina	2
Jan	1
Lina	1
Rob	1

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

Запит:

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

Результат запиту:

year	month	customers
2020	April	3
2020	February	1
2020	March	5
2020	NULL	9
NULL	NULL	9

3. Визначимо середню ціну страв в кондитерській.
Переглянемо вміст таблиці dish.

Запит:

```
SELECT * FROM confectionary.dish;
```

Результат запиту:

id	name	weight	price	kcal
1	tiramisu	200	150.00	460
2	cheesecake	500	200.00	1100
3	carrot cake	500	170.00	900
4	apple pie	600	150.00	920
5	oreo cheesecake	500	210.00	1000
6	blueberry cake	600	180.00	920
7	nutcake	700	190.00	950
8	cherry pie	600	140.00	910
NULL	NULL	NULL	NULL	NULL

Запит:

```
SELECT dish.name AS dessert, dish.price  
FROM confectionary.dish  
WHERE price > (SELECT AVG(price) FROM confectionary.dish )  
ORDER BY price DESC;
```

Результат запиту:

	dessert	price
	oreo cheesecake	210.00
	cheesecake	200.00
	nutcake	190.00
	blueberry cake	180.00

4. Визначимо найактивнішого клієнта минулого місяця.
Умова відбору WHERE буде відбирати тільки ті замовлення, які були зроблені за попередній місяць (MONTH(CURRENT_DATE)-1). Групування за customer_id потрібне для обчислення кількості замовлень кожного клієнта.

Запит:

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

Результат запиту:

	first_name	quantity
	Olena	2

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