**МІНІСТЕРСТВО ОСВІТИ І НАУКИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**


# Звіт до лабораторної роботи №9

З ПРЕДМЕТУ “ОРГАНІЗАЦІЯ БАЗ ДАНИХ ТА ЗНАНЬ”

**Виконав**:

ст. гр. КН-211

Шебеко Андрій

**Викладач:**

Якимишин Х.М.

Львів – 2020

Лабораторна робота №9
*з курсу* "ОБДЗ"
*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

1. З таблиць клієнт, продукт та коментар вибираємо опис товарів, куплених покупцем, емейл покупця та суму товарів, а також сумуємо ціну покупки та групуємо за емейлом.

```
select email, description, sum(price)
from (client inner join product) inner join comment
on comment.product_id = product.product_id
and comment.username = client.`first name`
group by email;
```

| email | description | sum(price) |
|---|---|---|
| andy@net | Big round-shape table | 998 |
| vik@ru | small light chair | 199 |
| maol@mail | Big square `1715` umbrella | 359 |
| surk@sky | Big round-shape table | 499 |
| zorik@mail | Big round-shape table | 499 |

2. Обраховуємо скільки користувачі зареєстровано за кожен місяць та рік з підсумком.

```
select email, year(register_date) as year, month(register_date) as month,
count(client_id) as users
from client
group by year, month with rollup;
```

| email | year | month | users |
|---|---|---|---|
| andy@net | 2020 | 3 | 2 |
| maol@mail | 2020 | 4 | 1 |
| smt@mail | 2020 | 5 | 7 |
| smt@mail | 2020 | NULL | 10 |
| smt@mail | NULL | NULL | 10 |

3. Обраховуємо середню довжину усіх коментарів, написаних користувачем.

```
select `first name`, email, avg(char_length(comment)) as avgcomment
from comment inner join client
on comment.username = client.`first name`
group by username;
```

| first name | email | avgcomment |
|---|---|---|
| Andrii | andy@net | 4.0000 |
| Vitya | vik@ru | 5.0000 |
| Lera | maol@mail | 8.0000 |
| Katya | surk@sky | 3.0000 |
| Zoya | zorik@mail | 8.0000 |

4. Обраховуємо рейтинг користувача за його активністю, 1 коментар = 1 рейтингу та 1 покупка = 2 рейтингу, сортуємо за рейтингом за спаданням.

```
select `first name`, count(distinct comment) as comments, ifnull(sum(distinct amount), 0) as products,
(count(distinct comment) + ifnull(sum(distinct amount), 0)*2) as rating
from client left join comment
on client.`first name` = comment.username
left join basket
on client.client_id = basket.client_id
group by  `first name`
order by rating DESC limit 3;
```

| first name | comments | products | rating |
|---|---|---|---|
| Andrii | 2 | 4 | 10 |
| Zoya | 1 | 0 | 1 |
| Lera | 1 | 0 | 1 |

Висновок: я навчився виконувати аналітичні та підсумкові запити з бази данних.