

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.

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

1. Визначити кількість повідомлень у кожній категорії.
2. Вивести кількість нових користувачів, повідомлень і коментарів за кожен місяць.
3. Визначити середню довжину повідомлень для кожного користувача.
4. Визначити найактивнішого в кожному місяці користувача.

1. Визначити кількість повідомлень у кожній категорії.

```
SELECT cname AS category, COUNT(message.messageID) AS messages  
FROM (message INNER JOIN message_category) INNER JOIN category  
ON message.messageID=message_category.messageID  
AND message_category.categoryID=category.categoryID  
GROUP BY cname;
```

Результати запиту:

category	messages
blogentry	2
FAQ	1
files	1
links	1
movies	1
news	2
pagecontent	2
profileinfo	2
questions	1
ua	1

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

```
SELECT YEAR(created) AS year, MONTHNAME(created) AS month,  
COUNT(login) AS users  
FROM author GROUP BY year, month WITH ROLLUP;
```

```
SELECT YEAR(posted) AS year, MONTHNAME(posted) AS month,  
COUNT(messageID) AS messages  
FROM message GROUP BY year, month WITH ROLLUP;
```

```
SELECT YEAR(posted) AS year, MONTHNAME(posted) AS month,  
COUNT(commentID) AS comments  
FROM comment GROUP BY year, month WITH ROLLUP;
```

Результати запитів:

year	month	users
2008	February	1
2008	January	1
2008	NULL	2
2009	February	3
2009	January	4
2009	March	1
2009	NULL	8
NULL	NULL	10

year	month	messages
2009	February	4
2009	January	3
2009	March	2
2009	NULL	9
NULL	NULL	9

year	month	comments
2009	February	1
2009	March	5
2009	NULL	6
NULL	NULL	6

3. Визначити середню довжину повідомлень для кожного користувача.

```
SELECT login AS user, AVG(CHAR_LENGTH(message)) AS avgmessage
FROM author INNER JOIN message ON
author.authorID=message.authorID
GROUP BY user;
```

Результати запиту:

user	avgmessage
admin	20.0000
guest2	35.0000
superuser2	34.0000
user1	24.3333
user2	9.0000
user4	34.0000

4. Визначити найактивнішого за останній місяць користувача.

Рейтинг активності буде визначатись за формулою: к-сть повідомлень*2 + к-сть коментарів. Умова відбору WHERE буде відбирати тільки ті повідомлення і коментарі, які були написані за попередній місяць (MONTH(CURRENT_DATE)-1). Групування за псевдонімом автора потрібне для агрегатного обчислення рейтингу кожного автора.

```
SELECT login AS bestauthor, (COUNT(DISTINCT message.messageID)
*2 + COUNT(DISTINCT comment.commentID)) AS rating
FROM (author INNER JOIN message) INNER JOIN comment ON
author.authorID = message.authorID AND author.authorID =
comment.authorID
WHERE MONTH(message.posted)=(MONTH(CURRENT_DATE)-1) AND
MONTH(comment.posted)=(MONTH(CURRENT_DATE)-1)
GROUP BY login ORDER BY rating DESC LIMIT 1;
```

Результат запиту:

bestauthor	rating
user1	6

Висновок: на цій лабораторній роботі було розглянуто методи застосування агрегатних функцій до результатів вибору даних з таблиць БД.