

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Інститут комп'ютерних наук та інформаційних технологій

Кафедра систем штучного інтелекту



Лабораторна робота №9

на тему “Аналітичні та підсумкові запити”

Виконав:

Романишин М.Р.

КН-211

Викладач:

Якимишин Х.М.

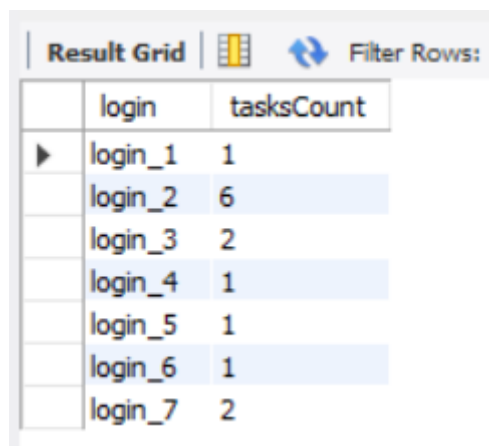
Лабораторна робота №9
з курсу “ОБДЗ”
на тему:
“Аналітичні та підсумкові запити”

Мета роботи: Розробити SQL запити для вибору записів з однієї чи кількох таблиць із застосуванням агрегатних функцій для отримання підсумкових значень полів.

Хід роботи

1. Визначаю кількість повідомлень кожного користувача

```
select login, count(task.task_id) as tasksCount
from (task inner join user_project_task)
inner join user
on task.task_id = user_project_task.task_id
and user_project_task.user_id = user.user_id
group by login;
```



The screenshot shows a 'Result Grid' window with a table containing two columns: 'login' and 'tasksCount'. The table has 8 rows of data. The first row is highlighted with a mouse cursor. The data is as follows:

login	tasksCount
login_1	1
login_2	6
login_3	2
login_4	1
login_5	1
login_6	1
login_7	2

2. Визначимо кількість коментарів та завдань за місяць, рік та за весь час.

```
select year(posted_time) as year, monthname(posted_time) as month,
count(comment_id) as comments
from task_comment group by year, month with rollup;
```

```
select year(deadline_time) as year, monthname(deadline_time) as month,
count(task_id) as tasks
from task group by year, month with rollup;
```

Result Grid			
	year	month	comments
▶	2019	March	1
	2019	NULL	1
	2020	April	3
	2020	January	2
	2020	July	1
	2020	March	11
	2020	May	4
	2020	NULL	21
	NULL	NULL	22

Result Grid			
	year	month	tasks
▶	2018	February	3
	2018	January	1
	2018	July	1
	2018	March	2
	2018	NULL	7
	2019	April	1
	2019	February	2
	2019	January	2
	2019	March	3
	2019	NULL	8
	NULL	NULL	15

3. Визначимо середню довжину коментарів для кожного користувача.

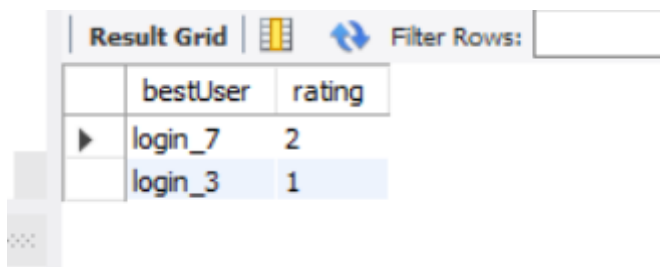
```
select login as user, avg(char_length(comment_text)) as avgdescription
from user inner join task_comment on
user.user_id = task_comment.user_id
group by user;
```

Result Grid		
	user	avgdescription
▶	login_1	14.1667
	login_2	13.0000
	login_3	13.0000
	login_4	14.0000
	login_5	14.0000
	login_6	14.0000
	login_7	14.0000

4. Визначимо двох найактивніших користувачів (рейтинг буде дорівнювати кількості написаних коментарів).

```
select login as bestUser, (count(task_comment.comment_id)) as rating
from user inner join task_comment
```

```
on user.user_id = task_comment.user_id
where month(task_comment.posted_time) = (month(current_date) - 1)
group by login order by rating desc limit 2;
```



The screenshot shows a 'Result Grid' window with a 'Filter Rows' input field. The grid contains two columns: 'bestUser' and 'rating'. There are two rows of data: the first row has 'login_7' and '2', and the second row has 'login_3' and '1'. The second row is highlighted in blue.

	bestUser	rating
▶	login_7	2
	login_3	1

Висновок: розробив SQL запити для вибору записів з однієї чи кількох таблиць із застосуванням агрегатних функцій для отримання підсумкових значень полів.