

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНИЙ  
УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Інститут комп'ютерних наук та інформаційних технологій

Кафедра систем штучного інтелекту



**Лабораторна робота №7**

**на тему “Запити на вибір даних з таблиць бази даних”**

**Виконав:**

Романишин М.Р.

КН-211

**Викладач:**

Якимишин Х.М.

Львів – 2020

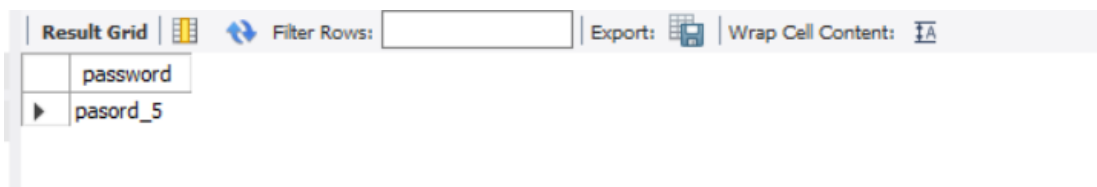
Лабораторна робота №7  
з курсу “ОБДЗ”  
на тему:  
“Запити на вибір даних з таблиць бази даних”

**Мета роботи:** Розробити SQL запити відбору даних з одиничних та з’єднаних таблиць, в тому числі з використанням підзапитів, натурального, умовного та лівого з’єднання, із застосуванням у критеріях вибірки функцій та операторів, в т. ч. LIKE, BETWEEN, IS NULL, IS NOT NULL, IN (...), NOT IN (...), ALL, SOME, ANY, EXISTS.

### Хід роботи

1. Знайдемо пароль користувача з user\_id = 5.

```
select user.password from user where user_id = 5;
```

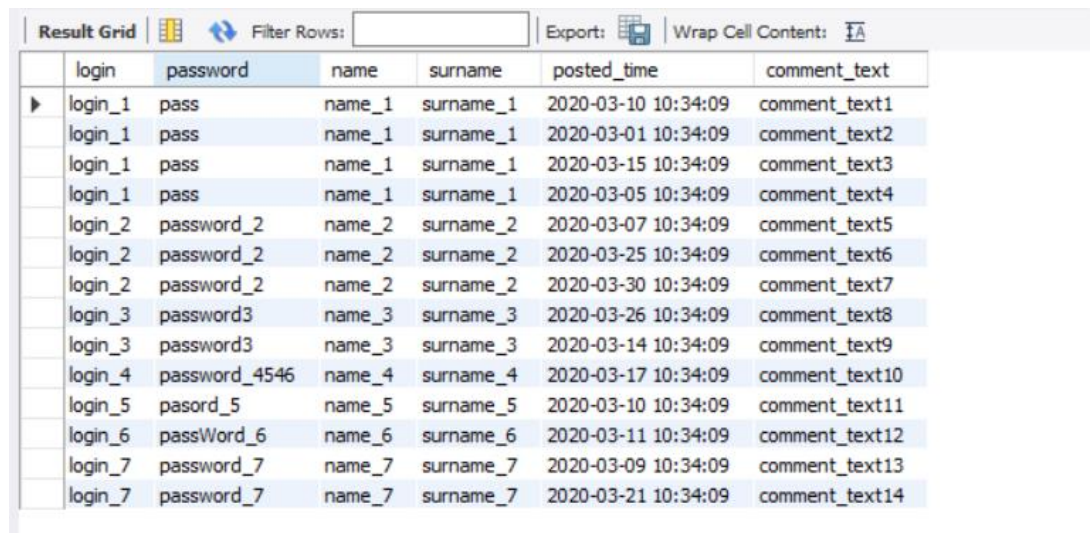


The screenshot shows a database interface with a 'Result Grid' tab. The grid contains one row with the column 'password' and the value 'pasord\_5'. The interface also includes a 'Filter Rows' field, an 'Export' button, and a 'Wrap Cell Content' checkbox.

password
pasord_5

2. Оберемо всіх користувачів з їхніми коментарями.

```
SELECT login, password, name, surname, task_comment.posted_time, task_comment.comment_text  
FROM user RIGHT JOIN task_comment ON  
user.user_id = task_comment.user_id;
```



The screenshot shows a database interface with a 'Result Grid' tab. The grid contains 14 rows of data. The columns are login, password, name, surname, posted\_time, and comment\_text. The data is as follows:

login	password	name	surname	posted_time	comment_text
login_1	pass	name_1	surname_1	2020-03-10 10:34:09	comment_text1
login_1	pass	name_1	surname_1	2020-03-01 10:34:09	comment_text2
login_1	pass	name_1	surname_1	2020-03-15 10:34:09	comment_text3
login_1	pass	name_1	surname_1	2020-03-05 10:34:09	comment_text4
login_2	password_2	name_2	surname_2	2020-03-07 10:34:09	comment_text5
login_2	password_2	name_2	surname_2	2020-03-25 10:34:09	comment_text6
login_2	password_2	name_2	surname_2	2020-03-30 10:34:09	comment_text7
login_3	password3	name_3	surname_3	2020-03-26 10:34:09	comment_text8
login_3	password3	name_3	surname_3	2020-03-14 10:34:09	comment_text9
login_4	password_4546	name_4	surname_4	2020-03-17 10:34:09	comment_text10
login_5	pasord_5	name_5	surname_5	2020-03-10 10:34:09	comment_text11
login_6	passWord_6	name_6	surname_6	2020-03-11 10:34:09	comment_text12
login_7	password_7	name_7	surname_7	2020-03-09 10:34:09	comment_text13
login_7	password_7	name_7	surname_7	2020-03-21 10:34:09	comment_text14

3. Оберемо всі коментарі які були написані пізніше за ‘2020-03-10 10:34:09’.

```
SELECT user.login, task_comment.posted_time , task_comment.comment_text  
FROM user JOIN task_comment ON task_comment.user_id = user.user_id  
WHERE task_comment.posted_time > '2020-03-10 10:34:09';
```

Result Grid				Filter Rows:	Export:	Wrap Cell Content:
	login	posted_time	comment_text			
▶	login_1	2020-03-15 10:34:09	comment_text3			
	login_2	2020-03-25 10:34:09	comment_text6			
	login_2	2020-03-30 10:34:09	comment_text7			
	login_3	2020-03-26 10:34:09	comment_text8			
	login_3	2020-03-14 10:34:09	comment_text9			
	login_4	2020-03-17 10:34:09	comment_text10			
	login_6	2020-03-11 10:34:09	comment_text12			
	login_7	2020-03-21 10:34:09	comment_text14			

4. Оберемо всіх користувачів зі завданнями в проектах, де вони виконують роль user.

```
SELECT user.login, user_project_task.user_role, task.description, task.title,
task.deadline_time
FROM (user JOIN user_project_task) INNER JOIN task
ON user.user_id = user_project_task.user_id
AND task.task_id = user_project_task.task_id
WHERE user_project_task.user_role = 'user';
```

Result Grid						Filter Rows:	Export:	Wrap Cell Content:
	login	user_role	description	title	deadline_time			
▶	login_2	user	task2_description	task2_title	2020-03-10 10:34:09			
	login_2	user	task3_description	task3_title	2020-03-15 10:34:09			
	login_2	user	task4_description	task4_title	2020-03-14 10:34:09			
	login_2	user	task5_description	task5_title	2020-03-18 10:34:09			
	login_2	user	task6_description	task6_title	2020-03-18 10:34:09			
	login_2	user	task7_description	task7_title	2020-03-05 10:34:09			
	login_3	user	task7_description	task7_title	2020-03-05 10:34:09			
	login_3	user	task7_description	task7_title	2020-03-05 10:34:09			
	login_4	user	task7_description	task7_title	2020-03-05 10:34:09			
	login_5	user	task7_description	task7_title	2020-03-05 10:34:09			
	login_6	user	task7_description	task7_title	2020-03-05 10:34:09			
	login_7	user	task7_description	task7_title	2020-03-05 10:34:09			
	login_7	user	task7_description	task7_title	2020-03-05 10:34:09			

5. Оберемо три останні коментарі до завдань від користувачів, які виконують роль user в проекті.

```
SELECT user.login, task_comment.comment_text, task_comment.posted_time
FROM user INNER JOIN task_comment

ON user.user_id = task_comment.user_id
WHERE user.user_id IN (SELECT user_project_task.user_id FROM user_project_task
WHERE user_project_task.user_role = 'user')
ORDER BY task_comment.posted_time DESC LIMIT 3;
```

	login	comment_text	posted_time
▶	login_6	comment_text7	2020-03-30 10:34:09
	login_7	comment_text7	2020-03-30 10:34:09
	login_2	comment_text7	2020-03-30 10:34:09

## 6. Оберемо користувачів, які не написали жодного коментаря.

```
SELECT user.login, user.name, user.surname FROM user
WHERE NOT EXISTS
(SELECT * FROM task_comment WHERE task_comment.user_id = user.user_id);
```

	login	name	surname
▶	login_8	name_8	surname_8

## 7. Оберемо користувачів, які не використовують пошту Gmail.

```
SELECT email, telegram
FROM contacts
WHERE email NOT REGEXP "^[a-z0-9](\.[a-z0-9]){5,}@g(oogle)?mail\.com$";
```

	email	telegram
▶	user5_email@ukr.net	https://t.me/user5_telegram
	user6_email@ukr.net	https://t.me/user6_telegram
	user7_email@ukr.net	https://t.me/user7_telegram
	user8_email@ukr.net	https://t.me/user8_telegram

## SQL-скріпт

```
use projectManagementSystem;
```

```
insert into project(project_name, customer_company) values('project1', 'qwerty');
```

```
insert into task (deadline_time, title, description) values ( '2020-03-18 10:34:09',
'task1_title', 'task1_description'),
('2020-03-10 10:34:09', 'task2_title', 'task2_description'),
('2020-03-15 10:34:09', 'task3_title', 'task3_description'),
('2020-03-14 10:34:09', 'task4_title', 'task4_description'),
('2020-03-18 10:34:09', 'task5_title', 'task5_description'),
('2020-03-18 10:34:09', 'task6_title', 'task6_description'),
('2020-03-05 10:34:09', 'task7_title', 'task7_description'),
('2020-03-06 10:34:09', 'task8_title', 'task8_description'),
('2020-03-01 10:34:09', 'task9_title', 'task9_description'),
('2020-03-08 10:34:09', 'task10_title', 'task10_description'),
('2020-03-19 10:34:09', 'task11_title', 'task11_description'),
('2020-03-20 10:34:09', 'task12_title', 'task12_description'),
```

```

('2020-03-25 10:34:09', 'task13_title', 'task13_description'),
('2020-03-30 10:34:09', 'task14_title', 'task14_description');

insert task_comment(user_id, task_id, comment_text, posted_time) values (1, 1, 'comment_text1',
'2020-03-10 10:34:09'),
    (1, 2, 'comment_text2', '2020-03-01 10:34:09'),
    (1, 3, 'comment_text3', '2020-03-15 10:34:09'),
    (1, 4, 'comment_text4', '2020-03-05 10:34:09'),
    (2, 5, 'comment_text5', '2020-03-07 10:34:09'),
    (2, 6, 'comment_text6', '2020-03-25 10:34:09'),
    (2, 7, 'comment_text7', '2020-03-30 10:34:09'),
    (3, 8, 'comment_text8', '2020-03-26 10:34:09'),
    (3, 9, 'comment_text9', '2020-03-14 10:34:09'),
    (4, 10, 'comment_text10', '2020-03-17 10:34:09'),
    (5, 11, 'comment_text11', '2020-03-10 10:34:09'),
    (6, 12, 'comment_text12', '2020-03-11 10:34:09'),
    (7, 13, 'comment_text13', '2020-03-09 10:34:09'),
    (7, 14, 'comment_text14', '2020-03-21 10:34:09');

insert into user_project_task(user_id, task_id, project_id, user_role) values (1,1,1,'admin'),
(2,2,1,'user'),
(2,3,1,'user'),
(2,4,1,'user'),
(2,5,1,'user'),
(2,6,1,'user'),
(2,7,1,'user'),
(3,7,1,'user'),
(3,7,1,'user'),
(4,7,1,'user'),
(5,7,1,'user'),
(6,7,1,'user'),
(7,7,1,'user'),
(7,7,1,'user');

select user.password from user where user_id = 5;

SELECT login, password, name, surname, task_comment.posted_time, task_comment.comment_text
FROM user RIGHT JOIN task_comment ON
user.user_id = task_comment.user_id;

SELECT user.login, task_comment.posted_time , task_comment.comment_text
FROM user JOIN task_comment ON task_comment.user_id = user.user_id
WHERE task_comment.posted_time > '2020-03-10 10:34:09';

SELECT user.login, user_project_task.user_role, task.description, task.title, task.deadline_time
FROM (user JOIN user_project_task) INNER JOIN task
ON user.user_id = user_project_task.user_id
AND task.task_id = user_project_task.task_id
WHERE user_project_task.user_role = 'user';

SELECT user.login, task_comment.comment_text, task_comment.posted_time
FROM user INNER JOIN task_comment
ON user.user_id = user.user_id
WHERE user.user_id IN (SELECT user_project_task.user_id FROM user_project_task
WHERE user_project_task.user_role = 'user')
ORDER BY task_comment.posted_time DESC LIMIT 3;

SELECT user.login, user.name, user.surname FROM user
WHERE NOT EXISTS
(SELECT * FROM task_comment WHERE task_comment.user_id = user.user_id);

SELECT email, telegram
FROM contacts
WHERE email NOT REGEXP "^[a-z0-9](\.[a-z0-9]){5,}@g(oogle)?mail\.com$";

```

**Висновок:** на цій лабораторній роботі було вивчено методи вибору даних зі з'єднаних таблиць БД засобами SQL та виконано запити до бази даних з використанням директив SELECT та JOIN, а також складних критеріїв в умові вибірки.