

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**

Інститут комп'ютерних наук та інформаційних технологій

Кафедра систем штучного інтелекту



Звіт до лабораторної роботи №7

з дисципліни

“Організація баз даних та знань”

Виконав:

ст. гр. КН-211

Петров Дмитро

Викладач:

Якимишин Х.М.

Львів – 2020

Тема: 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

Для вивчення роботи директив вибору даних з таблиць розробимо та виконаємо такі запити над таблицями user, delivery, order, comment.

1. Показати пароль заданого користувача.
2. Показати користувачів і їхні коментарі (ліве з'єднання таблиць).
3. Показати перелік користувачів у групі Guests (натуральне з'єднання).
4. Показати всі коментарі користувачів з груп Guests та SUgroup2 (умовне з'єднання).
5. Показати останні 3 коментарі користувачів з груп Guests та SUgroup2 (підзапит).
6. Визначити користувачів, які не написали жодного повідомлення.
7. Визначити користувачів, паролі яких не відповідають вимогам безпеки (менші за 8 символів або не містять цифр).

1. Знайдемо email користувача з номером 18. Для цього слід в умові відбору вказати номер потрібного користувача.

```
SELECT id, email FROM user WHERE id = 18;
```

```
mysql> SELECT id, email FROM user WHERE id = 18;
+----+-----+
| id | email                |
+----+-----+
| 18 | bazzka@hots.com     |
+----+-----+
1 row in set (0.00 sec)
```

2. Виберемо всіх користувачів з їхніми коментарями. Для цього потрібно виконати ліве з'єднання. Для користувачів, які не написали жодного коментаря в результатах буде відображено порожні значення, тому за допомогою функції EXISTS проводимо додаткову перевірку.

```
SELECT user.id, user.email, comment.text, comment.rate
```

```

FROM user
LEFT JOIN
  comment ON user.id = comment.user_id
WHERE
  EXISTS (SELECT * FROM comment WHERE user.id = comment.user_id);
mysql> SELECT user.id, user.email, comment.text, comment.rate
->      FROM user
->      LEFT JOIN
->      comment ON user.id = comment.user_id
->      WHERE
->      EXISTS (SELECT * FROM comment WHERE user.id = comment.user_id);
+-----+-----+-----+-----+
| id | email          | text                                     | rate |
+-----+-----+-----+-----+
| 16 | 1233@hots.com  | It works                               | 4     |
| 16 | 1233@hots.com  | OH MY GOD! This is incredible!         | 5     |
| 17 | new@hots.com   | DAMN! This thing even worse than I though | 3     |
| 18 | bazzka@hots.com | HOLY SHIT! GIVE MY MONEY BACK          | 1     |
| 19 | Smith@hots.com | Could be worse                         | 2     |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

```

3. Виберемо із коментарів лише ті, що стосуються завершених замовлень. Для цього виконаємо умовне з'єднання таблиць order і comment за атрибутом user_id, використовуючи директиву INNER JOIN.

```

SELECT order.user_id, order.id, order.status, comment.id, comment.text,
comment.rate
FROM db_shop.order INNER JOIN comment ON order.user_id = comment.user_id
WHERE order.status = 'accomplished';
mysql> SELECT order.user_id, order.id, order.status, comment.id, comment.text, comment.rate
-> FROM db_shop.order INNER JOIN comment ON order.user_id = comment.user_id
-> WHERE order.status = 'accomplished';
+-----+-----+-----+-----+-----+-----+
| user_id | id | status      | id | text                                     | rate |
+-----+-----+-----+-----+-----+-----+
| 17      | 13 | accomplished | 13 | DAMN! This thing even worse than I though | 3     |
| 18      | 14 | accomplished | 14 | HOLY SHIT! GIVE MY MONEY BACK          | 1     |
| 19      | 15 | accomplished | 15 | Could be worse                         | 2     |
| 19      | 16 | accomplished | 15 | Could be worse                         | 2     |
| 16      | 17 | accomplished | 11 | It works                               | 4     |
| 16      | 17 | accomplished | 12 | OH MY GOD! This is incredible!         | 5     |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

```

4. Виберемо всі коментарі користувачів до замовлень, що вже виконані, або доставляються. Для цього виконаємо умовне з'єднання таблиць user і order за атрибутом user_id, та таблиці comment використовуючи директиву INNER JOIN.

```

SELECT user.email, order.status, comment.text, comment.rate
FROM (user INNER JOIN db_shop.order)

```

INNER JOIN comment ON order.user_id = user.id AND comment.user_id = order.user_id

WHERE order.status IN ('accomplished', 'delivering');

```
mysql> SELECT user.email, order.status, comment.text, comment.rate
-> FROM (user INNER JOIN db_shop.order)
-> INNER JOIN comment ON order.user_id = user.id AND comment.user_id = order.user_id
-> WHERE order.status IN ('accomplished', 'delivering');
```

email	status	text	rate
1233@hots.com	delivering	It works	4
1233@hots.com	accomplished	It works	4
1233@hots.com	delivering	OH MY GOD! This is incredible!	5
1233@hots.com	accomplished	OH MY GOD! This is incredible!	5
new@hots.com	accomplished	DAMN! This thing even worse than I though	3
bazzka@hots.com	accomplished	HOLY SHIT! GIVE MY MONEY BACK	1
Smith@hots.com	accomplished	Could be worse	2
Smith@hots.com	accomplished	Could be worse	2

8 rows in set (0.00 sec)

5. Виберемо останні 3 коментарі замовлення, що не невідомо коли будуть доставлення через те, що саме перебувають в доставці до складу. Для цього замість дирактиви JOIN використаємо підзапит в умові відбору, який буде повертати номери потрібних товарів.

SELECT goods.depot, order.status, order.ordered

FROM db_shop.order

INNER JOIN goods ON order.id = goods.order_id

WHERE goods.id IN (SELECT goods.id FROM delivery WHERE status IN ('delivering', 'preparing'))

ORDER BY order.ordered DESC LIMIT 3;

```
mysql> SELECT goods.depot, order.status, order.ordered
-> FROM db_shop.order
-> INNER JOIN goods ON order.id = goods.order_id
-> WHERE goods.id IN (SELECT goods.id FROM delivery WHERE status IN ('delivering', 'preparing'))
-> ORDER BY order.ordered DESC LIMIT 3;
```

depot	status	ordered
Lviv, Sykhivska 09	preparing	2020-04-24 09:19:11
Lviv, Sykhivska 09	preparing	2020-04-24 09:19:11
Lviv, Lubliska 12	delivering	2020-04-24 09:09:01

3 rows in set (0.00 sec)

6. Визначимо користувачів, які не написали жодного повідомлення.

SELECT user.email FROM user WHERE NOT EXISTS (SELECT * FROM comment WHERE comment.user_id = user.id);

```
mysql> SELECT user.email FROM user
-> WHERE NOT EXISTS
-> (SELECT * FROM comment WHERE comment.user_id = user.id);
```

email
email@host.com
johnson@hots.com
BROWN@hots.com
DAVIS@hots.com
MILLER@hots.com

```
5 rows in set (0.00 sec)
```

7. Визначимо користувачів, паролі яких не відповідають вимогам безпеки (менші за 8 символів або не містять цифр).

```
SELECT email, password AS pass FROM user WHERE
CHAR_LENGTH(password)<8 OR password NOT REGEXP '[0-9]';
```

```
mysql> SELECT email, password AS pass FROM user
-> WHERE CHAR_LENGTH(password)<8 OR password NOT REGEXP '[0-9]';
```

email	pass
1233@hots.com	password
new@hots.com	123456
bazzka@hots.com	iamthelow
MILLER@hots.com	qwertyuiop

```
4 rows in set (0.00 sec)
```

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.