

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

**Інститут комп'ютерних наук та інформаційних технологій**

**Кафедра систем штучного інтелекту**



**Звіт до лабораторної роботи №11**

з дисципліни

“Організація Баз Даних та знань”

**Виконала:**

ст. гр. КН-210  
Заремба Вікторія

**Викладач:**

Мельникова Н.І.

Лабораторна робота №11  
з курсу “ОБДЗ”  
на тему:  
**“Розробка та застосування транзакцій”**

**Мета роботи:** Навчитися використовувати механізм транзакцій у СУБД MySQL. Розробити SQL запити, які виконуються як єдине ціле в рамках однієї транзакції.

**Короткі теоретичні відомості.**

Транзакція – це сукупність директив SQL, які виконуються як єдине ціле з можливістю відміни результатів їх виконання. Зміни в таблицях записуються у базу даних лише після успішного виконання всіх директив транзакції. Інакше, всі зроблені зміни ігноруються. Це дозволяє уникати помилок при маніпулюванні великими обсягами записів, зберігати цілісність даних при помилках під час додавання, видалення, модифікації значень у різних таблицях і полях тощо. СУБД MySQL також підтримує глобальні розподілені транзакції, які виконуються на декількох базах даних, або на різних серверах баз даних (XA-транзакції).

Для організації транзакцій в MySQL використовують такі директиви, як SET autocommit, START TRANSACTION, COMMIT і ROLLBACK.

START TRANSACTION Вказує на початок транзакції. Директива вимикає автоматичне збереження змін для всіх подальших запитів, поки не буде виконано команду COMMIT, або ROLLBACK.

COMMIT Зберегти зміни, зроблені даною транзакцією.

ROLLBACK Відмінити дану транзакцію і зроблені нею зміни у базі даних. Слід зауважити, що зміни у схемі бази даних не можна відмінити, тобто результат видалення, зміни або створення таблиці завжди зберігається.

SET autocommit=0 Вимикає автоматичне збереження змін для поточної сесії зв'язку з сервером БД. За замовчуванням, зміни зберігаються автоматично, тобто результат виконання запиту, який змінює таблицю, одразу записується на диск без можливості відміни операції.

AND CHAIN Одразу після завершення даної транзакції розпочати виконання наступної.

RELEASE Одразу після виконання даної транзакції завершити поточну сесію зв'язку з сервером.

Транзакції можна розбивати на окремі логічні частини, оголошуючи так звані точки збереження. Це дозволяє відмінити результати виконання не всієї транзакції, а лише тих запитів, які виконувались після оголошеної точки збереження (SAVEPOINT).

SAVEPOINT мітка Оголошує точку збереження всередині транзакції та задає її назву.

ROLLBACK TO [SAVEPOINT] мітка Відмінює результати виконання запитів, вказаних після даної точки збереження.

RELEASE SAVEPOINT мітка Видаляє точку збереження.

## Хід Роботи

В ході роботи, потрібно продемонструвати успішне і неуспішне виконання транзакції, а також збереження змін, відміни транзакції і точки збереження.

### 1. Початок транзакції.

Розпочнемо транзакцію та додамо кілька записів в таблицю department.

```
START TRANSACTION;  
INSERT INTO department VALUE (null, 'program', 24);  
INSERT INTO department VALUE (null, 'харчування', 24);  
INSERT INTO department VALUE (null, 'оптимізаційне забезпечення', 24);
```

### 2. Перевірка введених даних.

Відкриємо нову консоль і виберемо всі записи. Бачимо, що нові записи не додалися, оскільки транзакцію не закрито.

```
SELECT * FROM pm_system.department;
```

	id_department	dep_name	id_project
1	2	комунікації	23
2	3	програма	23
3	4	фандрейзинг	23
4	5	медіа	23
5	6	майстерня	23
6	7	програма	24
7	8	інтендант	24
8	9	зв'язки	24
9	10	зв'язківство	25
10	11	бунчужна	25
11	12	писар	25
12	13	інтендант	25
13	14	господарник	25

Якщо ввести аналогічний запит в консолі транзакції – зміни буде відображено, оскільки локально вони зберігаються.

### 3. Додамо точку збереження і введемо неправильні дані.

Додамо точку збереження SAVEPOINT one, після чого введемо неправильні дані в таблицю project і спробуємо закопітити. Результатом буде помилка.

```
SAVEPOINT one;  
INSERT INTO project VALUE (null, 'Dojo', '2020-05-28', 1);  
COMMIT;
```

Результат:




[2020-05-19 21:49:31] [21S01][1136] Column count doesn't match value count at row 1

### 4. Відміна результатів запитів після точки збереження, збереження і перевірка змін.

Відмінимо результати запитів, вказаних після точки збереження і закомітимо.

```
ROLLBACK TO SAVEPOINT one;  
COMMIT;
```

Ще раз відкриємо нову консоль і виберемо всі записи. Цього разу бачимо, що нові дані з'явилися у таблиці

	 id_department ▾	 dep_name ▾	 id_project ▾
1	2	комунікації	23
2	3	програма	23
3	4	фандрейзинг	23
4	5	медіа	23
5	6	майстерня	23
6	7	програма	24
7	8	інтендант	24
8	9	зв'язки	24
9	10	зв'язківство	25
10	11	бунчужна	25
11	12	писар	25
12	13	інтендант	25
13	14	господарник	25
14	18	rogram	24
15	19	харчування	24
16	20	оптимізаційне забезпечення	24

**Висновок:** На цій лабораторній роботі я ознайомився із механізмом транзакцій у СУБД MySQL.