

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Кафедра «Системи штучного інтелекту»



**Звіт**

**до лабораторної роботи №11**

**З дисципліни «Організація баз даних та знань»**

**Виконала:**

студентка групи КН-209

Міхняк Софія

**Прийняла:**

Мельникова Н. І.

*Львів-2020*

## Лабораторна робота №11

### “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 Відмінняє 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. Розробимо транзакцію, яка буде вносити дані в таблицю consumer і consumer\_order. Транзакція буде відміняти всі зміни у таблицях при виникненні помилки чи іншої суперечливості.

#### **1. Відміна транзакції.**

В першому рядку помилка – не вказано вік, отже, транзакція не буде виконаною

```
START TRANSACTION;
```

```
INSERT INTO consumer(name, email, password, gender, age) VALUE  
( 'user24',  
'user24@kml.com', 'u24pass', 'male');
```

```
INSERT INTO consumer(name, email, password, gender, age) VALUE  
( 'user533',  
'user533@kml.com', 'u533pass', 'female', 16);
```

```
INSERT INTO consumer(name, email, password, gender, age) VALUE  
( 'user37',  
'user37@kml.com', 'u37pass', 'male', 43);
```

```
INSERT INTO consumer(name, email, password, gender, age) VALUE  
( 'user42',  
'user42@kml.com', 'u42pass', 'male', 33);
```

```
INSERT INTO consumer_order (price, date, consumer_id) VALUE (500,  
now(), 5);
```

```
INSERT INTO consumer_order(price, date, consumer_id) VALUE (300,  
now(), 5);
```

```
INSERT INTO consumer_order(price, date, consumer_id) VALUE (150,  
now(), 5);
```

```
COMMIT;
```

Відповідь сервера:

Error Code: 1136. Column count doesn't match value count at row 1

Як бачимо на рис.1 нових клієнтів не додано.

id	name	email	password	gender	age
9	user3	user3@kml.com	u3pass	male	43
10	user4	user4@kml.com	u4pass	male	33
12	user532	user532@kml.com	u532pass	female	22
13	user41	user41@kml.com	u41pass	male	33
NULL	NULL	NULL	NULL	NULL	NULL

Рис.1 Таблиця consumer

## 2. Транзакція успішна

Результат транзакції після виправлення помилки:

✓	100	12:14:01	START TRANSACTION	0 row(s) affected
✓	101	12:14:01	INSERT INTO consumer(name, email, password, gender, age) VALUE ('user24', 'user24@kml.com', 'u24pass', ...	1 row(s) affected
✓	102	12:14:01	INSERT INTO consumer(name, email, password, gender, age) VALUE ('user533', 'user533@kml.com', 'u533pa...	1 row(s) affected
✓	103	12:14:01	INSERT INTO consumer(name, email, password, gender, age) VALUE ('user37', 'user37@kml.com', 'u37pass', ...	1 row(s) affected
✓	104	12:14:01	INSERT INTO consumer(name, email, password, gender, age) VALUE ('user42', 'user42@kml.com', 'u42pass', ...	1 row(s) affected
✓	105	12:14:02	INSERT INTO consumer_order(price, date, consumer_id) VALUE (500, now(), 5)	1 row(s) affected
✓	106	12:14:02	INSERT INTO consumer_order(price, date, consumer_id) VALUE (300, now(), 5)	1 row(s) affected
✓	107	12:14:02	INSERT INTO consumer_order(price, date, consumer_id) VALUE (150, now(), 5)	1 row(s) affected
✓	108	12:14:02	COMMIT	0 row(s) affected

У таблицю consumer додалися нові клієнти(рис.2) і у таблицю consumer\_order додалися нові замовлення(рис.3).

**Select \***

**From** consumer;

id	name	email	password	gender	age
12	user532	user532@kml.com	u532pass	female	22
13	user41	user41@kml.com	u41pass	male	33
15	user23	user23@kml.com	u23pass	male	26
16	user24	user24@kml.com	u24pass	male	26
17	user533	user533@kml.com	u533pass	female	16
18	user37	user37@kml.com	u37pass	male	43
19	user42	user42@kml.com	u42pass	male	33
NULL	NULL	NULL	NULL	NULL	NULL

Рис.2 Таблиця consumer

**Select \***

**From** consumer\_order;

id	price	date	consumer_id
1	200	2020-04-28 10:34:02	1
2	300	2020-04-28 10:34:08	1
3	150	2020-04-28 10:34:12	1
4	500	2020-04-28 11:29:08	1
6	500	2020-04-28 12:14:02	5
7	300	2020-04-28 12:14:02	5
8	150	2020-04-28 12:14:02	5
NULL	NULL	NULL	NULL

Рис.3 Таблиця consumer\_order

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