

МІНІСТЕРСТВО ОСВІТИ І НАУКИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»



Звіт до лабораторної роботи №11
З ПРЕДМЕТУ “ОРГАНІЗАЦІЯ БАЗ ДАНИХ ТА ЗНАНЬ”

Виконав:

ст. гр. КН-211

Шебеко Андрій

Викладач:

Якимишин Х.М.

Львів – 2020

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

Відмінняє результати виконання запитів, вказаних після даної точки збереження.

RELEASE SAVEPOINT мітка

Видаляє точку збереження.

Хід роботи

1. Відміна транзакції після виникнення помилки:

Розпочинаємо транзакцію, виключаємо авто збереження та передаємо данні.

```
start transaction;
set autocommit = 0;
insert into shop.stops value ('1', '0', 'Ternopil', 1);
insert into shop.stops value ('2', '0', 'White Church', 2);
insert into shop.stops value ('3', '1', 'Khmelnitsk', 1);
insert into shop.stops value ('4', '12', 'Moskow', 1);
rollback;
```

Під час виконання додавання останнього (4) рядка виникає помилка:

✖ 20 17:57:53 insert into shop.stops value ('4', '12', 'Moskow', 1) Error Code: 1452. Cannot add or update a child row: a foreign ke... 0.000 sec

stops_id	way_id	city	stop_que
1	0	Ternopil	1
2	0	White Church	2
3	1	Khmelnitsk	1
NULL	NULL	NULL	NULL

Ми виконуємо відміну транзакції.

stops_id	way_id	city	stop_que
NULL	NULL	NULL	NULL

2. Виконання транзакції з поверненням до попередніх станів за допомогою точок збереження.

Виконуємо транзакцію. Зберігаємо точку відновлення.

```
start transaction;
insert into shop.stops value ('1', '0', 'Ternopil', 1);
insert into shop.stops value ('2', '0', 'White Church', 2);
insert into shop.stops value ('3', '1', 'Khmelnitsk', 1);

savepoint one;

insert into shop.stops value ('4', '0', 'White Church', 3);
insert into shop.stops value ('5', '1', 'Khmelnitsk', 5);
```

stops_id	way_id	city	stop_que
1	0	Ternopil	1
2	0	White Church	2
3	1	Khmelnitsk	1
4	0	White Church	3
5	1	Khmelnitsk	5
NULL	NULL	NULL	NULL

Виконаємо повернення до точки one. Та створимо нову під назвою two.

```
rollback to savepoint one;  
savepoint two;
```

stops_id	way_id	city	stop_que
1	0	Ternopil	1
2	0	White Church	2
3	1	Khmelnitsk	1
NULL	NULL	NULL	NULL

Додамо ще 1 рядок до таблиці.

```
insert into shop.stops value ('6', '1', 'Khmelnitsk', 5);
```

stops_id	way_id	city	stop_que
1	0	Ternopil	1
2	0	White Church	2
3	1	Khmelnitsk	1
6	1	Khmelnitsk	5
NULL	NULL	NULL	NULL

Виконаємо повернення до точки two. Та збережемо зміни за допомогою команди commit.

stops_id	way_id	city	stop_que
1	0	Ternopil	1
2	0	White Church	2
3	1	Khmelnitsk	1
NULL	NULL	NULL	NULL

Висновок: я навчився розробляти та застосовувати транзакції.