

МІНІСТЕРСТВО ОСВІТИ І НАУКИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»
Інститут комп'ютерних наук та інформаційних технологій
Кафедра систем штучного інтелекту



Звіт до практичної роботи №11
з дисципліни “ ОБДЗ ”
На тему: «Розробка та застосування транзакцій»

Виконав:
ст. гр. КН-211
Качмарик Віктор
Викладач:
Якимишин Х. М.

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.

Розробимо транзакцію, яка буде вносити дані в таблицю medicine і all_medicine. Транзакція буде відміняти всі зміни у таблицях при виникненні помилки чи іншої суперечливості.

1. Відміна транзакції.

Транзакція складається з чотирьох запитів на додавання нових препаратів у таблицю medicine. При цьому, препарату 'Portex' в таблиці all_medicine не існує, а отже, транзакція не виконується.

```
START TRANSACTION;  
INSERT INTO medicine VALUE (NULL, 9, 'AntiCorona', 1, 430);  
INSERT INTO medicine VALUE (NULL, 9, 'Mezym', 1, 50);  
INSERT INTO medicine VALUE (NULL, 4, 'AntiCorona', 1, 460);  
INSERT INTO medicine VALUE (NULL, 9, 'Portex', 1, 430);  
COMMIT;
```

Відповідь сервера:

```
mysql> INSERT INTO medicine VALUE (NULL, 9, 'Portex', 1, 430);  
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint  
fails (`pharmacy`.`medicine`, CONSTRAINT `medicine_fk1` FOREIGN KEY (`medicine_  
name`) REFERENCES `all_medicine` (`medicine_name`) ON DELETE CASCADE ON UPDATE  
CASCADE)
```

2. Успішна транзакція:

Транзакція складається з запитів на додавання тих самих препаратів у таблицю medicine після створення у таблиці all_medicine препарату 'Portex'.

```
INSERT INTO all_medicine VALUE ('Portex', 2, Null, Null, Null);  
  
START TRANSACTION;  
INSERT INTO medicine VALUE (NULL, 9, 'AntiCorona', 1, 430);  
INSERT INTO medicine VALUE (NULL, 9, 'Mezym', 1, 50);  
INSERT INTO medicine VALUE (NULL, 4, 'AntiCorona', 1, 460);  
INSERT INTO medicine VALUE (NULL, 9, 'Portex', 1, 430);  
COMMIT;
```

Результат успішного додавання чотирьох препаратів у таблицю medicine показано нижче:

```
mysql> select * from medicine;
```

| medicine_id | pharmacy_id | medicine_name | quantity | price |
|-------------|-------------|------------------|----------|-------|
| 1 | 1 | Pentalgin | 10 | 100 |
| 2 | 2 | Pentalgin | 5 | 95 |
| 3 | 1 | Mezym | 7 | 55 |
| 4 | 4 | Mezym | 8 | 31 |
| 5 | 2 | Sorbex | 6 | 122 |
| 6 | 5 | Sorbex | 10 | 150 |
| 7 | 6 | Card | 9 | 123 |
| 8 | 7 | Ramtex | 7 | 220 |
| 9 | 3 | AntiCorona | 4 | 480 |
| 10 | 3 | Mixture | 20 | 68 |
| 11 | 10 | Adhesive plaster | 30 | 17 |
| 12 | 8 | Mixture | 25 | 70 |
| 13 | 8 | Adhesive plaster | 10 | 15 |
| 14 | 3 | Bandages | 7 | 20 |
| 15 | 10 | AntiCorona | 8 | 500 |
| 16 | 9 | AntiCorona | 1 | 430 |
| 17 | 9 | Mezym | 1 | 50 |
| 18 | 4 | AntiCorona | 1 | 460 |
| 20 | 9 | AntiCorona | 1 | 430 |
| 21 | 9 | Mezym | 1 | 50 |
| 22 | 4 | AntiCorona | 1 | 460 |
| 23 | 9 | Portex | 1 | 430 |

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