МІНІСТЕРСТВО ОСВІТИ І НАУКИ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Лабораторна робота №11
З дисципліни «Організація баз даних та знань»

Виконав:
студент групи КН-210
Черещук Любомир

Перевірив:
Кандидат тех. наук, ст. викладач
Мельникова Н. І.

Львів – 2020

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

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 відміняти 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 Відміняє результати виконання запитів, вказаних після даної точки збереження.

RELEASE SAVEPOINT *мітка*

Видаляє точку збереження.

Хід роботи

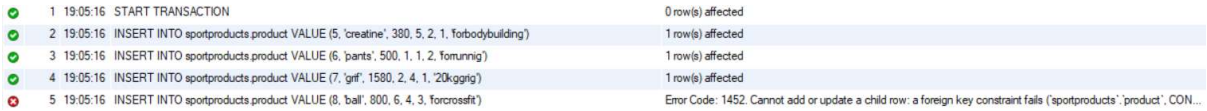
В ході роботи, потрібно продемонструвати успішне і неуспішне виконання транзакції.

Розробимо транзакцію, яка буде вносити дані в таблицю product.Транзакція буде відміняти всі зміни у таблицях при виникненні помилки чи іншої суперечливості. В таблиці product є 4 записів. Типів продуктів є 4, брендів 5 та категорій 3. Спробуємо здійснити транзакцію. Будемо додавати дані в таблицю, але одне з значень Brand_id поставимо 6, що буде некоректним значенням. Повинна отримуватись помилка.

Код транзакції:

```
START TRANSACTION;
    INSERT INTO sportproducts.product VALUE (5, 'creatine', 380, 5, 2, 1, 'forbodybuilding');
    INSERT INTO sportproducts.product VALUE (6, 'pants', 500, 1, 1, 2, 'forrunnig');
    INSERT INTO sportproducts.product VALUE (7, 'grif', 1580, 2, 4, 1, '20kggrig');
    INSERT INTO sportproducts.product VALUE (8, 'ball', 800, 6, 4, 3, 'forcrossfit');
COMMIT;
```

Результат:



1	19:05:16	START TRANSACTION	0 row(s) affected	
2	19:05:16	INSERT INTO sportproducts.product VALUE (5, 'creatine', 380, 5, 2, 1, 'forbodybuilding')	1 row(s) affected	
3	19:05:16	INSERT INTO sportproducts.product VALUE (6, 'pants', 500, 1, 1, 2, 'forrunnig')	1 row(s) affected	
4	19:05:16	INSERT INTO sportproducts.product VALUE (7, 'grif', 1580, 2, 4, 1, '20kggrig')	1 row(s) affected	
5	19:05:16	INSERT INTO sportproducts.product VALUE (8, 'ball', 800, 6, 4, 3, 'forcrossfit')	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (`sportproducts`.`product`, CON...	

Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails (`sportproducts`.`product`, CONSTRAINT `fk_Product_Brand` FOREIGN KEY (`Brand_id`) REFERENCES `brand` (`id`) ON DELETE CASCADE) 0.250 sec

SELECT * FROM sportproducts.product;

id	name	price	Brand_id	Type_id	Category_id	description
1	tshort	600.00	1	1	2	for running
2	sportshoes	1100.00	3	3	1	for bodybuilding
3	shorts	800.00	4	1	2	for running
4	protein	1200.00	1	2	1	for bodybuilding
5	creatine	380.00	5	2	1	forbodybuilding
6	pants	500.00	1	1	2	forrunnig
7	grif	1580.00	2	4	1	20kggrig

Отримали помилку, бо бренду з id 6 немає, а ми хочемо внести дані, які пов’язані з цим типом.

Записи, які не викликали помилки, додались у таблицю. Проте нам не потрібно, щоб в таблицю вносились якісь дані, якщо принаймні один запит дає помилку. Тому виконаємо команду ROLLBACK і зміни, які зробила попередня транзакція, відміняються.

Результат виконання попереднього запиту після ROLLBACK:

id	name	price	Brand_id	Type_id	Category_id	description
1	tshort	600.00	1	1	2	for running
2	sportshoes	1100.00	3	3	1	for bodybuilding
3	shorts	800.00	4	1	2	for running
4	protein	1200.00	1	2	1	for bodybuilding
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Тепер проведемо ту саму транзакцію, проте перед тим додамо в таблицю brand 6-й бренд. Код скрипта:

```
INSERT INTO `sportproducts`.`brand` (`id`, `name`, `country`) VALUES ('6', 'nike', 'usa');
START TRANSACTION;
    INSERT INTO sportproducts.product VALUE (5, 'creatine', 380, 5, 2, 1, 'forbodybuilding');
    INSERT INTO sportproducts.product VALUE (6, 'pants', 500, 1, 1, 2, 'forrunnig');
```

```
INSERT INTO sportproducts.product VALUE (7, 'grif', 1580, 2, 4, 1, '20kggrig');
INSERT INTO sportproducts.product VALUE (8, 'ball', 800, 6, 4, 3, 'forcrossfit');
COMMIT;
```

Результат:

10	19:20:51	INSERT INTO sportproducts.product VALUE (5, 'creatine', 380, 5, 2, 1, 'forbodybuilding')	
11	19:20:52	INSERT INTO sportproducts.product VALUE (6, 'pants', 500, 1, 1, 2, 'forrunnig')	
12	19:20:52	INSERT INTO sportproducts.product VALUE (7, 'grif', 1580, 2, 4, 1, '20kggrig')	
13	19:20:52	INSERT INTO sportproducts.product VALUE (8, 'ball', 800, 6, 4, 3, 'forcrossfit')	
14	19:20:52	COMMIT	

id	name	price	Brand_id	Type_id	Category_id	description
1	tshort	600.00	1	1	2	for running
2	sportshoes	1100.00	3	3	1	for bodybuilding
3	shorts	800.00	4	1	2	for running
4	protein	1200.00	1	2	1	for bodybuilding
5	creatine	380.00	5	2	1	forbodybuilding
6	pants	500.00	1	1	2	forrunnig
7	grif	1580.00	2	4	1	20kggrig
8	ball	800.00	6	4	3	forcrossfit

Як бачимо, транзакція пройшла успішно і всі команди виконались правильно.

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