

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА
ПОЛІТЕХНІКА”

Кафедра систем штучного інтелекту



Лабораторна робота 11
з організації баз даних та знань

Виконав:

Студент групи КН-208

Воробель Адріан

Викладач:

Якимишин Х. М.

Львів – 2020р.

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*

Відмінляє результати виконання запитів, вказаних після даної точки збереження.

RELEASE SAVEPOINT *мітка*

Видаляє точку збереження.

Хід роботи.

1. Відміна транзакції

```
start transaction;

savepoint befor_transaction;

insert      into      autoshop.cardistributor(car_id,
distrib_id) value (11,1);

insert      into      autoshop.cardistributor(car_id,
distrib_id) value (12,2);

insert      into      autoshop.cardistributor(car_id,
distrib_id) value (13,1);

insert      into      autoshop.cardistributor(car_id,
distrib_id) value (14,3);

insert      into      autoshop.cardistributor(car_id,
distrib_id) value (15,3);

insert      into      autoshop.cardistributor(car_id,
distrib_id) value (16,2);

insert      into      autoshop.cardistributor(car_id,
distrib_id) value (17,5);

insert      into      autoshop.cardistributor(car_id,
distrib_id) value (18,5);

insert      into      autoshop.cardistributor(car_id,
distrib_id) value (19,5);

insert      into      autoshop.cardistributor(car_id,
distrib_id) value (20,5);

commit;

rollback to befor_transaction;
```

Так як немає `distrib_id` із значенням 5, то `mysql` каже що виникла помилка.

```

1 • use autoshop;
2
3 • set autocommit = 0;
4   -- неуспінна операція
5 • start transaction;
6 • savepoint befor_transaction;
7 • insert into autoshop.cardistributor(car_id, distrib_id) value (11,1);
8 • insert into autoshop.cardistributor(car_id, distrib_id) value (12,2);
9 • insert into autoshop.cardistributor(car_id, distrib_id) value (13,1);
10 • insert into autoshop.cardistributor(car_id, distrib_id) value (14,3);
11 • insert into autoshop.cardistributor(car_id, distrib_id) value (15,3);
12 • insert into autoshop.cardistributor(car_id, distrib_id) value (16,2);
13 • insert into autoshop.cardistributor(car_id, distrib_id) value (17,5);
14 • insert into autoshop.cardistributor(car_id, distrib_id) value (18,5);
15 • insert into autoshop.cardistributor(car_id, distrib_id) value (19,5);
16 • insert into autoshop.cardistributor(car_id, distrib_id) value (20,5);
17 • commit;

```

Automatic context help disabled. Use the toolbar manually get help for the current caret position or toggle automatic help

Context Help Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 192	00:20:34	insert into autoshop.cardistributor(car_id, distrib_id) value (16,2)	1 row(s) affected	0.000 sec
✗ 193	00:20:34	insert into autoshop.cardistributor(car_id, distrib_id) value (17,5)	Error Code: 1452. Cannot add or update a child row: a foreign key constr...	0.000 sec

Але деяку частину воно все ж встигло записати.

	id	car_id	distrib_id
	7	7	1
	43	8	4
	44	9	4
	45	10	4
	53	11	1
	54	12	2
	55	13	1
	56	14	3
	57	15	3
	58	16	2
*	NULL	NULL	NULL

Відкотимо зміни за допомогою директиви rollback та завчасно створеної точки збереження.

	id	car_id	distrib_id
▶	1	1	3
	2	2	2
	3	3	3
	4	4	1
	5	5	2
	6	6	1
	7	7	1
	43	8	4
	44	9	4
	45	10	4

2. Успішна транзакція

```
start transaction;
```

```
savepoint befor_transaction;
```

```

insert      into      autoshop.cardistributor(car_id,
distrib_id) value (11,1);

insert      into      autoshop.cardistributor(car_id,
distrib_id) value (12,2);

insert      into      autoshop.cardistributor(car_id,
distrib_id) value (13,1);

insert      into      autoshop.cardistributor(car_id,
distrib_id) value (14,3);

insert      into      autoshop.cardistributor(car_id,
distrib_id) value (15,3);

insert      into      autoshop.cardistributor(car_id,
distrib_id) value (16,2);

insert      into      autoshop.cardistributor(car_id,
distrib_id) value (17,1);

insert      into      autoshop.cardistributor(car_id,
distrib_id) value (18,1);

insert      into      autoshop.cardistributor(car_id,
distrib_id) value (19,1);

insert      into      autoshop.cardistributor(car_id,
distrib_id) value (20,2);

commit;

```

	id	car_id	distrib_id
	43	8	4
	44	9	4
	45	10	4
	60	11	1
	61	12	2
	62	13	1
	63	14	3
	64	15	3
	65	16	2
	66	17	1
	67	18	1

Зміни успішно записані.

Висновок.

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