

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Інститут комп'ютерних наук та інформаційних технологій
Кафедра «Системи штучного інтелекту»



Лабораторна робота №11
з дисципліни: «ОБДЗ»

Виконав студент
групи КН-208
Жеребецький О. В.
Прийняла:
асистент
Якимішин Х.М.

Львів-2020

Хід роботи.

- Запам'ятаємо що в кінці транзакції варто ставити **commit** для збереження змін і **завершення транзакції**
- Автоматично стоїть `autocommit = 0` що означає, що коміт автоматично не відбувається.

1. Виконаємо транзакцію з заповненням даних.

```
start transaction;
```

```
INSERT INTO comment VALUES ("Very good product", '2020-02-01', 4, NULL);  
INSERT INTO comment VALUES ("Really chip product", '2020-05-17', 4, NULL);  
INSERT INTO comment VALUES ("Very convenient", '2020-02-15', 5, null);  
INSERT INTO comment VALUES ("Too expensive", '2020-08-23', 2, null);
```

	comment_text	comment_time	comment_mark	comment_id
	text of coment3	2020-05-03 00:00:00	3	3
	text of coment2	2020-10-01 00:00:00	4	4
	text of coment5	2020-01-15 00:00:00	3	5
	text of coment4	2020-09-14 00:00:00	2	6
	Very good product	2020-02-01 00:00:00	4	7
	Really chip product	2020-05-17 00:00:00	4	8
	Very convenient	2020-02-15 00:00:00	5	9
	Too expensive	2020-08-23 00:00:00	2	10

Дані відобразилися в таблицю, а тепер відмінимо дану транзакцією командою ROLLBACK.

```
rollback;
```

	comment_text	comment_time	comment_mark	comment_id
▶	text of coment1	2020-12-31 00:00:00	5	1
	text of coment1	2020-12-31 00:00:00	5	2
	text of coment3	2020-05-03 00:00:00	3	3
	text of coment2	2020-10-01 00:00:00	4	4
	text of coment5	2020-01-15 00:00:00	3	5
	text of coment4	2020-09-14 00:00:00	2	6
•	NULL	NULL	NULL	NULL

Таблиця містить тільки початкові дані.

2. Використовуючи SAVEPOINT відмінимо зміни лише для частини даних

```

start transaction;
INSERT INTO comment VALUES ("Very good product", '2020-02-01', 4, NULL);
INSERT INTO comment VALUES ("Really chip product", '2020-05-17', 4, NULL);
savepoint checkpoint;
INSERT INTO comment VALUES ("Very convenient", '2020-02-15', 5, null);
INSERT INTO comment VALUES ("Too expensive", '2020-08-23', 2, null);

```

	comment_text	comment_time	comment_mark	comment_id
	text of coment2	2020-10-01 00:00:00	4	4
	text of coment5	2020-01-15 00:00:00	3	5
	text of coment4	2020-09-14 00:00:00	2	6
	Very good product	2020-02-01 00:00:00	4	24
	Really chip product	2020-05-17 00:00:00	4	25
	Very convenient	2020-02-15 00:00:00	5	26
	Too expensive	2020-08-23 00:00:00	2	27
•	NULL	NULL	NULL	NULL

Всі зміни записані у таблиці, а тепер відмінімо дані в точці checkpoint.

```

rollback to checkpoint;
commit;

```

	comment_text	comment_time	comment_mark	comment_id
	text of coment1	2020-12-31 00:00:00	5	2
	text of coment3	2020-05-03 00:00:00	3	3
	text of coment2	2020-10-01 00:00:00	4	4
	text of coment5	2020-01-15 00:00:00	3	5
	text of coment4	2020-09-14 00:00:00	2	6
	Very good product	2020-02-01 00:00:00	4	24
	Really chip product	2020-05-17 00:00:00	4	25
•	NULL	NULL	NULL	NULL

Дані збереглися але лише частина, так як іншу половину ми відмінили командою ROLLBACK TO.

- Покажемо перекривання savepoint –ів. Створимо 3 точки збереження і видалимо спочатку останню а потім першу і побачимо що все пов'язане з середньою точкою видалилося також. Будемо додавати дані.

- Які дані були спочатку

	comment_text	comment_time	comment_mark	comment_id
▶	text of coment1	2020-12-31 00:00:00	5	1
	text of coment1	2020-12-31 00:00:00	5	2
	text of coment2	2020-10-01 00:00:00	4	4
✱	NULL	NULL	NULL	NULL

- Код (створимо три savepoint)

```
start transaction;
savepoint before_insert;
INSERT INTO comment VALUES ("Very good product", '2020-02-01', 4, NULL);
INSERT INTO comment VALUES ("Really chip product", '2020-05-17', 4, NULL);
savepoint two_inserted;
INSERT INTO comment VALUES ("Very convenient", '2020-02-15', 5, null);
savepoint last_point;
INSERT INTO comment VALUES ("Too expensive", '2020-08-23', 2, null);
```

- Наповнення після виконання коду

	comment_text	comment_time	comment_mark	comment_id
▶	text of coment1	2020-12-31 00:00:00	5	1
	text of coment1	2020-12-31 00:00:00	5	2
	text of coment2	2020-10-01 00:00:00	4	4
	Very good product	2020-02-01 00:00:00	4	40
	Really chip product	2020-05-17 00:00:00	4	41
	Very convenient	2020-02-15 00:00:00	5	42
	Too expensive	2020-08-23 00:00:00	2	43
✱	NULL	NULL	NULL	NULL

- Вернемся до моменту перед тим як додали останній елемент.

rollback to last_point;

	comment_text	comment_time	comment_mark	comment_id
▶	text of coment1	2020-12-31 00:00:00	5	1
	text of coment1	2020-12-31 00:00:00	5	2
	text of coment2	2020-10-01 00:00:00	4	4
	Very good product	2020-02-01 00:00:00	4	40
	Really chip product	2020-05-17 00:00:00	4	41
	Very convenient	2020-02-15 00:00:00	5	42
✱	NULL	NULL	NULL	NULL

- Тепер же використаємо перший savepoint для повернення до моменту коли ще нічого не добавилось. Отримаємо такі дані як і були на початку:

`rollback to before_insert;`

	comment_text	comment_time	comment_mark	comment_id
▶	text of coment1	2020-12-31 00:00:00	5	1
	text of coment1	2020-12-31 00:00:00	5	2
	text of coment2	2020-10-01 00:00:00	4	4
•	NULL	NULL	NULL	NULL

4. Спробуємо створити точку збереження в транзакції. І поспробуємо спочатку зробити відкат всієї транзакції, після чого викласти rollback до точки збереження.

- Початкові дані в таблиці

	comment_text	comment_time	comment_mark	comment_id
▶	text of coment1	2020-12-31 00:00:00	5	1
	text of coment1	2020-12-31 00:00:00	5	2
	text of coment2	2020-10-01 00:00:00	4	4
•	NULL	NULL	NULL	NULL

- Код

```
start transaction;
INSERT INTO comment VALUES ("Very good product", '2020-02-01', 4, NULL);
INSERT INTO comment VALUES ("Really chip product", '2020-05-17', 4, NULL);
INSERT INTO comment VALUES ("Very convenient", '2020-02-15', 5, null);
savepoint last_point;
INSERT INTO comment VALUES ("Too expensive", '2020-08-23', 2, null);

rollback;
rollback to last_point;
select * from comment;
```

- При виконанні успішно виконується відміна усієї транзакції. Але повернення до save point не успішне оскільки створення цього поінту також стерлося.

✓	48	01:47:35	rollback	0 row(s) affected
✗	49	01:47:35	rollback to last_point	Error Code: 1305. SAVEPOINT last_point does not exist

- Отож дані після операцій

	comment_text	comment_time	comment_mark	comment_id
▶	text of coment1	2020-12-31 00:00:00	5	1
	text of coment1	2020-12-31 00:00:00	5	2
	text of coment2	2020-10-01 00:00:00	4	4
*	NULL	NULL	NULL	NULL

5. Відтворимо ситуацію коли ти щось випадково не то видалив. Али маєш сейф поінт і можеш відкотити все до видалення.

- Виведемо всі дані з таблиці comment.

	comment_text	comment_time	comment_mark	comment_id
▶	text of coment1	2020-12-31 00:00:00	5	1
	text of coment1	2020-12-31 00:00:00	5	2
	text of coment2	2020-10-01 00:00:00	4	4
	Very good product	2020-02-01 00:00:00	4	48
	Really chip product	2020-05-17 00:00:00	4	49
	Very convenient	2020-02-15 00:00:00	5	50
	Too expensive	2020-08-23 00:00:00	2	51
	Verv convenient	2020-12-11 00:00:00	2	52

- Код

```
start transaction;
INSERT INTO comment VALUES ("Very convenient", '2020-12-11', 2, null);
savepoint before_delete;
delete from comment where comment_mark<=4;
|
rollback to before_delete;
select * from comment;
```

- Що лишилось після випадкового видалення не тих даних

	comment_text	comment_time	comment_mark	comment_id
▶	text of coment1	2020-12-31 00:00:00	5	1
	text of coment1	2020-12-31 00:00:00	5	2
	Very convenient	2020-02-15 00:00:00	5	50
★	NULL	NULL	NULL	NULL

- Тепер запустимо команду rollback to , що дасть нам змогу вернутись до моменту перед видаленням. В результаті отримаємо назад усі дані:

	comment_text	comment_time	comment_mark	comment_id
▶	text of coment1	2020-12-31 00:00:00	5	1
	text of coment1	2020-12-31 00:00:00	5	2
	text of coment2	2020-10-01 00:00:00	4	4
	Very good product	2020-02-01 00:00:00	4	48
	Really chip product	2020-05-17 00:00:00	4	49
	Very convenient	2020-02-15 00:00:00	5	50
	Too expensive	2020-08-23 00:00:00	2	51
	Verv convenient	2020-12-11 00:00:00	2	52

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