ЛАБОРАТОРНА РОБОТА 3. РОБОТА З ТРАНЗАКЦІЯМИ

Мета роботи: вивчити основи роботи з механізмом транзакцій на прикладі СУБД MySQL.

Хід роботи

Увага! Перш ніж перейти до виконання лабораторної роботи, необхідно створити тимчасову базу даних, використовуючи запити, використані у лабораторній роботі 2. В усіх подальших пунктах даної лабораторної роботи передбачається використання тимчасової бази даних.

1. Створити запит, що ілюструє роботу механізму транзакцій при додаванні даних в одну таблицю

Розглянемо послідовність дій при створенні та використанні запиту, за допомогою якого запускається транзакція, до таблиці supplied додається новий запис, а потім імітується ситуація некоректного або коректного завершення транзакції. Стан таблиці контролюється до початку транзакції, під час виконання транзакції та після її завершення. Для цього необхідно виконати наступну послідовність дій.

```
SELECT supplied.contract_number, supplied.supplied_product, supplied.supplied_cost, supplied.supplied_amount,
   supplier.supplier_address, contract.contract_date
FROM supplied, contract, supplier
WHERE contract.contract_number = supplied.contract_number AND supplier.supplier_id = contract.supplier_id
AND contract.contract_number = 1;

SET AUTOCOMMIT = 0;
START TRANSACTION;
INSERT INTO supplied VALUES (1, 'Vacuum cleaner', 22, 390);

SELECT supplied.contract_number, supplied.supplied_product, supplied.supplied_cost, supplied.supplied_amount,
   supplier.supplier_address, contract.contract_date
FROM supplied, contract, supplier
WHERE contract.contract_number = supplied.contract_number AND supplier.supplier_id = contract.supplier_id
AND contract.contract_number = 1;

ROLLBACK;

SELECT supplied.contract_number, supplied.supplied_product, supplied.supplied_cost, supplied.supplied_amount,
   supplier.supplier_address, contract.contract_date
FROM supplied, contract, supplier
WHERE contract.contract_number = supplied.contract_number AND supplier.supplier_id = contract.supplier_id
AND contract.contract_number = 1;
```

Запити SELECT дозволяють вивести дані, які ілюструють стан таблиці до початку транзакції (рисунок 3.1), в процесі виконання транзакції та після завершення транзакції.

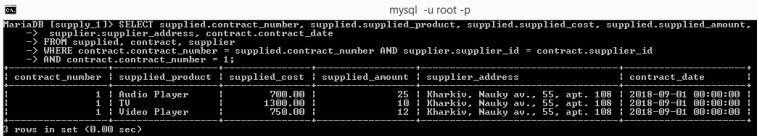

Рисунок 3.1

Як видно з наведених даних, новий запис у таблиці з'являється (рисунок 3.2), а потім зникає (рисунок 3.3).

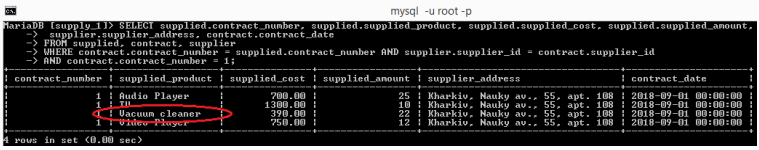

Рисунок 3.2

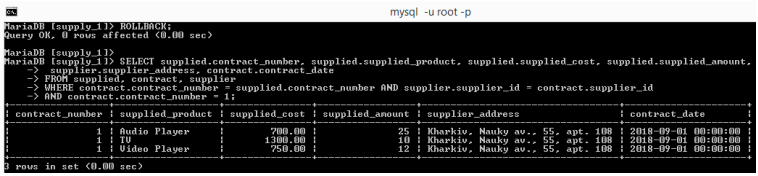

Рисунок 3.3

Тепер необхідно розглянути ситуацію коректного завершення транзакції. Для цього у наведеному тексті запиту необхідно змінити оператор ROLLBACK на COMMIT. Виконати запит та проаналізувати отримані результати.

2. Створити запит, що ілюструє роботу механізму транзакцій при додаванні даних в декілька таблиць

Розглянемо послідовність дій при створенні та використання запиту, за допомогою якого запускається транзакція, а потім створюється новий постачальник, з цим постачальником укладається договір на постачання, за цим договором поставляється продукція. Імітується ситуація некоректного або коректного завершення транзакції. Стан таблиць контролюється до початку транзакції, в процесі виконання транзакції та після завершення транзакції. Для цього необхідно виконати наступну послідовність дій.

```
SELECT * FROM supplier;
SELECT * FROM contract;
SELECT * FROM supplied;

SET AUTOCOMMIT = 0;
START TRANSACTION;
INSERT INTO supplier (supplier_id, supplier_address, supplier_phone)
   VALUES (6, 'Kyiv, Velyka Vasylkivska st., 55', '');
INSERT INTO contract (contract_date, supplier_id, contract_note)
   VALUES ('2018-12-12', 6, '');
INSERT INTO supplied VALUES (6, 'Vacuum cleaner', 22, 390);
INSERT INTO supplied VALUES (6, 'Coffee machine', 33, 90);

SELECT * FROM supplier;
SELECT * FROM contract;
SELECT * FROM supplied;

ROLLBACK;

SELECT * FROM supplier;
SELECT * FROM contract;
SELECT * FROM supplied;
```

Запити SELECT дозволяють вивести дані, які ілюструють стан таблиць до початку транзакції, в процесі виконання транзакції та після завершення транзакції. Як буде видно з отриманих даних, нові записи у таблицях з'являються, а потім зникають.

Тепер необхідно розглянути ситуацію коректного завершення транзакції. Для цього у наведеному тексті запиту необхідно змінити оператор ROLLBACK на COMMIT. Виконати запит та проаналізувати отримані результати.

3. Створити запит, що ілюструє роботу механізму транзакцій при зміненні даних в декількох таблицях

Розглянемо послідовність дій при створенні та використанні запиту, за допомогою якого запускається транзакція, потім змінюються дані, введені у таблиці при виконанні попереднього запиту. Імітується ситуація некоректного або коректного завершення транзакції. Стан таблиць контролюється до початку транзакції, в процесі виконання транзакції та після завершення транзакції. Для цього необхідно виконати наступну послідовність дій.

```sql
ALTER TABLE contract
DROP FOREIGN KEY contract_ibfk_1;

ALTER TABLE contract
ADD CONSTRAINT contract_ibfk_1 FOREIGN KEY (supplier_id) REFERENCES supplier(supplier_id) ON DELETE CASCADE ON UPDATE CASCADE;

SELECT * FROM supplier;
SELECT * FROM contract;
SELECT * FROM supplied;

SET AUTOCOMMIT = 0;
START TRANSACTION;
UPDATE supplier SET supplier_id = 22 WHERE supplier_id = 6;
UPDATE supplied SET supplied_cost = supplied_cost * 1.1 WHERE contract_number = 8;

SELECT * FROM supplier;
SELECT * FROM contract;
SELECT * FROM supplied WHERE contract_number = 8;

ROLLBACK;

SELECT * FROM supplier;
SELECT * FROM contract;
SELECT * FROM supplied WHERE contract_number = 8;
```

Запити SELECT дозволяють вивести дані, які ілюструють стан таблиць до початку транзакції, в процесі виконання транзакції та після завершення транзакції. Як буде видно з отриманих даних, оновлені записи у таблицях з'являються, а потім зникають.

Тепер необхідно розглянути ситуацію коректного завершення транзакції. Для цього у наведеному тексті запиту необхідно змінити оператор ROLLBACK на COMMIT. Виконати запит та проаналізувати отримані результати.

4. Створити запит, що ілюструє роботу механізму транзакцій при видаленні даних з декількох таблиць

Розглянемо послідовність дій при створенні та використанні запиту, за допомогою якого запускається транзакція, в рамках якої видаляється постачальник, який був створений при виконанні запиту 2 та дані якого були змінені при виконанні запиту 3. З урахуванням механізму контролю посилкової цілісності, що використовується (CASCADE), дані будуть видалені у декількох таблицях. Імітується ситуація некоректного або коректного завершення транзакції. Стан таблиць контролюється до початку транзакції, в процесі виконання транзакції та після завершення транзакції. Для цього необхідно виконати наступну послідовність дій.

```
ALTER TABLE supplied
DROP FOREIGN KEY supplied_ibfk_1;

ALTER TABLE supplied
ADD CONSTRAINT supplied_ibfk_1 FOREIGN KEY (contract_number) REFERENCES contract(contract_number) ON DELETE CASCADE ON UPDATE CASCADE;

SELECT * FROM supplier;
SELECT * FROM contract;
SELECT * FROM supplied;

SET AUTOCOMMIT = 0;
START TRANSACTION;
DELETE FROM supplier WHERE supplier_id = 22;

SELECT * FROM supplier;
SELECT * FROM contract;
SELECT * FROM supplied;

ROLLBACK;

SELECT * FROM supplier;
SELECT * FROM contract;
SELECT * FROM supplied;
```

Запити SELECT дозволяють вивести дані, які ілюструють стан таблиць до початку транзакції, в процесі виконання транзакції та після завершення транзакції. Як буде видно з отриманих даних, видалені записи у таблицях зникають, а потім з'являються.

Тепер необхідно розглянути ситуацію коректного завершення транзакції. Для цього у наведеному тексті запиту необхідно змінити оператор ROLLBACK на COMMIT. Виконати запит та проаналізувати отримані результати.

5. Оформити звіт з лабораторної роботи

У звіт включити основні етапи виконання лабораторної роботи та знімки екрану, що їх демонструють.

6. Питання для самоконтролю

1. Що таке транзакція?
2. Таблиці яких типів у СУБД MySQL підтримують транзакції?
3. Таблиці яких типів у СУБД MySQL не підтримують транзакції?
4. Яким чином у СУБД MySQL можна відключити режим автоматичного завершення транзакцій?
5. Який оператор використовується для завершення транзакції?
6. Який оператор використовується для відкату змін, виконаних транзакцією?
7. За допомогою якої команди у СУБД MySQL можна включити режим автоматичного завершення транзакцій для окремої послідовності операторів?
8. З таблицями якого типу можуть бути використані оператори SAVEPOINT та ROLLBACK TO SAVEPOINT?
9. Яке призначення операторів SAVEPOINT та ROLLBACK TO SAVEPOINT?

10. З якими проблемами пов’язане паралельне виконання транзакцій?

11. Які існують рівні ізоляції транзакцій та які проблеми кожен з цих рівнів дозволяє вирішити?

12. Який тип таблиць використовується у MySQL за замовченням (починаючи з версії 5.5)?

13. Які рівні ізоляції транзакцій підтримує InnoDB?

14. Який рівень ізоляції транзакцій за замовченням використовується у InnoDB?