Міністерство освіти і науки України
Одеський національний політехнічний університет
Інститут комп’ютерних систем
Кафедра інформаційних систем

Лабораторна робота №10
з дисципліни «Операційні Системи»

Тема: «Керування процесами-транзакціями в базах даних. Частина 2»

Виконав:
студент групи АІ-204
Плаксивий Д.В.

Перевірив:
Блажко О. А.
Дрозд М.О.

Одеса – 2021

Мета роботи: дослідити поведінку процесів-транзакцій в базах даних та засоби керуванням ними через механізм блокування з використанням сучасних систем керування базами даних.

Завдання

Для кожної транзакції підготував окремий термінал, в якому виконав команду доступу до вашої БД з використанням утиліти psql.

Завдання 1. Аналіз роботи багато версійного протоколу

В завданні 1 рішення попередньої лабораторної роботи було створено таблицю з декількома рядками.

Підготував чотири транзакції за прикладом з рисунку 2:

– T1 – отримання номеру транзакції, внесення нового рядка в таблицю та перегляд вмісту таблиці;

– T2 – постійний перегляд вмісту таблиці

```
plaksivij_danilo=> START TRANSACTION
plaksivij_danilo-> ;
START TRANSACTION
plaksivij_danilo=> INSERT INTO person VALUES (3, 'Artemenko', '01/09/2000');
INSERT 0 1
plaksivij_danilo=> SELECT xmin, xmax, name FROM person;
 xmin | xmax |         name
------+------+----------------------
 2113 |    0 | Plaksivij
 2120 |    0 | Borshakov
 3045 |    0 | Artemenko
(3 rows)

plaksivij_danilo=> COMMIT;
COMMIT
plaksivij_danilo=>
```

```
plaksivij_danilo=> START TRANSACTION;
START TRANSACTION
plaksivij_danilo=> START TRANSACTION;
WARNING:  there is already a transaction in progress
START TRANSACTION
plaksivij_danilo=> SELECT xmin, xmax, name FROM person;
 xmin | xmax |         name
------+------+----------------------
 2113 |    0 | Plaksivij
 2120 |    0 | Borshakov
(2 rows)

plaksivij_danilo=> SELECT xmin, xmax, name FROM person;
 xmin | xmax |         name
------+------+----------------------
 2113 |    0 | Plaksivij
 2120 |    0 | Borshakov
 3045 |    0 | Artemenko
(3 rows)

plaksivij_danilo=>
```

– Т3 – видалення рядку з наступною відміною цієї операції;

```
plaksivij_danilo=> START TRANSACTION;
START TRANSACTION
plaksivij_danilo=> DELETE FROM person WHERE p_id = 3;
DELETE 1
plaksivij_danilo=> SELECT xmin, xmax, name FROM person;
 xmin | xmax |         name
------+------+----------------------
 2113 |    0 | Plaksivij
 2120 |    0 | Borshakov
(2 rows)

plaksivij_danilo=> ABORT;
ROLLBACK
plaksivij_danilo=>
```

```
plaksivij_danilo=> SELECT xmin, xmax, name FROM person;
 xmin | xmax |         name
------+------+----------------------
 2113 |    0 | Plaksivij
 2120 |    0 | Borshakov
 3045 |    0 | Artemenko
(3 rows)

plaksivij_danilo=> SELECT xmin, xmax, name FROM person;
 xmin | xmax |         name
------+------+----------------------
 2113 |    0 | Plaksivij
 2120 |    0 | Borshakov
 3045 | 3047 | Artemenko
(3 rows)
```

– Т4 – зміна значення однієї з колонок рядка.

```
plaksivij_danilo=> START TRANSACTION;
START TRANSACTION
plaksivij_danilo=> UPDATE person SET name = 'Artemov' WHERE p_id = 3;
UPDATE 1
plaksivij_danilo=> SELECT xmin, xmax, name FROM person;
 xmin | xmax |         name
------+------+----------------------
 2113 |    0 | Plaksivij
 2120 |    0 | Borshakov
 3049 |    0 | Artemov
(3 rows)

plaksivij_danilo=> COMMIT;
COMMIT
plaksivij_danilo=>
```

```
plaksivij_danilo=> SELECT xmin, xmax, name FROM person;
 xmin | xmax |         name
------+------+----------------------
 2113 |    0 | Plaksivij
 2120 |    0 | Borshakov
 3045 | 3049 | Artemenko
(3 rows)

plaksivij_danilo=> SELECT xmin, xmax, name FROM person;
 xmin | xmax |         name
------+------+----------------------
 2113 |    0 | Plaksivij
 2120 |    0 | Borshakov
 3049 |    0 | Artemov
(3 rows)
```

Завдання 2. Аналіз стану транзакцій на різних рівнях багаторівневого блокування

Виконав послідовно в двох терміналах наступні комбінації блокувань таблиці: IX-IS, SIX-IX, SIX-IS.

IX-IS

```
plaksivij_danilo=> START TRANSACTION;
START TRANSACTION
plaksivij_danilo=> LOCK TABLE person IN ROW EXCLUSIVE MODE;
LOCK TABLE
plaksivij_danilo=>
```

```
START TRANSACTION
plaksivij_danilo=> LOCK TABLE person IN ROW SHARE MODE;
LOCK TABLE
plaksivij_danilo=>
```

```
plaksivij_danilo=> SELECT virtualtransaction,locktype,relation,pid,mode,granted
FROM pg_locks WHERE locktype = 'relation' AND relation = 16678;
 virtualtransaction | locktype | relation | pid  |         mode          | grant
ed
--------------------+----------+----------+------+-----------------------+------
---
 37/628             | relation |    16678 | 6270 | AccessShareLock       | t
 37/628             | relation |    16678 | 6270 | RowExclusiveLock      | f
 4/104087           | relation |    16678 | 1369 | ShareRowExclusiveLock | t
 39/422             | relation |    16678 | 6278 | AccessShareLock       | t
 39/422             | relation |    16678 | 6278 | RowExclusiveLock      | f
 42/12              | relation |    16678 | 6824 | AccessShareLock       | t
 42/12              | relation |    16678 | 6824 | RowExclusiveLock      | f
 33/1991            | relation |    16678 | 1464 | RowShareLock          | t
(8 rows)
```

SIX-IX

```
plaksivij_danilo=> START TRANSACTION;
START TRANSACTION
plaksivij_danilo=> LOCK TABLE person IN SHARE ROW EXCLUSIVE MODE;
LOCK TABLE
```

```
START TRANSACTION
plaksivij_danilo=> LOCK TABLE person IN ROW EXCLUSIVE MODE;
```

```
plaksivij_danilo=> SELECT virtualtransaction,locktype,relation,pid,mode,granted
FROM pg_locks WHERE locktype = 'relation' AND relation = 16678;
 virtualtransaction | locktype | relation | pid  |         mode          | grant
ed
--------------------+----------+----------+------+-----------------------+------
---
 37/628             | relation |    16678 | 6270 | AccessShareLock       | t
 37/628             | relation |    16678 | 6270 | RowExclusiveLock      | f
 4/104087           | relation |    16678 | 1369 | ShareRowExclusiveLock | t
 39/422             | relation |    16678 | 6278 | AccessShareLock       | t
 39/422             | relation |    16678 | 6278 | RowExclusiveLock      | f
 42/14              | relation |    16678 | 6824 | RowExclusiveLock      | f
 33/1991            | relation |    16678 | 1464 | RowShareLock          | t
 41/674             | relation |    16678 | 6817 | RowExclusiveLock      | f
(8 rows)

plaksivij_danilo=>
```

SIX – IS

```
plaksivij_danilo=> START TRANSACTION;
START TRANSACTION
plaksivij_danilo=> LOCK TABLE person in SHARE ROW EXCLUSIVE MODE;
```

```
plaksivij_danilo=> Start transaction;
START TRANSACTION
plaksivij_danilo=> LOCK TABLE person in ROW SHARE MODE;
LOCK TABLE
```

```
plaksivij_danilo=> SELECT virtualtransaction,locktype,relation,pid,mode,granted
FROM pg_locks WHERE locktype = 'relation' AND relation = 16678;
 virtualtransaction | locktype | relation | pid  |         mode          | grant
ed
--------------------+----------+----------+------+-----------------------+------
---
 37/628             | relation |    16678 | 6270 | AccessShareLock       | t
 37/628             | relation |    16678 | 6270 | RowExclusiveLock      | f
 4/104087           | relation |    16678 | 1369 | ShareRowExclusiveLock | t
 39/422             | relation |    16678 | 6278 | AccessShareLock       | t
 39/422             | relation |    16678 | 6278 | RowExclusiveLock      | f
 33/1991            | relation |    16678 | 1464 | RowShareLock          | t
(6 rows)

plaksivij_danilo=>
```

Завдання 3. Керування квазіпаралельним виконанням транзакцій на різних рівнях ізоляції транзакцій

Підготував транзакції, які було створено у завданні 3.1 рішення попередньої лабораторної роботи, а саме, створив дві транзакції, кожна з яких повинна включати такі операції:

- операція читання першого рядку таблиці;

- операція редагування однієї із змінних таблиці в першому рядку;

- повторна операція читання першого рядку таблиці;

- операція фіксації всіх змін.

1.1 Виконав роботу транзакцій при умові їх роботи на рівні ізоляції READ COMMITED. Проаналізував реакцію СКБД на операцію UPDATE 2-ї транзакції (яка виконується пізніше) та дав свої висновки.

```
plaksivij_danilo=> START TRANSACTION;
START TRANSACTION
plaksivij_danilo=> SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
SET
plaksivij_danilo=> SELECT * FROM person;
 p_id |         name          |     bd
------+-----------------------+------------
    2 | Plaksivij             | 2000-03-02
    1 | Borshakov             | 2000-03-04
    3 | Artemov               | 2000-01-09
(3 rows)

plaksivij_danilo=> UPDATE person SET name = 'Karpova' WHERE p_id = 1;
UPDATE 1
plaksivij_danilo=> SELECT * FROM person;
 p_id |         name          |     bd
------+-----------------------+------------
    2 | Plaksivij             | 2000-03-02
    3 | Artemov               | 2000-01-09
    1 | Karpova               | 2000-03-04
(3 rows)

plaksivij_danilo=> COMMIT;
COMMIT
```

```
plaksivij_danilo=> START TRANSACTION;
START TRANSACTION
plaksivij_danilo=> SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
SET
plaksivij_danilo=> SELECT * FROM person;
 p_id |         name          |     bd
------+-----------------------+------------
    2 | Plaksivij             | 2000-03-02
    1 | Borshakov             | 2000-03-04
    3 | Artemov               | 2000-01-09
(3 rows)

plaksivij_danilo=> UPDATE person SET bd = '07/01/2000' WHERE p_id = 1;
UPDATE 1
plaksivij_danilo=> SELECT * FROM person;
 p_id |         name          |     bd
------+-----------------------+------------
    2 | Plaksivij             | 2000-03-02
    3 | Artemov               | 2000-01-09
    1 | Karpova               | 2000-07-01
(3 rows)

plaksivij_danilo=>
```

Під час зміни 2 транзакції, СКБД направило транзакцію у стан очікування, поки не зберіглася 1 транзакція.

1.2 Повторив роботу транзакцій при умові їх роботи на рівні ізоляції REPEATABLE READ. Проаналізував реакцію СКБД на операцію UPDATE 2-ї транзакції (яка виконується пізніше) та дайте свої висновки.

```
START TRANSACTION
plaksivij_danilo=> SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
SET
plaksivij_danilo=> SELECT * FROM person;
 p_id |         name          |     bd
------+-----------------------+------------
    2 | Plaksivij             | 2000-03-02
    3 | Artemov               | 2000-01-09
    1 | Karpova               | 2000-07-01
(3 rows)

plaksivij_danilo=> UPDATE person SET name = 'Bohdanov' WHERE p_id = 1;
UPDATE 1
plaksivij_danilo=> SELECT * FROM person;
 p_id |         name          |     bd
------+-----------------------+------------
    2 | Plaksivij             | 2000-03-02
    3 | Artemov               | 2000-01-09
    1 | Bohdanov              | 2000-07-01
(3 rows)

plaksivij_danilo=> COMMIT;
COMMIT
plaksivij_danilo=>
```

```
plaksivij_danilo=> START TRANSACTION;
START TRANSACTION
plaksivij_danilo=> SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
SET
plaksivij_danilo=> SELECT * FROM person;
 p_id |         name          |     bd
------+-----------------------+------------
    2 | Plaksivij             | 2000-03-02
    3 | Artemov               | 2000-01-09
    1 | Karpova               | 2000-07-01
(3 rows)

plaksivij_danilo=> UPDATE person SET bd = '04/04/2000' WHERE p_id = 1;
ERROR:  could not serialize access due to concurrent update
plaksivij_danilo=> UPDATE person SET bd = '04/04/2000' WHERE p_id = 1;
ERROR:  current transaction is aborted, commands ignored until end of transactio
n block
plaksivij_danilo=>
```

На рівні ізоляції REPEATABLE READ також 2 транзакцію направило у стан очікування до збереження змін 1 транзакції, після збереження видає помилку.

1.3 Повторив роботу транзакцій при умові їх роботи на рівні ізоляції SERIALIZABLE. Проаналізував реакцію СКБД на операцію UPDATE 2-ї транзакції (яка виконується пізніше) та дав свої висновки.

```
START TRANSACTION
plaksivij_danilo=> SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
SET
plaksivij_danilo=> SELECT * FROM person;
 p_id |         name          |     bd
------+-----------------------+------------
    2 | Plaksivij             | 2000-03-02
    3 | Artemov               | 2000-01-09
    1 | Bohdanov              | 2000-07-01
(3 rows)

plaksivij_danilo=> UPDATE person SET name = 'Alekseenko' WHERE p_id = 1;
UPDATE 1
plaksivij_danilo=> SELECT * FROM person;
 p_id |         name          |     bd
------+-----------------------+------------
    2 | Plaksivij             | 2000-03-02
    3 | Artemov               | 2000-01-09
    1 | Alekseenko            | 2000-07-01
(3 rows)

plaksivij_danilo=> COMMIT;
COMMIT
plaksivij_danilo=>
```

```
plaksivij_danilo=> START TRANSACTION;
START TRANSACTION
plaksivij_danilo=> SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
SET
plaksivij_danilo=> UPDATE person SET bd = '01/01/2000' WHERE p_id = 1;
ERROR:  could not serialize access due to concurrent update
plaksivij_danilo=> UPDATE person SET bd = '04/04/2000' WHERE p_id = 1;
ERROR:  current transaction is aborted, commands ignored until end of transactio
n block
plaksivij_danilo=>
```

Неможливо змінювати дані паралельно в ізоляції SERIALIZABLE.

Завдання 4. Керування квазіпаралельним виконанням транзакцій при наявності тупикових ситуацій.

3.1 Виконав модифікацію транзакцій так, щоб вони призводили до тупикової ситуації.

3.2 Виконайте дві модифіковані транзакції.

```
plaksivij_danilo=> START TRANSACTION;
START TRANSACTION
plaksivij_danilo=> UPDATE person SET name = 'Borshecky' WHERE p_id = 1;
UPDATE 1
plaksivij_danilo=> UPDATE person SET bd = '09/09/2000' WHERE p_id = 2;
UPDATE 1
plaksivij_danilo=> SELECT * FROM person;
 p_id |         name         |     bd
------+----------------------+------------
    3 | Artemov              | 2000-01-09
    1 | Borshecky            | 2000-07-01
    2 | Plaksivij            | 2000-09-09
(3 rows)

plaksivij_danilo=>
```

```
plaksivij_danilo=> START TRANSACTION;
START TRANSACTION
plaksivij_danilo=> UPDATE person SET bd = '08/08/2000' WHERE p_id = 2;
UPDATE 1
plaksivij_danilo=> UPDATE person SET name = 'Karpanov' WHERE p_id = 1;
ERROR:  deadlock detected
DETAIL:  Process 9067 waits for ShareLock on transaction 3097; blocked by proces
s 2765.
Process 2765 waits for ShareLock on transaction 3098; blocked by process 9067.
HINT:  See server log for query details.
CONTEXT:  while updating tuple (0,17) in relation "person"
plaksivij_danilo=>
```

Проаналізував реакцію СКБД на операцію UPDATE 2-ї транзакції (яка виконується пізніше) та яка призвела до тупику. Дав свої висновки з урахуванням:

– ідентифікаторів процесів

– номерів транзакцій.

Під час виконання вправи виник тупик, та СКБД скасувала 2 транзакцію, через те, що вона призвела до тупика, а 1 навпаки зберегла.

```
[plaksivij_danilo@vpsj3IeQ ~]$ ps -u postgres -o pid,ppid,stat,cmd | egrep "plak
sivij_danilo"
 2765  8763 Ss   postgres: plaksivij_danilo plaksivij_danilo [local] idle
 9067  8763 Ss   postgres: plaksivij_danilo plaksivij_danilo [local] idle
[plaksivij_danilo@vpsj3IeQ ~]$
```

Висновок: під час виконання лабораторної роботи н 10 дослідив поведінку процесів-транзакцій в базах даних та засоби керуванням ними через механізм блокування з використанням сучасних систем керування базами даних. Найважчим для мене завданням було завдання 4(Керування квазіпаралельним виконанням транзакцій при наявності тупикових ситуацій.)