Міністерство освіти і науки України

Державний Університет Одеська Політехника

Інститут комп’ютерних систем

Кафедра інформаційних систем

Лабораторна робота №10

з дисципліни «Операційні Системи»

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

Виконав:

ст. гр. АІ-204

Бериславський В.Р

Перевірив:

Блажко О. А.

Дрозд М.О.

Одеса – 2021

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

Хід роботи:

Завдання 1: Аналіз роботи багато версійного протоколу

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

Підготуйте 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

– T3 – видалення рядку з наступною відміною цієї операції;

– T4 – зміна значення однієї з колонок рядка.

В операцію читання рядка таблиці додайте системні колонки xmin, xmax.

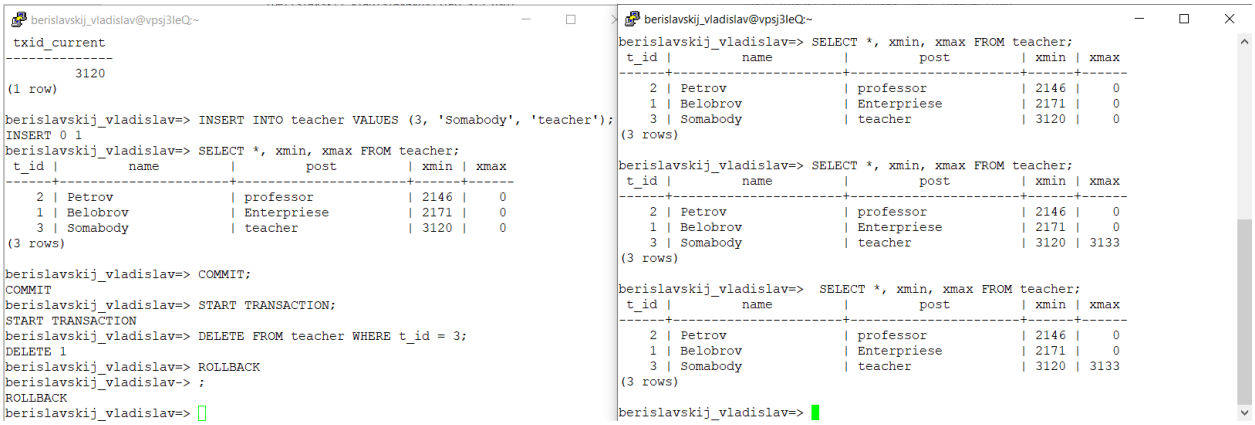
На кожному кроці виконання транзакції переглядайте значення колонок xmin, xmax та зробіть відповідні висновки.

T1

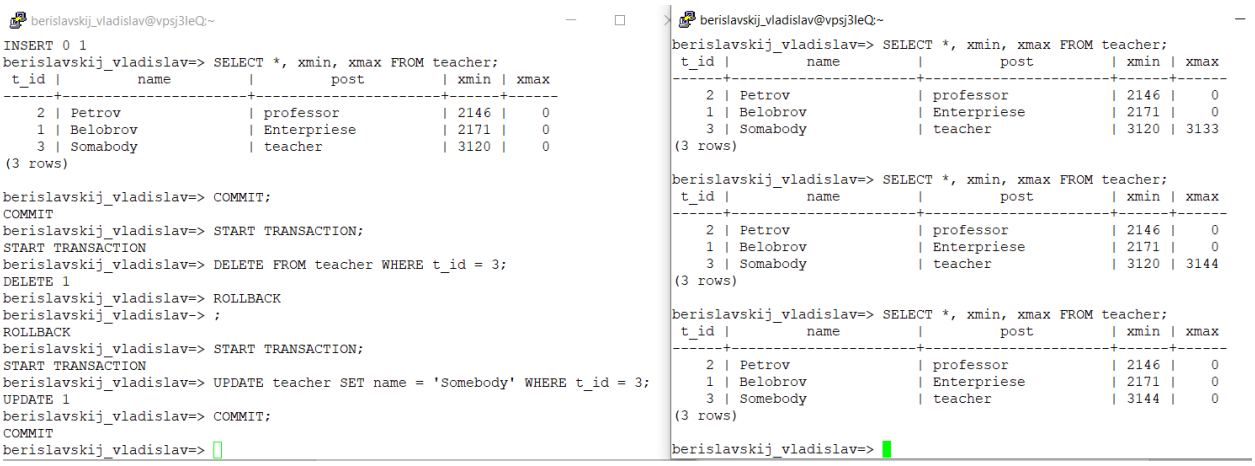
```
berislavskij_vladislav@vpsj3IeQ:~
[berislavskij_vladislav@vpsj3IeQ ~]$ psql
psql (9.5.25)
Type "help" for help.

berislavskij_vladislav=> START TRANSACTION;
START TRANSACTION
berislavskij_vladislav=> SELECT txid_current();
 txid_current
--------------
         3120
(1 row)

berislavskij_vladislav=> INSERT INTO teacher VALUES (3, 'Somabody', 'teacher');
INSERT 0 1
berislavskij_vladislav=> SELECT *, xmin, xmax FROM teacher;
 t_id |         name         |         post         | xmin | xmax
------+----------------------+----------------------+------+------
    2 | Petrov               | professor            | 2146 |    0
    1 | Belobrov             | Enterpriese          | 2171 |    0
    3 | Somabody             | teacher              | 3120 |    0
(3 rows)

berislavskij_vladislav=> COMMIT;
COMMIT
berislavskij_vladislav=>
```

```
berislavskij_vladislav@vpsj3IeQ:~
berislavskij_vladislav=> START TRANSACTION;
START TRANSACTION
berislavskij_vladislav=> SELECT *, xmin, xmax FROM teacher;
 t_id |         name         |         post         | xmin | xmax
------+----------------------+----------------------+------+------
    2 | Petrov               | professor            | 2146 |    0
    1 | Belobrov             | Enterpriese          | 2171 |    0
(2 rows)

berislavskij_vladislav=> SELECT *, xmin, xmax FROM teacher;
 t_id |         name         |         post         | xmin | xmax
------+----------------------+----------------------+------+------
    2 | Petrov               | professor            | 2146 |    0
    1 | Belobrov             | Enterpriese          | 2171 |    0
(2 rows)

berislavskij_vladislav=> SELECT *, xmin, xmax FROM teacher;
 t_id |         name         |         post         | xmin | xmax
------+----------------------+----------------------+------+------
    2 | Petrov               | professor            | 2146 |    0
    1 | Belobrov             | Enterpriese          | 2171 |    0
    3 | Somabody             | teacher              | 3120 |    0
(3 rows)

berislavskij_vladislav=>
```

T3



T4



Завдання 2. Аналіз стану транзакцій на різних рівнях багаторівневого блокування.

Виконайте послідовно в двох терміналах наступні комбінації блокувань таблиці:

IX-IS, SIX-IX, SIX-IS. Надайте висновки про сумісність блокувань.

Для кожної комбінації блокувань перед завершенням 1-ї транзакції (яка розпочалася раніше) в додатковому терміналі через команду psql отримайте данні про стан транзакцій (таблиця pg_locs).

Для початку визначимо OID бази данних:

```
berislavskij_vladislav=> SELECT oid, datname FROM  pg_database WHERE datname = 'berislavskij_vladislav';
  oid  |        datname
-------+------------------------
 16516 | berislavskij_vladislav
(1 row)
```

```
berislavskij_vladislav=> SELECT database, relation, locktype, virtualtransaction, pid, mode, granted FROM  pg_locks
WHERE locktype = 'relation' AND database = 16516;
 database | relation | locktype | virtualtransaction | pid  |      mode       | granted
----------+----------+----------+--------------------+------+-----------------+---------
    16516 |    11673 | relation | 17/2875            | 8113 | AccessShareLock | t
(1 row)

berislavskij_vladislav=>
```

За замовченням працює 1 транзакція

IX-IS

```
berislavskij_vladislav=> START TRANSACTION;
START TRANSACTION
berislavskij_vladislav=> LOCK TABLE teacher IN ROW EXCLUSIVE MODE;
LOCK TABLE
berislavskij_vladislav=>
```

```
berislavskij_vladislav=> START TRANSACTION;
START TRANSACTION
berislavskij_vladislav=> LOCK TABLE teacher IN ROW SHARE MODE;
LOCK TABLE
berislavskij_vladislav=>
```

```
berislavskij_vladislav=> SELECT database, relation, locktype, virtualtransaction, pid, mode, granted FROM  pg_locks
WHERE locktype = 'relation' AND database = 16516;
 database | relation | locktype | virtualtransaction | pid  |       mode       | granted
----------+----------+----------+--------------------+------+------------------+---------
    16516 |    16639 | relation | 40/198             | 6921 | RowShareLock     | t
    16516 |    11673 | relation | 17/2876            | 8113 | AccessShareLock  | t
    16516 |    16639 | relation | 9/20215            | 6795 | RowExclusiveLock | t
(3 rows)
```

Блокування сумісні

SIX-IX

```
START TRANSACTION
berislavskij_vladislav=> LOCK TABLE teacher IN SHARE ROW EXCLUSIVE MODE;
LOCK TABLE
berislavskij_vladislav=>
```

```
berislavskij_vladislav=> START TRANSACTION;
START TRANSACTION
berislavskij_vladislav=> LOCK TABLE teacher IN ROW EXCLUSIVE MODE;
```

```
berislavskij_vladislav=> SELECT database, relation, locktype, virtualtransaction, pid, mode, granted FROM  pg_locks
WHERE locktype = 'relation' AND database = 16516;
 database | relation | locktype | virtualtransaction | pid  |         mode          | granted
----------+----------+----------+--------------------+------+-----------------------+---------
    16516 |    11673 | relation | 17/2878            | 8113 | AccessShareLock       | t
    16516 |    16639 | relation | 9/20216            | 6795 | ShareRowExclusiveLock | t
    16516 |    16639 | relation | 40/199             | 6921 | RowExclusiveLock      | f
(3 rows)
```

Блокування несумісні

SIX-IS

```
berislavskij_vladislav=> START TRANSACTION;
START TRANSACTION
berislavskij_vladislav=> LOCK TABLE teacher IN SHARE ROW EXCLUSIVE MODE;
LOCK TABLE
berislavskij vladislav=>
```

```
START TRANSACTION
berislavskij_vladislav=>
berislavskij_vladislav=> LOCK TABLE teacher IN ROW SHARE MODE;
LOCK TABLE
berislavskij_vladislav=>
```

```
berislavskij_vladislav=> SELECT database, relation, locktype, virtualtransaction, pid, mode, granted FROM  pg_locks
WHERE locktype = 'relation' AND database = 16516;
 database | relation | locktype | virtualtransaction | pid  |      mode       | granted
----------+----------+----------+--------------------+------+-----------------+---------
    16516 |    11673 | relation | 17/2879            | 8113 | AccessShareLock | t
(1 row)
```

Блокування сумісні

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

Підготуйте транзакції, які було створено у завданні 3.1 рішення попередньої лабораторної роботи, а саме, створіть 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 Виконайте роботу транзакцій при умові їх роботи на рівні ізоляції READ COMMITED. Проаналізуйте реакцію СКБД на операцію UPDATE 2-ї транзакції (яка виконується пізніше) та дайте свої висновки.

```
berislavskij_vladislav=> START TRANSACTION;
START TRANSACTION
berislavskij_vladislav=> SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
SET
berislavskij_vladislav=> SELECT * FROM teacher WHERE t_id = 1;
 t_id |         name         |         post
------+----------------------+----------------------
    1 | Belobrov             | Enterpriese
(1 row)

berislavskij_vladislav=> UPDATE teacher SET name = 'Arthur Belobrov'
berislavskij_vladislav-> WHERE t_id = 1;
UPDATE 1
berislavskij_vladislav=> SELECT * FROM teacher WHERE t_id = 1;
 t_id |         name         |         post
------+----------------------+----------------------
    1 | Arthur Belobrov      | Enterpriese
(1 row)

berislavskij_vladislav=> COMMIT;
```

```
berislavskij_vladislav=> START TRANSACTION;
START TRANSACTION
berislavskij_vladislav=>
berislavskij_vladislav=> LOCK TABLE teacher IN ROW SHARE MODE;
LOCK TABLE
berislavskij_vladislav=> COMMIT;
COMMIT
berislavskij_vladislav=> START TRANSACTION;
START TRANSACTION
berislavskij_vladislav=> SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
SET
berislavskij_vladislav=> SELECT * FROM teacher WHERE t_id = 1;
 t_id |         name         |         post
------+----------------------+----------------------
    1 | Belobrov             | Enterpriese
(1 row)

berislavskij_vladislav=> UPDATE teacher SET post = 'Enterpriese worker'
berislavskij_vladislav-> WHERE t_id = 1;
```

Ізоляція встановлена таким чином, що ми не маємо можливості провести оновлення данних другою транзакцією, так як перша ще не виконала COMMIT;

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

```
berislavskij_vladislav=> START TRANSACTION;
START TRANSACTION
berislavskij_vladislav=> SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
SET
berislavskij_vladislav=> SELECT * FROM teacher WHERE t_id = 1;
 t_id |         name         |         post
------+----------------------+----------------------
    1 | Arthur Belobrov      | Enterpriese worker
(1 row)

berislavskij_vladislav=> UPDATE teacher SET name = 'Arthur' WHERE t_id = 1;
UPDATE 1
berislavskij_vladislav=> SELECT * FROM teacher WHERE t_id = 1;
 t_id |         name         |         post
------+----------------------+----------------------
    1 | Arthur               | Enterpriese worker
(1 row)

berislavskij_vladislav=> COMMIT;
COMMIT
berislavskij vladislav=>
```

```
    1 | Arthur Belobrov      | Enterpriese worker
(1 row)

berislavskij_vladislav=> COMMIT;
COMMIT
berislavskij_vladislav=> START TRANSACTION;
START TRANSACTION
berislavskij_vladislav=> SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
SET
berislavskij_vladislav=> SELECT * FROM teacher WHERE t_id = 1;
 t_id |         name         |         post
------+----------------------+----------------------
    1 | Arthur Belobrov      | Enterpriese worker
(1 row)

berislavskij_vladislav=> UPDATE teacher SET post = 'worker' WHERE t_id = 1;
ERROR:  could not serialize access due to concurrent update
berislavskij_vladislav=> SELECT * FROM teacher WHERE t_id = 1;
ERROR:  current transaction is aborted, commands ignored until end of transac
tion block
berislavskij vladislav=>
```

При REPEATABLE READ ми спостерігали схожу ситуацію, але по завершенню першої транзакції друга викликає помилку та блокується.

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

```
berislavskij_vladislav=> START TRANSACTION;
START TRANSACTION
berislavskij_vladislav=> SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
SET
berislavskij_vladislav=> SELECT * FROM teacher WHERE t_id = 1;
 t_id |         name         |         post
------+----------------------+----------------------
    1 | Arthur               | Enterpriese worker
(1 row)

berislavskij_vladislav=> UPDATE teacher SET name = 'Arthur Belobrov'
berislavskij_vladislav-> WHERE t_id = 1;
UPDATE 1
berislavskij_vladislav=> SELECT * FROM teacher WHERE t_id = 1;
 t_id |         name         |         post
------+----------------------+----------------------
    1 | Arthur Belobrov      | Enterpriese worker
(1 row)

berislavskij_vladislav=> COMMIT;
COMMIT
berislavskij_vladislav=>
```

```
berislavskij_vladislav=> SELECT * FROM teacher WHERE t_id = 1;
ERROR:  current transaction is aborted, commands ignored until end of transac
tion block
berislavskij_vladislav=> COMMIT;
ROLLBACK
berislavskij_vladislav=> START TRANSACTION;
START TRANSACTION
berislavskij_vladislav=> SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
SET
berislavskij_vladislav=> SELECT * FROM teacher WHERE t_id = 1;
 t_id |         name         |         post
------+----------------------+----------------------
    1 | Arthur               | Enterpriese worker
(1 row)

berislavskij_vladislav=> UPDATE teacher SET post = 'Enterpreize'
berislavskij_vladislav-> WHERE t_id = 1;
ERROR:  could not serialize access due to concurrent update
berislavskij_vladislav=> SELECT * FROM teacher WHERE t_id = 1;
ERROR:  current transaction is aborted, commands ignored until end of transac
tion block
berislavskij_vladislav=>
```

Ми не можемо виконувати зміни одночасно, тому друга транзакція викликала помилку та відмінила операцію.

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

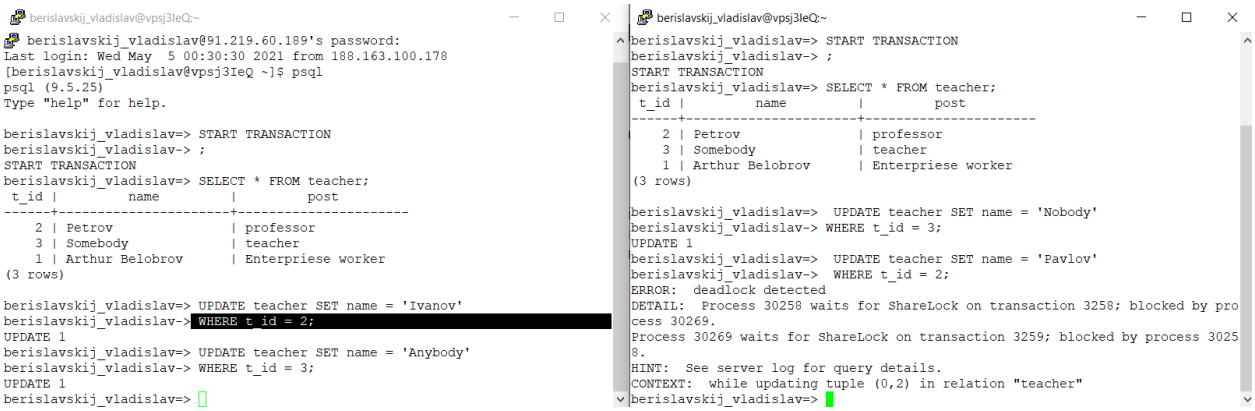
3.1 Виконайте модифікацію транзакцій так, щоб вони призводили до тупикової ситуації.

3.2 Виконайте дві модифіковані транзакції.

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

– ідентифікаторів процесів

– номерів транзакцій.



У ході виконання транзакцій виникла тупикова ситуація, і тому 2 транзакція була скасована для того, щоб надати першій транзакції можливість успішно завершити роботу.

```
[berislavskij_vladislav@vpsj3IeQ ~]$ ps -u postgres -o pid,ppid,stat,cmd | egrep "berislavskij_vladislav"
 6795  8763 Ss   postgres: berislavskij_vladislav berislavskij_vladislav [local] idle
 6921  8763 Ss   postgres: berislavskij_vladislav berislavskij_vladislav [local] idle in transaction (aborted)
30258  8763 Ss   postgres: berislavskij_vladislav berislavskij_vladislav [local] idle in transaction (aborted)
30269  8763 Ss   postgres: berislavskij_vladislav berislavskij_vladislav [local] idle in transaction
```

Висновок : у ході виконання лабораторної роботи були проведені різна операції з керування транзакціями. Найскладнішим виявилось 2 завання, так як вони потребувало деякої додаткової інформації з визначення oid бази даних та виведення інформації за ним.