

Державний університет «Одеська Політехніка»

Інститут комп'ютерних систем

Кафедра «Інформаційних систем»

Лабораторна робота №10

З дисципліни

«Операційні системи»

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

Виконала: Студентка групи AI-204

Озарчук А.С.

Перевірили: Блажко О. А

Дрозд М. О.

Завдання

Для кожної транзакції підготуйте окремий термінал, в якому виконайте 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.

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.

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_locks`).

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

COMMITTED. Проаналізуйте реакцію СКБД на операцію UPDATE 2-ї транзакції (яка виконується пізніше) та дайте свої висновки.

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.

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.

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.

Виконання завдань

Для кожної транзакції я підготувала окремий термінал, в якому виконала команду доступу до моєї 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.

Я підготувала 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 додала системні колонки xmin, xmax.

На кожному кроці виконання транзакції переглядала значення колонок xmin, xmax, та зробила відповідні висновки.

1) Транзакція 1 та частина транзакції 2

Ми бачимо, що транзакція 1 (номер 3109) додала до таблиці university рядок 3 (за колонкою xmin). Зміни можливо побачити після успішного завершення 1 транзакції.

```
ozarchuk_anna@vpsj3IeQ:~  
psql (9.5.25)  
Type "help" for help.  
  
ozarchuk_anna=> START TRANSACTION;  
START TRANSACTION  
ozarchuk_anna=> select txid_current();  
txid_current  
-----  
3109  
(1 row)  
  
ozarchuk_anna=> INSERT INTO university VALUES (3,'KNTEU',2008);  
INSERT 0 1  
ozarchuk_anna=> SELECT xmin,xmax,* FROM university;  
xmin | xmax | u_id | name | year  
-----+-----+-----+-----+-----  
2181 | 0 | 1 | NUOUA | 2005  
3107 | 0 | 2 | DUOP | 2020  
3109 | 0 | 3 | KNTEU | 2008  
(3 rows)  
  
ozarchuk_anna=> COMMIT;  
COMMIT  
ozarchuk_anna=>
```

```
ozarchuk_anna@vpsj3IeQ:~  
ozarchuk_anna@91.219.60.189's password:  
Last login: Tue May 4 22:47:22 2021 from 176.105.192.32  
[ozarchuk_anna@vpsj3IeQ ~]$ psql  
psql (9.5.25)  
Type "help" for help.  
  
ozarchuk_anna=> START TRANSACTION;  
START TRANSACTION  
ozarchuk_anna=> SELECT xmin,xmax,* FROM university;  
xmin | xmax | u_id | name | year  
-----+-----+-----+-----+-----  
2181 | 0 | 1 | NUOUA | 2005  
3107 | 0 | 2 | DUOP | 2020  
(2 rows)  
  
ozarchuk_anna=> SELECT xmin,xmax,* FROM university;  
xmin | xmax | u_id | name | year  
-----+-----+-----+-----+-----  
2181 | 0 | 1 | NUOUA | 2005  
3107 | 0 | 2 | DUOP | 2020  
3109 | 0 | 3 | KNTEU | 2008  
(3 rows)  
ozarchuk_anna=>
```

2) Транзакція 3 та частина транзакції 2

Бачимо, що в колонці xmin для u_id = 3, транзакція 3 (номер 3110) намагається видалити рядок. Транзакція 3 скасовується, але значення xmax залишається незмінним.

```
ozarchuk_anna@vpsj3IeQ:~  
login as: ozarchuk_anna  
ozarchuk_anna@91.219.60.189's password:  
Last login: Tue May 4 22:47:58 2021 from 176.105.192.32  
[ozarchuk_anna@vpsj3IeQ ~]$ psql  
psql (9.5.25)  
Type "help" for help.  
  
ozarchuk_anna=> START TRANSACTION;  
START TRANSACTION  
ozarchuk_anna=> DELETE FROM university WHERE u_id = 3;  
DELETE 1  
ozarchuk_anna=> ROLLBACK;  
ROLLBACK  
ozarchuk_anna=>   
  
ozarchuk_anna=> SELECT xmin,xmax,* FROM university;  
xmin | xmax | u_id | name | year  
-----+-----+-----+-----+-----  
2181 | 0 | 1 | NUOUA | 2005  
3107 | 0 | 2 | DUOP | 2020  
3109 | 3110 | 3 | KNTEU | 2008  
(3 rows)  
  
ozarchuk_anna=> SELECT xmin,xmax,* FROM university;  
xmin | xmax | u_id | name | year  
-----+-----+-----+-----+-----  
2181 | 0 | 1 | NUOUA | 2005  
3107 | 0 | 2 | DUOP | 2020  
3109 | 3110 | 3 | KNTEU | 2008  
(3 rows)  
  
ozarchuk_anna=>   

```

3) Транзакція 4 та частина транзакції 2

Бачимо, що в колонці xmax для u_id = 3, транзакція 4 (номер 3110) намагається змінити дані. Транзакція 4 успішно завершується, а значення xmax змінюється на 0.

```
ozarchuk_anna=> START TRANSACTION;  
START TRANSACTION  
ozarchuk_anna=> UPDATE university SET name = 'RDGU' WHERE u_id = 3;  
UPDATE 1  
ozarchuk_anna=> COMMIT;  
COMMIT  
ozarchuk_anna=>   

```

```
ozarchuk_anna=> SELECT xmin,xmax,* FROM university;  
xmin | xmax | u_id | name | year  
-----+-----+-----+-----+-----  
2181 | 0 | 1 | NUOUA | 2005  
3107 | 0 | 2 | DUOP | 2020  
3109 | 3110 | 3 | KNTEU | 2008  
(3 rows)  

```

```
ozarchuk_anna=> START TRANSACTION;  
START TRANSACTION  
ozarchuk_anna=> SELECT xmin,xmax,* FROM university;  
xmin | xmax | u_id | name | year  
-----+-----+-----+-----+-----  
2181 | 0 | 1 | NUOUA | 2005  
3107 | 0 | 2 | DUOP | 2020  
3112 | 0 | 3 | RDGU | 2008  
(3 rows)  
  
ozarchuk_anna=>   

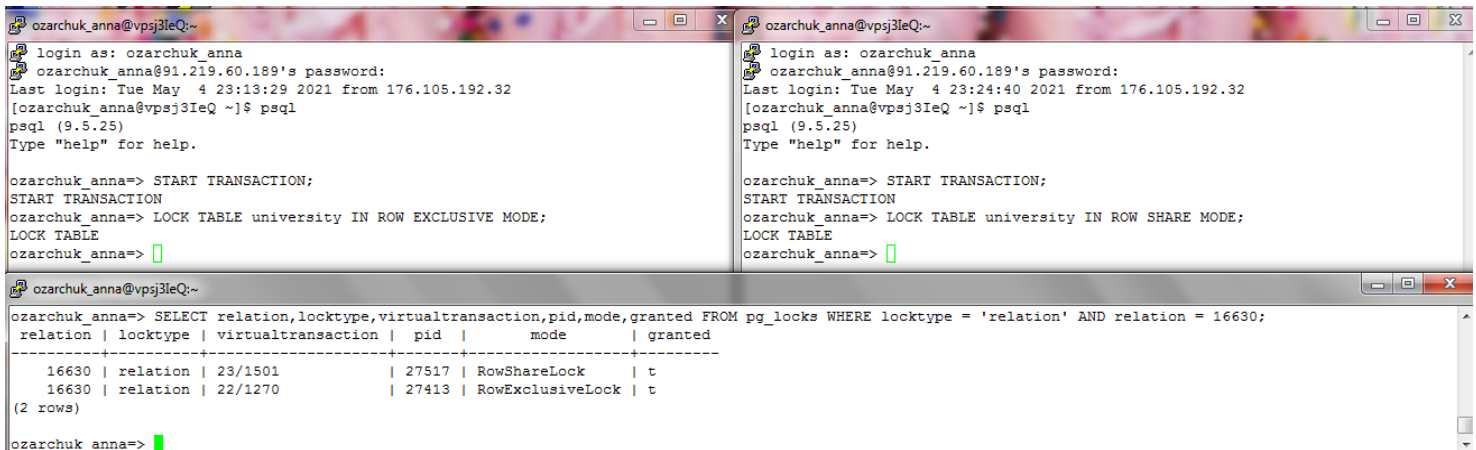
```

Завдання 2. Аналіз стану транзакцій на різних рівнях багаторівневого блокування

Виконала послідовно в двох терміналах наступні комбінації блокувань таблиці: IX-IS, SIX-IX, SIX-IS. Надала висновки про сумісність блокувань.

Для кожної комбінації блокувань перед завершенням 1-ї транзакції (яка розпочалася раніше) в додатковому терміналі через команду `psql` отримала дані про стан транзакцій (таблиця `pg_locks`).

1) Блокування IX-IS є сумісними, тому транзакції проходять без проблем. У колонці `granted` стоїть `t`.



```
ozarchuk_anna@vpsj3IeQ:~$ psql
psql (9.5.25)
Type "help" for help.

ozarchuk_anna=> START TRANSACTION;
START TRANSACTION
ozarchuk_anna=> LOCK TABLE university IN ROW EXCLUSIVE MODE;
LOCK TABLE
ozarchuk_anna=>

ozarchuk_anna@vpsj3IeQ:~$ psql
psql (9.5.25)
Type "help" for help.

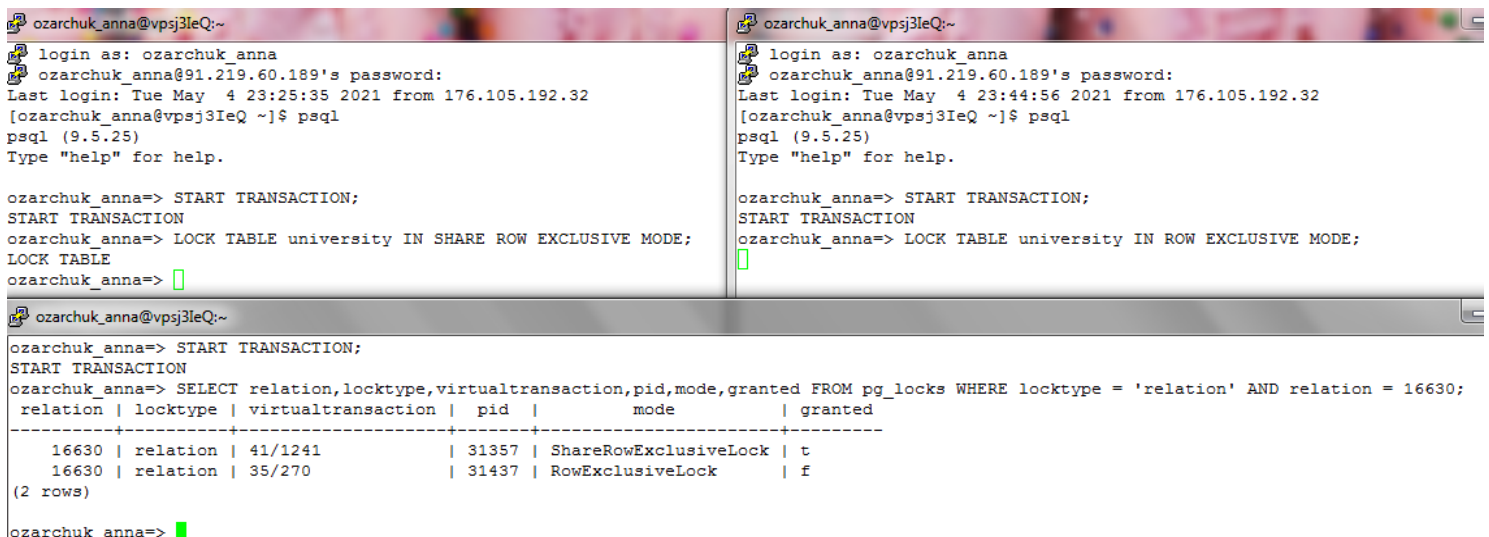
ozarchuk_anna=> START TRANSACTION;
START TRANSACTION
ozarchuk_anna=> LOCK TABLE university IN ROW SHARE MODE;
LOCK TABLE
ozarchuk_anna=>

ozarchuk_anna@vpsj3IeQ:~$ psql
psql (9.5.25)
Type "help" for help.

ozarchuk_anna=> SELECT relation,locktype,virtualtransaction,pid,mode,granted FROM pg_locks WHERE locktype = 'relation' AND relation = 16630;
relation | locktype | virtualtransaction | pid | mode | granted
-----+-----+-----+-----+-----+-----
16630 | relation | 23/1501 | 27517 | RowShareLock | t
16630 | relation | 22/1270 | 27413 | RowExclusiveLock | t
(2 rows)

ozarchuk_anna=>
```

2) Блокування SIX-IX є несумісними, тому транзакція 2 чекає на кінець першої. У колонці `granted` стоїть `f` проти IX.



```
ozarchuk_anna@vpsj3IeQ:~$ psql
psql (9.5.25)
Type "help" for help.

ozarchuk_anna=> START TRANSACTION;
START TRANSACTION
ozarchuk_anna=> LOCK TABLE university IN SHARE ROW EXCLUSIVE MODE;
LOCK TABLE
ozarchuk_anna=>

ozarchuk_anna@vpsj3IeQ:~$ psql
psql (9.5.25)
Type "help" for help.

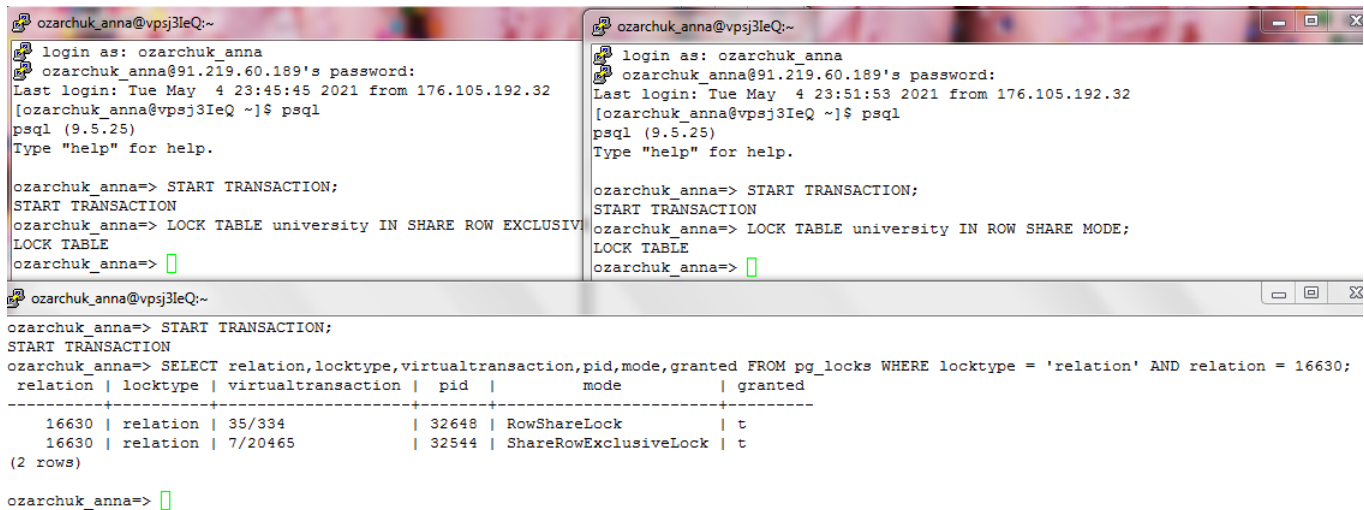
ozarchuk_anna=> START TRANSACTION;
START TRANSACTION
ozarchuk_anna=> LOCK TABLE university IN ROW EXCLUSIVE MODE;
LOCK TABLE
ozarchuk_anna=>

ozarchuk_anna@vpsj3IeQ:~$ psql
psql (9.5.25)
Type "help" for help.

ozarchuk_anna=> SELECT relation,locktype,virtualtransaction,pid,mode,granted FROM pg_locks WHERE locktype = 'relation' AND relation = 16630;
relation | locktype | virtualtransaction | pid | mode | granted
-----+-----+-----+-----+-----+-----
16630 | relation | 41/1241 | 31357 | ShareRowExclusiveLock | t
16630 | relation | 35/270 | 31437 | RowExclusiveLock | f
(2 rows)

ozarchuk_anna=>
```

3) Блокування SIX-IS є сумісними, тому транзакції проходять без проблем. У колонці granted стоїть t



```
ozarchuk_anna@vpsj3IeQ:~  
login as: ozarchuk_anna  
ozarchuk_anna@91.219.60.189's password:  
Last login: Tue May 4 23:45:45 2021 from 176.105.192.32  
[ozarchuk_anna@vpsj3IeQ ~]$ psql  
psql (9.5.25)  
Type "help" for help.  
  
ozarchuk_anna=> START TRANSACTION;  
START TRANSACTION  
ozarchuk_anna=> LOCK TABLE university IN SHARE ROW EXCLUSIVE  
LOCK TABLE  
ozarchuk_anna=>   
  
ozarchuk_anna@vpsj3IeQ:~  
login as: ozarchuk_anna  
ozarchuk_anna@91.219.60.189's password:  
Last login: Tue May 4 23:51:53 2021 from 176.105.192.32  
[ozarchuk_anna@vpsj3IeQ ~]$ psql  
psql (9.5.25)  
Type "help" for help.  
  
ozarchuk_anna=> START TRANSACTION;  
START TRANSACTION  
ozarchuk_anna=> LOCK TABLE university IN ROW SHARE MODE;  
LOCK TABLE  
ozarchuk_anna=>   
  
ozarchuk_anna@vpsj3IeQ:~  
ozarchuk_anna=> START TRANSACTION;  
START TRANSACTION  
ozarchuk_anna=> SELECT relation,locktype,virtualtransaction,pid,mode,granted FROM pg_locks WHERE locktype = 'relation' AND relation = 16630;  
 relation | locktype | virtualtransaction | pid | mode | granted  
-----+-----+-----+-----+-----+-----  
 16630 | relation | 35/334 | 32648 | RowShareLock | t  
 16630 | relation | 7/20465 | 32544 | ShareRowExclusiveLock | t  
(2 rows)  
  
ozarchuk_anna=>   

```

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

Підготувала транзакції, які було створено у завданні 3.1 рішення попередньої лабораторної роботи, а саме, створила 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 Виконала роботу транзакцій при умові їх роботи на рівні ізоляції READ COMMITED.

Реалізує безпеку від брудного читання. Транзакції бачать лише зафіксовані зміни (командою commit), тому update 2 транзакції не бачить змінених даних. Допоки транзакція 1 не завершиться, T2 перебуває в стані WAIT

```
ozarchuk_anna@vpsj3leQ:~  
ERROR: syntax error at or near "TRANSACTION"  
LINE 1: START TRANSACTION;  
^  
ozarchuk_anna=> START TRANSACTION;  
START TRANSACTION  
ozarchuk_anna=> SET TRANSACTION ISOLATION LEVEL READ COMMITTED;  
SET  
ozarchuk_anna=> SELECT * FROM university WHERE u_id = 1;  
u_id | name | year  
-----+-----+-----  
1 | NUOVA | 2005  
(1 row)  
  
ozarchuk_anna=> UPDATE university SET name = 'SVU' WHERE u_id = 1;  
UPDATE 1  
ozarchuk_anna=> SELECT * FROM university WHERE u_id = 1;  
u_id | name | year  
-----+-----+-----  
1 | SVU | 2005  
(1 row)  
  
ozarchuk_anna=> COMMIT;  
COMMIT  
ozarchuk_anna=> █  
  
ozarchuk_anna@vpsj3leQ:~  
ozarchuk_anna=> START TRANSACTION;  
START TRANSACTION  
ozarchuk_anna=> SET TRANSACTION ISOLATION LEVEL READ COMMITTED;  
SET  
ozarchuk_anna=> SELECT * FROM university WHERE u_id = 1;  
u_id | name | year  
-----+-----+-----  
1 | NUOVA | 2005  
(1 row)  
  
ozarchuk_anna=> UPDATE university SET year = 2004 WHERE u_id = 1;  
UPDATE 1  
ozarchuk_anna=> SELECT * FROM university WHERE u_id = 1;  
u_id | name | year  
-----+-----+-----  
1 | SVU | 2004  
(1 row)  
  
ozarchuk_anna=> COMMIT;  
COMMIT  
ozarchuk_anna=> █
```

1.2 Повторила роботу транзакцій при умові їх роботи на рівні ізоляції REPEATABLE READ.

Реалізує безпеку від неповторного читання. Коли зміни першої транзакції були зафіксовані, то REPEATABLE READ відмінює другу транзакцію з помилкою ERROR: could not serialize access due to concurrent update, тому що REPEATABLE READ не може змінювати дані, змінені іншою транзакцією.

```
ozarchuk_anna@vpsj3leQ:~  
ozarchuk_anna=> START TRANSACTION;  
START TRANSACTION  
ozarchuk_anna=> SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;  
SET  
ozarchuk_anna=> SELECT * FROM university WHERE u_id = 1;  
u_id | name | year  
-----+-----+-----  
1 | SVU | 2004  
(1 row)  
  
ozarchuk_anna=> UPDATE university SET name = 'LNU' WHERE u_id = 1;  
UPDATE 1  
ozarchuk_anna=> SELECT * FROM university WHERE u_id = 1;  
u_id | name | year  
-----+-----+-----  
1 | LNU | 2004  
(1 row)  
  
ozarchuk_anna=> COMMIT;  
COMMIT  
ozarchuk_anna=> █  
  
psql (9.5.25)  
Type "help" for help.  
  
ozarchuk_anna=> START TRANSACTION;  
START TRANSACTION  
ozarchuk_anna=> SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;  
SET  
ozarchuk_anna=> SELECT * FROM university WHERE u_id = 1;  
u_id | name | year  
-----+-----+-----  
1 | SVU | 2004  
(1 row)  
  
ozarchuk_anna=> UPDATE university SET year = 2001 WHERE u_id = 1;  
ERROR: could not serialize access due to concurrent update  
ozarchuk_anna=> SELECT * FROM university WHERE u_id = 1;  
ERROR: current transaction is aborted, commands ignored until end of transaction block  
ozarchuk_anna=> COMMIT;  
ROLLBACK  
ozarchuk_anna=> █
```

1.3 Повторила роботу транзакцій при умові їх роботи на рівні ізоляції SERIALIZABLE.

Транзакції працюють так, ніби окрім них не існує. Так як 1 транзакція вже зафіксувала свої зміни, 2 завершилася без фіксації своїх змін. Рівні ізоляції Serializable заборонено виконувати паралельно зміни одних даних.


```
ozarchuk_anna@vpsj3IeQ:~  
START TRANSACTION  
ozarchuk_anna=> SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;  
SET  
ozarchuk_anna=> SELECT * FROM university WHERE u_id = 1;  
u_id | name | year  
-----+-----+-----  
1 | ONPU | 2004  
(1 row)  
  
ozarchuk_anna=> UPDATE university SET name = 'DUOP' WHERE u_id = 1;  
UPDATE 1  
ozarchuk_anna=> SELECT * FROM university WHERE u_id = 1;  
u_id | name | year  
-----+-----+-----  
1 | DUOP | 2004  
(1 row)  
  
ozarchuk_anna=> COMMIT;  
COMMIT  
ozarchuk_anna=>   
  
ozarchuk_anna@vpsj3IeQ:~  
Last login: Wed May 5 02:30:14 2021 from 176.105.192.32  
[ozarchuk_anna@vpsj3IeQ ~]$ psql  
psql (9.5.25)  
Type "help" for help.  
  
ozarchuk_anna=> START TRANSACTION;  
START TRANSACTION  
ozarchuk_anna=> SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;  
SET  
ozarchuk_anna=> SELECT * FROM university WHERE u_id = 1;  
u_id | name | year  
-----+-----+-----  
1 | ONPU | 2004  
(1 row)  
  
ozarchuk_anna=> UPDATE university SET year = 2002 WHERE u_id = 1;  
ERROR: could not serialize access due to concurrent update  
ozarchuk_anna=> COMMIT;  
ROLLBACK  
ozarchuk_anna=>   

```

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

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

3.2 Виконала дві модифіковані транзакції.

Дві транзакції зависають і СКБД сама вирішує, яку з них видалити. Транзакція 1 заблокована та транзакція 2 заблокована (стан процесу Ss). Кожна з них чекає на завершення попередньої, тому отримуємо цикл.

```
ozarchuk_anna@vpsj3IeQ:~  
ozarchuk_anna=> START TRANSACTION;  
START TRANSACTION  
ozarchuk_anna=> SELECT * FROM university;  
u_id | name | year  
-----+-----+-----  
3 | RDGU | 2008  
1 | ONPU | 2004  
2 | KNTEU | 2020  
(3 rows)  
  
ozarchuk_anna=> UPDATE university SET name = 'DUOP' WHERE u_id = 1;  
UPDATE 1  
ozarchuk_anna=> UPDATE university SET name = 'KNU' WHERE u_id = 2;  
UPDATE 1  
ozarchuk_anna=>   
  
ozarchuk_anna@vpsj3IeQ:~  
psql (9.5.25)  
Type "help" for help.  
  
ozarchuk_anna=> START TRANSACTION;  
START TRANSACTION  
ozarchuk_anna=> UPDATE university SET year = 2003 WHERE u_id = 2;  
UPDATE 1  
ozarchuk_anna=> UPDATE university SET name = 2002 WHERE u_id = 1;  
ERROR: deadlock detected  
DETAIL: Process 23050 waits for ShareLock on transaction 3242; blocked by process 22935.  
Process 22935 waits for ShareLock on transaction 3243; blocked by process 23050.  
HINT: See server log for query details.  
CONTEXT: while updating tuple (0,27) in relation "university"  
ozarchuk_anna=>   
  
[ozarchuk_anna@vpsj3IeQ ~]$ ps -u postgres -o pid,ppid,stat,cmd | egrep "ozarchuk_anna"  
22935 8763 Ss postgres: ozarchuk_anna ozarchuk_anna [local] idle in transaction  
23050 8763 Ss postgres: ozarchuk_anna ozarchuk_anna [local] idle in transaction (aborted)  
[ozarchuk_anna@vpsj3IeQ ~]$   

```

Висновок

Під час виконання Лабораторної роботи № 10 мною було досліджено поведінку процесів-транзакцій в базах даних та засоби керування ними через механізм блокування з використанням сучасних систем керування базами даних.