

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»  
Варіант 3

Виконала:  
Студентка групи AI-202  
Неживих М.О.  
Перевірили:  
Блажко О.А

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

### Хід роботи

#### Завдання 1 Аналіз роботи багато версійного протоколу:

Готуємо 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 додаємо системні колонки xmin, xmax.

На кожному кроці виконання транзакції переглядаємо значення колонок xmin, xmax.

№	T1	T2	T3	T4
1		start transaction;		
2		select*,xmin,xmax from university;		
3	start transaction;			
4	select txid_current();			
5	Insert into university values(3, 'ONUM', 1900);			
6	select*,xmin,xmax from university;			
7		select*,xmin,xmax from university;		
8	commit;			

9		select*,xmin,xmax from university;		
10			start transaction;	
11			delete from university where u_id = 3;	
12		select*,xmin,xmax from university;		
13			rollback;	
14		select*,xmin,xmax from university;		
15				start transaction;
16				UPDATE university SET year = 1850 WHERE u_id = 3;
17		select*,xmin,xmax from university;		
18				commit;
19		select * xmin,xmax from university;		
20		commit;		

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

nezhivih_mariya=> start transaction;
START TRANSACTION
nezhivih_mariya=> select*,xmin,xmax from university;
 u_id |      name      | year | xmin | xmax
-----+-----+-----+-----+-----
    2 | DUOP           | 2021 | 2509 |    0
    1 | ONPU           | 1918 | 3020 |    0
(2 rows)
```

```

nezhivih_mariya=> start transaction;
START TRANSACTION
nezhivih_mariya=> select txid_current();
txid_current
-----
3032
(1 row)

nezhivih_mariya=> Insert into university values(3, 'ONUM', 1900);
INSERT 0 1
nezhivih_mariya=> select*,xmin,xmax from university;
 u_id |          name          | year | xmin | xmax
-----+-----+-----+-----+-----
  2 | DUOP                   | 2021 | 2509 |    0
  1 | ONPU                   | 1918 | 3020 |    0
  3 | ONUM                   | 1900 | 3032 |    0
(3 rows)

```

```

nezhivih_mariya=> select*,xmin,xmax from university;
 u_id |          name          | year | xmin | xmax
-----+-----+-----+-----+-----
  2 | DUOP                   | 2021 | 2509 |    0
  1 | ONPU                   | 1918 | 3020 |    0
(2 rows)

```

```

nezhivih_mariya=> commit;
COMMIT

```

```

nezhivih_mariya=> select*,xmin,xmax from university;
 u_id |          name          | year | xmin | xmax
-----+-----+-----+-----+-----
  2 | DUOP                   | 2021 | 2509 |    0
  1 | ONPU                   | 1918 | 3020 |    0
  3 | ONUM                   | 1900 | 3032 | 3038
(3 rows)

```

```

nezhivih_mariya=> start transaction;
START TRANSACTION
nezhivih_mariya=> delete from university where u_id = 3;
DELETE 1

```

```

nezhivih_mariya=> select*,xmin,xmax from university;
 u_id |          name          | year | xmin | xmax
-----+-----+-----+-----+-----
  2 | DUOP                   | 2021 | 2509 |    0
  1 | ONPU                   | 1918 | 3020 |    0
  3 | ONUM                   | 1900 | 3032 | 3038
(3 rows)

```

```

nezhivih_mariya=> rollback;
ROLLBACK

```

```

nezhivih_mariya=> select*,xmin,xmax from university;
 u_id |          name          | year | xmin | xmax
-----+-----+-----+-----+-----
  2 | DUOP                   | 2021 | 2509 |    0
  1 | ONPU                   | 1918 | 3020 |    0
  3 | ONUM                   | 1900 | 3032 | 3038
(3 rows)

```

```

nezhivih_mariya=> start transaction;
START TRANSACTION
nezhivih_mariya=> UPDATE university SET year =
nezhivih_mariya-> 1850 WHERE u_id = 3;
UPDATE 1

```

```

nezhivih_mariya=> select*,xmin,xmax from university;
u_id |          name          | year | xmin | xmax
-----+-----+-----+-----+-----
    2 | DUOP                   | 2021 | 2509 |    0
    1 | ONPU                   | 1918 | 3020 |    0
    3 | ONUM                   | 1900 | 3032 | 3039
(3 rows)

```

```

nezhivih_mariya=> commit;
COMMIT

```

```

nezhivih_mariya=> select*,xmin,xmax from university;
u_id |          name          | year | xmin | xmax
-----+-----+-----+-----+-----
    2 | DUOP                   | 2021 | 2509 |    0
    1 | ONPU                   | 1918 | 3020 |    0
    3 | ONUM                   | 1850 | 3039 |    0
(3 rows)

nezhivih_mariya=> commit;
COMMIT

```

## **Завдання 2 Аналіз стану транзакцій на різних рівнях багаторівневого блокування:**

Виконуємо послідовно в двох терміналах наступні комбінації блокувань таблиці: IX-IS, SIX-IX, SIX-IS. Для кожної комбінації блокувань перед завершенням 1-ї транзакції (яка розпочалася раніше) в додатковому терміналі через команду `psql` отримуємо дані про стан транзакцій.

```

nezhivih_mariya=> START TRANSACTION;
START TRANSACTION
nezhivih_mariya=> lock table university in row share mode;
LOCK TABLE

```

```

nezhivih_mariya=> START TRANSACTION;
START TRANSACTION
nezhivih_mariya=> lock table university in row exclusive mode;
LOCK TABLE

```

```

nezhivih_mariya=> select relation,locktype,virtualtransaction,pid,mode,granted
nezhivih_mariya-> from pg_locks
nezhivih_mariya-> where locktype = 'relation';

```

relation	locktype	virtualtransaction	pid	mode	granted
16621	relation	40/148	3918	RowExclusiveLock	t
16765	relation	31/386	781	RowExclusiveLock	t
11673	relation	9/20042	5547	AccessShareLock	t
16759	relation	12/15833	21626	AccessShareLock	t
11673	relation	38/160	3342	AccessShareLock	t
16759	relation	10/10310	18325	AccessShareLock	t
16636	relation	22/1175	30599	AccessShareLock	t
16621	relation	27/956	3910	RowShareLock	t
16765	relation	30/1421	761	RowShareLock	t
16624	relation	36/295	3046	AccessShareLock	t
16759	relation	16/1944	27903	AccessShareLock	t
16687	relation	28/775	4889	AccessShareLock	t
16678	relation	4/104087	1369	ShareRowExclusiveLock	t
16678	relation	33/1991	1464	RowShareLock	t

(14 rows)

```

nezhivih_mariya=> START TRANSACTION;
START TRANSACTION
nezhivih_mariya=> lock table university in share row exclusive mode;
LOCK TABLE

```

```

nezhivih_mariya=> START TRANSACTION;
START TRANSACTION
nezhivih_mariya=> lock table university in row exclusive mode;

```

```

nezhivih_mariya=> select relation,locktype,virtualtransaction,pid,mode,granted
from pg_locks
where locktype = 'relation';

```

relation	locktype	virtualtransaction	pid	mode	granted
11673	relation	9/20043	5547	AccessShareLock	t
16759	relation	12/15833	21626	AccessShareLock	t
11673	relation	38/160	3342	AccessShareLock	t
16759	relation	16/1944	27903	AccessShareLock	t
16687	relation	28/775	4889	AccessShareLock	t
16678	relation	37/628	6270	AccessShareLock	t
16678	relation	37/628	6270	RowExclusiveLock	f
16621	relation	40/150	3918	RowExclusiveLock	f
16765	relation	30/1422	761	ShareRowExclusiveLock	t
16732	relation	43/56	4602	RowShareLock	t
16678	relation	4/104087	1369	ShareRowExclusiveLock	t
16678	relation	39/422	6278	AccessShareLock	t
16678	relation	39/422	6278	RowExclusiveLock	f
16732	relation	15/3802	4505	ShareRowExclusiveLock	t
16765	relation	31/387	781	RowExclusiveLock	f
16678	relation	33/1991	1464	RowShareLock	t
16621	relation	27/957	3910	ShareRowExclusiveLock	t

(17 rows)

```

nezhivih_mariya=> START TRANSACTION;
START TRANSACTION
nezhivih_mariya=> lock table university in share row exclusive mode;
LOCK TABLE

```

```

nezhivih_mariya=> START TRANSACTION;
START TRANSACTION
nezhivih_mariya=> lock table university in row share mode;
LOCK TABLE

```

```

nezhivih_mariya=> select relation,locktype,virtualtransaction,pid,mode,granted
from pg_locks
where locktype = 'relation';

```

relation	locktype	virtualtransaction	pid	mode	granted
11673	relation	9/20044	5547	AccessShareLock	t
16759	relation	12/15833	21626	AccessShareLock	t
11673	relation	38/160	3342	AccessShareLock	t
16636	relation	3/85146	7470	RowShareLock	t
16636	relation	5/83721	7153	RowExclusiveLock	t
11673	relation	34/1344	2765	AccessShareLock	t
16759	relation	16/1944	27903	AccessShareLock	t
16687	relation	28/775	4889	AccessShareLock	t
16678	relation	37/628	6270	AccessShareLock	t
16678	relation	37/628	6270	RowExclusiveLock	f
16621	relation	40/150	3918	RowExclusiveLock	f
16624	relation	36/296	3046	RowExclusiveLock	f
16765	relation	30/1423	761	ShareRowExclusiveLock	t
16732	relation	43/56	4602	RowShareLock	t
16678	relation	4/104087	1369	ShareRowExclusiveLock	t
16624	relation	34/1344	2765	ShareRowExclusiveLock	t
16678	relation	39/422	6278	AccessShareLock	t
16678	relation	39/422	6278	RowExclusiveLock	f
16732	relation	15/3802	4505	ShareRowExclusiveLock	t

--More--

### **Завдання 3 Керування квазіпаралельним виконанням транзакцій на різних рівнях ізоляції транзакцій:**

Готуємо транзакції, які було створено у завданні 3.1 рішення попередньої лабораторної роботи, а саме, створюємо 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.

№	Команди T1	Команди T2
1	START TRANSACTION;	
2		START TRANSACTION;
	SET TRANSACTION ISOLATION LEVEL (Потрібний режим блокування);	
		SET TRANSACTION ISOLATION LEVEL (Потрібний режим блокування);
3	SELECT * FROM university WHERE u_id = 1;	
4		SELECT * FROM university WHERE u_id = 1;
6	UPDATE university SET year = 1974 WHERE u_id = 1;	
		UPDATE university SET year = 1975 WHERE u_id = 1;
9	SELECT * FROM university WHERE u_id = 1;	
10	COMMIT;	
12		SELECT * FROM university WHERE u_id = 1;
13		COMMIT;

Виконуємо роботу транзакцій при умові їх роботи на рівні ізоляції READ COMMITTED.



```

nezhivih_mariya=> START TRANSACTION;
START TRANSACTION
nezhivih_mariya=> SET TRANSACTION ISOLATION
nezhivih_mariya-> LEVEL
nezhivih_mariya-> read committed;
SET
nezhivih_mariya=> SELECT * FROM university WHERE u_id = 1;
  u_id |          name          | year
-----+-----+-----
      1 | ONPU                   | 1918
(1 row)

nezhivih_mariya=> UPDATE university SET year = 1974 WHERE u_id = 1;
UPDATE 1
nezhivih_mariya=> SELECT * FROM university WHERE u_id = 1;
  u_id |          name          | year
-----+-----+-----
      1 | ONPU                   | 1974
(1 row)

nezhivih_mariya=> COMMIT;
COMMIT

```

```

nezhivih_mariya=> START TRANSACTION;
START TRANSACTION
nezhivih_mariya=> SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
SET
nezhivih_mariya=> SELECT * FROM university WHERE u_id = 1;
  u_id |          name          | year
-----+-----+-----
      1 | ONPU                   | 1918
(1 row)

nezhivih_mariya=> UPDATE university SET year = 1975 WHERE u_id = 1;
UPDATE 1
nezhivih_mariya=> SELECT * FROM university WHERE u_id = 1;
  u_id |          name          | year
-----+-----+-----
      1 | ONPU                   | 1975
(1 row)

nezhivih_mariya=> COMMIT;
COMMIT

```

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

```

nezhivih_mariya=> START TRANSACTION;
START TRANSACTION
nezhivih_mariya=> SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
SET
nezhivih_mariya=> SELECT * FROM university WHERE u_id = 1;
  u_id |          name          | year
-----+-----+-----
      1 | ONPU                   | 1975
(1 row)

nezhivih_mariya=> UPDATE university SET year = 1974 WHERE u_id = 1;
UPDATE 1
nezhivih_mariya=> SELECT * FROM university WHERE u_id = 1;
  u_id |          name          | year
-----+-----+-----
      1 | ONPU                   | 1974
(1 row)

nezhivih_mariya=> COMMIT;
COMMIT

```

```

nezhivih_mariya=> START TRANSACTION;
START TRANSACTION
nezhivih_mariya=> SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
SET
nezhivih_mariya=> SELECT * FROM university WHERE u_id = 1;
  u_id |          name          | year
-----+-----+-----
      1 | ONPU                   | 1975
(1 row)

nezhivih_mariya=> UPDATE university SET year = 1975 WHERE u_id = 1;
ERROR:  could not serialize access due to concurrent update
nezhivih_mariya=> SELECT * FROM university WHERE u_id = 1;
ERROR:  current transaction is aborted, commands ignored until end of transaction block
nezhivih_mariya=> COMMIT;
ROLLBACK

```

Повторюємо роботу транзакцій при умові їх роботи на рівні ізоляції  
SERIALIZABLE.

```

nezhivih_mariya=> START TRANSACTION;
START TRANSACTION
nezhivih_mariya=> SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
SET
nezhivih_mariya=> SELECT * FROM university WHERE u_id = 1;
 u_id |          name          | year
-----+-----+-----
      1 | ONPU                   | 1974
(1 row)

nezhivih_mariya=> UPDATE university SET year = 1974 WHERE u_id = 1;
UPDATE 1
nezhivih_mariya=> SELECT * FROM university WHERE u_id = 1;
 u_id |          name          | year
-----+-----+-----
      1 | ONPU                   | 1974
(1 row)

nezhivih_mariya=> COMMIT;
COMMIT

```

```

nezhivih_mariya=> START TRANSACTION;
START TRANSACTION
nezhivih_mariya=> SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
SET
nezhivih_mariya=> SELECT * FROM university WHERE u_id = 1;
 u_id |          name          | year
-----+-----+-----
      1 | ONPU                   | 1974
(1 row)

nezhivih_mariya=> UPDATE university SET year = 1975 WHERE u_id = 1;
ERROR:  could not serialize access due to concurrent update
nezhivih_mariya=> SELECT * FROM university WHERE u_id = 1;
ERROR:  current transaction is aborted, commands ignored until end of transaction block
nezhivih_mariya=> COMMIT;
ROLLBACK

```

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

Виконуємо модифікацію транзакцій так, щоб вони утворювали тупикову ситуацію.

№	Команди T1	Команди T2
1	START TRANSACTION;	
2		START TRANSACTION;
	SET TRANSACTION ISOLATION	

	LEVEL (Потрібний режим блокування);	
		SET TRANSACTION ISOLATION LEVEL (Потрібний режим блокування);
3	SELECT * FROM university WHERE u_id = 1;	
4		SELECT * FROM university WHERE u_id = 1;
6	UPDATE university SET year = 1974 WHERE u_id = 1;	
		UPDATE university SET year = 1975 WHERE u_id = 2;
9	UPDATE university SET year = 1974 WHERE u_id = 2;	
10		UPDATE university SET year = 1975 WHERE u_id = 1;
12	COMMIT;	
13		COMMIT;

```

nezhivih_mariya=> START TRANSACTION;
START TRANSACTION
nezhivih_mariya=> SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
SET
nezhivih_mariya=> SELECT * FROM university WHERE u_id = 1;
  u_id |          name          | year
-----+-----+-----
      1 | ONPU                   | 1974
(1 row)

nezhivih_mariya=> UPDATE university SET year = 1974 WHERE u_id = 1;
UPDATE 1
nezhivih_mariya=> UPDATE university SET year = 1974 WHERE u_id = 2;
UPDATE 1
nezhivih_mariya=> COMMIT;
COMMIT

```

```

nezhivih_mariya=> START TRANSACTION;
START TRANSACTION
nezhivih_mariya=> SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
SET
nezhivih_mariya=> SELECT * FROM university WHERE u_id = 1;
 u_id |          name          | year
-----+-----+-----
      1 | ONPU                   | 1974
(1 row)

nezhivih_mariya=> UPDATE university SET year = 1975 WHERE u_id = 2;
UPDATE 1
nezhivih_mariya=> UPDATE university SET year = 1975 WHERE u_id = 1;
ERROR:  deadlock detected
DETAIL:  Process 781 waits for ShareLock on transaction 3085; blocked by process
       761.
Process 761 waits for ShareLock on transaction 3086; blocked by process 781.
HINT:   See server log for query details.
CONTEXT:  while updating tuple (0,13) in relation "university"
nezhivih_mariya=> COMMIT;
ROLLBACK

```

```

[nezhivih_mariya@vpsj3IeQ ~]$ ps -u postgres -o pid,ppid,stat,cmd
PID  PPID  STAT  CMD
761   8763  Ss    postgres: nezhivih_mariya nezhivih_mariya [local] idle
781   8763  Ss    postgres: nezhivih_mariya nezhivih_mariya [local] idle

```

**Висновок:** У ході лабораторної роботи було досліджено поведінку процесів-транзакцій в базах даних та засоби керування ними через механізм блокування з використанням сучасних систем керування базами даних.