

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ ПОЛІТЕХНІЧНИЙ УНІВЕРСИТЕТ

Інститут комп'ютерних систем
Кафедра інформаційних систем

Варіант №3

Лабораторна робота №10

По дисципліні “Операційні системи”

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

Виконав:

Студент групи AI-202

Боднар І.В.

Перевірили:

Блажко О.А

Одеса 2021

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:

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 airplane;		
3	start transaction;			
4	select txid_current();			
5	insert into airplane value(3, 'TU-34', 2003);			
6	select*,xmin,xmax from airplane;			
7		select*,xmin,xmax from airplane;		
8	commit;			
9		select*,xmin,xmax		

		from airplane;		
10			start transaction;	
11			delete from airplane where a_id = 3;	
12		select*,xmin,xmax from airplane;		
13			rollback;	
14		select*,xmin,xmax from airplane;		
15				start transaction;
16				UPDATE airplane SET year = 2000 WHERE a_id = 3;
17		select*,xmin,xmax from airplane;		
18				commit;
19		select*,xmin,xmax from airplane;		
20		commit;		

```

bodnar_illya=> start transaction;
START TRANSACTION
bodnar_illya=> select txid_current();
txid_current
-----
2895
(1 row)

bodnar_illya=> insert into airplane values(3,'TU-34',2003);
INSERT 0 1
bodnar_illya=> select *,xmin,xmax from airplane;
 a_id |          name          | year | xmin | xmax
-----+-----+-----+-----+-----
  2   | AN-124                 | 1984 | 2393 |    0
  1   | TU-107                 | 1986 | 2397 |    0
  3   | TU-34                  | 2003 | 2895 |    0
(3 rows)

bodnar_illya=> commit;
COMMIT
bodnar_illya=> █

```

T1:

```

bodnar_illya=> start transaction;
START TRANSACTION
bodnar_illya=> select *,xmin,xmax from airplane;
 a_id |          name          | year | xmin | xmax
-----+-----+-----+-----+-----
  2   | AN-124                 | 1984 | 2393 |    0
  1   | TU-107                 | 1986 | 2397 |    0
(2 rows)

bodnar_illya=> select *,xmin,xmax from airplane;
 a_id |          name          | year | xmin | xmax
-----+-----+-----+-----+-----
  2   | AN-124                 | 1984 | 2393 |    0
  1   | TU-107                 | 1986 | 2397 |    0
(2 rows)

```

T2:

```

bodnar_illya=> █

```

```

bodnar_illya=> start transaction;
START TRANSACTION
bodnar_illya=> delete from airplane where a_id = 3;
DELETE 1
bodnar_illya=> rollback;
ROLLBACK
bodnar_illya=> 

```

T3:

```

bodnar_illya=> select *,xmin,xmax from airplane;
a_id |          name          | year | xmin | xmax
-----+-----+-----+-----+-----
    2 | AN-124                 | 1984 | 2393 |    0
    1 | TU-107                 | 1986 | 2397 |    0
    3 | TU-34                  | 2003 | 2895 | 2898
(3 rows)

```

```

bodnar_illya=> select *,xmin,xmax from airplane;
a_id |          name          | year | xmin | xmax
-----+-----+-----+-----+-----
    2 | AN-124                 | 1984 | 2393 |    0
    1 | TU-107                 | 1986 | 2397 |    0
    3 | TU-34                  | 2003 | 2895 | 2898
(3 rows)

```

T2:

```

bodnar_illya=> 

```

```

bodnar_illya=> UPDATE airplane SET year =
bodnar_illya-> 2000 WHERE a_id = 3;
UPDATE 1
bodnar_illya=> commit;
COMMIT

```

T4:

```

bodnar_illya=> 

```

```

bodnar_illya=> select *,xmin,xmax from airplane;
a_id |          name          | year | xmin | xmax
-----+-----+-----+-----+-----
      2 | AN-124                  | 1984 | 2393 |    0
      1 | TU-107                  | 1986 | 2397 |    0
      3 | TU-34                   | 2003 | 2895 | 2899
(3 rows)

bodnar_illya=> select *,xmin,xmax from airplane;
a_id |          name          | year | xmin | xmax
-----+-----+-----+-----+-----
      2 | AN-124                  | 1984 | 2393 |    0
      1 | TU-107                  | 1986 | 2397 |    0
      3 | TU-34                   | 2000 | 2899 |    0
(3 rows)

bodnar_illya=> commit;
COMMIT
bodnar_illya=> █

```

T2:

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.

IX-IS:

```

bodnar_illya=> start transaction;
START TRANSACTION
bodnar_illya=> lock table airplane in row share mode;
LOCK TABLE
bodnar_illya=> █

```

```

bodnar_illya=> start transaction;
START TRANSACTION
bodnar_illya=> lock table airplane in row exclusive mode;
LOCK TABLE
bodnar_illya=> 

```

```

bodnar_illya-> from pg_locks
bodnar_illya-> where locktype = 'relation';

```

relation	locktype	virtualtransaction	pid	mode	granted
16711	relation	8/26065	19625	RowShareLock	t
16696	relation	10/9344	11451	AccessShareLock	t
16711	relation	7/19193	19594	RowExclusiveLock	t
16606	relation	11/7709	12199	RowExclusiveLock	t
16690	relation	17/2328	25041	RowExclusiveLock	t
16651	relation	12/15491	12628	AccessShareLock	t
16606	relation	13/3904	10530	RowShareLock	t
3455	relation	13/3904	10530	AccessShareLock	t
2663	relation	13/3904	10530	AccessShareLock	t
2662	relation	13/3904	10530	AccessShareLock	t
2685	relation	13/3904	10530	AccessShareLock	t
2684	relation	13/3904	10530	AccessShareLock	t
2615	relation	13/3904	10530	AccessShareLock	t
1259	relation	13/3904	10530	AccessShareLock	t
11673	relation	23/831	1899	AccessShareLock	t
16687	relation	9/16367	24025	AccessShareLock	t
16687	relation	9/16367	24025	RowExclusiveLock	t
16687	relation	22/1165	1374	AccessShareLock	t
16687	relation	22/1165	1374	RowExclusiveLock	f

--More--

SIX-IX:

```

bodnar_illya=> start transaction;
START TRANSACTION
bodnar_illya=> lock table airplane in share row exclusive mode;
LOCK TABLE
bodnar_illya=> 

```

```

bodnar_illya=> start transaction;
START TRANSACTION
bodnar_illya=> lock table airplane in row exclusive mode;

```

2615	relation	13/3904	10530	AccessShareLock	t
1259	relation	13/3904	10530	AccessShareLock	t
11673	relation	23/832	1899	AccessShareLock	t
16687	relation	9/16367	24025	AccessShareLock	t
16687	relation	9/16367	24025	RowExclusiveLock	t
16711	relation	8/26066	19625	RowExclusiveLock	f
16687	relation	3/82551	3616	AccessShareLock	t
16687	relation	3/82551	3616	ShareRowExclusiveLock	f
16687	relation	22/1165	1374	AccessShareLock	t
16687	relation	22/1165	1374	RowExclusiveLock	f
16711	relation	7/19194	19594	ShareRowExclusiveLock	t
16687	relation	21/2981	1362	AccessShareLock	t
16687	relation	21/2981	1362	ShareRowExclusiveLock	f
16687	relation	6/66565	19017	AccessShareLock	t
16687	relation	6/66565	19017	RowShareLock	t
16687	relation	6/66565	19017	RowExclusiveLock	t
16687	relation	16/1924	24042	AccessShareLock	t
16687	relation	16/1924	24042	RowShareLock	t
16687	relation	16/1924	24042	RowExclusiveLock	t
16687	relation	2/788731	23181	AccessShareLock	t
16687	relation	2/788731	23181	RowExclusiveLock	t
16687	relation	4/100992	19003	AccessShareLock	t
16687	relation	4/100992	19003	RowExclusiveLock	t
16687	relation	15/3748	3661	AccessShareLock	t
16687	relation	15/3748	3661	RowExclusiveLock	f

SIX-IS:

```
bodnar_illya=> start transaction;
START TRANSACTION
bodnar_illya=> lock table airplane in share row exclusive mode;
LOCK TABLE
bodnar_illya=> █
```

```
bodnar_illya=> start transaction;
START TRANSACTION
bodnar_illya=> lock table airplane in row share mode;
LOCK TABLE
bodnar_illya=> █
```


16696	relation	10/9344	11451	AccessShareLock	t
16606	relation	11/7709	12199	RowExclusiveLock	t
16651	relation	12/15491	12628	AccessShareLock	t
16606	relation	13/3904	10530	RowShareLock	t
3455	relation	13/3904	10530	AccessShareLock	t
2663	relation	13/3904	10530	AccessShareLock	t
2662	relation	13/3904	10530	AccessShareLock	t
2685	relation	13/3904	10530	AccessShareLock	t
2684	relation	13/3904	10530	AccessShareLock	t
2615	relation	13/3904	10530	AccessShareLock	t
1259	relation	13/3904	10530	AccessShareLock	t
11673	relation	23/833	1899	AccessShareLock	t
16687	relation	9/16367	24025	AccessShareLock	t
16687	relation	9/16367	24025	RowExclusiveLock	t
16711	relation	8/26067	19625	RowShareLock	t
16687	relation	3/82551	3616	AccessShareLock	t
16687	relation	3/82551	3616	ShareRowExclusiveLock	f
16687	relation	22/1165	1374	AccessShareLock	t
16687	relation	22/1165	1374	RowExclusiveLock	f
16711	relation	7/19195	19594	ShareRowExclusiveLock	t
16687	relation	21/2981	1362	AccessShareLock	t
16687	relation	21/2981	1362	ShareRowExclusiveLock	f
16687	relation	6/66565	19017	AccessShareLock	t
16687	relation	6/66565	19017	RowShareLock	t
16687	relation	6/66565	19017	RowExclusiveLock	t
16687	relation	16/1924	24042	AccessShareLock	t
16687	relation	16/1924	24042	RowShareLock	t

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 airplane WHERE a_id = 1;	
4		SELECT * FROM airplane WHERE a_id = 1;
6	UPDATE airplane SET year = 1967 WHERE a_id = 1;	
		UPDATE airplane SET year = 1987 WHERE a_id = 1;
9	SELECT * FROM airplane WHERE a_id = 1;	
10	COMMIT;	
12		SELECT * FROM airplane WHERE a_id = 1;
13		COMMIT;

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

T1:

```
bodnar_illya=> start transaction;
START TRANSACTION
bodnar_illya=> SET TRANSACTION ISOLATION
LEVEL
read committed;
SET
bodnar_illya=> UPDATE airplane SET year = 1967 WHERE a_id = 1;
UPDATE 1
bodnar_illya=> SELECT * FROM airplane WHERE a_id = 1;
a_id | name | year
-----+-----+-----
1 | TU-107 | 1967
(1 row)

bodnar_illya=> commit;
COMMIT
bodnar_illya=> 
```

T2:

```
bodnar_illya=> start transaction;
START TRANSACTION
bodnar_illya=> SET TRANSACTION ISOLATION
LEVEL
read committed;
SET
bodnar_illya=> UPDATE airplane SET year = 1967 WHERE a_id = 1;
UPDATE 1
bodnar_illya=> SELECT * FROM airplane WHERE a_id = 1;
a_id | name | year
-----+-----+-----
1 | TU-107 | 1967
(1 row)

bodnar_illya=> commit;
COMMIT
bodnar_illya=> 
```

- 3.2. Повторіть роботу транзакцій при умові їх роботи на рівні ізоляції REPEATABLE READ.

T1:

```
START TRANSACTION
bodnar_illya=> SET TRANSACTION ISOLATION
LEVEL repeatable read;
SET
bodnar_illya=> UPDATE airplane SET year = 1967 WHERE a_id = 1;
UPDATE 1
bodnar_illya=> SELECT * FROM airplane WHERE a_id = 1;
a_id | name | year
-----+-----+-----
1 | TU-107 | 1967
(1 row)

bodnar_illya=> commit;
COMMIT
bodnar_illya=> █
```

T2:

```
bodnar_illya=> start transaction;
START TRANSACTION
bodnar_illya=> SET TRANSACTION ISOLATION
LEVEL repeatable read;
SET
bodnar_illya=> UPDATE airplane SET year = 1967 WHERE a_id = 1;
ERROR: could not serialize access due to concurrent update
bodnar_illya=> █
```

3.3. Повторіть роботу транзакцій при умові їх роботи на рівні ізоляції **SERIALIZABLE**.

T1:

```
bodnar_illya=> start transaction
bodnar_illya-> ;
START TRANSACTION
bodnar_illya=> SET TRANSACTION ISOLATION
bodnar_illya-> LEVEL SERIALIZABLE;
SET
bodnar_illya=> SELECT * FROM airplane WHERE a_id = 1;
a_id |          name          | year
-----+-----+-----
    1 | TU-107                 | 1967
(1 row)

bodnar_illya=> UPDATE airplane SET year = 1967 WHERE a_id = 1;
UPDATE 1
bodnar_illya=> SELECT * FROM airplane WHERE a_id = 1;
a_id |          name          | year
-----+-----+-----
    1 | TU-107                 | 1967
(1 row)

bodnar_illya=> commit;
COMMIT
bodnar_illya=> █
```

T2:

```
bodnar_illya=> start transaction;
START TRANSACTION
bodnar_illya=>
bodnar_illya=> SET TRANSACTION ISOLATION
bodnar_illya-> LEVEL SERIALIZABLE;
SET
bodnar_illya=> SELECT * FROM airplane WHERE a_id = 1;
a_id |          name          | year
-----+-----+-----
    1 | TU-107                 | 1967
(1 row)

bodnar_illya=> UPDATE airplane SET year = 1967 WHERE a_id = 1;
ERROR:  could not serialize access due to concurrent update
bodnar_illya=> █
```

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 airplane WHERE a_id = 1;	
4		SELECT * FROM airplane WHERE a_id = 1;
6	UPDATE airplane SET year = 1967 WHERE a_id = 1;	
		UPDATE airplane SET year = 1987 WHERE a_id = 2;
9	UPDATE airplane SET year = 1967 WHERE a_id = 2;	
10		UPDATE airplane SET year = 1967 WHERE a_id = 1;
12	COMMIT;	
13		COMMIT;

T1:

```
bodnar_illya=> start transaction
;
START TRANSACTION
bodnar_illya=> SET TRANSACTION ISOLATION
LEVEL SERIALIZABLE;
SET
bodnar_illya=> SELECT * FROM airplane WHERE a_id = 1;
a_id |          name          | year
-----+-----+-----
    1 | TU-107                 | 1967
(1 row)

bodnar_illya=> UPDATE airplane SET year  = 1967 WHERE a_id = 1;
UPDATE 1
bodnar_illya=> UPDATE airplane SET year  = 1967 WHERE a_id = 2;
UPDATE 1
bodnar_illya=> []
```

T2:

```
bodnar_illya=> start transaction;
START TRANSACTION
bodnar_illya=> SET TRANSACTION ISOLATION
LEVEL serializable;
SET
bodnar_illya=> SELECT * FROM airplane WHERE a_id = 1;
a_id |          name          | year
-----+-----+-----
    1 | TU-107                 | 1967
(1 row)

bodnar_illya=> UPDATE airplane SET year  = 1967 WHERE a_id = 2;
UPDATE 1
bodnar_illya=> UPDATE airplane SET year  = 1967 WHERE a_id = 1;
ERROR:  deadlock detected
DETAIL:  Process 19625 waits for ShareLock on transaction 2928; blocked by process 21005.
Process 21005 waits for ShareLock on transaction 2929; blocked by process 19625.
HINT:  See server log for query details.
CONTEXT:  while updating tuple (0,21) in relation "airplane"
bodnar_illya=> []
```

T3:

```
19625  8763 Ss  postgres: bodnar_illya bodnar_illya [local] idle in transaction (abort
20258  8763 Ss  postgres: markovskij_danilo markovskij_danilo [local] idle
20313  8763 Ss  postgres: markovskij_danilo markovskij_danilo [local] idle in transact
21005  8763 Ss  postgres: bodnar_illya bodnar_illya [local] idle in transaction
```

Висновок: У процесі виконання лабораторної роботи ми дослідили поведінку процесів-транзакцій в базах даних та засоби керування ними через механізм блокування з використанням сучасних систем керування базами даних.