

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ПОЛІТЕХНІКА»
ІНСТИТУТ КОМП'ЮТЕРНИХ СИСТЕМ
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ

Лабораторна робота №10
з дисципліни
“Операційні системи”
Тема
««Програмування керуванням процесами в ОС Unix»
Варіант 6

Виконав:
Студент групи АІ-203
Шуляк М.Р.

Одеса 2021

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

Завдання для виконання:

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.

Хід роботи

Завдання 1. Аналіз роботи багато версійного протоколу

T1

```
shulyak_mikola=> SELECT * FROM teacher
```

```
shulyak_mikola-> ;
```

p_id	name	post
2	Shulyak	Professor
1	Andryuxa	docent

(2 rows)

```
shulyak_mikola=> █
```

```
shulyak_mikola=> START TRANSACTION;
```

```
START TRANSACTION
```

```
shulyak_mikola=> SELECT txid_current();
```

txid_current
3467

(1 row)

```
shulyak_mikola=> INSERT INTO teacher VALUES(3,'Simakov','Magistr');
```

```
INSERT 0 1
```

```
shulyak_mikola=> SELECT xmin,xmax,p_id,name,post FROM teacher;
```

xmin	xmax	p_id	name	post
3456	0	2	Shulyak	Professor
3462	0	1	Andryuxa	docent
3467	0	3	Simakov	Magistr

(3 rows)

```
shulyak_mikola=> commit;
```

```
COMMIT
```

```
shulyak_mikola=> █
```

T2

```

shulyak_mikola=> START TRANSACTION;
START TRANSACTION
shulyak_mikola=> SELECT xmin,xmax,p_id,name,post FROM teacher ;
  xmin | xmax | p_id |      name      |      post
-----+-----+-----+-----+-----
 3456 |    0 |    2 | Shulyak        | Professor
 3462 |    0 |    1 | Andryuxa       | docent
 3467 |    0 |    3 | Simakov        | Magistr
(3 rows)

shulyak_mikola=> SELECT xmin, xmax,p_id,name,post FROM teacher ;
  xmin | xmax | p_id |      name      |      post
-----+-----+-----+-----+-----
 3456 |    0 |    2 | Shulyak        | Professor
 3462 |    0 |    1 | Andryuxa       | docent
 3467 |    0 |    3 | Simakov        | Magistr
(3 rows)

shulyak_mikola=> █

```

T2 не бачить зміни здійснені T1 до їх фіксації. Після фіксації до таблиці додається новий рядок, xmin якого дорівнює 3149 - номеру T1, яка виконала зміни.

T3

```
shulyak_mikola=> START TRANSACTION ;
START TRANSACTION
shulyak_mikola=> DELETE FROM teacher WHERE p_id = 2;
DELETE 1
shulyak_mikola=> rollback;
ROLLBACK
shulyak_mikola=> █
```

T2

```
shulyak_mikola=> SELECT xmin,xmax,p_id,name,post FROM teacher ;
  xmin | xmax | p_id |      name      |      post
-----+-----+-----+-----+-----
  3456 | 3471 |    2 | Shulyak        | Professor
  3462 |    0 |    1 | Andryuxa       | docent
  3467 |    0 |    3 | Simakov        | Magistr
(3 rows)
```

```
shulyak_mikola=> SELECT xmin,xmax,p_id,name,post FROM teacher ;
  xmin | xmax | p_id |      name      |      post
-----+-----+-----+-----+-----
  3456 | 3471 |    2 | Shulyak        | Professor
  3462 |    0 |    1 | Andryuxa       | docent
  3467 |    0 |    3 | Simakov        | Magistr
(3 rows)
```

Після видалення рядку та відміни операції xmax рядку змінився на 3471, що показує, що над цим рядком здійснювалися операції транзакцією з номером 3471.

T4

```
shulyak_mikola=> START TRANSACTION ;
START TRANSACTION
shulyak_mikola=> UPDATE teacher SET post =
shulyak_mikola-> ^C
shulyak_mikola=> UPDATE teacher SET post = 'Uvolen' WHERE p_id = 3;
UPDATE 1
shulyak_mikola=> COMMIT;
COMMIT
shulyak_mikola=> █
```

T2

```
shulyak_mikola=> SELECT xmin,xmax,p_id,name,post FROM teacher ;
xmin | xmax | p_id |      name      |      post
-----+-----+-----+-----+-----
3456 | 3471 |    2 | Shulyak        | Professor
3462 |    0 |    1 | Andryuxa       | docent
3474 |    0 |    3 | Simakov       | Uvolen
(3 rows)
```

```
shulyak_mikola=> SELECT xmin,xmax,p_id,name,post FROM teacher ;
xmin | xmax | p_id |      name      |      post
-----+-----+-----+-----+-----
3456 | 3471 |    2 | Shulyak        | Professor
3462 |    0 |    1 | Andryuxa       | docent
3474 | 3476 |    3 | Simakov       | Uvolen
(3 rows)
```

```
shulyak_mikola=> SELECT xmin,xmax,p_id,name,post FROM teacher ;
xmin | xmax | p_id |      name      |      post
-----+-----+-----+-----+-----
3456 | 3471 |    2 | Shulyak        | Professor
3462 |    0 |    1 | Andryuxa       | docent
3474 |    0 |    3 | Simakov       | Uvolen
(3 rows)
```

xmax рядку 3 змінився на 3476, що означає виконання дій над ним, а після фіксації цих змін xmin = 3747, xmax = 0 - поточне значення було створено транзакцією з номером 3149, але поки немає нових версій, створених іншими транзакціями.

2. Аналіз стану транзакцій на різних рівнях багаторівневого блокування

IX - IS

```

login as: shulyak_mikola
shulyak_mikola@91.219.60.189's password:
Access denied
shulyak_mikola@91.219.60.189's password:
Last failed login: Wed May  5 23:06:17 EEST 2021 from 78.26.173.150 on ssh:notty
There was 1 failed login attempt since the last successful login.
Last login: Wed May  5 22:58:48 2021 from 78.26.173.150
[shulyak_mikola@vpsj3IeQ ~]$ psql
psql (9.5.25)
Type "help" for help.

```

```

shulyak_mikola=> START TRANSACTION ;
START TRANSACTION
shulyak_mikola=> lock table teacher in row exclusive mode;
LOCK TABLE
shulyak_mikola=> select relation,locktype,virtualtransaction,pid,mode,granted fr
om pg_locks where locktype = 'relation';

```

relation	locktype	virtualtransaction	pid	mode	granted
16858	relation	4/139895	9385	AccessShareLock	t
11673	relation	9/26750	11176	AccessShareLock	t
16855	relation	9/26750	11176	RowExclusiveLock	t
3455	relation	9/26750	11176	AccessShareLock	t
2663	relation	9/26750	11176	AccessShareLock	t
2662	relation	9/26750	11176	AccessShareLock	t
2685	relation	9/26750	11176	AccessShareLock	t
2684	relation	9/26750	11176	AccessShareLock	t
2615	relation	9/26750	11176	AccessShareLock	t
1259	relation	9/26750	11176	AccessShareLock	t
16837	relation	8/30389	10340	AccessShareLock	t
16702	relation	3/94520	6750	RowShareLock	t

```

(12 rows)

```

```

shulyak_mikola=> 

```

```

2663 | relation | 9/26750 | 11176 | AccessShareLock | t
2615 | relation | 9/26750 | 11176 | AccessShareLock | t
1259 | relation | 9/26750 | 11176 | AccessShareLock | t
16702 | relation | 3/94520 | 6750 | RowShareLock | t
(11 rows)

```

```

shulyak_mikola=> 

```

Блокування IX та IS сумісні.

SIX-IX

```
shulyak_mikola@91.219.60.189's password:
Last login: Wed May  5 23:06:20 2021 from 78.26.173.150
[shulyak_mikola@vpsj3IeQ ~]$ psql
psql (9.5.25)
Type "help" for help.
```

```
shulyak_mikola=> start transaction ;
START TRANSACTION
```

```
shulyak_mikola=> lock table teacher in share row exclusive mode;
sele
```

```
^Ccancel request sent
```

```
ERROR: canceling statement due to user request
```

```
shulyak_mikola=> select relation,locktype,virtualtransaction,pid,mode,granted, f
rom pg_locks where locktype='relation';
```

```
ERROR: syntax error at or near "from"
```

```
LINE 1: ...ion,locktype,virtualtransaction,pid,mode,granted, from pg_lo...
                                     ^
```

```
shulyak_mikola=> select relation,locktype,virtualtransaction,pid,mode,granted,from pg_
ktype='relation';
```

```
ERROR: syntax error at or near "from"
```

```
LINE 1: ...tion,locktype,virtualtransaction,pid,mode,granted,from pg_l...
                                     ^
```

```
shulyak_mikola=> select relation,locktype,virtualtransaction,pid,mode,granted from pg_l
type='relation';
```

```
ERROR: current transaction is aborted, commands ignored until end of transaction block
```

```
shulyak_mikola=> rollback;
```

```
ROLLBACK
```

```
shulyak_mikola=> select relation,locktype,virtualtransaction,pid,mode,granted from pg_l
type='relation';
```

relation	locktype	virtualtransaction	pid	mode	granted
16837	relation	8/30559	12683	RowExclusiveLock	t
16858	relation	4/139895	9385	AccessShareLock	t
11673	relation	9/26750	11176	AccessShareLock	t
3455	relation	9/26750	11176	AccessShareLock	t
2663	relation	9/26750	11176	AccessShareLock	t
2662	relation	9/26750	11176	AccessShareLock	t
2685	relation	9/26750	11176	AccessShareLock	t
2684	relation	9/26750	11176	AccessShareLock	t
2615	relation	9/26750	11176	AccessShareLock	t
1259	relation	9/26750	11176	AccessShareLock	t
16837	relation	7/23489	12677	AccessShareLock	t
11673	relation	5/88042	12370	AccessShareLock	t
16702	relation	3/94520	6750	RowShareLock	t
16855	relation	9/26750	11176	RowExclusiveLock	t

```
(14 rows)
```

```
shulyak mikola=> █
```

```

16687 | relation | 52/34          | 30003 | ShareRowExclusiveLock | f
16630 | relation | 35/334        | 32648 | RowShareLock          | t
16687 | relation | 23/1591       | 31825 | ShareRowExclusiveLock | f
16687 | relation | 13/7428       | 2290  | ShareRowExclusiveLock | f
16723 | relation | 3/85913       | 24022 | ShareRowExclusiveLock | t
16630 | relation | 7/20465       | 32544 | ShareRowExclusiveLock | t
16687 | relation | 44/374        | 2645  | ShareRowExclusiveLock | f
16687 | relation | 15/4090       | 24544 | ShareRowExclusiveLock | t
16687 | relation | 28/897        | 32340 | ShareRowExclusiveLock | f
16687 | relation | 31/1015       | 29331 | RowExclusiveLock      | f
(33 rows)

```

```

shulyak_mikola=> start transaction ;
START TRANSACTION
shulyak_mikola=> lock table teacher in row exclusive mode;
LOCK TABLE
shulyak_mikola=> █

```

Блокування SIX та IX не сумісні.

```

shulyak_mikola=> start transaction;
WARNING:  there is already a transaction in progress
START TRANSACTION
shulyak_mikola=> lock table teacher in row exclusive mode;
LOCK TABLE
shulyak_mikola=> select relation,locktype,virtualtransaction,pid,mode,granted fr
om pg_locks where locktype='relation';
 relation | locktype | virtualtransaction | pid | mode | granted
-----+-----+-----+-----+-----+-----
 16858 | relation | 4/139895          | 9385 | AccessShareLock | t
 11673 | relation | 9/26751           | 11176 | AccessShareLock | t
 16855 | relation | 9/26751           | 11176 | RowExclusiveLock | t
 16837 | relation | 7/23489           | 12677 | AccessShareLock | t
 16837 | relation | 10/13629          | 13055 | RowExclusiveLock | t
 16702 | relation | 3/94520           | 6750 | RowShareLock    | t
(6 rows)

```

```

shulyak_mikola=> █
shulyak_mikola=> start transaction ;
START TRANSACTION
shulyak_mikola=> lock table teacher in row share mode;
LOCK TABLE
shulyak_mikola=> commit;
COMMIT
shulyak_mikola=> █

```

SIX та IS сумісні.

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

T1

```
shulyak_mikola=> start transaction;
START TRANSACTION
shulyak_mikola=> set transaction isolation level read committed;
SET
shulyak_mikola=> select * from teacher where p_id = 1;
p_id |          name          |          post
-----+-----+-----
    1 | Andryuxa               | docent
(1 row)

shulyak_mikola=> update teacher set name = 'Nickolay' where p_id = 1;
UPDATE 1
shulyak_mikola=> select * from teacher where p_id = 1;
p_id |          name          |          post
-----+-----+-----
    1 | Nickolay               | docent
(1 row)

shulyak_mikola=> █
```

T2

```
shulyak_mikola=> start transaction;
START TRANSACTION
shulyak_mikola=> set transaction isolation level read committed;
SET
shulyak_mikola=> select * from teacher where p_id = 1;
p_id |          name          |          post
-----+-----+-----
    1 | Andryuxa               | docent
(1 row)

shulyak_mikola=> update teacher set post = 'bakalavr' where p_id = 1;
UPDATE 1
shulyak_mikola=> select * from teacher where p_id = 1;
p_id |          name          |          post
-----+-----+-----
    1 | Nickolay               | bakalavr
(1 row)

shulyak_mikola=> █
```

При виконанні операції update у T2 вона переходить в режим очікування і після завершення T1 успішно змінює дані.

Повторіть роботу транзакцій при умові їх роботи на рівні ізоляції

REPEATABLE READ:

T1

```
shulyak_mikola=> start transaction ;
START TRANSACTION
shulyak_mikola=> set transaction isolation level repeatable read;
SET
shulyak_mikola=> select * from teacher where p_id = 1;
p_id |          name          |          post
-----+-----+-----
    1 | Nickolay               | bakalavr
(1 row)

shulyak_mikola=> update teacher set name = 'Sasha' where p_id = 1;
UPDATE 1
shulyak_mikola=> select * from teacher where p_id = 1;
p_id |          name          |          post
-----+-----+-----
    1 | Sasha                  | bakalavr
(1 row)

shulyak_mikola=> commit;
COMMIT
shulyak mikola=> █
```

T2

```
shulyak_mikola=> start transaction;
START TRANSACTION
shulyak_mikola=> set transaction isolation level repeatable read;
SET
shulyak_mikola=> select * from teacher where p_id = 1;
p_id |          name          |          post
-----+-----+-----
    1 | Sasha                  | bakalavr
(1 row)

shulyak_mikola=> update teacher set name = 'Igor' where p_id = 1;
ERROR:  could not serialize access due to concurrent update
shulyak_mikola=> █
```

При виконанні операції update у T2 вона переходить в режим очікування і після завершення T1 повідомляє про помилку та завершує транзакцію без зміни даних.

Повторіть роботу транзакцій при умові їх роботи на рівні ізоляції SERIALIZABLE:

T1

```
START TRANSACTION
shulyak_mikola=> set transaction isolation level serializable;
SET
shulyak_mikola=> select * from teacher where p_id = 1;
p_id |          name          |          post
-----+-----+-----
    1 | Sasha                  | magistr
(1 row)

shulyak_mikola=> update teacher set name = 'Ivan' where p_id = 1;
UPDATE 1
shulyak_mikola=> select * from teacher where p_id = 1;
p_id |          name          |          post
-----+-----+-----
    1 | Ivan                   | magistr
(1 row)

shulyak_mikola=> commit;
COMMIT
shulyak_mikola=> █
```

T2

```
shulyak_mikola=> start transaction;
START TRANSACTION
shulyak_mikola=> set transaction isolation level serializable;
SET
shulyak_mikola=> select * from teacher where p_id = 1;
p_id |          name          |          post
-----+-----+-----
    1 | Sasha                  | magistr
(1 row)

shulyak_mikola=> update teacher set post = 'docent' where p_id = 1;
ERROR:  canceling statement due to user request
```

При виконанні операції update у T2 вона переходить в режим очікування і після завершення T1 повідомляє про помилку та завершує транзакцію без зміни даних.

4. Керування квазіпаралельним виконанням транзакцій при наявності тупикових ситуацій.

```
shulyak_mikola=> start transaction ;
START TRANSACTION
shulyak_mikola=> update teacher set name = 'Ruslan
shulyak_mikola'> ^C
shulyak_mikola=> update teacher set name = 'Ruslan' where p_id =1;
UPDATE 1
shulyak_mikola=> select * from teacher where p_id = 1;
 p_id |          name          |          post
-----+-----+-----
    1 | Ruslan                 | magistr
(1 row)

shulyak_mikola=> update teacher set name = 'Artem' where p_id = 2;
UPDATE 1
shulyak_mikola=> █

shulyak_mikola@vpsj3leQ:~
ROLLBACK
shulyak_mikola=> start transaction;
START TRANSACTION
shulyak_mikola=> set transaction isolation level serializable;
SET
shulyak_mikola=> select * from teacher where p_id = 1;
 p_id |          name          |          post
-----+-----+-----
    1 | Sasha                 | magistr
(1 row)

shulyak_mikola=> update teacher set post = 'docent' where p_id = 1;
^CCancel request sent
ERROR:  canceling statement due to user request
CONTEXT:  while updating tuple (0,12) in relation "teacher"
shulyak_mikola=> commit;
ROLLBACK
shulyak_mikola=> start transaction;
START TRANSACTION
shulyak_mikola=> update teacher set name = 'Shamil' where p_id = 2;
ERROR:  deadlock detected
DETAIL:  Process 27106 waits for ShareLock on transaction 3214; blocked by process 24022.
Process 24022 waits for ShareLock on transaction 3215; blocked by process 27106.
HINT:  See server log for query details.
CONTEXT:  while updating tuple (0,19) in relation "worker"
```

При виконанні операції update у T2 вона отримала повідомлення про помилку, так як очікувала завершення T1, а T1 очікувала завершення T2 - це призвело до тупіка, тому необхідно було примусово завершити транзакцію, що призвело до нього, тобто T2.

Висновок:

Найважчим у даній роботі виявилась робота із транзакціями.