

МІНІСТЕРСТВО ОСВІТИ Й НАУКИ УКРАЇНИ  
ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ ПОЛІТЕХНІЧНИЙ УНІВЕРСИТЕТ  
Інститут комп'ютерних систем  
Кафедра інформаційних систем

Лабораторна робота № 9  
За дисципліною: "Операційні системи"  
Тема: ««Керування процесами-транзакціями в базах даних.  
Частина 1»»

Виконав:  
Студент групи АІ-205  
Шостак Р.С.  
Перевірили:  
Блажко О.А.  
Дрозд М.О.

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.

## 2 Завдання

### Завдання 1 Проектування транзакцій

Нехай задані три транзакції, приклади яких представлено в таблиці 15.

8	1	$T1 = R[D] \ W[D] \ C1$ $T2 = R[A] \ R[B] \ W[A] \ C2$ $T3 = W[D] \ R[B] \ W[B] \ C3$
---	---	---

У відповідності з вашим варіантом виконайте наступні теоретичні завдання.

1.1 Створіть історії квазіпаралельного виконання транзакцій для протоколу 1-го ступеня блокування з описом таблиці блокування транзакцій.

$H_{T1T2T3} = R1[D]R2[A]X3[D]W3[D]X1[D] - WAIT \ R2[B]R3[B]X2[A]W2[A]X2[B]W2[B]C2C3X1[D] \ W1[D]C1$

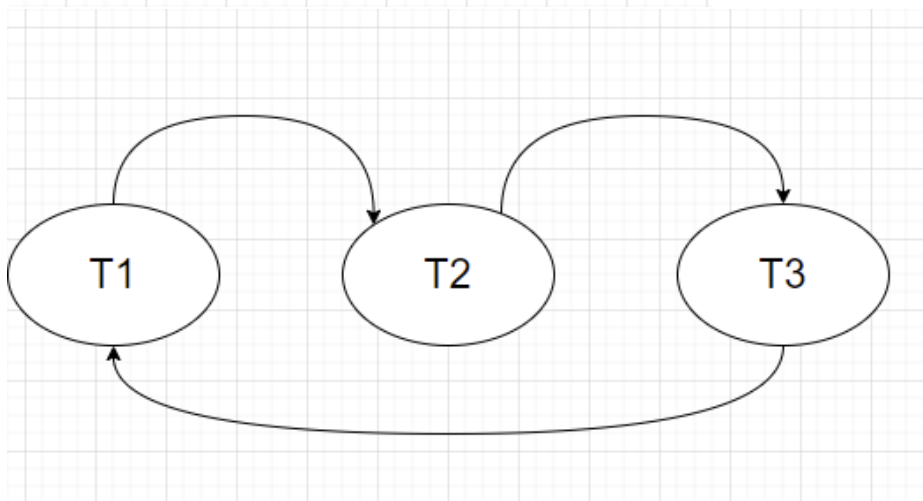
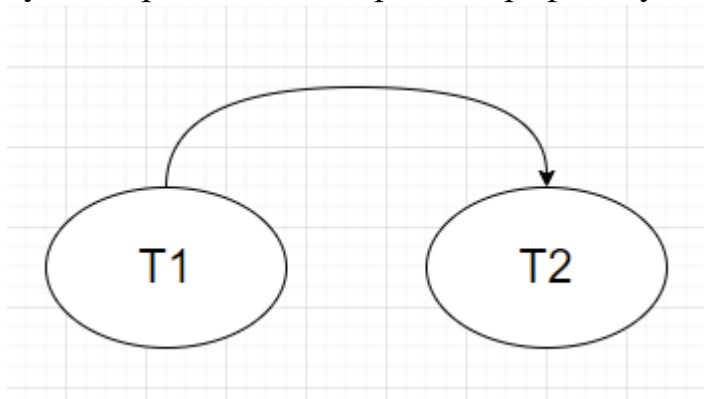
N	T1	T2	T3	Result
1	R1[D]			
2		R2[A]		
3			X3[D]	
4			W3[D]	
5	X1[D]			WAIT
6		R2[B]		
7			R3[B]	
8		X2[A]		
9		W2[A]		
10			X2[B]	
11			W2[B]	
12		C2		
13			C3	
14	X1[D]			
15	W1[D]			
16	C1			

1.2 Повторіть попереднє завдання з використанням протоколу 2-го ступеня блокування.

$H_{T1T2T3} = S1[D]R1[D] \ S2[A]R2[A]X3[D]W3[D]X1[D] - WAIT \ S2[B]R2[B]S3[B]R3[B]X2[A]W2[A]X2[B]W2[B]C2C3X1[D] \ W1[D]C1$

N	T1	T2	T3	Result
1	S1[D]			
2	R1[D]			
3		S2[A]		
4		R2[A]		
5			X3[D]	
6			W3[D]	
7	X1[D]			WAIT
8		S2[B]		
9		R2[B]		
10			S3[B]	
11			R3[B]	
12		X2[A]		
13		W2[A]		
14			X2[B]	
15			W2[B]	
16		C2		
17			C3	
18	X1[D]			
19	W1[D]			
20	C1			

1.3 Для створених історій у 1-му та 2-му пунктів завдань визначте наявність тупика транзакції, створивши граф очікування транзакцій.



У жодному варіанті тупіку немає

## Завдання 2 Налаштування бази даних

Нехай існує база даних, назва якої співпадає з іменем вашого користувача в ОС Linux.

Приклад команди створення реляційної таблиці в базі даних наведено в таблиці 16.

8	1	Create table university ( u_id integer, name char(20), year integer);	Insert into university values (1, 'ONPU', 1918);
---	---	---	---

Також в таблиці 1 наведено приклад команди внесення одного рядка в реляційну таблицю бази даних.

### 1.1 Встановіть з'єднання з вашою базою даних.

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

```
shostak_roman=> █
```

### 1.2 У відповідності із варіантом з таблиці 1 створіть реляційну таблицю в базі даних.

```
shostak_roman=> start transaction;
START TRANSACTION
shostak_roman=> create table university ( u_id integer, name char(20),year integer);
CREATE TABLE
shostak_roman=> █
```

### 1.3 У відповідності із варіантом з таблиці 1 додайте рядок в реляційну таблицю, створену у попередньому пункті завдання.

```
shostak_roman=> insert into university values (1,'ONPU',1918);
INSERT 0 1
```

### 1.4 Створіть ще одну операцію внесення рядка в таблицю, який буде відрізнятися значеннями всіх змінних від прикладу з варіанту.

```
shostak_roman=> insert into university values (2,'ONEU',1930);
INSERT 0 1
shostak_roman=> select* from university;
 u_id |          name          | year
-----+-----+-----
    1 | ONPU                   | 1918
    2 | ONEU                   | 1930
(2 rows)
```

## Завдання 3. Керування квазіпаралельним виконанням транзакцій з використанням команд блокування

### 1 Створіть дві транзакції, кожна з яких повинна включати такі операції:

- операція читання першого рядку таблиці;

```
shostak_roman=> select * from university where u_id=1;
u_id |          name          | year
-----+-----+-----
    1 | ONPU                   | 1918
(1 row)
```

- операція зміни однієї із змінних таблиці в першому рядку;
- повторна операція читання першого рядку таблиці;

```
shostak_roman=> update university set year = 1900 where u_id=1;
UPDATE 1
shostak_roman=> select * from university where u_id=1;
u_id |          name          | year
-----+-----+-----
    1 | ONPU                   | 1900
(1 row)
```

- операція фіксації всіх змін.

```
shostak_roman=> commit;
COMMIT
```

2 При створенні транзакцій включіть відповідні операції блокування для протоколу 1-го ступеня блокування.

3 У двох терміналах виконайте операції транзакцій при їх квазіпаралельному режимі роботи за умови, що одна з транзакцій стартує першою.

```
shostak_roman=> start transaction ;
START TRANSACTION
shostak_roman=> select * from university where u_id=1;
u_id |          name          | year
-----+-----+-----
    1 | ONPU                   | 1900
(1 row)

shostak_roman=> lock table university in exclusive mode;
LOCK TABLE
shostak_roman=> update university set year = 1905 where u_id=1;
UPDATE 1
shostak_roman=> select * from university where u_id=1;
u_id |          name          | year
-----+-----+-----
    1 | ONPU                   | 1905
(1 row)

shostak_roman=> commit;
COMMIT
shostak_roman=>
```

```
shostak_roman=> start transaction;
START TRANSACTION
shostak_roman=> select * from university where u_id = 1;
u_id |          name          | year
-----+-----+-----
    1 | ONPU                   | 1900
(1 row)

shostak_roman=> lock table university in exclusive mode ;
LOCK TABLE
shostak_roman=> select * from university where u_id = 1;
u_id |          name          | year
-----+-----+-----
    1 | ONPU                   | 1905
(1 row)

shostak_roman=> update university set year = 1920 where u_id=1;
UPDATE 1
shostak_roman=> select * from university where u_id = 1;
u_id |          name          | year
-----+-----+-----
    1 | ONPU                   | 1920
(1 row)

shostak_roman=> commit;
COMMIT
shostak_roman=>
```

4 Повторіть роботу транзакцій, але в першій транзакції замість операції фіксації виконайте операцію відміни.

```

shostak_roman=> start transaction ;
START TRANSACTION
shostak_roman=> select * from university where u_id=1;
u_id | name | year
-----+-----+-----
1 | ONPU | 1920
(1 row)

shostak_roman=> lock table university in exclusive mode;
LOCK TABLE
shostak_roman=> update university set year = 1927 where u_id=1;
UPDATE 1
shostak_roman=> select * from university where u_id=1;
u_id | name | year
-----+-----+-----
1 | ONPU | 1927
(1 row)

shostak_roman=> rollback;
ROLLBACK
shostak_roman=> 

```

```

shostak_roman=> start transaction;
START TRANSACTION
shostak_roman=> select * from university where u_id = 1;
u_id | name | year
-----+-----+-----
1 | ONPU | 1920
(1 row)

shostak_roman=> lock table university in exclusive mode ;
LOCK TABLE
shostak_roman=> select * from university where u_id = 1;
u_id | name | year
-----+-----+-----
1 | ONPU | 1920
(1 row)

shostak_roman=> update university set year = 1940 where u_id=1;
UPDATE 1
shostak_roman=> select * from university where u_id = 1;
u_id | name | year
-----+-----+-----
1 | ONPU | 1940
(1 row)

shostak_roman=> commit;
COMMIT
shostak_roman=> 

```

5 Повторіть пункти 3 та 4 але з використанням протоколу 2-го ступеня блокування.

```

shostak_roman=> start TRANSACTION ;
START TRANSACTION
shostak_roman=> lo
load lock
shostak_roman=> lock TABLE university in share mode;
LOCK TABLE
shostak_roman=> select * from university where u_id=1;
u_id | name | year
-----+-----+-----
1 | ONPU | 1940
(1 row)

shostak_roman=> lock table university in exclusive mode;
LOCK TABLE
shostak_roman=> 

```

```

START TRANSACTION
shostak_roman=> lock TABLE university in share mode;
LOCK TABLE
shostak_roman=> select * from university where u_id=1;
u_id | name | year
-----+-----+-----
1 | ONPU | 1940
(1 row)

shostak_roman=> lock TABLE university in exclusive mode ;
ERROR: deadlock detected
DETAIL: Process 1803 waits for ExclusiveLock on relation 16825 of database 16586; blocked by process 1798.
Process 1798 waits for ExclusiveLock on relation 16825 of database 16586; blocked by process 1803.
HINT: See server log for query details.
shostak_roman=> 

```

Получили ошибку DEADLOCK DETECTED