

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ  
ПОЛІТЕХНІЧНИЙ УНІВЕРСИТЕТ

Інститут комп'ютерних систем

Кафедра інформаційних систем

Лабораторна робота № 9

з дисципліни

«Операційні системи»

Тема: «Керування процесами-транзакціями в базах даних. Частина 1»

Виконав:

Студент групи AI-202

Сідельніков М. В.

Перевірив:

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.

## **Завдання:**

### **Завдання 1. Проектування транзакцій**

Нехай задані три транзакції, приклади яких представлено в таблиці 15. У відповідності з вашим варіантом виконайте наступні теоретичні завдання.

1. Створіть історії квазіпаралельного виконання транзакцій для протоколу 1-го ступеня блокування з описом таблиці блокування транзакцій.
2. Повторіть попереднє завдання з використанням протоколу 2-го ступеня блокування.
3. Для створених історій у 1-му та 2-му пунктів завдань визначте наявність тупика транзакції, створивши граф очікування транзакцій.

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

1. Встановіть з'єднання з вашою базою даних.
2. У відповідності із варіантом з таблиці 1 створіть реляційну таблицю в базі даних.
3. У відповідності із варіантом з таблиці 1 додайте рядок в реляційну таблицю, створену у попередньому пункті завдання.
4. Створіть ще одну операцію внесення рядка в таблицю, який буде відрізнятися значеннями всіх змінних від прикладу з варіанту.

### **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;
  - операція зміни однієї із змінних таблиці в першому рядку;
  - повторна операція читання першого рядку таблиці;
  - операція фіксації всіх змін.
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.
4. Повторіть роботу транзакцій, але в першій транзакції замість операції фіксації виконайте операцію відміни.
5. Повторіть пункти 3 та 4 але з використанням протоколу 2-го ступеня блокування.

#### **Хід роботи**

### **Завдання 1. Проектування транзакцій**

Нехай задані три транзакції

$T1 = R[D] R[B] W[D] C1$

$T2 = W[A] W[D] C2$

$T3 = W[A] R[B] W[D] C3.$

У відповідності з вашим варіантом виконайте наступні теоретичні завдання.

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

2. Повторіть попереднє завдання з використанням протоколу 2-го ступеня блокування.
3. Для створених історій у 1-му та 2-му пунктів завдань визначте наявність тупика транзакції, створивши граф очікування транзакцій.

1.  $H_{T1,T2,T3} = R1[D] X2[A] W2[A] X3[A] - \text{Wait } R1[B] X2[D] W2[D] R3[B] X1[D] - \text{Wait } U2 C2 X3[A] W3[A] X1[D] W1[D] X3[D] - \text{Wait } U1 C1 X3[D] W3[D] U3 C3$

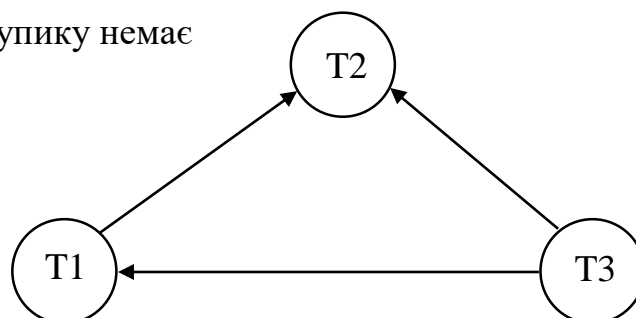
Назва змінної	Перелік встановлених блокувань	Перелік запитів на блокування
A	X2X3	X3
B	-	-
D	X2X1X3	X1X3

2.  $H_{T1,T2,T3} = S1[D] R1[D] X2[A] W2[A] X3[A] - \text{Wait } S1[B] R1[B] X2[D] - \text{Wait } X1[D] - \text{Wait: Тупиковий стан.}$

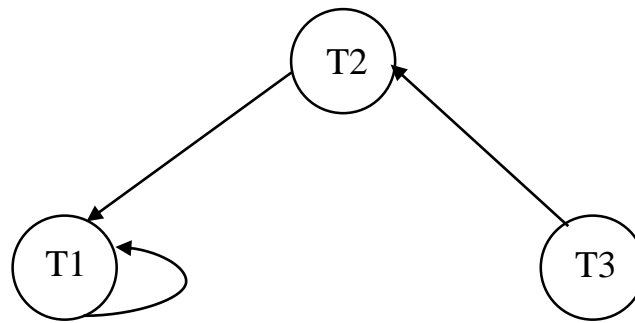
Назва змінної	Перелік встановлених блокувань	Перелік запитів на блокування
A	X2	X3
B	S1	-
D	S1	X2X1

- 3.

Тупику немає



Тупик присутній



## 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

*Create table student ( s\_id integer, name char(20), kurs integer);*

Приклад команди внесення одного рядка в реляційну таблицю бази даних

*Insert into student values (1, 'Petrov', 4);*

1. Встановіть з'єднання з вашою базою даних.
2. Створіть реляційну таблицю в базі даних.
3. Додайте рядок в реляційну таблицю, створену у попередньому пункті завдання.
4. Створіть ще одну операцію внесення рядка в таблицю, який буде відрізнятися значеннями всіх змінних від прикладу з варіанту.

```
sidelnikov_mikita@vpsj3IeQ:~  
[sidelnikov_mikita@vpsj3IeQ ~]$ psql sidelnikov_mikita sidelnikov_mikita  
psql (9.5.25)  
Type "help" for help.  
  
sidelnikov_mikita=> Create table student (s_id integer, name char(20), kurs integer);  
CREATE TABLE  
sidelnikov_mikita=> Insert into student values (1, 'Petrov', 4);  
INSERT 0 1  
sidelnikov_mikita=> Insert into student values (2, 'Bashirov', 3);  
INSERT 0 1  
sidelnikov_mikita=> Select * from student  
sidelnikov_mikita-> ;  
 s_id |      name      | kurs  
-----+-----+-----  
    1 | Petrov         |    4  
    2 | Bashirov       |    3  
(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;
  - операція зміни однієї із змінних таблиці в першому рядку;
  - повторна операція читання першого рядку таблиці;
  - операція фіксації всіх змін.
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 транзакція стартує першою.

```
sidelnikov_mikita@vpsj3IeQ:~  
[sidelnikov_mikita@vpsj3IeQ ~]$ psql sidelnikov_mikita sidelnikov_mikita  
psql (9.5.25)  
Type "help" for help.  
  
sidelnikov_mikita=> Start transaction;  
START TRANSACTION  
sidelnikov_mikita=> Select * from student where s_id = 1;  
 s_id |      name      | kurs  
-----+-----  
    1 | Petrov         |    4  
(1 row)  
  
sidelnikov_mikita=> Lock table student in exclusive mode;  
LOCK TABLE  
sidelnikov_mikita=> Update student set kurs = 5 where s_id = 1;  
UPDATE 1  
sidelnikov_mikita=> Select * from student where s_id = 1;  
 s_id |      name      | kurs  
-----+-----  
    1 | Petrov         |    5  
(1 row)  
  
sidelnikov_mikita=> Commit;  
COMMIT  
sidelnikov_mikita=> █
```

```
sidelnikov_mikita@vpsj3IeQ:~  
[sidelnikov_mikita@vpsj3IeQ ~]$ psql sidelnikov_mikita sidelnikov_mikita  
psql (9.5.25)  
Type "help" for help.  
  
sidelnikov_mikita=> Start transaction;  
START TRANSACTION  
sidelnikov_mikita=> Select * from student where s_id = 1;  
 s_id |      name      | kurs  
-----+-----  
    1 | Petrov         |    5  
(1 row)  
  
sidelnikov_mikita=> Update student set name = 'Petrovich' where s_id = 1;  
UPDATE 1  
sidelnikov_mikita=> Select * from student where s_id = 1;  
 s_id |      name      | kurs  
-----+-----  
    1 | Petrovich      |    5  
(1 row)  
  
sidelnikov_mikita=> Commit  
sidelnikov_mikita-> ;  
COMMIT  
sidelnikov_mikita=> █
```

При виконанні, LOCK у другому терміналі перейшов до стану очікування через блокування першим терміналом. Після COMMIT у першому терміналі, другий продовжив роботу.

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

```
sidelnikov_mikita@vpsj3IeQ:~  
[sidelnikov_mikita@vpsj3IeQ ~]$ psql sidelnikov_mikita sidelnikov_mikita  
psql (9.5.25)  
Type "help" for help.  
  
sidelnikov_mikita=> Start transaction;  
START TRANSACTION  
sidelnikov_mikita=> Select * from student where s_id = 1;  
 s_id |      name      | kurs  
-----+-----+-----  
    1 | Petrovich      |    5  
(1 row)  
  
sidelnikov_mikita=> Lock table student in exclusive mode;  
LOCK TABLE  
sidelnikov_mikita=> Update student set name = 'Rabinovich' where s_id = 1;  
UPDATE 1  
sidelnikov_mikita=> Select * from student where s_id = 1;  
 s_id |      name      | kurs  
-----+-----+-----  
    1 | Rabinovich     |    5  
(1 row)  
  
sidelnikov_mikita=> Abort;  
ROLLBACK  
sidelnikov_mikita=> █
```



```
sidelnikov_mikita@vpsj3IeQ:~  
[sidelnikov_mikita@vpsj3IeQ ~]$ psql sidelnikov_mikita sidelnikov_mikita  
psql (9.5.25)  
Type "help" for help.  
  
sidelnikov_mikita=> Start transaction;  
START TRANSACTION  
sidelnikov_mikita=> Select * from student where s_id = 1;  
 s_id |      name      | kurs  
-----+-----+-----  
    1 | Petrovich      |    5  
(1 row)  
  
sidelnikov_mikita=> Lock table student in exclusive mode;  
LOCK TABLE  
sidelnikov_mikita=> Update student set kurs = 1 where s_id = 1;  
UPDATE 1  
sidelnikov_mikita=> Select * from student where s_id = 1;  
 s_id |      name      | kurs  
-----+-----+-----  
    1 | Petrovich      |    1  
(1 row)  
  
sidelnikov_mikita=> Commit;  
COMMIT  
sidelnikov_mikita=> █
```

При виконанні, LOCK у другому терміналі перейшов до стану очікування через блокування першим терміналом. Після ABORT у першому терміналі, другий продовжив роботу.

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

```
sidelnikov_mikita@vpsj3IeQ:~  
[sidelnikov_mikita@vpsj3IeQ ~]$ psql sidelnikov_mikita sidelnikov_mikita  
psql (9.5.25)  
Type "help" for help.  
  
sidelnikov_mikita=> Start transaction;  
START TRANSACTION  
sidelnikov_mikita=> Lock table student in share mode;  
LOCK TABLE  
sidelnikov_mikita=> Select * from student where s_id = 1;  
 s_id |      name      | kurs  
-----+-----+-----  
    1 | Rabinovich     |    3  
(1 row)  
  
sidelnikov_mikita=> Lock table student in exclusive mode;  
LOCK TABLE  
sidelnikov_mikita=> █
```

```
sidelnikov_mikita@vpsj3IeQ:~  
[sidelnikov_mikita@vpsj3IeQ ~]$ psql sidelnikov_mikita sidelnikov_mikita  
psql (9.5.25)  
Type "help" for help.  
  
sidelnikov_mikita=> Start transaction;  
START TRANSACTION  
sidelnikov_mikita=> Lock table student in share mode;  
LOCK TABLE  
sidelnikov_mikita=> Select * from student where s_id = 1;  
 s_id |      name      | kurs  
-----+-----+-----  
    1 | Rabinovich     |    3  
(1 row)  
  
sidelnikov_mikita=> Lock table student in exclusive mode;  
ERROR:  deadlock detected  
DETAIL:  Process 20271 waits for ExclusiveLock on relation 16801 of database 16446; blocked by process 20217.  
Process 20217 waits for ExclusiveLock on relation 16801 of database 16446; blocked by process 20271.  
HINT:  See server log for query details.  
sidelnikov_mikita=> █
```

Ми отримали помилку через тупик, що з'явився після X1[teacher] X2[teacher].

**Висновки:** під час виконання лабораторної роботи було досліджено поведінку процесів-транзакцій в базах даних та засоби керування ними через механізм блокування з використанням сучасних систем керування базами даних.