

Міністерство освіти і науки України  
Державний університет «Одеська політехніка»  
Інститут комп'ютерних систем  
Кафедра інформаційних систем

Лабораторна робота №9  
з дисципліни «Операційні Системи»

Тема: «Керування процесами-транзакціями в базах даних»

Виконав:  
ст. гр. АІ-204  
Бабич А.Ю

Перевірив:  
Блажко О. А.

**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:**

#### **Завдання 1.**

Нехай задані три транзакції, приклади яких представлено в таблиці 13.

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.

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

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

№ команди	№ учасника	Приклади транзакцій
1	1	T1= R[A] R[B] W[A] C1 T2= W[D] R[B] W[B] C2 T3= R[D] W[D] C3

#### **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 1.

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

1.1 Встановіть з'єднання з вашою базою даних.

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

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

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

№ команди	№ учасника	Операція створення реляційної таблиці в базі даних	Операції внесення даних в реляційну таблицю в базі даних
1	1	Create table person ( p_id integer, name char(20), bd date);	Insert into person values (1, 'Ivanov', '01/04/2000');

**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.

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

Задано:

T1= R[D] R[B] W[D] C2

T2= W[A] R[B] W[B] C1

T3= R[A] W[A] C3

Створюють історію квазіпаралельного виконання транзакцій для протоколу 1-го ступеня, з описом таблиці блокування транзакцій:

$H_{T_1, T_2, T_3} = R_1[A], X_2[D], W_2[D], R_3[D], R_1[B], R_2[B], X_3[D] - \text{Wait}, X_1[A], W_1[A], X_2[B], W_2[B], U_1, C_1, U_2, C_2, X_3[D], W_3[D], U_3, C_3$

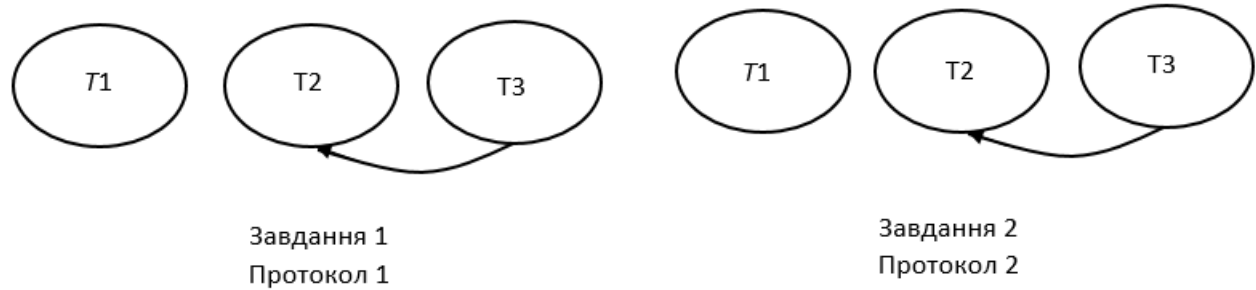
Назва змінної	Перелік встановлених блокувань	Перелік запитів
A		X1 U1
B		X2 U2
D	X2	X2 U2 X3 U3

Повторюють попереднє завдання з використанням протоколу 2-го ступеня блокування:

$H_{T_1, T_2, T_3} = S_1[A], R_1[A], X_2[D], W_2[D], S_3[D] - \text{wait}, S_1[B], R_1[B], S_2[B], R_2[B], X_1[A], W_1[A], X_2[B], W_2[B], U_1, C_1, U_2, C_2, R_3[D], X_3[D], W_3[D], U_3, C_3$

Назва змінної	Перелік встановлених блокувань	Перелік запитів
A		S1 X1 U1
B		S1 S2 X2 U1 U2
D	X2	X2 S3 U2 X3 U3

Для створених історій у 1-му та 2-му пунктів завдань визнаємо наявність тупика тразакції, створивши граф очікування транзакцій.



## Завдання 2

### 1.1 Встановлюють з'єднання з базою даних.

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

babich_artem=> █
```

№ коман ди	№ учасни ка	Операція створення реляційної таблиці в базі даних	Операції внесення даних в реляційну таблицю в базі даних
1	1	Create table person ( p_id integer, name char(20), bd date);	Insert into person values (1, 'Ivanov', '01/04/2000');

1.2 У відповідності із варіантом з таблиці створюють реляційну таблицю в базі даних.

```
babich_artem=> CREATE TABLE person
babich_artem-> (p_id integer, name char(20), bd date);
CREATE TABLE
babich_artem=> █
```

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

```
babich_artem=> INSERT INTO person VALUES (1, 'Ivanov', '01/04/2000');
INSERT 0 1
babich_artem=> █
```

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

```
babich_artem=> INSERT INTO person VALUES (2, 'Tarasov', '07/04/2003');
INSERT 0 1
babich_artem=> █
```

1.5 Перевіряють вміст таблиці.

```
babich_artem=> SELECT * FROM person;
 p_id |      name      |      bd
-----+-----+-----
    1 | Ivanov         | 2000-01-04
    2 | Tarasov        | 2003-07-04
(2 rows)

babich_artem=> █
```

### Завдання 3

1. Створюють 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.

Транзакція 1	Транзакція 2
START TRANSACTION	START TRANSACTION
SELECT * FROM person WHERE p_id = 1;	SELECT * FROM person WHERE p_id = 1;
UPDATE person SET name = 'Babich' WHERE p_id = 1;	UPDATE person SET bd = '07/02/2003' WHERE p_id = 1;
SELECT * FROM person WHERE p_id = 1;	SELECT * FROM person WHERE p_id = 1;
COMMIT;	COMMIT;

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

Транзакція 1	Транзакція 2
START TRANSACTION	START TRANSACTION
SELECT * FROM person WHERE p_id = 1;	SELECT * FROM person WHERE p_id = 1;
LOCK TABLE person IN EXCLUSIVE MODE;	LOCK TABLE person IN EXCLUSIVE MODE;
UPDATE person SET name = 'Babich' WHERE p_id = 1;	UPDATE person SET bd = '07/02/2003' WHERE p_id = 1;
SELECT * FROM person WHERE p_id = 1;	SELECT * FROM person WHERE p_id = 1;
COMMIT;	COMMIT;

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

babich\_artem@vpsj3leQ:~
^
babich\_artem=> START TRANSACTION;
START TRANSACTION
babich\_artem=> SELECT \*
babich\_artem-> FROM person
babich\_artem-> WHERE p\_id = 1;
p\_id | name | bd
-----+-----+-----
1 | Ivanov | 2000-01-04
(1 row)

babich\_artem=> LOCK TABLE person in EXCLUSIVE MODE
babich\_artem-> ;
LOCK TABLE
babich\_artem=> UPDATE person
babich\_artem-> SET name = 'Babich'
babich\_artem-> WHERE p\_id = 1;
UPDATE 1
babich\_artem=> SELECT \*
babich\_artem-> FROM person
babich\_artem-> WHERE p\_id = 1;
p\_id | name | bd
-----+-----+-----
1 | Babich | 2000-01-04
(1 row)

babich\_artem=> COMMIT;
COMMIT
babich\_artem=> █

babich\_artem@vpsj3leQ:~
babich\_artem=> START TRANSACTION;
START TRANSACTION
babich\_artem=> SELECT \*
babich\_artem-> FROM person
babich\_artem-> WHERE p\_id = 1;
p\_id | name | bd
-----+-----+-----
1 | Babich | 2000-01-04
(1 row)

babich\_artem=> LOCK TABLE person in EXCLUSIVE MODE;
LOCK TABLE
babich\_artem=>
babich\_artem=> UPDATE person
babich\_artem-> SET bd = '07/02/2003'
babich\_artem-> WHERE p\_id = 1;
UPDATE 1
babich\_artem=> SELECT \*
babich\_artem-> FROM person
babich\_artem-> WHERE p\_id = 1;
p\_id | name | bd
-----+-----+-----
1 | Babich | 2003-07-02
(1 row)

babich\_artem=> COMMIT;
COMMIT
babich\_artem=> █

4. Повторюють роботу транзакцій, але в першій транзакції замість операції фіксації виконують операцію відміни.



PuTTY (inactive)

babich\_artem=> START TRANSACTION;  
START TRANSACTION  
babich\_artem=> SELECT \*  
babich\_artem-> FROM person  
babich\_artem-> WHERE p\_id = 1;  

p_id	name	bd
1	Babich	2003-07-02

  
(1 row)  
  
babich\_artem=> LOCK TABLE person in EXCLUSIVE MODE;  
LOCK TABLE  
babich\_artem=> UPDATE person  
babich\_artem-> SET name = 'Ivanov'  
babich\_artem-> WHERE p\_id = 1;  
UPDATE 1  
babich\_artem=> SELECT \*  
babich\_artem-> FROM person  
babich\_artem-> WHERE p\_id = 1;  

p_id	name	bd
1	Ivanov	2003-07-02

  
(1 row)  
  
babich\_artem=> ROLLBACK;  
ROLLBACK  
babich\_artem=>

babich\_artem@vpsj3leQ:~

babich\_artem=> START TRANSACTION;  
START TRANSACTION  
babich\_artem=> SELECT \*  
babich\_artem-> FROM person  
babich\_artem-> WHERE p\_id = 1;  

p_id	name	bd
1	Babich	2003-07-02

  
(1 row)  
  
babich\_artem=> LOCK TABLE person in EXCLUSIVE MODE;  
LOCK TABLE  
babich\_artem=> UPDATE person  
babich\_artem-> SET bd = '01/04/2000'  
babich\_artem-> WHERE p\_id = 1;  
UPDATE 1  
babich\_artem=> SELECT \*  
babich\_artem-> FROM person  
babich\_artem-> WHERE p\_id = 1;  

p_id	name	bd
1	Babich	2000-01-04

  
(1 row)  
  
babich\_artem=> ROLLBACK;  
ROLLBACK  
babich\_artem=>

6 Повторюють пункти 3 та 4 але з використанням протоколу 2-го ступеня блокування.

babich\_artem@vpsj3leQ:~

START TRANSACTION  
babich\_artem=> LOCK TABLE person in SHARE MODE;  
LOCK TABLE  
babich\_artem=> SELECT \*  
babich\_artem-> FROM person  
babich\_artem-> WHERE p\_id = 1;  

p_id	name	bd
1	Ivanov	2003-07-02

  
(1 row)  
  
babich\_artem=> LOCK TABLE person in EXCLUSIVE MODE;  
LOCK TABLE  
babich\_artem=> UPDATE person  
babich\_artem-> SET name = 'Babich'  
babich\_artem-> WHERE p\_id = 1;  
UPDATE 1  
babich\_artem=> SELECT \*  
babich\_artem-> FROM person  
babich\_artem-> WHERE p\_id = 1;  

p_id	name	bd
1	Babich	2003-07-02

  
(1 row)  
  
babich\_artem=> COMMIT;  
COMMIT  
babich\_artem=>

babich\_artem@vpsj3leQ:~

START TRANSACTION  
babich\_artem=> LOCK TABLE person in SHARE MODE;  
LOCK TABLE  
babich\_artem=> SELECT \*  
babich\_artem-> FROM person  
babich\_artem-> WHERE p\_id = 1;  

p_id	name	bd
1	Ivanov	2003-07-02

  
(1 row)  
  
babich\_artem=> LOCK TABLE person in EXCLUSIVE MODE;  
  
^Ccancel request sent  
ERROR: canceling statement due to user request  
babich\_artem=> LOCK TABLE person in EXCLUSIVE MODE;  
ERROR: current transaction is aborted, commands ignored until end of transaction block  
babich\_artem=> UPDATE person  
babich\_artem-> SET bd = '07/02/2003'  
babich\_artem-> WHERE p\_id = 1;  
ERROR: current transaction is aborted, commands ignored until end of transaction block  
babich\_artem=>

Транзакція 2-го терміналу була заблокована, а 1-му терміналу було надано можливість виконати блокування таблиці. Подальші операції у 2-му терміналі є неможливими.

**Висновки:** в ході 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.