

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  
Лобко Данііл Віталійович

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.

### **Перелік завдань:**

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

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

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.

Таблиця 15 – Варіанти завдань з прикладами 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
1	2	T1= W[A] R[B] W[B] C1 T2= R[D] R[B] W[D] C2 T3= R[A] W[A] C3
2	1	T1= R[A] W[A] C1 T2= R[D] R[B] W[D] C2 T3= W[A] R[B] W[A] C3
2	2	T1= W[A] W[D] C1 T2= R[D] R[B] W[D] C2 T3= W[A] R[B] W[D] C3
3	1	T1= W[A] W[B] C1 T2= R[D] R[A] W[D] C2 T3= W[A] R[B] W[D] C3
3	2	T1= W[D] W[A] C1 T2= R[D] R[A] W[D] C2 T3= W[A] R[B] W[D] C3
4	1	T1= W[D] W[A] C1 T2= R[D] R[A] W[D] C2 T3= W[B] R[A] W[D] C3
4	2	T1= W[D] R[B] W[B] C1 T2= R[A] R[B] W[A] C2 T3= R[D] W[D] C3

Таблиця 15 – Продовження таблиці

№ команди	№ учасника	Приклади транзакцій
5	1	T1= R[D] R[B] W[D] C1 T2= W[A] R[B] W[B] C2 T3= R[A] W[A] C3
5	2	T1= R[D] R[B] W[D] C1 T2= R[A] W[A] C2 T3= W[A] R[B] W[A] C3
6	1	T1= R[D] R[B] W[D] C1 T2= W[A] W[D] C2 T3= W[A] R[B] W[D] C3
6	2	T1= R[D] R[A] W[D] C1 T2= W[A] W[B] C2 T3= W[A] R[B] W[D] C3
7	1	T1= R[D] R[A] W[D] C1 T2= W[D] W[A] C2 T3= W[A] R[B] W[D] C3
7	2	T1= R[D] R[A] W[D] C1 T2= W[D] W[A] C2 T3= W[B] R[A] W[D] C3
8	1	T1= R[D] W[D] C1 T2= R[A] R[B] W[A] C2 T3= W[D] R[B] W[B] C3
8	2	T1= R[A] W[A] C1 T2= W[A] R[B] W[B] C2 T3= R[D] R[B] W[D] C3
9	1	T1= W[A] R[B] W[A] C1 T2= R[A] W[A] C2 T3= R[D] R[B] W[D] C3
9	2	T1= W[A] R[B] W[D] C1 T2= W[A] W[D] C2 T3= R[D] R[B] W[D] C3
10	1	T1= W[A] R[B] W[D] C1 T2= W[A] W[B] C2 T3= R[D] R[A] W[D] C3
10	2	T1= W[A] R[B] W[D] C1 T2= W[D] W[A] C2 T3= R[D] R[A] 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 16.

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.

Таблиця 16 - Варіанти завдань з командами створення бази даних та внесення даних

№ команди	№ учасника	Операція створення реляційної таблиці в базі даних	Операції внесення даних в реляційну таблицю в базі даних
1	1	<i>Create table person ( p_id integer, name char(20), bd date);</i>	<i>Insert into person values (1, 'Ivanov', '01/04/2000');</i>
1	2	<i>Create table student ( s_id integer, name char(20), kurs integer);</i>	<i>Insert into student values (1, 'Petrov', 4);</i>
2	1	<i>Create table teacher ( t_id integer, name char(20), post char(20));</i>	<i>Insert into teacher values (1, 'Ivanov', 'docent');</i>
2	2	<i>Create table department ( d_id integer, name char(20), faculty char(20));</i>	<i>Insert into department values (1, 'SPO', 'IKS');</i>
3	1	<i>Create table airplane (a_id integer, model char(20), year integer);</i>	<i>Insert into airplane values (1, 'TU-107', 1960);</i>
3	2	<i>Create table university ( u_id integer, name char(20), year integer);</i>	<i>Insert into university values (1, 'ONPU', 1918);</i>
4	1	<i>Create table auto ( a_id integer, name char(20), year integer);</i>	<i>Insert into auto values (1, 'BMW 5', 2003);</i>
4	2	<i>Create table employer (e_id integer, name char(20), salary integer);</i>	<i>Insert into employer values (1, 'Ivanov', 200);</i>

Таблиця 16 - продовження

№ команди	№ учасника	Операція створення таблиці в базі даних	Операції внесення даних в базі даних
5	1	Create table worker ( p_id integer, name char(20), bd date);	Insert into worker values (1, 'Ivanov', '01/04/2000');
5	2	Create table person ( p_id integer, name char(20), bd date);	Insert into person values (1, 'Ivanov', '01/04/2000');
6	1	Create table student ( s_id integer, name char(20), kurs integer);	Insert into student values (1, 'Petrov', 4);
6	2	Create table teacher ( t_id integer, name char(20), post char(20));	Insert into teacher values (1, 'Ivanov', 'docent');
7	1	Create table department ( d_id integer, name char(20), faculty char(20));	Insert into department values (1, 'SPO', 'IKS');
7	2	Create table airplane (a_id integer, model char(20), year integer);	Insert into airplane values (1, 'TU- 107', 1960);
8	1	Create table university ( u_id integer, name char(20), year integer);	Insert into university values (1, 'ONPU', 1918);
8	2	Create table auto ( a_id integer, name char(20), year integer);	Insert into auto values (1, 'BMW 5', 2003);
9	1	Create table employer (e_id integer, name char(20), salary integer);	Insert into employer values (1, 'Ivanov', 200);
9	2	Create table worker ( p_id integer, name char(20), bd date);	Insert into worker values (1, 'Ivanov', '01/04/2000');
10	1	Create table person ( p_id integer, name char(20), bd date);	Insert into person values (1, 'Ivanov', '01/04/2000');
10	2	Create table student ( s_id integer, name char(20), kurs integer);	Insert into student values (1, 'Petrov', 4);

### **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, приклади яких представлено в таблиці 15.

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 транзакцій, створивши граф очікування транзакцій.



4	2	T1= W[D] R[B] W[B] C1 T2= R[A] R[B] W[A] C2 T3= R[D] W[D] C3
---	---	--

1)  $H_{T1T2T3}$  I ступеня блокування =

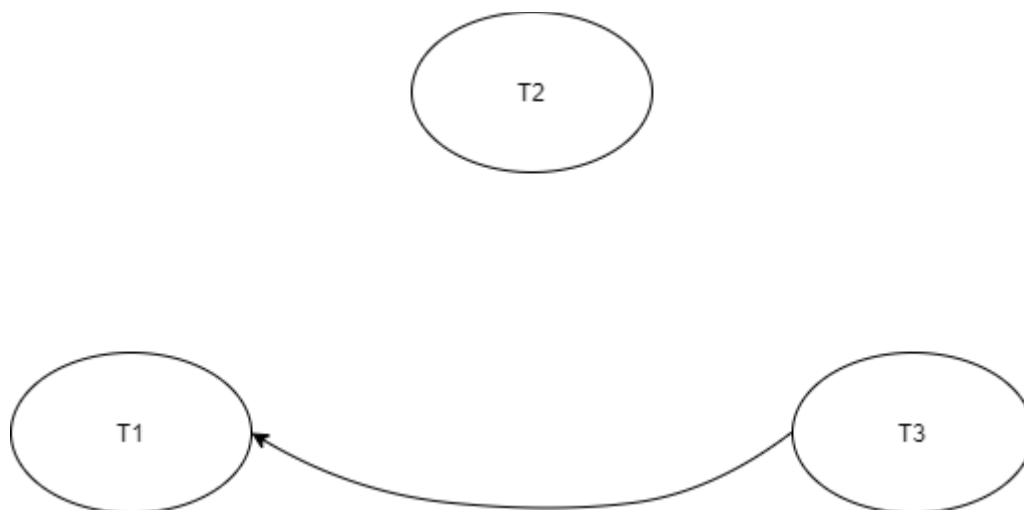
X1[D] W1[D] R[A] R[D]

R1[B] R2[B] X3[D] – wait A1 X3[D] W3[D]

X1[B] W1[B] X2[A] W2[A] C3

C1 C2

Назва змінної	Перелік встановлених блокувань	Перелік запитів на блокування
A	X2	-
B	X1	-
D	X1	X3



2)  $H_{T1T2T3}$  II ступеня блокування =

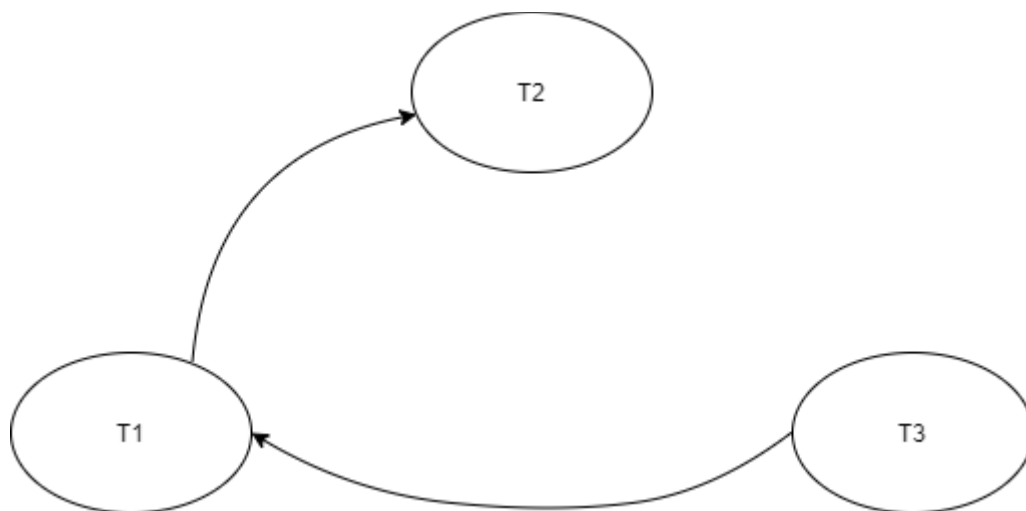
X1[D] W1[D] S2[A] R2[A] S3[D]-wait A1 U1 S3[D] R3[D]

S1[B] R1[B] S2[B] R2[B] X3[D] W3[D]

X1[B]-wait A2 U2 X1[B] W1[B] X2[A] W2[A] C3

C1 C2

Назва змінної	Перелік встановлених блокувань	Перелік запитів на блокування
A	X2S2	-
B	X1S1S2	X1
D	X1X3S3	S3



## 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.

1.1 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.

```
psql (9.5.25)
Type "help" for help.

lobko_daniil=> CREATE TABLE employer (e_id integer, name char(20), salary integer);
CREATE TABLE
lobko_daniil=> INSERT INTO employer VALUES (1,'Ivanov',200);
INSERT 0 1
lobko_daniil=> INSERT INTO employer VALUES (2,'NotIvanov',400);
INSERT 0 1
```

4	2	Create table employer (e_id integer, name char(20), salary integer);	Insert into employer values (1, 'Ivanov', 200);
---	---	--	--

## 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.

```
lobko_daniil@vpsj3leQ:~  
lobko_daniil=> SELECT * FROM employer WHERE e_id=1;  
e_id | name | salary  
-----  
1 | Ivanov | 200  
(1 row)  
  
lobko_daniil=> ROLLBACK;  
ROLLBACK  
lobko_daniil=> START TRANSACTION;  
START TRANSACTION  
lobko_daniil=> SELECT * FROM employer WHERE e_id=1;  
e_id | name | salary  
-----  
1 | Ivanov | 200  
(1 row)  
  
lobko_daniil=> UPDATE employer SET salary=2000 WHERE e_id=1;  
UPDATE 1  
lobko_daniil=> SELECT * FROM employer WHERE e_id=1;  
e_id | name | salary  
-----  
1 | Ivanov | 2000  
(1 row)  
  
lobko_daniil=> commit;  
COMMIT  
lobko_daniil=>
```

```
lobko_daniil@vpsj3leQ:~  
ROLLBACK  
lobko_daniil=> START TRANSACTION;  
START TRANSACTION  
lobko_daniil=> LOCK TABLE employer IN EXCLUSIVE MODE;  
LOCK TABLE  
lobko_daniil=> ROLLBACK;  
ROLLBACK  
lobko_daniil=> START TRANSACTION;  
START TRANSACTION  
lobko_daniil=> SELECT * FROM employer WHERE e_id=1;  
e_id | name | salary  
-----  
1 | Ivanov | 2000  
(1 row)  
  
lobko_daniil=> UPDATE employer SET salary=20000 WHERE e_id=1;  
UPDATE 1  
lobko_daniil=> SELECT * FROM employer WHERE e_id=1;  
e_id | name | salary  
-----  
1 | Ivanov | 20000  
(1 row)  
  
lobko_daniil=> commit;  
COMMIT  
lobko_daniil=>
```

2 При створенні транзакцій включіть відповідні операції блокування для протоколу

1-го ступеня блокування.

```
lobko_daniil@vpsj3leQ:~  
e_id | name | salary  
-----  
1 | Ivanov | 2000  
(1 row)  
  
lobko_daniil=> commit;  
COMMIT  
lobko_daniil=> START TRANSACTION;  
START TRANSACTION  
lobko_daniil=> SELECT * FROM employer WHERE e_id=1;  
e_id | name | salary  
-----  
1 | Ivanov | 20000  
(1 row)  
  
lobko_daniil=> LOCK TABLE employer IN EXCLUSIVE MODE;  
LOCK TABLE  
lobko_daniil=> UPDATE employer SET salary=2002 WHERE e_id=1;  
UPDATE 1  
lobko_daniil=> SELECT * FROM employer WHERE e_id=1;  
e_id | name | salary  
-----  
1 | Ivanov | 2002  
(1 row)  
  
lobko_daniil=> commit;  
COMMIT  
lobko_daniil=>
```

```
lobko_daniil@vpsj3leQ:~  
e_id | name | salary  
-----  
1 | Ivanov | 20000  
(1 row)  
  
lobko_daniil=> commit;  
COMMIT  
lobko_daniil=> START TRANSACTION;  
START TRANSACTION  
lobko_daniil=> SELECT * FROM employer WHERE e_id=1;  
e_id | name | salary  
-----  
1 | Ivanov | 2002  
(1 row)  
  
lobko_daniil=> LOCK TABLE employer IN EXCLUSIVE MODE;  
LOCK TABLE  
lobko_daniil=> UPDATE employer SET salary=20002 WHERE e_id=1;  
UPDATE 1  
lobko_daniil=> SELECT * FROM employer WHERE e_id=1;  
e_id | name | salary  
-----  
1 | Ivanov | 20002  
(1 row)  
  
lobko_daniil=> commit;  
COMMIT  
lobko_daniil=>
```

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

```
lobko_daniil@vpsj3leQ:~  
-----  
1 | Ivanov | 2003  
(1 row)  
  
lobko_daniil=> commit;  
COMMIT  
lobko_daniil=> START TRANSACTION;  
START TRANSACTION  
lobko_daniil=> SELECT * FROM employer WHERE e_id=1;  
e_id | name | salary  
-----  
1 | Ivanov | 2003  
(1 row)  
  
lobko_daniil=> LOCK TABLE employer IN EXCLUSIVE MODE;  
LOCK TABLE  
lobko_daniil=> UPDATE employer SET salary=2003 WHERE e_id=1;  
UPDATE 1  
lobko_daniil=> SELECT * FROM employer WHERE e_id=1;  
e_id | name | salary  
-----  
1 | Ivanov | 2003  
(1 row)  
  
lobko_daniil=> commit;  
COMMIT  
lobko_daniil=>   
  
lobko_daniil@vpsj3leQ:~  
-----  
COMMIT  
lobko_daniil=> START TRANSACTION;  
START TRANSACTION  
lobko_daniil=> SELECT * FROM employer WHERE e_id=1;  
e_id | name | salary  
-----  
1 | Ivanov | 20002  
(1 row)  
  
lobko_daniil=> LOCK TABLE employer IN EXCLUSIVE MODE;  
LOCK TABLE  
lobko_daniil=> START TRANSACTION;  
WARNING: there is already a transaction in progress  
START TRANSACTION  
lobko_daniil=> ROLLBACK;  
ROLLBACK  
lobko_daniil=> START TRANSACTION;  
START TRANSACTION  
lobko_daniil=> SELECT * FROM employer WHERE e_id=1;  
e_id | name | salary  
-----  
1 | Ivanov | 2003  
(1 row)  
  
lobko_daniil=> LOCK TABLE employer IN EXCLUSIVE MODE;  
LOCK TABLE  
lobko_daniil=>   

```

У другому (правому) терміналі ми потрапили в тупик через квазіпаралельне виконання блокування(робота терміналу відновилося через commit у першому терміналі).

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

```
lobko_daniil@vpsj3leQ:~  
-----  
1 | Ivanov | 2003  
(1 row)  
  
lobko_daniil=> commit;  
COMMIT  
lobko_daniil=> START TRANSACTION;  
START TRANSACTION  
lobko_daniil=> SELECT * FROM employer WHERE e_id=1;  
e_id | name | salary  
-----  
1 | Ivanov | 2003  
(1 row)  
  
lobko_daniil=> LOCK TABLE employer IN EXCLUSIVE MODE;  
LOCK TABLE  
lobko_daniil=> UPDATE employer SET salary=200344 WHERE e_id=1;  
UPDATE 1  
lobko_daniil=> SELECT * FROM employer WHERE e_id=1;  
e_id | name | salary  
-----  
1 | Ivanov | 200344  
(1 row)  
  
lobko_daniil=> rollback;  
ROLLBACK  
lobko_daniil=>   
  
lobko_daniil@vpsj3leQ:~  
-----  
WARNING: there is already a transaction in progress  
START TRANSACTION  
lobko_daniil=> ROLLBACK;  
ROLLBACK  
lobko_daniil=> START TRANSACTION;  
START TRANSACTION  
lobko_daniil=> SELECT * FROM employer WHERE e_id=1;  
e_id | name | salary  
-----  
1 | Ivanov | 2003  
(1 row)  
  
lobko_daniil=> LOCK TABLE employer IN EXCLUSIVE MODE;  
LOCK TABLE  
lobko_daniil=> rollback;  
ROLLBACK  
lobko_daniil=> START TRANSACTION;  
START TRANSACTION  
lobko_daniil=> SELECT * FROM employer WHERE e_id=1;  
e_id | name | salary  
-----  
1 | Ivanov | 2003  
(1 row)  
  
lobko_daniil=> LOCK TABLE employer IN EXCLUSIVE MODE;  
^[[B^[[ALOCK TABLE  
lobko_daniil=> LOCK TABLE employer IN EXCLUSIVE MODE;  

```

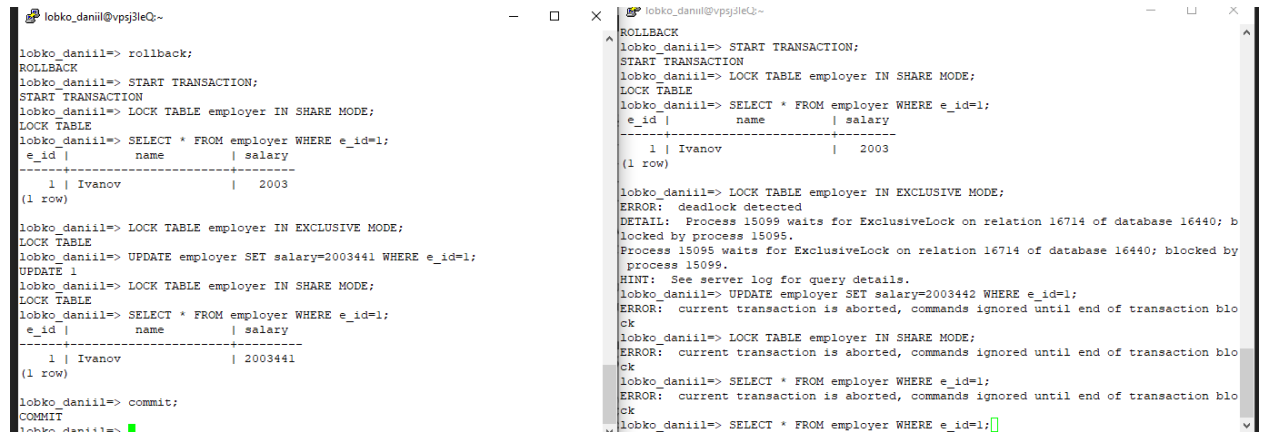
5 Повторіть пункти 3 та 4 але з використанням протоколу 2-го ступеня блокування.

```
lobko_daniil@vpsj3leQ:~  
-----  
1 | Ivanov | 2003  
(1 row)  
  
lobko_daniil=> LOCK TABLE employer IN EXCLUSIVE MODE;  
LOCK TABLE  
lobko_daniil=> UPDATE employer SET salary=200344 WHERE e_id=1;  
UPDATE 1  
lobko_daniil=> SELECT * FROM employer WHERE e_id=1;  
e_id | name | salary  
-----  
1 | Ivanov | 200344  
(1 row)  
  
lobko_daniil=> rollback;  
ROLLBACK  
lobko_daniil=> START TRANSACTION;  
START TRANSACTION  
lobko_daniil=> LOCK TABLE employer IN SHARE MODE;  
LOCK TABLE  
lobko_daniil=> SELECT * FROM employer WHERE e_id=1;  
e_id | name | salary  
-----  
1 | Ivanov | 2003  
(1 row)  
  
lobko_daniil=> LOCK TABLE employer IN EXCLUSIVE MODE;  
LOCK TABLE  
lobko_daniil=>   
  
lobko_daniil@vpsj3leQ:~  
-----  
lobko_daniil=> SELECT * FROM employer WHERE e_id=1;  
e_id | name | salary  
-----  
1 | Ivanov | 2003  
(1 row)  
  
lobko_daniil=> LOCK TABLE employer IN EXCLUSIVE MODE;  
^[[B^[[ALOCK TABLE  
lobko_daniil=> rollback;  
ROLLBACK  
lobko_daniil=> START TRANSACTION;  
START TRANSACTION  
lobko_daniil=> LOCK TABLE employer IN SHARE MODE;  
LOCK TABLE  
lobko_daniil=> SELECT * FROM employer WHERE e_id=1;  
e_id | name | salary  
-----  
1 | Ivanov | 2003  
(1 row)  
  
lobko_daniil=> LOCK TABLE employer IN EXCLUSIVE MODE;  
ERROR: deadlock detected  
DETAIL: Process 15099 waits for ExclusiveLock on relation 16714 of database 16440; b  
locked by process 15095.  
Process 15095 waits for ExclusiveLock on relation 16714 of database 16440; blocked by  
process 15099.  
HINT: See server log for query details.  
lobko_daniil=>   

```

Як ми можемо бачити, ми отримали так званий “deadlock”. (перший термінал під час заблокування X mode потрапив у очікування).

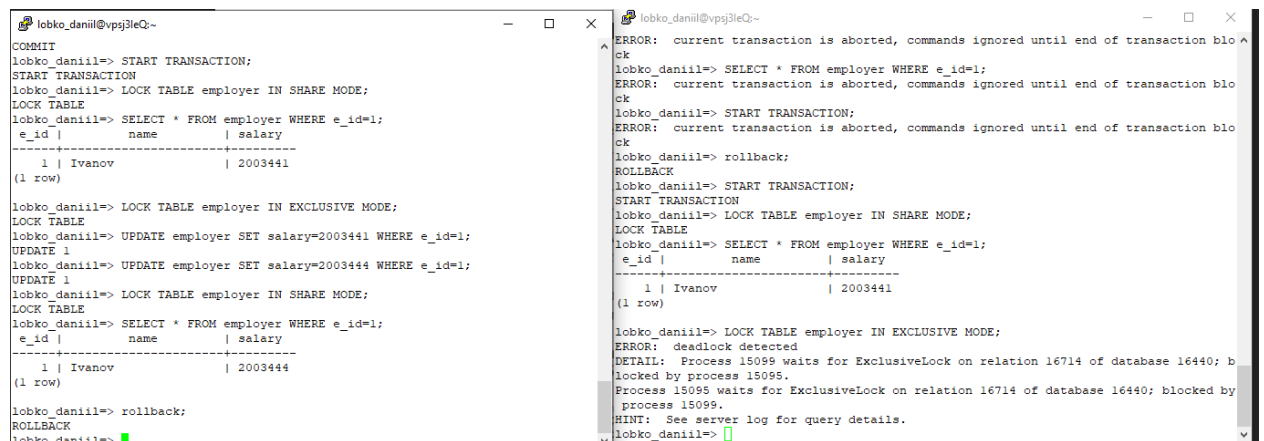
Commit:



```
lobko_daniil@vpsj3leQ:~  
lobko_daniil=> rollback;  
ROLLBACK  
lobko_daniil=> START TRANSACTION;  
START TRANSACTION  
lobko_daniil=> LOCK TABLE employer IN SHARE MODE;  
LOCK TABLE  
lobko_daniil=> SELECT * FROM employer WHERE e_id=1;  
e_id | name | salary  
-----+-----+-----  
1 | Ivanov | 2003  
(1 row)  
  
lobko_daniil=> LOCK TABLE employer IN EXCLUSIVE MODE;  
LOCK TABLE  
lobko_daniil=> UPDATE employer SET salary=2003441 WHERE e_id=1;  
UPDATE 1  
lobko_daniil=> LOCK TABLE employer IN SHARE MODE;  
LOCK TABLE  
lobko_daniil=> SELECT * FROM employer WHERE e_id=1;  
e_id | name | salary  
-----+-----+-----  
1 | Ivanov | 2003441  
(1 row)  
  
lobko_daniil=> commit;  
COMMIT  
lobko_daniil=>
```

```
lobko_daniil@vpsj3leQ:~  
ROLLBACK  
lobko_daniil=> START TRANSACTION;  
START TRANSACTION  
lobko_daniil=> LOCK TABLE employer IN SHARE MODE;  
LOCK TABLE  
lobko_daniil=> SELECT * FROM employer WHERE e_id=1;  
e_id | name | salary  
-----+-----+-----  
1 | Ivanov | 2003  
(1 row)  
  
lobko_daniil=> LOCK TABLE employer IN EXCLUSIVE MODE;  
ERROR: deadlock detected  
DETAIL: Process 15099 waits for ExclusiveLock on relation 16714 of database 16440; b  
locked by process 15095.  
Process 15095 waits for ExclusiveLock on relation 16714 of database 16440; blocked by  
process 15099.  
HINT: See server log for query details.  
lobko_daniil=> UPDATE employer SET salary=2003442 WHERE e_id=1;  
ERROR: current transaction is aborted, commands ignored until end of transaction blo  
ck  
lobko_daniil=> LOCK TABLE employer IN SHARE MODE;  
ERROR: current transaction is aborted, commands ignored until end of transaction blo  
ck  
lobko_daniil=> SELECT * FROM employer WHERE e_id=1;  
ERROR: current transaction is aborted, commands ignored until end of transaction blo  
ck  
lobko_daniil=> SELECT * FROM employer WHERE e_id=1;
```

Rollback:



```
lobko_daniil@vpsj3leQ:~  
COMMIT  
lobko_daniil=> START TRANSACTION;  
START TRANSACTION  
lobko_daniil=> LOCK TABLE employer IN SHARE MODE;  
LOCK TABLE  
lobko_daniil=> SELECT * FROM employer WHERE e_id=1;  
e_id | name | salary  
-----+-----+-----  
1 | Ivanov | 2003441  
(1 row)  
  
lobko_daniil=> LOCK TABLE employer IN EXCLUSIVE MODE;  
LOCK TABLE  
lobko_daniil=> UPDATE employer SET salary=2003441 WHERE e_id=1;  
UPDATE 1  
lobko_daniil=> UPDATE employer SET salary=2003444 WHERE e_id=1;  
UPDATE 1  
lobko_daniil=> LOCK TABLE employer IN SHARE MODE;  
LOCK TABLE  
lobko_daniil=> SELECT * FROM employer WHERE e_id=1;  
e_id | name | salary  
-----+-----+-----  
1 | Ivanov | 2003444  
(1 row)  
  
lobko_daniil=> rollback;  
ROLLBACK  
lobko_daniil=>
```

```
lobko_daniil@vpsj3leQ:~  
ERROR: current transaction is aborted, commands ignored until end of transaction blo  
ck  
lobko_daniil=> SELECT * FROM employer WHERE e_id=1;  
ERROR: current transaction is aborted, commands ignored until end of transaction blo  
ck  
lobko_daniil=> START TRANSACTION;  
START TRANSACTION  
lobko_daniil=> LOCK TABLE employer IN SHARE MODE;  
LOCK TABLE  
lobko_daniil=> SELECT * FROM employer WHERE e_id=1;  
e_id | name | salary  
-----+-----+-----  
1 | Ivanov | 2003441  
(1 row)  
  
lobko_daniil=> LOCK TABLE employer IN EXCLUSIVE MODE;  
ERROR: deadlock detected  
DETAIL: Process 15099 waits for ExclusiveLock on relation 16714 of database 16440; b  
locked by process 15095.  
Process 15095 waits for ExclusiveLock on relation 16714 of database 16440; blocked by  
process 15099.  
HINT: See server log for query details.  
lobko_daniil=>
```

**Висновки:** Виконуючи цю лабораторну роботи ми закріпили навички роботи з керування процесами-транзакціями в базах даних

Найскладнішим було завдання 1.