

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 АІ-202  
Лукашак Д. О.

Перевірив:  
Блажко О. А.

**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.

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.

### **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.

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

2.1 Встановіть з'єднання з вашою базою даних.

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

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

2.4 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**

3.1 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.

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

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

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

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

## Виконання

### Завдання 1. Проектування транзакцій

Нехай задані три транзакції:

$$T_1 = R[D] \ R[A] \ W[D] \ C1$$

$$T_2 = W[A] \ W[B] \ C2$$

$$T_3 = W[A] \ R[B] \ W[D] \ C3$$

Виконайте наступні теоретичні завдання:

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

$$H_{T_1, T_2, T_3} = R1[D] \ X2[A] \ W2[A] \ X3[A] - Wait \ R1[A] \ X2[B] \ W2[B] \ X1[D] \ W1[D] \ U2 \ C2 \ X3[A] \ W3[A] \ U1 \ C1 \ R3[B] \ X3[D] \ W3[D] \ U3 \ C3$$

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

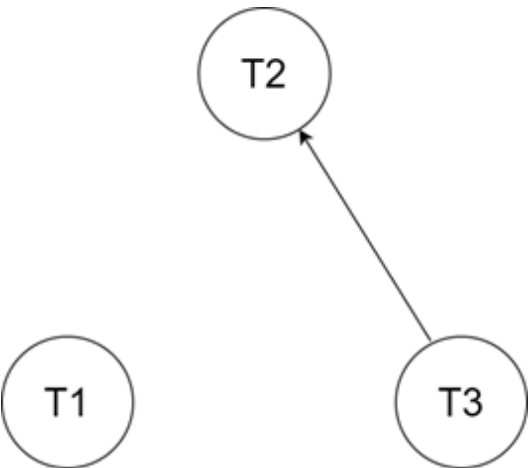
1.2 Повторіть попереднє завдання з використанням протоколу 2-го ступеня блокування.

$H_{T_1,T_2,T_3} = S1[D] \ R1[D] \ X2[A] \ W2[A] \ X3[A] - Wait \ S1[A] - Wait \ X2[B] \ W2[B] \ U2 \ C2 \ X3[A] \ W3[A] \ S3[B] \ R3[B] \ X3[D] - Wait >$  тупиковий стан

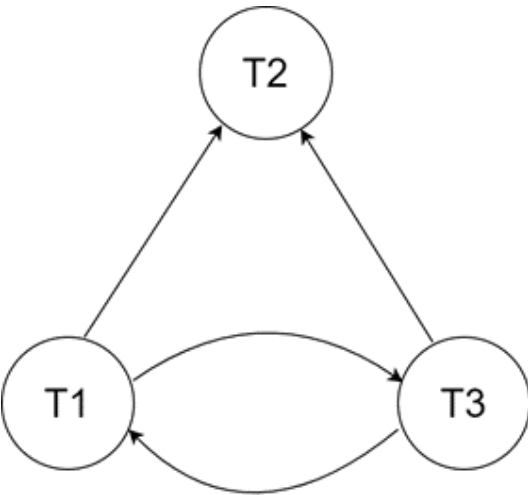
Назва змінної	Перелік встановлених блокувань	Перелік запитів на блокування
A	X2 X3	X3 S1
B	X2 S3	-
D	S1	X3

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

1. Тупику немає



2. 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 наведено в таблиці.

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

Операція створення таблиці в базі даних	Операції внесення даних в базі даних
Create table teacher (t_id integer, name char(20), post char(20));	Insert into teacher values (1, 'Ivanov', 'docent');

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

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

lukashak_daniil=> █
```

### 2.2 Створіть реляційну таблицю в базі даних.

```
lukashak_daniil=> CREATE TABLE teacher (t_id integer, name char(20), post char(20));
CREATE TABLE
lukashak_daniil=> █
```

### 2.3 Додайте рядок в реляційну таблицю, створену у попередньому пункті завдання.

```
lukashak_daniil=> INSERT INTO teacher VALUES(1, 'Ivanov', 'docent');
INSERT 0 1
lukashak_daniil=> █
```

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

```
lukashak_daniil=> INSERT INTO teacher VALUES(2, 'Lukashak', 'ne docent');
INSERT 0 1
lukashak_daniil=> █
```

```
lukashak_daniil=> SELECT * FROM teacher;
```

t_id	name	post
1	Ivanov	docent
2	Lukashak	ne docent

(2 rows)

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

3.1 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.

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

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

#### 1) Перший термінал:

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

lukashak_daniil=> START TRANSACTION;
START TRANSACTION
lukashak_daniil=> SELECT * FROM teacher WHERE t_id = 1;
```

t_id	name	post
1	Ivanov	docent

(1 row)

```
lukashak_daniil=> LOCK TABLE teacher in EXCLUSIVE MODE;
LOCK TABLE
lukashak_daniil=> UPDATE teacher SET name = 'Ivanova' WHERE t_id = 1;
UPDATE 1
lukashak_daniil=> SELECT * FROM teacher WHERE t_id = 1;
```

t_id	name	post
1	Ivanova	docent

(1 row)

```
lukashak_daniil=> COMMIT;
COMMIT
lukashak_daniil=>
```

#### 2) Другий термінал:

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

lukashak_daniil=> START TRANSACTION;
START TRANSACTION
lukashak_daniil=> SELECT * FROM teacher WHERE t_id = 1;
 t_id |      name      |      post
-----+-----+-----
  1   | Ivanov         | docent
(1 row)

lukashak_daniil=> LOCK TABLE teacher in EXCLUSIVE MODE;
LOCK TABLE
lukashak_daniil=> UPDATE teacher SET name = 'Ivanov' WHERE t_id = 1;
UPDATE 1
lukashak_daniil=> SELECT * FROM teacher WHERE t_id = 1;
 t_id |      name      |      post
-----+-----+-----
  1   | Ivanov         | docent
(1 row)

lukashak_daniil=> COMMIT;
COMMIT
lukashak_daniil=> █
```

Під час виконання, LOCK у другому терміналі перейшов до стану очікування, так як таблиця вже була заблокована першим терміналом. Після COMMIT у першому, з'явилася змога продовжити виконання й у другому.

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

#### 1) Перший термінал:

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

lukashak_daniil=> START TRANSACTION;
START TRANSACTION
lukashak_daniil=> SELECT * FROM teacher WHERE t_id = 1;
 t_id |      name      |      post
-----+-----+-----
  1   | Ivanov         | docent
(1 row)

lukashak_daniil=> LOCK TABLE teacher in EXCLUSIVE MODE;
LOCK TABLE
lukashak_daniil=> UPDATE teacher SET name = 'Ivanova' WHERE t_id = 1;
UPDATE 1
lukashak_daniil=> SELECT * FROM teacher WHERE t_id = 1;
 t_id |      name      |      post
-----+-----+-----
  1   | Ivanova        | docent
(1 row)

lukashak_daniil=> ROLLBACK;
ROLLBACK
lukashak_daniil=> █
```

#### 2) Другий термінал:

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

lukashak_daniil=> START TRANSACTION;
START TRANSACTION
lukashak_daniil=> SELECT * FROM teacher WHERE t_id = 1;
 t_id |      name      |      post
-----+-----+-----
    1 | Ivanov         | docent
(1 row)

lukashak_daniil=> LOCK TABLE teacher in EXCLUSIVE MODE;
LOCK TABLE
lukashak_daniil=> UPDATE teacher SET name = 'Ivanovna' WHERE t_id = 1;
UPDATE 1
lukashak_daniil=> SELECT * FROM teacher WHERE t_id = 1;
 t_id |      name      |      post
-----+-----+-----
    1 | Ivanovna       | docent
(1 row)

lukashak_daniil=> COMMIT;
COMMIT
lukashak_daniil=> █
```

Під час виконання, LOCK у другому терміналі перейшов до стану очікування, так як таблиця вже була заблокована першим терміналом. Після ROLLBACK у першому, з'явилася змога продовжити виконання й у другому.

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

1.

1) Перший термінал:

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

lukashak_daniil=> START TRANSACTION;
START TRANSACTION
lukashak_daniil=> LOCK TABLE teacher in SHARE MODE;
LOCK TABLE
lukashak_daniil=> SELECT * FROM teacher WHERE t_id = 1;
 t_id |      name      |      post
-----+-----+-----
    1 | Ivanov         | docent
(1 row)

lukashak_daniil=> LOCK TABLE teacher in EXCLUSIVE MODE;
LOCK TABLE
lukashak_daniil=> []
```

2) Другий термінал:



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

lukashak_daniil=> START TRANSACTION;
START TRANSACTION
lukashak_daniil=> LOCK TABLE teacher in SHARE MODE;
LOCK TABLE
lukashak_daniil=> SELECT * FROM teacher WHERE t_id = 1;
 t_id |      name      |      post
-----+-----+-----
    1 | Ivanov         | docent
(1 row)

lukashak_daniil=> LOCK TABLE teacher in EXCLUSIVE MODE;
ERROR:  deadlock detected
DETAIL:  Process 26188 waits for ExclusiveLock on relation 16675 of database 16448; blocked by process 26185.
Process 26185 waits for ExclusiveLock on relation 16675 of database 16448; blocked by process 26188.
HINT:   See server log for query details.
lukashak_daniil=> []
```

2.

### 1) Перший термінал:

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

lukashak_daniil=> START TRANSACTION;
START TRANSACTION
lukashak_daniil=> LOCK TABLE teacher in SHARE MODE;
LOCK TABLE
lukashak_daniil=> SELECT * FROM teacher WHERE t_id = 1;
 t_id |      name      |      post
-----+-----+-----
    1 | Ivanovna       | docent
(1 row)

lukashak_daniil=> LOCK TABLE teacher in EXCLUSIVE MODE;
LOCK TABLE
lukashak_daniil=> []
```

### 2) Другий термінал:

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

lukashak_daniil=> START TRANSACTION;
START TRANSACTION
lukashak_daniil=> LOCK TABLE teacher in SHARE MODE;
LOCK TABLE
lukashak_daniil=> SELECT * FROM teacher WHERE t_id = 1;
 t_id |      name      |      post
-----+-----+-----
    1 | Ivanovna       | docent
(1 row)

lukashak_daniil=> LOCK TABLE teacher in EXCLUSIVE MODE;
ERROR:  deadlock detected
DETAIL:  Process 27886 waits for ExclusiveLock on relation 16675 of database 16448; blocked by process 27883.
Process 27883 waits for ExclusiveLock on relation 16675 of database 16448; blocked by process 27886.
HINT:   See server log for query details.
lukashak_daniil=> █
```

Між двома завданнями нема різниці, так як ми не дійдемо до фіксації чи відміни через тупик, що зустрівся після X1[teacher] та X2[teacher].

## **Висновки:**

Ми дослідили поведінку процесів-транзакцій в базах даних та засоби керування ними через механізм блокування з використанням сучасних систем керування базами даних. Найскладніше завдання було на написання історій транзакцій, так як легко заплутатися хто кого очікує.