

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ПОЛІТЕХНІКА»
ІНСТИТУТ КОМП'ЮТЕРНИХ СИСТЕМ
КАФЕДРА «ІНФОРМАЦІЙНИ СИСТЕМИ»

Лабораторна робота №9

з дисципліни

«Операційні системи»

Тема

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

Виконав:

Студент групи AI-202

Узун Михайло

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.

1.1 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
8	2	T1= R[A] W[A] C1 T2= W[A] R[B] W[B] C2 T3= R[D] R[B] 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
8	2	<i>Create table auto (a_id integer, name char(20), year integer);</i>	<i>Insert into auto values (1, 'BMW 5', 2003);</i>

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.

Таблиця 15 – Варіанти завдань з прикладами транзакцій

Таблиця 1.2. Варіанти завдань з прикладами транзакцій

№ команди	№ учасника	Приклади транзакцій
8	2	T1= R[A] W[A] C1 T2= W[A] R[B] W[B] C2 T3= R[D] R[B] W[D] 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.

$H_{T1,T2,T3} = R1[A] \text{ X2}[A] W2[A] R3[D] \text{ X1}[A] - \text{Wait } R2[B] R3[B] X2[B] W2[B] X3[D] W3[D] \text{ U2 } C2 U3 C3 X1[A] W1[A] U1 C1$

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

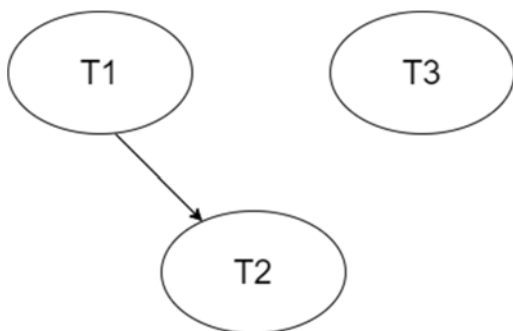
1.2 Повторіть попереднє завдання з використанням протоколу 2-го ступеня блокування.

$H_{T1,T2,T3} = S1[A] R1[A] \text{ X2}[A] W2[A] S3[D] R3[D] \text{ X1}[A] - \text{Wait } \text{S2}[B] R2[B] \text{ S3}[B] - \text{Wait } X2[B] W2[B] U2 C2 S3[B] R3[B] X1[A] W1[A] X3[D] W3[D] U1 C1 U3 C3$

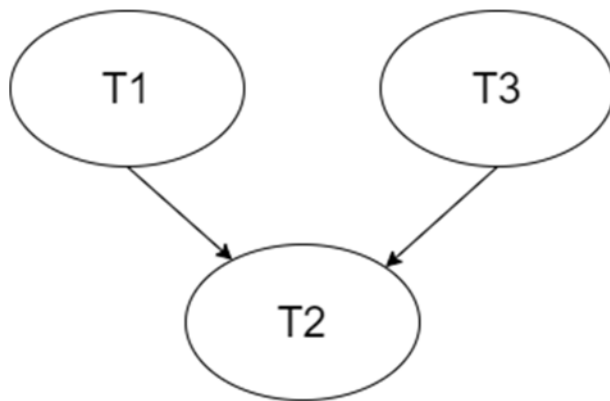
Назва змінної	Перелік встановлених блокувань	Перелік запитів на блокування
A	X1 X2 S1	X1
B	X2 S2 S3	S3
D	X3 S3	-

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

1)



2)



Тупиков не знайдено.

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.

№ команд	№ учасника	Операція створення реляційної таблиці в базі даних	Операції внесення даних в реляційну таблицю в базі даних
8	2	<i>Create table auto (a_id integer, name char(20), year integer);</i>	<i>Insert into auto values (1, 'BMW 5', 2003);</i>

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

1.1 Встановіть з'єднання з вашою базою даних.

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

uzun_mihajlo=> █
```

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

```
uzun_mihajlo=> CREATE TABLE auto (a_id integer, name char(20), year integer);
CREATE TABLE
uzun_mihajlo=> █
```

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

```
uzun_mihajlo=> INSERT INTO auto VALUES (1, 'Toyota', 2004);  
INSERT 0 1
```

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

```
uzun_mihajlo=> INSERT INTO auto VALUES (2, 'Mercedes', 2010);  
INSERT 0 1
```

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.

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

```
uzun_mihajlo=> START TRANSACTION;
START TRANSACTION
uzun_mihajlo=> SELECT * FROM auto WHERE a_id = 1;
 a_id |      name      | year
-----+-----+-----
    1 | Toyota         | 2004
(1 row)
```

```
uzun_mihajlo=> LOCK TABLE auto IN EXCLUSIVE MODE;
LOCK TABLE
uzun_mihajlo=> UPDATE auto SET name = 'Honda' WHERE a_id = 1;
UPDATE 1
uzun_mihajlo=> SELECT * FROM auto WHERE a_id = 1;
 a_id |      name      | year
-----+-----+-----
    1 | Honda          | 2004
(1 row)
```

```
uzun_mihajlo=> COMMIT;
COMMIT
uzun_mihajlo=>
```

 uzun_mihajlo@vpsj3IeQ:~

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

```
uzun_mihajlo=> START TRANSACTION;
START TRANSACTION
uzun_mihajlo=> SELECT * FROM auto WHERE a_id = 1;
 a_id |      name      | year
-----+-----+-----
    1 | Toyota         | 2004
(1 row)

uzun_mihajlo=> LOCK TABLE auto IN EXCLUSIVE MODE;
LOCK TABLE
uzun_mihajlo=> UPDATE auto SET name = 'Mazda' WHERE a_id = 1;
UPDATE 1
uzun_mihajlo=> SELECT * FROM auto WHERE a_id = 1;
 a_id |      name      | year
-----+-----+-----
    1 | Mazda          | 2004
(1 row)

uzun_mihajlo=> COMMIT
```

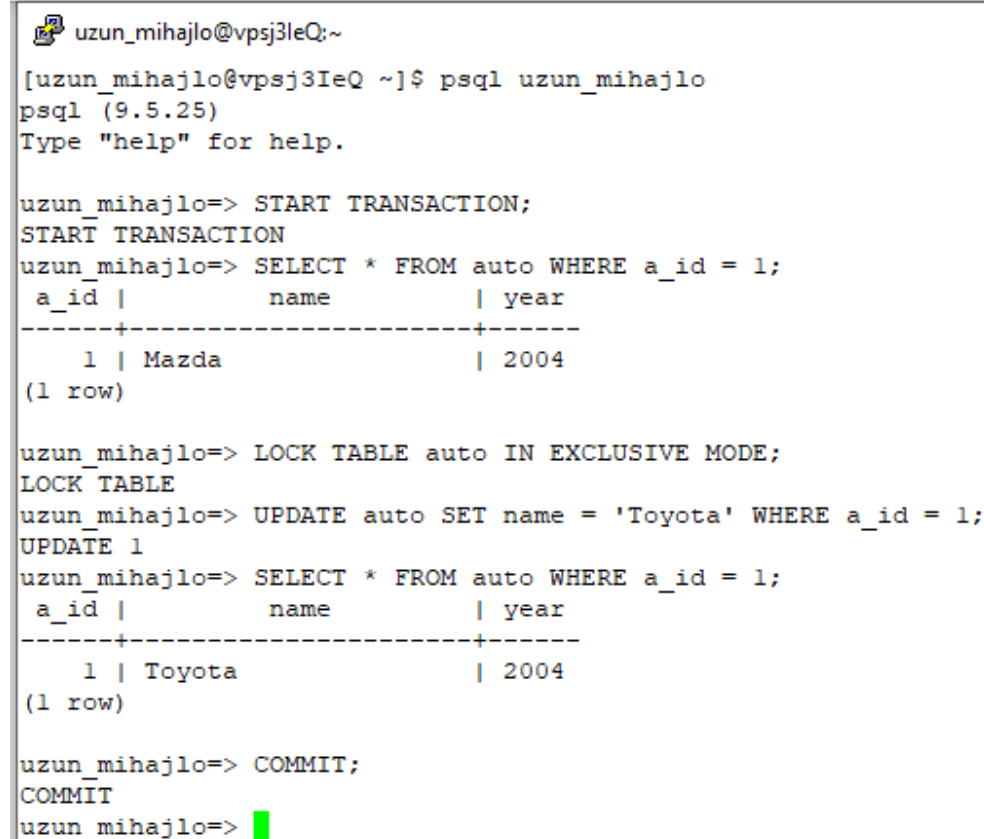
4 Повторіть роботу транзакцій, але в першій транзакції замість операції фіксації виконайте операцію відміни.

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

uzun_mihajlo=> START TRANSACTION;
START TRANSACTION
uzun_mihajlo=> SELECT * FROM auto WHERE a_id = 1;
 a_id |          name          | year
-----+-----+-----
    1 | Mazda                  | 2004
(1 row)

uzun_mihajlo=> LOCK TABLE auto IN EXCLUSIVE MODE;
LOCK TABLE
uzun_mihajlo=> UPDATE auto SET name = 'Honda' WHERE a_id = 1;
UPDATE 1
uzun_mihajlo=> SELECT * FROM auto WHERE a_id = 1;
 a_id |          name          | year
-----+-----+-----
    1 | Honda                  | 2004
(1 row)

uzun_mihajlo=> ROLLBACK;
ROLLBACK
uzun_mihajlo=> 
```



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

uzun_mihajlo=> START TRANSACTION;
START TRANSACTION
uzun_mihajlo=> SELECT * FROM auto WHERE a_id = 1;
 a_id |          name          | year
-----+-----+-----
    1 | Mazda                  | 2004
(1 row)

uzun_mihajlo=> LOCK TABLE auto IN EXCLUSIVE MODE;
LOCK TABLE
uzun_mihajlo=> UPDATE auto SET name = 'Toyota' WHERE a_id = 1;
UPDATE 1
uzun_mihajlo=> SELECT * FROM auto WHERE a_id = 1;
 a_id |          name          | year
-----+-----+-----
    1 | Toyota                 | 2004
(1 row)

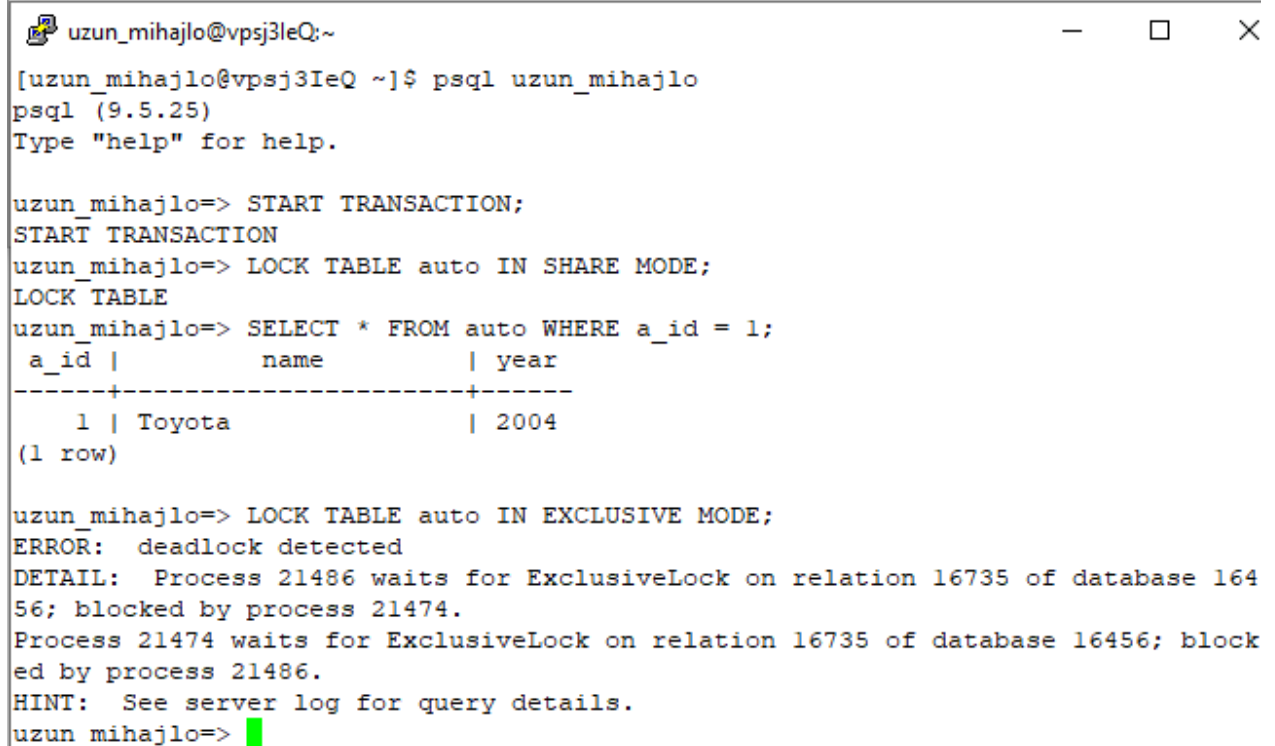
uzun_mihajlo=> COMMIT;
COMMIT
uzun_mihajlo=> 
```


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

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

uzun_mihajlo=> START TRANSACTION;
START TRANSACTION
uzun_mihajlo=> LOCK TABLE auto IN SHARE MODE;
LOCK TABLE
uzun_mihajlo=> SELECT * FROM auto WHERE a_id = 1;
 a_id |          name          | year
-----+-----+-----
    1 | Toyota                 | 2004
(1 row)

uzun_mihajlo=> LOCK TABLE auto IN EXCLUSIVE MODE;
LOCK TABLE
uzun_mihajlo=> 
```



```
ERROR: deadlock detected
DETAIL: Process 21486 waits for ExclusiveLock on relation 16735 of database 164
56; blocked by process 21474.
Process 21474 waits for ExclusiveLock on relation 16735 of database 16456; block
ed by process 21486.
HINT: See server log for query details.
uzun_mihajlo=> 
```

Формується тупикова ситуація.

Висновок: в процесі виконання цієї лабораторної роботи були досліджені поведінка процесів-транзакцій в базах даних та засоби керування ними через механізм блокування з використанням сучасних систем керування базами даних.