

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. Частина 1»

Виконала: Студентка групи AI-204

Озарчук А.С.

Перевірили: Блажко О. А

Дрозд М. О.

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.

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.

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.

Виконання завдань

Завдання 1. Проектування транзакцій

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

8	1	T1= R[D] W[D] C1 T2= R[A] R[B] W[A] C2 T3= W[D] R[B] W[B] C3
---	---	--

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

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

Таблиця блокувань

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

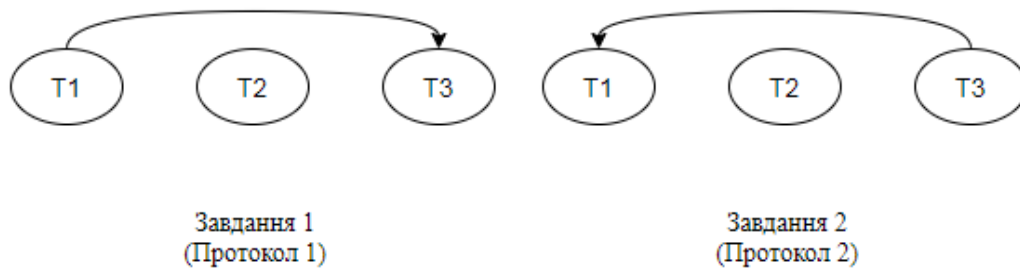
1.2 Повторила попереднє завдання з використанням протоколу 2-го ступеня блокування.

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

Таблиця блокувань

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

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



Тупиків не має.

Завдання 2. Налаштування бази даних

Нехай існує база даних, назва якої співпадає з іменем мого користувача в ОС Linux.

8	1	Create table university (u_id integer, name char(20), year integer);	Insert into university values (1, 'ONPU', 1918);
---	---	---	---

1.1 Я встановила з'єднання з моєю базою даних.

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

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

```
ozarchuk_anna=> Create table university
ozarchuk_anna-> (u_id integer, name char(20), year integer);
CREATE TABLE
```

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

```
ozarchuk_anna=> Insert into university values (1, 'ONPU', 1918);
INSERT 0 1
```

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

```
ozarchuk_anna=> Insert into university values (3, 'DUOP', 2020);
INSERT 0 1
```

1.5 Перевірила вміст таблиці

```
ozarchuk_anna=> SELECT * FROM university;
 u_id |      name      | year
-----+-----+-----
    1 | ONPU           | 1918
    3 | DUOP           | 2020
(2 rows)

ozarchuk_anna=> █
```

Завдання 3. Керування квазіпаралельним виконанням транзакцій з використанням команд блокування

1. Я створила 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 university WHERE u_id = 1;	SELECT * FROM university WHERE u_id = 1;
UPDATE university SET name = 'DUOP' WHERE u_id = 1;	UPDATE university SET year = 2020 WHERE u_id = 1;
SELECT * FROM university WHERE u_id = 1;	SELECT * FROM university WHERE u_id = 1;
COMMIT;	COMMIT;

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

Транзакція 1	Транзакція 2
START TRANSACTION	START TRANSACTION
SELECT * FROM university WHERE u_id = 1;	SELECT * FROM university WHERE u_id = 1;
LOCK TABLE university IN EXCLUSIVE MODE;	LOCK TABLE university IN EXCLUSIVE MODE;
UPDATE university SET name = 'DUOP' WHERE u_id = 1;	UPDATE university SET year = 2020 WHERE u_id = 1;
SELECT * FROM university	SELECT * FROM university

WHERE u_id = 1;	WHERE u_id = 1;
COMMIT;	COMMIT;

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

```

ozarchuk_anna@vpsj3leQ:~
psql (9.5.25)
Type "help" for help.

ozarchuk_anna=> START TRANSACTION;
START TRANSACTION
ozarchuk_anna=> SELECT * FROM university WHERE u_id = 1;
 u_id |      name      | year
-----+-----+-----
  1   | ONPU           | 1918
(1 row)

ozarchuk_anna=> LOCK TABLE university IN EXCLUSIVE MODE;
LOCK TABLE
ozarchuk_anna=> UPDATE university SET name = 'DUOP' WHERE u_id = 1;
UPDATE 1
ozarchuk_anna=> SELECT * FROM university WHERE u_id = 1;
 u_id |      name      | year
-----+-----+-----
  1   | DUOP           | 1918
(1 row)

ozarchuk_anna=> COMMIT;
COMMIT
ozarchuk_anna=> █

```

```

ozarchuk_anna@vpsj3leQ:~
psql (9.5.25)
Type "help" for help.

ozarchuk_anna=> START TRANSACTION;
START TRANSACTION
ozarchuk_anna=> SELECT * FROM university WHERE u_id = 1;
 u_id |      name      | year
-----+-----+-----
  1   | ONPU           | 1918
(1 row)

ozarchuk_anna=> LOCK TABLE university IN EXCLUSIVE MODE;
LOCK TABLE
ozarchuk_anna=> UPDATE university SET year = 2020 WHERE u_id = 1;
UPDATE 1
ozarchuk_anna=> SELECT * FROM university WHERE u_id = 1;
 u_id |      name      | year
-----+-----+-----
  1   | DUOP           | 2020
(1 row)

ozarchuk_anna=> COMMIT;
COMMIT
ozarchuk_anna=> █

```

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

```

ozarchuk_anna@vpsj3leQ:~
psql (9.5.25)
Type "help" for help.

ozarchuk_anna=> START TRANSACTION;
START TRANSACTION
ozarchuk_anna=> SELECT * FROM university WHERE u_id = 1;
 u_id |      name      | year
-----+-----+-----
  1   | DUOP           | 2020
(1 row)

ozarchuk_anna=> LOCK TABLE university IN EXCLUSIVE MODE;
LOCK TABLE
ozarchuk_anna=> UPDATE university SET name = 'ONU' WHERE u_id = 1;
UPDATE 1
ozarchuk_anna=> SELECT * FROM university WHERE u_id = 1;
 u_id |      name      | year
-----+-----+-----
  1   | ONU            | 2020
(1 row)

ozarchuk_anna=> ROLLBACK;
ROLLBACK
ozarchuk_anna=> █

```

```

ozarchuk_anna@vpsj3leQ:~
login as: ozarchuk_anna
ozarchuk_anna@91.219.60.189's password:
Last login: Mon Apr 26 00:23:05 2021 from 176.105.192.126
[ozarchuk_anna@vpsj3leQ ~]$ psql
psql (9.5.25)
Type "help" for help.

ozarchuk_anna=> START TRANSACTION;
START TRANSACTION
ozarchuk_anna=> LOCK TABLE university IN EXCLUSIVE MODE;
LOCK TABLE
ozarchuk_anna=> UPDATE university SET year = 2005 WHERE u_id = 1;
UPDATE 1
ozarchuk_anna=> SELECT * FROM university WHERE u_id = 1;
 u_id |      name      | year
-----+-----+-----
  1   | DUOP           | 2005
(1 row)

ozarchuk_anna=> COMMIT;
COMMIT
ozarchuk_anna=> █

```

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

Транзакція 1	Транзакція 2
START TRANSACTION	START TRANSACTION
LOCK TABLE university IN SHARE MODE;	LOCK TABLE university IN SHARE MODE;
SELECT * FROM university	SELECT * FROM university

WHERE u_id = 1;	WHERE u_id = 1;
LOCK TABLE university IN EXCLUSIVE MODE;	LOCK TABLE university IN EXCLUSIVE MODE;
UPDATE university SET name = 'NUOUA' WHERE u_id = 1;	UPDATE university SET year = 2015 WHERE u_id = 1;
LOCK TABLE university IN SHARE MODE;	LOCK TABLE university IN SHARE MODE;
SELECT * FROM university WHERE u_id = 1;	SELECT * FROM university WHERE u_id = 1;
COMMIT;	COMMIT;

```

ozarchuk_anna@vpsj3leQ:~
login as: ozarchuk_anna
ozarchuk_anna@91.219.60.189's password:
Last login: Mon Apr 26 00:44:40 2021 from 176.105.192.126
[ozarchuk_anna@vpsj3leQ ~]$ psql
psql (9.5.25)
Type "help" for help.

ozarchuk_anna=> START TRANSACTION;
START TRANSACTION
ozarchuk_anna=> LOCK TABLE university IN SHARE MODE;
LOCK TABLE
ozarchuk_anna=> SELECT * FROM university WHERE u_id = 1;
 u_id |      name      | year
-----+-----+-----
  1   | DUOP           | 2005
(1 row)

ozarchuk_anna=> LOCK TABLE university IN EXCLUSIVE MODE;
LOCK TABLE
ozarchuk_anna=> UPDATE university SET name = 'NUOUA' WHERE u_id = 1;
UPDATE 1
ozarchuk_anna=> LOCK TABLE university IN SHARE MODE;
LOCK TABLE
ozarchuk_anna=> SELECT * FROM university WHERE u_id = 1;
 u_id |      name      | year
-----+-----+-----
  1   | NUOUA          | 2005
(1 row)

ozarchuk_anna=> COMMIT;
COMMIT
ozarchuk_anna=> █

```

```

ozarchuk_anna@vpsj3leQ:~
login as: ozarchuk_anna
ozarchuk_anna@91.219.60.189's password:
Last login: Mon Apr 26 00:47:53 2021 from 176.105.192.126
[ozarchuk_anna@vpsj3leQ ~]$ psql
psql (9.5.25)
Type "help" for help.

ozarchuk_anna=> START TRANSACTION;
START TRANSACTION
ozarchuk_anna=> LOCK TABLE university IN SHARE MODE;
LOCK TABLE
ozarchuk_anna=> SELECT * FROM university WHERE u_id = 1;
 u_id |      name      | year
-----+-----+-----
  1   | DUOP           | 2005
(1 row)

ozarchuk_anna=> LOCK TABLE university IN EXCLUSIVE MODE;
ERROR:  deadlock detected
DETAIL:  Process 8870 waits for ExclusiveLock on relation 16630 of database 16540; blocked by process 8790.
Process 8790 waits for ExclusiveLock on relation 16630 of database 16540; blocked by process 8870.
HINT:  See server log for query details.
ozarchuk_anna=> LOCK TABLE university IN EXCLUSIVE MODE;
ERROR:  current transaction is aborted, commands ignored until end of transaction block
ozarchuk_anna=> LOCK TABLE university IN EXCLUSIVE MODE;
ERROR:  current transaction is aborted, commands ignored until end of transaction block
ozarchuk_anna=> UPDATE university SET year = 2015 WHERE u_id = 1;
ERROR:  current transaction is aborted, commands ignored until end of transaction block
ozarchuk_anna=> █

```

Під час виконання протоколу 2-го рівня виникла помилка, тому транзакція 2 була завершена. Через те, що виник цикл, виникла помилка. Обидві транзакції спочатку встановили SHARE LOCK, а потім намагалися встановити EXCLUSIVE LOCK. Для того, щоб вийти з цього циклу, транзакцію було завершено.

Висновок

Під час виконання Лабораторної роботи №9 мною було досліджено поведінку процесів-транзакцій в базах даних та засоби керування ними через механізм блокування з використанням сучасних систем керування базами даних.