

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»

Виконала:

Студентка групи АІ-202
Гребенік Анжеліка Олександрівна

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.

4	1	T1= W[D] W[A] C1 T2= R[D] R[A] W[D] C2 T3= W[B] 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.

4	1	Create table auto (a_id integer, name char(20), year integer);	Insert into auto values (1, 'BMW 5', 2003);
---	---	---	--

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.

4	1	T1= W[D] W[A] C1 T2= R[D] R[A] W[D] C2 T3= W[B] R[A] W[D] C3
---	---	--

$H_{T1,T2,T3} =$

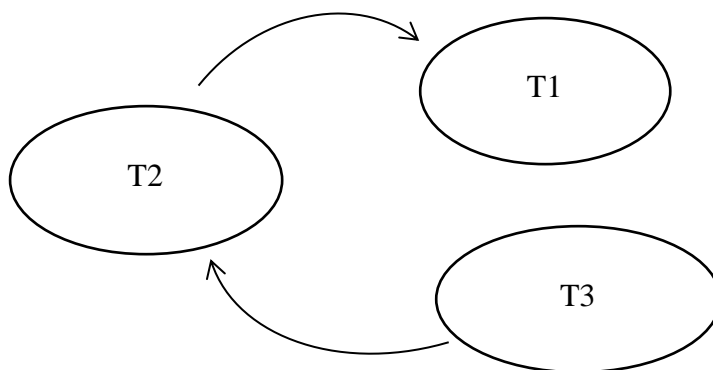
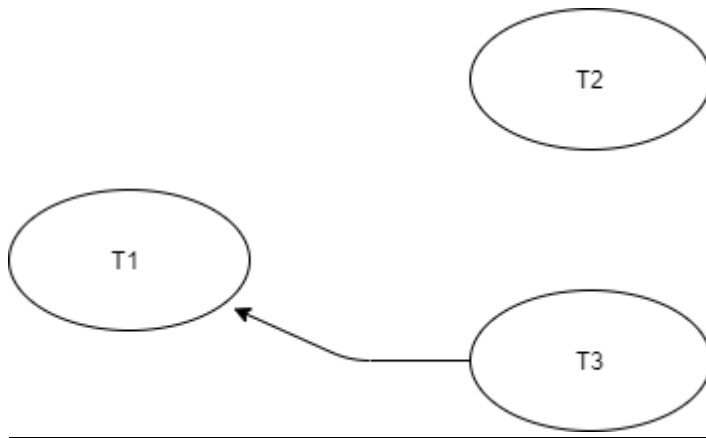
X1[D] W1[D] R2[D] X3[B] W3[B] X1[A] W1[A] R2[A] R3[A] U1 C1 X2[D] W2[D]
X3[D] – wait U2 C2 X3[D] W3[D] U3 C3

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

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

$H_{T_1, T_2, T_3} = X_1[D] \ W_1[D] \ S_2[D] - \text{wait } A_1 \ U_1 \ S_2[D] \ R_2[D] \ X_3[B] \ W_3[B]$
 $X_1[A] \ W_1[A] \ S_2[A] - \text{wait } A_1 \ U_1 \ S_2[A] \ R_2[A] \ S_3[A] \ R_3[A] \ U_1 \ C_1 \ X_2[D]$
 $W_2[D] \ X_3[D] - \text{wait } A_2 \ U_2 \ X_3[D] \ W_3[D] \ U_2 \ C_2 \ U_3 \ C_3$

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.

4	1	Create table auto (a_id integer, name char(20), year integer);	Insert into auto values (1, 'BMW 5', 2003);
---	---	---	--

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

grebenik_anzhelika=> CREATE TABLE auto (a_id integer, name char(20), year integer);
CREATE TABLE
grebenik_anzhelika=> INSERT INTO auto values (1, 'BMW 5', 2003);
INSERT 0 1
grebenik_anzhelika=> INSERT INTO auto values (2, 'Audi', 2006);
INSERT 0 1
grebenik_anzhelika=> █
```

Встановлюємо з'єднання з базою даних

psql grebenik_anzhelika

Вводимо команди для створення таблиці та додаємо рядки

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.

```
grebenik_anzhelika@vpsj3IeQ:~
Last login: Wed Apr 21 12:04:07 2021 from 46.149.53.142
[grebenik_anzhelika@vpsj3IeQ ~]$ psql grebenik_anzhelika
psql (9.5.25)
Type "help" for help.

grebenik_anzhelika=> CREATE TABLE auto (a_id integer, name char(20), year integer);
CREATE TABLE
grebenik_anzhelika=> INSERT INTO auto values (1, 'BMW 5', 2003);
INSERT 0 1
grebenik_anzhelika=> INSERT INTO auto values (2, 'Audi', 2006);
INSERT 0 1
grebenik_anzhelika=> start transaction;
START TRANSACTION
grebenik_anzhelika=> select * from auto where a_id=1;
 a_id |      name      | year 
-----+-----+-----
  1   | BMW 5          | 2003 
(1 row)

grebenik_anzhelika=> ^C
grebenik_anzhelika=> commit;
COMMIT
grebenik_anzhelika=> █

grebenik_anzhelika@vpsj3IeQ:~
login as: grebenik_anzhelika
grebenik_anzhelika@91.219.60.189's password:
Last login: Tue Apr 27 23:22:03 2021 from 46.149.53.142
[grebenik_anzhelika@vpsj3IeQ ~]$ psql grebenik_anzhelika
psql (9.5.25)
Type "help" for help.

grebenik_anzhelika=> start transaction;
START TRANSACTION
grebenik_anzhelika=> select * from auto where a_id=1;
 a_id |      name      | year 
-----+-----+-----
  1   | BMW 5          | 2003 
(1 row)

grebenik_anzhelika=> commit;
COMMIT
grebenik_anzhelika=> █
```

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

```
grebenik_anzhelika@vpsj3leQ:~  
grebenik_anzhelika=> ^C  
grebenik_anzhelika=> commit;  
COMMIT  
grebenik_anzhelika=> start transaction;  
START TRANSACTION  
grebenik_anzhelika=> select * from auto where a_id=1;  
a_id | name | year  
-----+-----+-----  
1 | BMW 5 | 2003  
(1 row)  
  
grebenik_anzhelika=> lock table auto in exclusive mode;  
LOCK TABLE  
grebenik_anzhelika=> update auto set year=2005 where a_id=1;  
UPDATE 1  
grebenik_anzhelika=> select * from auto where a_id=1;  
a_id | name | year  
-----+-----+-----  
1 | BMW 5 | 2005  
(1 row)  
  
grebenik_anzhelika=> commit;  
COMMIT  
grebenik_anzhelika=> █
```

```
grebenik_anzhelika@vpsj3leQ:~  
START TRANSACTION  
grebenik_anzhelika=> select * from auto where a_id=1;  
a_id | name | year  
-----+-----+-----  
1 | BMW 5 | 2003  
(1 row)  
  
grebenik_anzhelika=> commit;  
COMMIT  
grebenik_anzhelika=> start transaction;  
START TRANSACTION  
grebenik_anzhelika=> lock table auto in exclusive mode;  
LOCK TABLE  
grebenik_anzhelika=> update auto set year=2005 where a_id=1;  
UPDATE 1  
grebenik_anzhelika=> select * from auto where a_id=1;  
a_id | name | year  
-----+-----+-----  
1 | BMW 5 | 2005  
(1 row)  
  
grebenik_anzhelika=> commit;  
COMMIT  
grebenik_anzhelika=> █
```

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

```
grebenik_anzhelika@vpsj3leQ:~  
grebenik_anzhelika=> commit;  
WARNING: there is no transaction in progress  
COMMIT  
grebenik_anzhelika=> start transaction;  
START TRANSACTION  
grebenik_anzhelika=> select * from auto where a_id=1;  
a_id | name | year  
-----+-----+-----  
1 | BMW 5 | 2006  
(1 row)  
  
grebenik_anzhelika=> lock table auto in exclusive mode;  
LOCK TABLE  
grebenik_anzhelika=> update auto set year=2008 where a_id=1;  
UPDATE 1  
grebenik_anzhelika=> select * from auto where a_id=1;  
a_id | name | year  
-----+-----+-----  
1 | BMW 5 | 2008  
(1 row)  
  
grebenik_anzhelika=> commit;  
COMMIT  
grebenik_anzhelika=> █
```

```
grebenik_anzhelika@vpsj3leQ:~  
grebenik_anzhelika=> lock table auto in exclusive mode;  
LOCK TABLE  
grebenik_anzhelika=> update auto set year=2006 where a_id=1;  
UPDATE 1  
grebenik_anzhelika=> select * from auto where a_id=1;  
a_id | name | year  
-----+-----+-----  
1 | BMW 5 | 2006  
(1 row)  
  
grebenik_anzhelika=> commit;  
COMMIT  
grebenik_anzhelika=> start transaction;  
START TRANSACTION  
grebenik_anzhelika=> select * from auto where a_id=1;  
a_id | name | year  
-----+-----+-----  
1 | BMW 5 | 2006  
(1 row)  
  
grebenik_anzhelika=> lock table auto in exclusive mode;  
LOCK TABLE  
grebenik_anzhelika=> █
```

У другому терміналі ми потрапляємо в тупик використовуючи блокування, робота відновляється після команди commit в першому терміналі.

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

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

grebenik_anzhelika=> start transaction;
START TRANSACTION
grebenik_anzhelika=> select * from auto where a_id=1;
 a_id | name | year
-----+-----+-----
 1 | BMW 5 | 2005
(1 row)

grebenik_anzhelika=> lock table auto in exclusive mode;
LOCK TABLE
grebenik_anzhelika=> update auto set year=2006 where a_id=1;
UPDATE 1
grebenik_anzhelika=> select * from auto where a_id=1;
 a_id | name | year
-----+-----+-----
 1 | BMW 5 | 2006
(1 row)

grebenik_anzhelika=> rollback;
ROLLBACK
grebenik_anzhelika=> █
```

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

grebenik_anzhelika=> start transaction;
START TRANSACTION
grebenik_anzhelika=> select * from auto where a_id=1;
 a_id | name | year
-----+-----+-----
 1 | BMW 5 | 2005
(1 row)

grebenik_anzhelika=> lock table auto in exclusive mode;
LOCK TABLE
grebenik_anzhelika=> update auto set year=2006 where a_id=1;
UPDATE 1
grebenik_anzhelika=> select * from auto where a_id=1;
 a_id | name | year
-----+-----+-----
 1 | BMW 5 | 2006
(1 row)

grebenik_anzhelika=> commit;
COMMIT
grebenik_anzhelika=> █
```

У другому терміналі ми потрапляємо в тупик використовуючи блокування, робота відновлюється після команди rollback в першому терміналі.

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

```
grebenik_anzhelika@vpsj3leQ:~
START TRANSACTION
grebenik_anzhelika=> lock table auto in exclusive mode;
LOCK TABLE
grebenik_anzhelika=> start transaction;
WARNING: there is already a transaction in progress
START TRANSACTION
grebenik_anzhelika=> lock table auto in share mode;
LOCK TABLE
grebenik_anzhelika=> rollback;
ROLLBACK
grebenik_anzhelika=> start transaction;
START TRANSACTION
grebenik_anzhelika=> lock table auto in share mode;
LOCK TABLE
grebenik_anzhelika=> select * from auto where a_id=1;
 a_id | name | year
-----+-----+-----
 1 | BMW 5 | 2008
(1 row)

grebenik_anzhelika=> lock table auto in exclusive mode;
LOCK TABLE
grebenik_anzhelika=> █
```

```
grebenik_anzhelika@vpsj3leQ:~
grebenik_anzhelika@91.219.60.189's password:
Last login: Wed Apr 28 00:38:04 2021 from 46.149.53.142
[grebenik_anzhelika@vpsj3leQ ~]$ psql grebenik_anzhelika
psql (9.5.25)
Type "help" for help.

grebenik_anzhelika=> start transaction;
START TRANSACTION
grebenik_anzhelika=> lock table auto in share mode;
LOCK TABLE
grebenik_anzhelika=> select * from auto where a_id=1;
 a_id | name | year
-----+-----+-----
 1 | BMW 5 | 2008
(1 row)

grebenik_anzhelika=> lock table auto in exclusive mode;
ERROR: deadlock detected
DETAIL: Process 27962 waits for ExclusiveLock on relation 16729 of database 16438; blocked by process 22818.
Process 22818 waits for ExclusiveLock on relation 16729 of database 16438; blocked by process 27962.
HINT: See server log for query details.
grebenik_anzhelika=> █
```

Перший термінал потрапляє в очікування після exclusive mode, а другий термінал видає deadlock, який в цей час виводить перший з свого стану.

```
grebenik_anzhelika@vpsj3leQ:~
START TRANSACTION
grebenik_anzhelika=> lock table auto in share mode;
LOCK TABLE
grebenik_anzhelika=> select * from auto where a_id=1;
 a_id | name | year
-----+-----+-----
 1 | BMW 5 | 2008
(1 row)

grebenik_anzhelika=> lock table auto in exclusive mode;
LOCK TABLE
grebenik_anzhelika=> update auto set year=2010 where a_id=1;
UPDATE 1
grebenik_anzhelika=> lock table auto in share mode;
LOCK TABLE
grebenik_anzhelika=> select * from auto where a_id=1;
 a_id | name | year
-----+-----+-----
 1 | BMW 5 | 2010
(1 row)

grebenik_anzhelika=> commit;
COMMIT
grebenik_anzhelika=> █
```

```
grebenik_anzhelika@vpsj3leQ:~
ROLLBACK
grebenik_anzhelika=> start transaction;
START TRANSACTION
grebenik_anzhelika=> lock table auto in share mode;
LOCK TABLE
grebenik_anzhelika=> select * from auto where a_id=1;
 a_id | name | year
-----+-----+-----
 1 | BMW 5 | 2008
(1 row)

grebenik_anzhelika=> lock table auto in exclusive mode;
ERROR: deadlock detected
DETAIL: Process 27962 waits for ExclusiveLock on relation 16729 of database 16438; blocked by process 22818.
Process 22818 waits for ExclusiveLock on relation 16729 of database 16438; blocked by process 27962.
HINT: See server log for query details.
grebenik_anzhelika=> update auto set year=2010 where a_id=1;
ERROR: current transaction is aborted, commands ignored until end of transaction block
grebenik_anzhelika=> lock table auto in share mode;
ERROR: current transaction is aborted, commands ignored until end of transaction block
```

```
grebenik_anzhelika@vpsj3leQ:~  
START TRANSACTION  
grebenik_anzhelika=> lock table auto in share mode;  
LOCK TABLE  
grebenik_anzhelika=> select * from auto where a_id=1;  
a_id | name | year  
-----+-----+-----  
1 | BMW 5 | 2010  
(1 row)  
  
grebenik_anzhelika=> lock table auto in exclusive mode;  
LOCK TABLE  
grebenik_anzhelika=> update auto set year=2011 where a_id=1;  
UPDATE 1  
grebenik_anzhelika=> lock table auto in share mode;  
LOCK TABLE  
grebenik_anzhelika=> select * from auto where a_id=1;  
a_id | name | year  
-----+-----+-----  
1 | BMW 5 | 2011  
(1 row)  
  
grebenik_anzhelika=> rollback;  
ROLLBACK  
grebenik_anzhelika=>  
  
grebenik_anzhelika@vpsj3leQ:~  
n block  
grebenik_anzhelika=> select * from auto where a_id=1;  
ERROR: current transaction is aborted, commands ignored until end of transaction  
n block  
grebenik_anzhelika=> rollback;  
ROLLBACK  
grebenik_anzhelika=> start transaction;  
START TRANSACTION  
grebenik_anzhelika=> lock table auto in share mode;  
LOCK TABLE  
grebenik_anzhelika=> select * from auto where a_id=1;  
a_id | name | year  
-----+-----+-----  
1 | BMW 5 | 2010  
(1 row)  
  
grebenik_anzhelika=> lock table auto in exclusive mode;  
ERROR: deadlock detected  
DETAIL: Process 27962 waits for ExclusiveLock on relation 16729 of database 16438; blocked by process 22818.  
Process 22818 waits for ExclusiveLock on relation 16729 of database 16438; blocked by process 27962.  
HINT: See server log for query details.  
grebenik_anzhelika=>
```

Висновок: під час лабораторної роботи ми дослідили поведінку процесів-транзакцій в базах даних та засоби керування ними через механізм блокування з використанням сучасних систем керування базами даних. Найскладнішим було перше завдання.