

Міністерство освіти і науки України  
Одеський національний політехнічний університет  
Інститут комп'ютерних систем  
Кафедра інформаційних систем

Лабораторна робота №10  
З дисципліни "Операційні системи"  
Тема : «Керування процесами-транзакціями в базах даних. Частина 2»

Виконав  
Ст.гр. AI-204  
Костецький Б.В.  
Перевірили:  
Блажко О.А.  
Дрозд М.О.

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.** Аналіз роботи багатоверсійного протоколу

В завданні 1 рішення попередньої лабораторної роботи було створено таблицю з декількома рядками.

Підготуйте чотири транзакції за прикладом з рисунку 2:

- T1 – отримання номеру транзакції, внесення нового рядка в таблицю та перегляд вмісту таблиці;
- T2 – постійний перегляд вмісту таблиці
- T3 – видалення рядку з наступною відміною цієї операції;
- T4 – зміна значення однієї з колонок рядка.

В операцію читання рядка таблиці додайте системні колонки `xmin`, `xmax`.

На кожному кроці виконання транзакції переглядайте значення колонок `xmin`, `xmax` та зробіть відповідні висновки.

### **Завдання 2.** Аналіз стану транзакцій на різних рівнях багаторівневого блокування

Виконайте послідовно в двох терміналах наступні комбінації блокувань таблиці:

IX-IS, SIX-IX, SIX-IS. Надайте висновки про сумісність блокувань.

Для кожної комбінації блокувань перед завершенням 1-ї транзакції (яка розпочалася раніше) в додатковому терміналі через команду `psql` отримайте дані про стан транзакцій (таблиця `pg_locks`).

### **Завдання 3.** Керування квазіпаралельним виконанням транзакцій на різних рівнях ізоляції транзакцій

Підготуйте транзакції, які було створено у завданні 3.1 рішення попередньої лабораторної роботи, а саме, створіть дві транзакції, кожна з яких повинна включати такі операції:

- операція читання першого рядку таблиці;
- операція редагування однієї із змінних таблиці в першому рядку;
- повторна операція читання першого рядку таблиці;
- операція фіксації всіх змін.

1.1 Виконайте роботу транзакцій при умові їх роботи на рівні ізоляції READ

COMMITTED. Проаналізуйте реакцію СКБД на операцію UPDATE 2-ї транзакції (яка виконується пізніше) та дайте свої висновки.

1.2 Повторіть роботу транзакцій при умові їх роботи на рівні ізоляції REPEATABLE

READ. Проаналізуйте реакцію СКБД на операцію UPDATE 2-ї транзакції (яка виконується

пізніше) та дайте свої висновки.

1.3 Повторіть роботу транзакцій при умові їх роботи на рівні ізоляції

SERIALIZABLE. Проаналізуйте реакцію СКБД на операцію UPDATE 2-ї транзакції (яка

виконується пізніше) та дайте свої висновки.

**Завдання 4.** Керування квазіпаралельним виконанням транзакцій при наявності тупикових ситуацій.

1.1 Виконайте модифікацію транзакцій так, щоб вони призводили до тупикової ситуації.

1.2 Виконайте дві модифіковані транзакції.

Проаналізуйте реакцію СКБД на операцію UPDATE 2-ї транзакції (яка виконується пізніше) та яка призвела до тупику. Дайте свої висновки з урахуванням:

- ідентифікаторів процесів
- номерів транзакцій.

## Хід роботи:

### Завдання 1

- T1 – отримання номеру транзакції, внесення нового рядка в таблицю та перегляд вмісту таблиці;
- T2 – постійний перегляд вмісту таблиці

```
kostetskij_bogdan@vpsj3leQ:~  
Last login: Sun Apr 25 21:58:00 2021 from 213.231.23.64.pool.breezei  
[kostetskij_bogdan@vpsj3leQ ~]$ psql  
psql (9.5.25)  
Type "help" for help.  
  
kostetskij_bogdan=> START TRANSACTION;  
START TRANSACTION  
kostetskij_bogdan=> INSERT INTO department VALUES (1, 'SPO', 'IKS');  
INSERT 0 1  
kostetskij_bogdan=> SELECT xmin, xmax, faculty FROM department;  
      xmin | xmax | faculty  
-----+-----+-----  
 2123 |    0 | IMI  
 2132 | 2135 | IMI  
 3440 |    0 | IKS  
(3 rows)  
  
kostetskij_bogdan=> COMMIT;  
COMMIT  
kostetskij_bogdan=>   
  
kostetskij_bogdan@vpsj3leQ:~  
2123 |    0 | IMI  
2132 | 2135 | IMI  
(2 rows)  
  
kostetskij_bogdan=> SELECT xmin, xmax, faculty FROM department;  
      xmin | xmax | faculty  
-----+-----+-----  
 2123 |    0 | IMI  
 2132 | 2135 | IMI  
(2 rows)  
  
kostetskij_bogdan=> SELECT xmin, xmax, faculty FROM department;  
      xmin | xmax | faculty  
-----+-----+-----  
 2123 |    0 | IMI  
 2132 | 2135 | IMI  
 3440 |    0 | IKS  
(3 rows)  
  
kostetskij_bogdan=>   

```

У T2 нема наявного 3-го рядка , так як T1 не було завершено. Після завершення T1 у T2 рядко відображався під номером 3440.

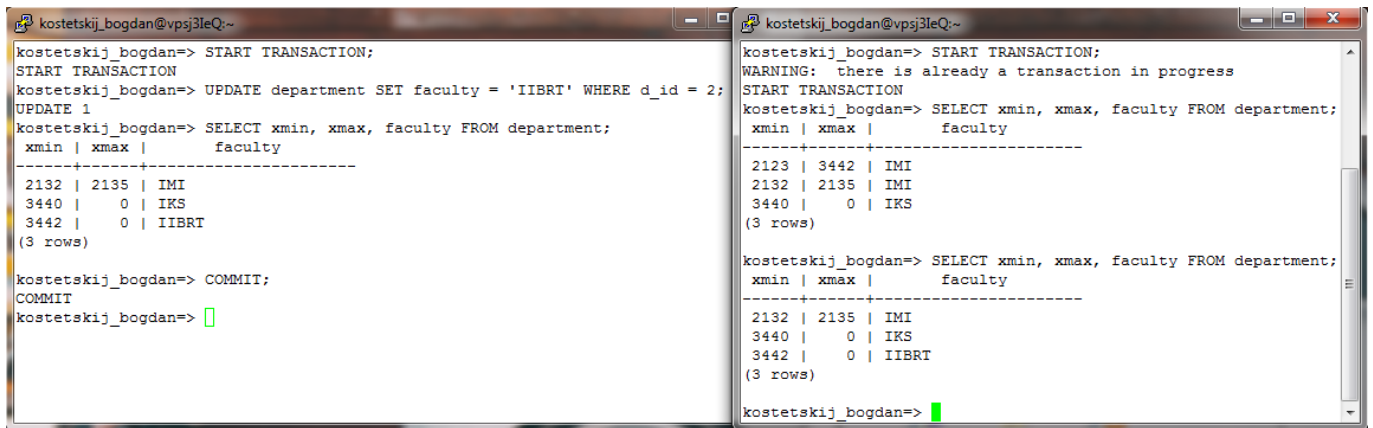
- T3 – видалення рядку з наступною відміною цієї операції;

```
kostetskij_bogdan@vpsj3leQ:~  
kostetskij_bogdan=> START TRANSACTION;  
WARNING: there is already a transaction in progress  
START TRANSACTION  
kostetskij_bogdan=> COMMIT;  
COMMIT  
kostetskij_bogdan=> START TRANSACTION;  
START TRANSACTION  
kostetskij_bogdan=> DELETE FROM department WHERE d_id = 2;  
DELETE 1  
kostetskij_bogdan=> SELECT xmin, xmax, faculty FROM department;  
      xmin | xmax | faculty  
-----+-----+-----  
 2132 | 2135 | IMI  
 3440 |    0 | IKS  
(2 rows)  
  
kostetskij_bogdan=> ABORT;  
ROLLBACK  
kostetskij_bogdan=>   
  
kostetskij_bogdan@vpsj3leQ:~  
kostetskij_bogdan=> SELECT xmin, xmax, faculty FROM department;  
      xmin | xmax | faculty  
-----+-----+-----  
 2123 | 3441 | IMI  
 2132 | 2135 | IMI  
 3440 |    0 | IKS  
(3 rows)  
  
kostetskij_bogdan=>   

```

Видалення рядка відбулося у T3, але T2 все одно відображав цей рядок під номером 3441.

– T4 – зміна значення однієї з колонок рядка.



```
kostetskij_bogdan@vpsj3leQ:~  
kostetskij_bogdan=> START TRANSACTION;  
START TRANSACTION  
kostetskij_bogdan=> UPDATE department SET faculty = 'IIBRT' WHERE d_id = 2;  
UPDATE 1  
kostetskij_bogdan=> SELECT xmin, xmax, faculty FROM department;  
xmin | xmax | faculty  
-----  
2132 | 2135 | IMI  
3440 | 0 | IKS  
3442 | 0 | IIBRT  
(3 rows)  
kostetskij_bogdan=> COMMIT;  
COMMIT  
kostetskij_bogdan=> █
```

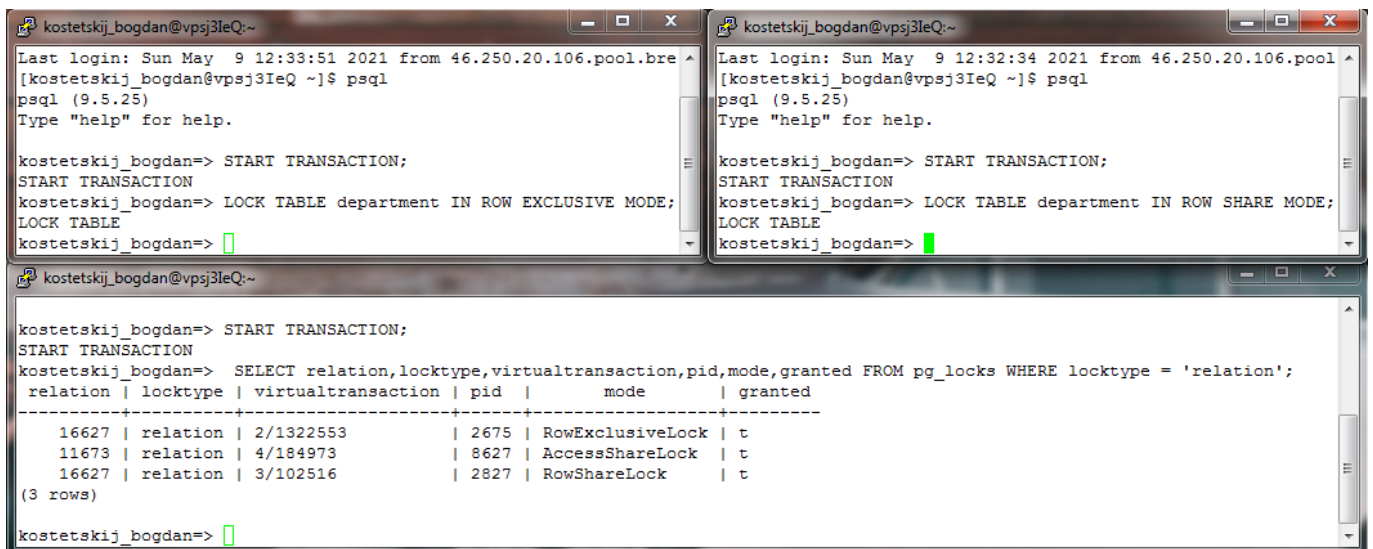
```
kostetskij_bogdan@vpsj3leQ:~  
kostetskij_bogdan=> START TRANSACTION;  
WARNING: there is already a transaction in progress  
START TRANSACTION  
kostetskij_bogdan=> SELECT xmin, xmax, faculty FROM department;  
xmin | xmax | faculty  
-----  
2123 | 3442 | IMI  
2132 | 2135 | IMI  
3440 | 0 | IKS  
(3 rows)  
kostetskij_bogdan=> SELECT xmin, xmax, faculty FROM department;  
xmin | xmax | faculty  
-----  
2132 | 2135 | IMI  
3440 | 0 | IKS  
3442 | 0 | IIBRT  
(3 rows)  
kostetskij_bogdan=> █
```

Під час першого считування у T2 відображається лише зміна номеру транзакції – 2132 . Після завершення T4 у T2 можна вже побачити зміну із номером транзакції 3442.

## Завдання 2

Виконали послідовно в двох терміналах наступні комбінації блокувань таблиці: IX-IS, SIX-IX, SIX-IS. Надали висновки про сумісність блокувань.

### IX-IS



```
kostetskij_bogdan@vpsj3leQ:~  
Last login: Sun May 9 12:33:51 2021 from 46.250.20.106.pool.bre  
[kostetskij_bogdan@vpsj3leQ ~]$ psql  
psql (9.5.25)  
Type "help" for help.  
kostetskij_bogdan=> START TRANSACTION;  
START TRANSACTION  
kostetskij_bogdan=> LOCK TABLE department IN ROW EXCLUSIVE MODE;  
LOCK TABLE  
kostetskij_bogdan=> █
```

```
kostetskij_bogdan@vpsj3leQ:~  
Last login: Sun May 9 12:32:34 2021 from 46.250.20.106.pool  
[kostetskij_bogdan@vpsj3leQ ~]$ psql  
psql (9.5.25)  
Type "help" for help.  
kostetskij_bogdan=> START TRANSACTION;  
START TRANSACTION  
kostetskij_bogdan=> LOCK TABLE department IN ROW SHARE MODE;  
LOCK TABLE  
kostetskij_bogdan=> █
```

```
kostetskij_bogdan@vpsj3leQ:~  
kostetskij_bogdan=> START TRANSACTION;  
START TRANSACTION  
kostetskij_bogdan=> SELECT relation,locktype,virtualtransaction,pid,mode,granted FROM pg_locks WHERE locktype = 'relation';  
relation | locktype | virtualtransaction | pid | mode | granted  
-----  
16627 | relation | 2/1322553 | 2675 | RowExclusiveLock | t  
11673 | relation | 4/184973 | 8627 | AccessShareLock | t  
16627 | relation | 3/102516 | 2827 | RowShareLock | t  
(3 rows)  
kostetskij_bogdan=> █
```

Після виконання даних блокувань можна визначити , що тип блокування IX та IS сумісні.

## SIX-IX

```
kostetskij_bogdan@vpsj3leQ:~  
kostetskij_bogdan=> START TRANSACTION;  
START TRANSACTION  
kostetskij_bogdan=> LOCK TABLE department IN SHARE ROW EXCLUSIVE MODE;  
LOCK TABLE  
kostetskij_bogdan=> █
```

```
kostetskij_bogdan@vpsj3leQ:~  
Type "help" for help.  
kostetskij_bogdan=> START TRANSACTION;  
START TRANSACTION  
kostetskij_bogdan=> LOCK TABLE department IN ROW EXCLUSIVE MODE;  
█
```

```
kostetskij_bogdan@vpsj3leQ:~  
kostetskij_bogdan=> START TRANSACTION;  
START TRANSACTION  
kostetskij_bogdan=> SELECT relation,locktype,virtualtransaction,pid,mode,granted FROM pg_locks WHERE locktype = 'relation';  
relation | locktype | virtualtransaction | pid | mode | granted  
-----+-----+-----+-----+-----+-----  
11673 | relation | 4/185127 | 15016 | AccessShareLock | t  
16627 | relation | 3/102636 | 14809 | RowExclusiveLock | f  
16627 | relation | 2/1345056 | 14687 | ShareRowExclusiveLock | t  
(3 rows)  
kostetskij_bogdan=> █
```

Звертаючи увагу на таблицю з даними , можна визначити що блокування IX знаходиться у стані очікування через вже виконане блокування SIX. Звідси можна зробити висновок , що дані блокування не сумісні.

## SIX-IS

```
kostetskij_bogdan@vpsj3leQ:~  
kostetskij_bogdan=> START TRANSACTION;  
START TRANSACTION  
kostetskij_bogdan=> LOCK TABLE department IN SHARE ROW EXCLUSIVE MODE;  
LOCK TABLE  
kostetskij_bogdan=> █
```

```
kostetskij_bogdan@vpsj3leQ:~  
kostetskij_bogdan=> START TRANSACTION;  
START TRANSACTION  
kostetskij_bogdan=> LOCK TABLE department IN ROW SHARE MODE;  
LOCK TABLE  
kostetskij_bogdan=> █
```

```
kostetskij_bogdan@vpsj3leQ:~  
kostetskij_bogdan=> SELECT relation,locktype,virtualtransaction,pid,mode,granted FROM pg_locks WHERE locktype = 'relation';  
relation | locktype | virtualtransaction | pid | mode | granted  
-----+-----+-----+-----+-----+-----  
11673 | relation | 4/185299 | 16317 | AccessShareLock | t  
16627 | relation | 3/102846 | 16077 | RowShareLock | t  
16627 | relation | 2/1345236 | 15846 | ShareRowExclusiveLock | t  
(3 rows)  
kostetskij_bogdan=> █
```

Перша і друга транзакції заблокували таблиці. Дивлячись у таблицю з даними про стан транзакцій можна зрозуміти , що блокування є сумісними.

## Завдання 3

Підготуйте транзакції, які було створено у завданні 3.1 рішення попередньої лабораторної роботи, а саме, створіть дві транзакції, кожна з яких повинна включати такі операції:

- операція читання першого рядку таблиці;
- операція редагування однієї із змінних таблиці в першому рядку;
- повторна операція читання першого рядку таблиці;
- операція фіксації всіх змін.

**1.1** Виконайте роботу транзакцій при умові їх роботи на рівні ізоляції READ COMMITTED. Проаналізуйте реакцію СКБД на операцію UPDATE 2-ї транзакції (яка виконується пізніше) та дайте свої висновки.

Left terminal window:

```
kostetskij_bogdan@vps3leQ:~$
kostetskij_bogdan=> START TRANSACTION;
START TRANSACTION
kostetskij_bogdan=>
kostetskij_bogdan=> SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
SET
kostetskij_bogdan=> SELECT * FROM department;
d_id | name | faculty
-----+-----+-----
1 | DOT | IMI
2 | NKN | IIBRT
3 | SPO | IKS
(3 rows)

kostetskij_bogdan=> UPDATE department SET name = 'BNB' WHERE d_id = 3;
UPDATE 1
kostetskij_bogdan=> SELECT * FROM department;
d_id | name | faculty
-----+-----+-----
1 | DOT | IMI
2 | NKN | IIBRT
3 | BNB | IKS
(3 rows)

kostetskij_bogdan=> COMMIT;
COMMIT
kostetskij_bogdan=>
```

Right terminal window:

```
Type "help" for help.

kostetskij_bogdan=> START TRANSACTION;
START TRANSACTION
kostetskij_bogdan=> SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
SET
kostetskij_bogdan=> SELECT * FROM department;
d_id | name | faculty
-----+-----+-----
1 | DOT | IMI
2 | NKN | IIBRT
3 | BNB | IKS
(3 rows)

kostetskij_bogdan=> UPDATE department SET faculty = 'IIIR' WHERE d_id = 3;
UPDATE 1
kostetskij_bogdan=> SELECT * FROM department;
d_id | name | faculty
-----+-----+-----
1 | DOT | IMI
2 | NKN | IIBRT
3 | BNB | IIIR
(3 rows)

kostetskij_bogdan=> COMMIT;
COMMIT
```

Усі дії оновлюються у T2 лише після завершення T1

**1.2** Повторіть роботу транзакцій при умові їх роботи на рівні ізоляції REPEATABLE READ. Проаналізуйте реакцію СКБД на операцію UPDATE 2-ї транзакції (яка виконується пізніше) та дайте свої висновки.

Left terminal window:

```
kostetskij_bogdan@vps3leQ:~$
Type "help" for help.

kostetskij_bogdan=> START TRANSACTION;
START TRANSACTION
kostetskij_bogdan=> SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
SET
kostetskij_bogdan=> SELECT * FROM department;
d_id | name | faculty
-----+-----+-----
2 | EOS | IKS
1 | DOT | IMI
3 | KBV | IIIR
(3 rows)

kostetskij_bogdan=> UPDATE department SET name = 'KSM' WHERE d_id = 3;
UPDATE 1
kostetskij_bogdan=> SELECT * FROM department;
d_id | name | faculty
-----+-----+-----
2 | EOS | IKS
1 | DOT | IMI
3 | KSM | IIIR
(3 rows)

kostetskij_bogdan=> COMMIT;
```

Right terminal window:

```
kostetskij_bogdan@vps3leQ:~$
START TRANSACTION
kostetskij_bogdan=> SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
SET
kostetskij_bogdan=> SELECT * FROM department;
d_id | name | faculty
-----+-----+-----
2 | EOS | IKS
1 | DOT | IMI
3 | KBV | IIIR
(3 rows)

kostetskij_bogdan=> UPDATE department SET faculty = 'IMBT' WHERE d_id = 3;

ERROR: could not serialize access due to concurrent update
kostetskij_bogdan=>
kostetskij_bogdan=>
kostetskij_bogdan=>
kostetskij_bogdan=> SELECT * FROM department;
ERROR: current transaction is aborted, commands ignored until end of transaction block
kostetskij_bogdan=> ABORT;
```

Як і у попередньому випадку T2 оновлюється лише після завершення T1, але після цього завершення виникає помилка та блокується транзакція.

**1.3** Повторіть роботу транзакцій при умові їх роботи на рівні ізоляції SERIALIZABLE. Проаналізуйте реакцію СКБД на операцію UPDATE 2-ї транзакції (яка виконується пізніше) та дайте свої висновки.

Left terminal window:

```
kostetskij_bogdan@vps3leQ:~$
kostetskij_bogdan=> START TRANSACTION;
START TRANSACTION
kostetskij_bogdan=> SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
SET
kostetskij_bogdan=> SELECT * FROM department;
d_id | name | faculty
-----+-----+-----
2 | EOS | IKS
1 | DOT | IMI
3 | KSM | IIIR
(3 rows)

kostetskij_bogdan=> UPDATE department SET name = 'NKN' WHERE d_id = 2;
UPDATE 1
kostetskij_bogdan=> SELECT * FROM department;
d_id | name | faculty
-----+-----+-----
1 | DOT | IMI
3 | KSM | IIIR
2 | NKN | IKS
(3 rows)

kostetskij_bogdan=> COMMIT;
COMMIT
kostetskij_bogdan=>
```

Right terminal window:

```
kostetskij_bogdan@vps3leQ:~$
kostetskij_bogdan=> START TRANSACTION;
START TRANSACTION
kostetskij_bogdan=> SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
SET
kostetskij_bogdan=> SELECT * FROM department;
d_id | name | faculty
-----+-----+-----
2 | EOS | IKS
1 | DOT | IMI
3 | KSM | IIIR
(3 rows)

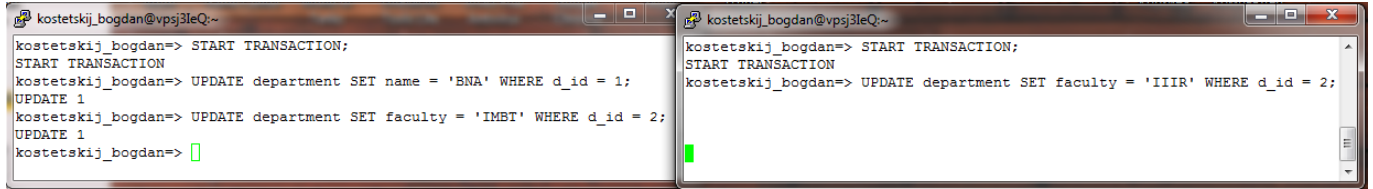
kostetskij_bogdan=> UPDATE department SET faculty = 'IMBT' WHERE d_id = 2;
ERROR: could not serialize access due to concurrent update
kostetskij_bogdan=> SELECT * FROM department;
ERROR: current transaction is aborted, commands ignored until end of transaction block
kostetskij_bogdan=> ABORT;
ROLLBACK
kostetskij_bogdan=>
```

На рівні ізоляції SERIALIZABLE не вдається виконати паралельні зміни.

## Завдання 4

1.1 Виконали модифікацію транзакцій так, щоб вони призводили до тупикової ситуації.

1.2 Виконайте дві модифіковані транзакції.

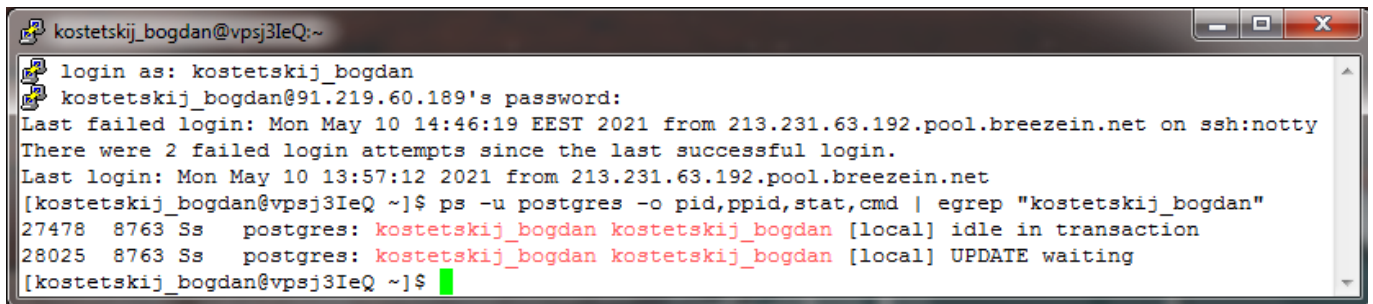


```
kostetskij_bogdan@vpsj3IeQ:~  
kostetskij_bogdan=> START TRANSACTION;  
START TRANSACTION  
kostetskij_bogdan=> UPDATE department SET name = 'BNA' WHERE d_id = 1;  
UPDATE 1  
kostetskij_bogdan=> UPDATE department SET faculty = 'IMBT' WHERE d_id = 2;  
UPDATE 1  
kostetskij_bogdan=> █  
  
kostetskij_bogdan@vpsj3IeQ:~  
kostetskij_bogdan=> START TRANSACTION;  
START TRANSACTION  
kostetskij_bogdan=> UPDATE department SET faculty = 'IIIR' WHERE d_id = 2;  
█
```

Проаналізували реакцію СКБД на операцію UPDATE 2-ї транзакції (яка виконується пізніше) та яка призвела до тупику. Дайте свої висновки з урахуванням:

- ідентифікаторів процесів
- номерів транзакцій.

Аналізуючи результати роботи переглянули таблицю процесів.



```
kostetskij_bogdan@vpsj3IeQ:~  
login as: kostetskij_bogdan  
kostetskij_bogdan@91.219.60.189's password:  
Last failed login: Mon May 10 14:46:19 EEST 2021 from 213.231.63.192.pool.breezein.net on ssh:notty  
There were 2 failed login attempts since the last successful login.  
Last login: Mon May 10 13:57:12 2021 from 213.231.63.192.pool.breezein.net  
[kostetskij_bogdan@vpsj3IeQ ~]$ ps -u postgres -o pid,ppid,stat,cmd | egrep "kostetskij_bogdan"  
27478  8763 Ss   postgres: kostetskij_bogdan kostetskij_bogdan [local] idle in transaction  
28025  8763 Ss   postgres: kostetskij_bogdan kostetskij_bogdan [local] UPDATE waiting  
[kostetskij_bogdan@vpsj3IeQ ~]$ █
```

Висновок: під час виконання лабораторної роботи було досліджено поведінку процесів-транзакцій в базах даних та засоби керування ними через механізм блокування з використанням сучасних систем керування базами даних.