МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ “ОДЕСЬКА ПОЛІТЕХНІКА”
Інститут комп'ютерних систем
Кафедра інформаційних систем

Лабораторна робота №10

з дисципліни “Операційні системи”.
Тема: «Керування процесами-транзакціями в базах даних. Частина 2».

Виконала:
студентка групи АІ-204
Томчук Вікторія Борисівна

Перевірив:
Блажко О.А.

Одеса-2021

Мета роботи: дослідити поведінку процесів-транзакцій в базах даних та засоби керуванням ними через механізм блокування з використанням сучасних систем керування базами даних.

2 Завдання

Завдання 1. Аналіз роботи багато версійного протоколу
В завданні 1 рішення попередньої лабораторної роботи було створено таблицю з декількома рядками.

− 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 данні про стан транзакцій (таблиця pg_locs).

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.

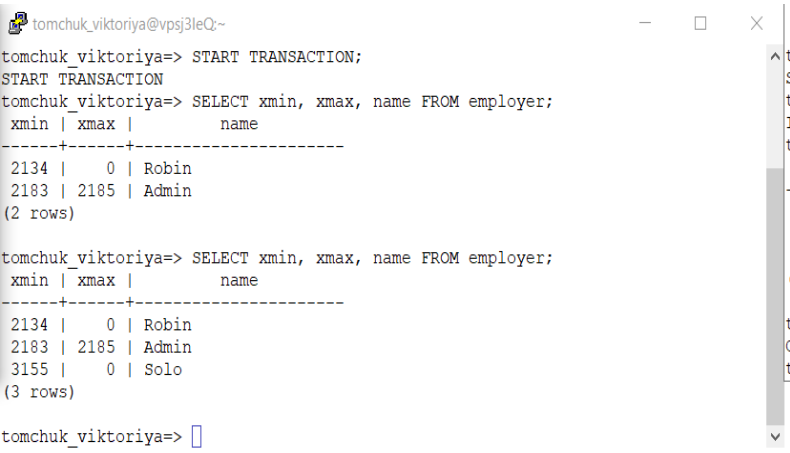
1.1 Виконайте роботу транзакцій при умові їх роботи на рівні ізоляції READ COMMITED. Проаналізуйте реакцію СКБД на операцію UPDATE 2-ї транзакції (яка виконується пізніше) та дайте свої висновки.
1.2 Повторіть роботу транзакцій при умові їх роботи на рівні ізоляції REPEATABLE READ. Проаналізуйте реакцію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

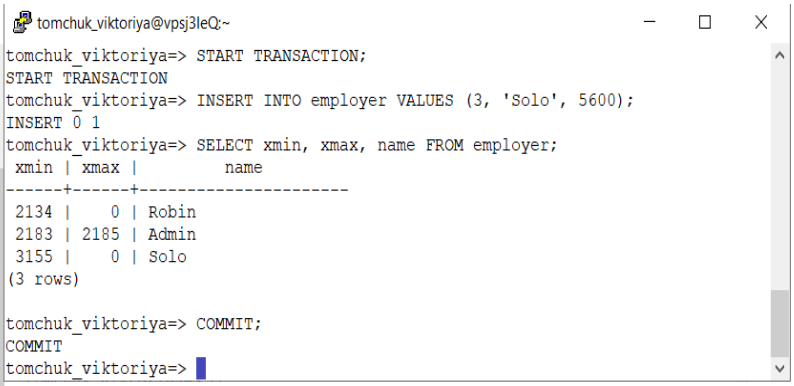
Завдання 4. Керування квазіпаралельним виконанням транзакцій при наявності тупикових ситуацій.
3.1 Виконайте модифікацію транзакцій так, щоб вони призводили до тупикової ситуації.
3.2 Виконайте дві модифіковані транзакції.
Проаналізуйте реакцію СКБД на операцію UPDATE 2-ї транзакції (яка виконується пізніше) та яка призвела до тупику. Дайте свої висновки з урахуванням:
− ідентифікаторів процесів
− номерів транзакцій.

Завдання 1.

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

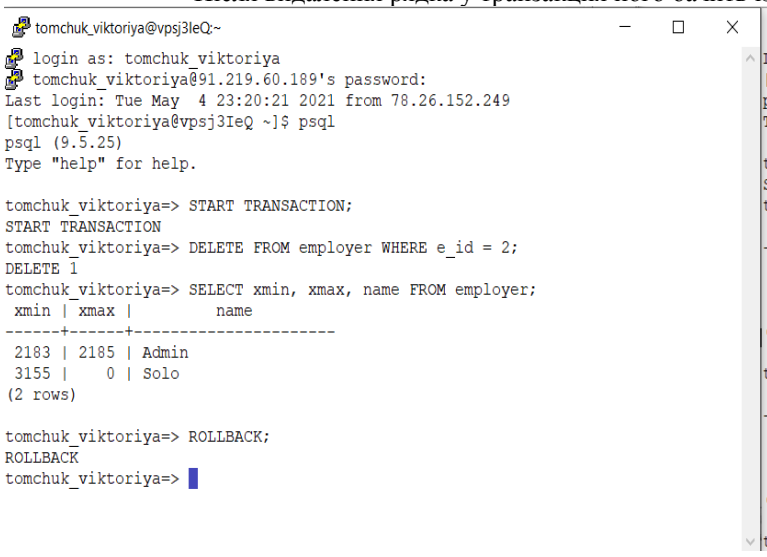
В даних транзакціях можна помітити, як xmax має відмінне від інших значення. До цього була робота з даними рядками, і збирається видалити рядок 2185. Спочатку не бачить рядка, тому що T1 не було завершено, після успішного завершення T1, у T2 починає відображатись створений рядок.

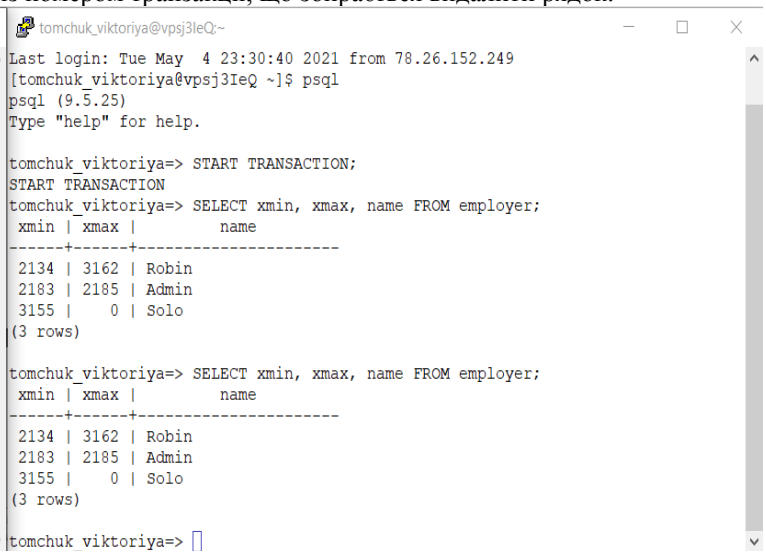




– T3 – видалення рядку з наступною відміною цієї операції;

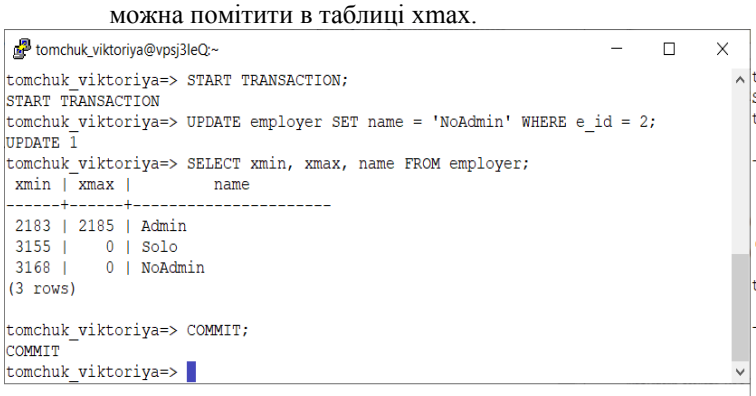
Після видалення рядка у транзакция його бачить із номером транзакції, що збирається видалити рядок.

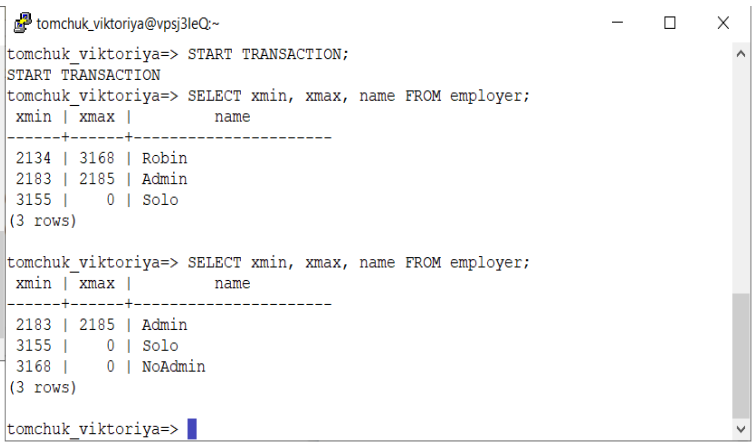




– T4 – зміна значення однієї з колонок рядка.

Після прочитання, до наступної транзакції приходить інформація про зміни рядка з транзакцією, що можна помітити в таблиці xmax.

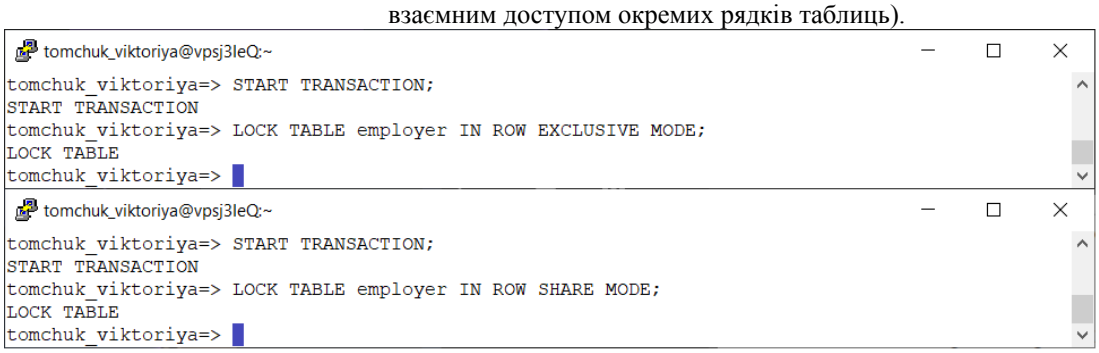


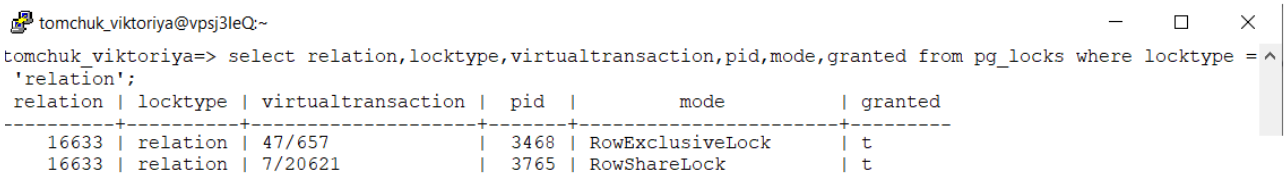


Завдання 2. *Виконайте послідовно в двох терміналах наступні комбінації блокувань таблиці: IX-IS, SIX-IX, SIX-IS. Надайте висновки про сумісність блокувань.*

IX-IS:

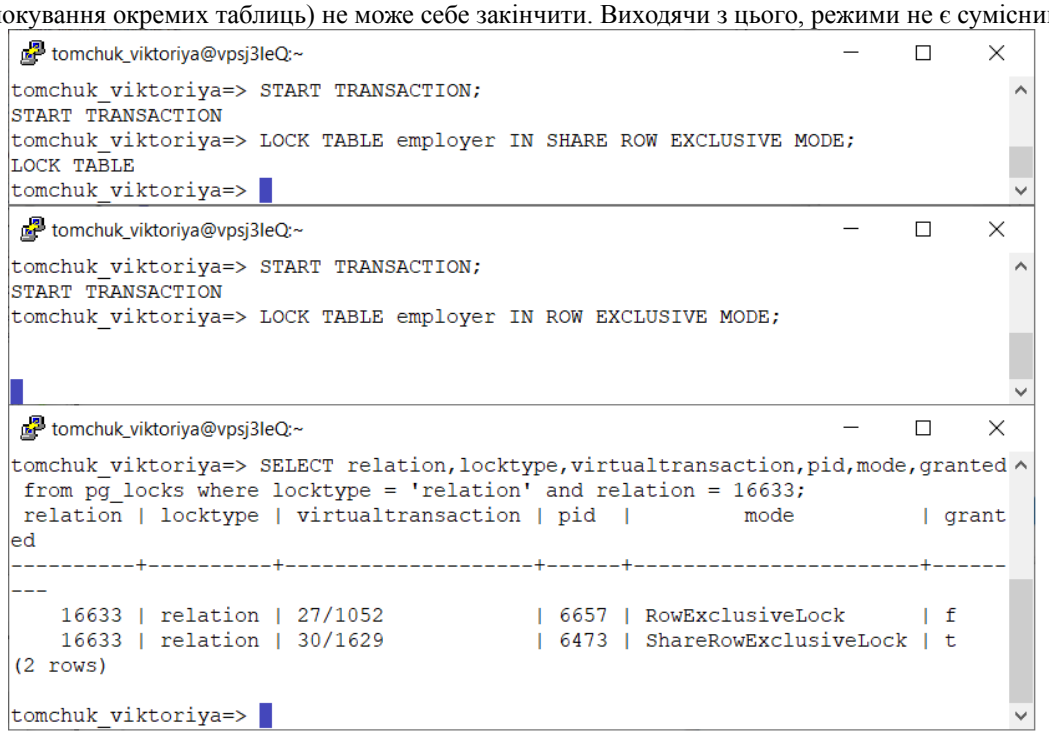
Як можна побачити, режим IX (блокування окремих таблиць) сумісний з режимом IS (блокування з взаємним доступом окремих рядків таблиць).





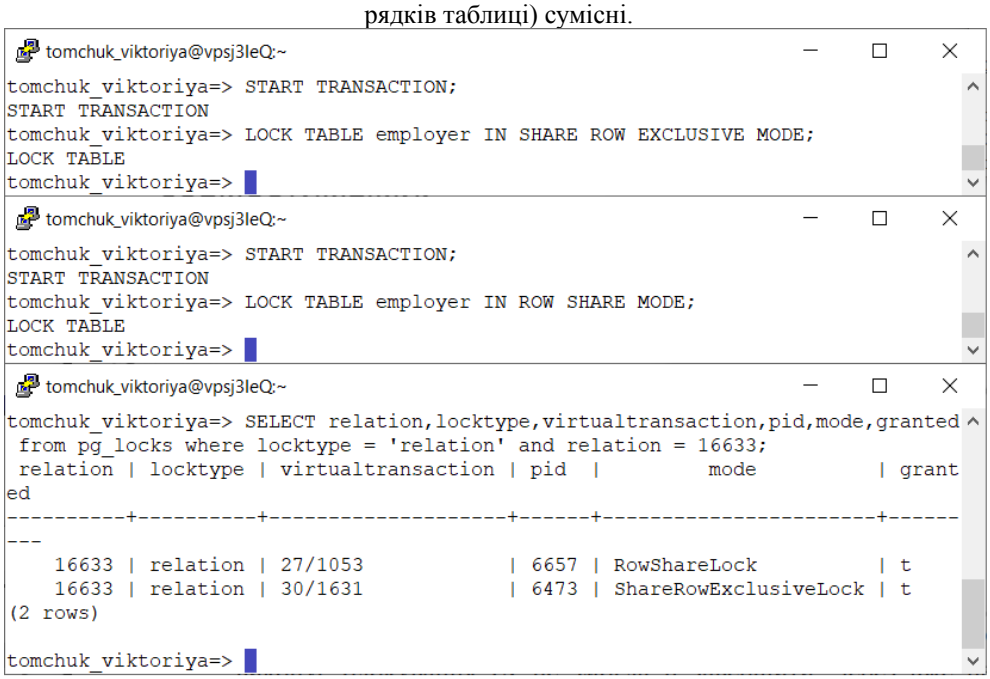
SIX-IX:

Як можна помітити, через вплив блокування SIX (блокування зі взаємним доступом блокування всіх рядків таблиці та монопольне блокування окремих рядків таблиці) на першій транзакції, друга транзакція з IX (блокування окремих таблиць) не може себе закінчити. Виходячи з цього, режими не є сумісними.



SIX-IS:

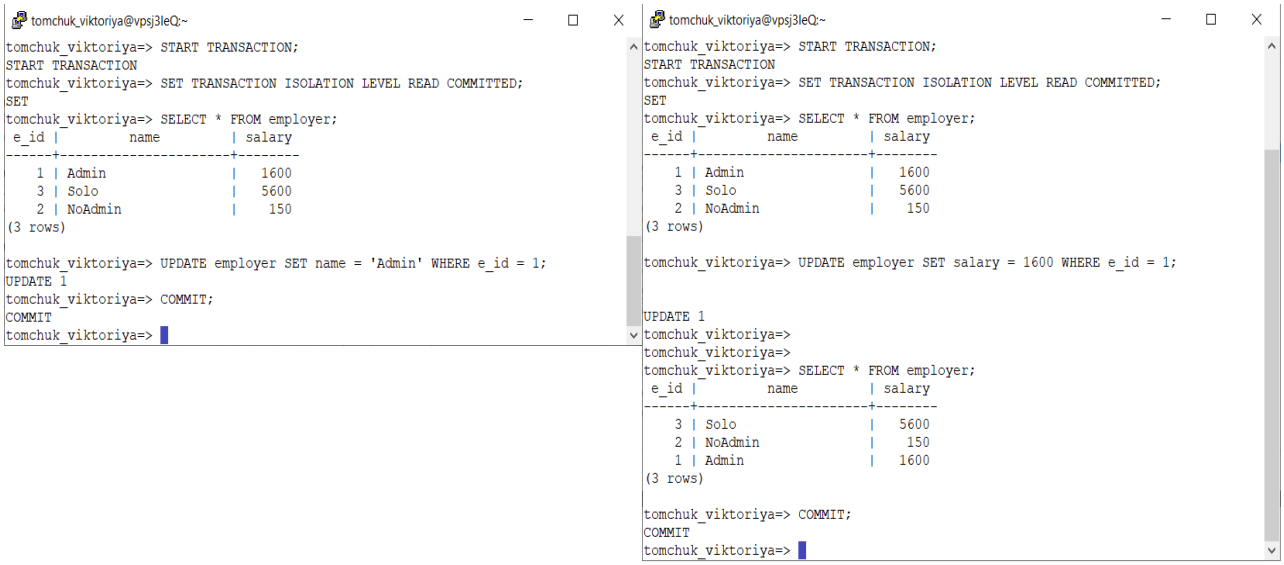
Транзакції вдало заблокували таблиці. SIX (блокування зі взаємним доступом блокування всіх рядків таблиці та монопольне блокування окремих рядків таблиці) – IS (блокування зі взаємним доступом окремих рядків таблиці) сумісні.



Завдання 3. *Підготовлюють транзакції, які було створено у завданні 3.1 рішення попередньої лабораторної роботи, а саме, створюють дві транзакції, кожна з яких повинна включати операції:*

Транзакція 1	Транзакція 2
START TRANSACTION	START TRANSACTION
SELECT * FROM employer WHERE e_id = 1;	SELECT * FROM employer WHERE e_id = 1;
UPDATE employer SET name = 'Admin' WHERE e_id = 1;	UPDATE employer SET salary = 1600 WHERE e_id = 1;
SELECT * FROM employer WHERE e_id = 1;	SELECT * FROM employer WHERE e_id = 1;
COMMIT;	COMMIT;

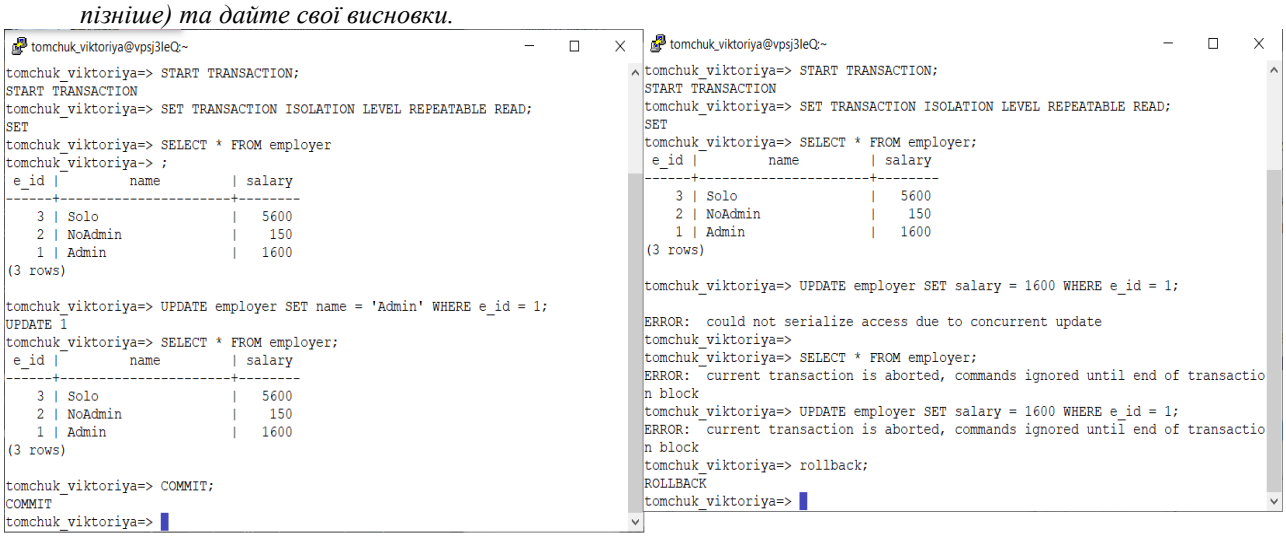
Завдання 3.1. *Виконайте роботу транзакцій при умові їх роботи на рівні ізоляції READ COMMITED. Проаналізуйте реакцію СКБД на операцію UPDATE 2-ї транзакції (яка виконується пізніше) та дайте свої висновки.*



```
tomchuk_viktoriya@vpsj3leQ:~
tomchuk_viktoriya=> START TRANSACTION;
START TRANSACTION
tomchuk_viktoriya=> SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
SET
tomchuk_viktoriya=> SELECT * FROM employer;
 e_id |         name         | salary
------+----------------------+--------
    1 | Admin                |   1600
    3 | Solo                 |   5600
    2 | NoAdmin              |    150
(3 rows)

tomchuk_viktoriya=> UPDATE employer SET name = 'Admin' WHERE e_id = 1;
UPDATE 1
tomchuk_viktoriya=> COMMIT;
COMMIT
tomchuk_viktoriya=>
```

```
tomchuk_viktoriya@vpsj3leQ:~
tomchuk_viktoriya=> START TRANSACTION;
START TRANSACTION
tomchuk_viktoriya=> SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
SET
tomchuk_viktoriya=> SELECT * FROM employer;
 e_id |         name         | salary
------+----------------------+--------
    1 | Admin                |   1600
    3 | Solo                 |   5600
    2 | NoAdmin              |    150
(3 rows)

tomchuk_viktoriya=> UPDATE employer SET salary = 1600 WHERE e_id = 1;


UPDATE 1
tomchuk_viktoriya=>
tomchuk_viktoriya=>
tomchuk_viktoriya=> SELECT * FROM employer;
 e_id |         name         | salary
------+----------------------+--------
    3 | Solo                 |   5600
    2 | NoAdmin              |    150
    1 | Admin                |   1600
(3 rows)

tomchuk_viktoriya=> COMMIT;
COMMIT
tomchuk_viktoriya=>
```

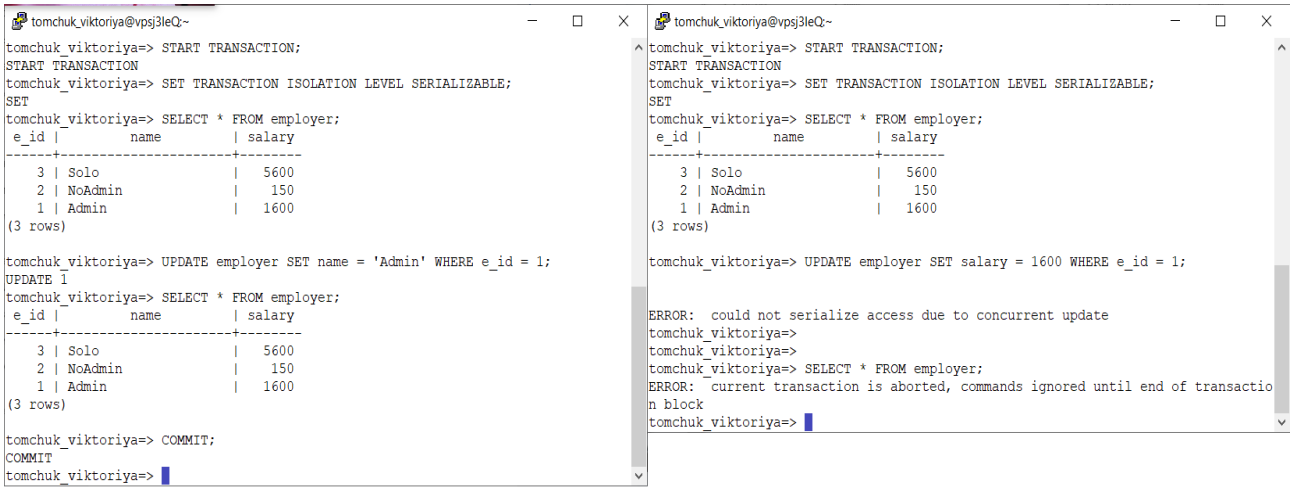
Завдання 3.2. *Повторіть роботу транзакцій при умові їх роботи на рівні ізоляції REPEATABLE READ. Проаналізуйте реакцію СКБД на операцію UPDATE 2-ї транзакції (яка виконується пізніше) та дайте свої висновки.*



```
tomchuk_viktoriya@vpsj3leQ:~
tomchuk_viktoriya=> START TRANSACTION;
START TRANSACTION
tomchuk_viktoriya=> SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
SET
tomchuk_viktoriya=> SELECT * FROM employer
tomchuk_viktoriya-> ;
 e_id |         name         | salary
------+----------------------+--------
    3 | Solo                 |   5600
    2 | NoAdmin              |    150
    1 | Admin                |   1600
(3 rows)

tomchuk_viktoriya=> UPDATE employer SET name = 'Admin' WHERE e_id = 1;
UPDATE 1
tomchuk_viktoriya=> SELECT * FROM employer;
 e_id |         name         | salary
------+----------------------+--------
    3 | Solo                 |   5600
    2 | NoAdmin              |    150
    1 | Admin                |   1600
(3 rows)

tomchuk_viktoriya=> COMMIT;
COMMIT
tomchuk_viktoriya=>
```

```
tomchuk_viktoriya@vpsj3leQ:~
tomchuk_viktoriya=> START TRANSACTION;
START TRANSACTION
tomchuk_viktoriya=> SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
SET
tomchuk_viktoriya=> SELECT * FROM employer;
 e_id |         name         | salary
------+----------------------+--------
    3 | Solo                 |   5600
    2 | NoAdmin              |    150
    1 | Admin                |   1600
(3 rows)

tomchuk_viktoriya=> UPDATE employer SET salary = 1600 WHERE e_id = 1;

ERROR:  could not serialize access due to concurrent update
tomchuk_viktoriya=>
tomchuk_viktoriya=> SELECT * FROM employer;
ERROR:  current transaction is aborted, commands ignored until end of transactio
n block
tomchuk_viktoriya=> UPDATE employer SET salary = 1600 WHERE e_id = 1;
ERROR:  current transaction is aborted, commands ignored until end of transactio
n block
tomchuk_viktoriya=> rollback;
ROLLBACK
tomchuk_viktoriya=>
```

Завдання 3.3. *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.*



```
tomchuk_viktoriya@vpsj3leQ:~
tomchuk_viktoriya=> START TRANSACTION;
START TRANSACTION
tomchuk_viktoriya=> SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
SET
tomchuk_viktoriya=> SELECT * FROM employer;
 e_id |         name         | salary
------+----------------------+--------
    3 | Solo                 |   5600
    2 | NoAdmin              |    150
    1 | Admin                |   1600
(3 rows)

tomchuk_viktoriya=> UPDATE employer SET name = 'Admin' WHERE e_id = 1;
UPDATE 1
tomchuk_viktoriya=> SELECT * FROM employer;
 e_id |         name         | salary
------+----------------------+--------
    3 | Solo                 |   5600
    2 | NoAdmin              |    150
    1 | Admin                |   1600
(3 rows)

tomchuk_viktoriya=> COMMIT;
COMMIT
tomchuk_viktoriya=>
```

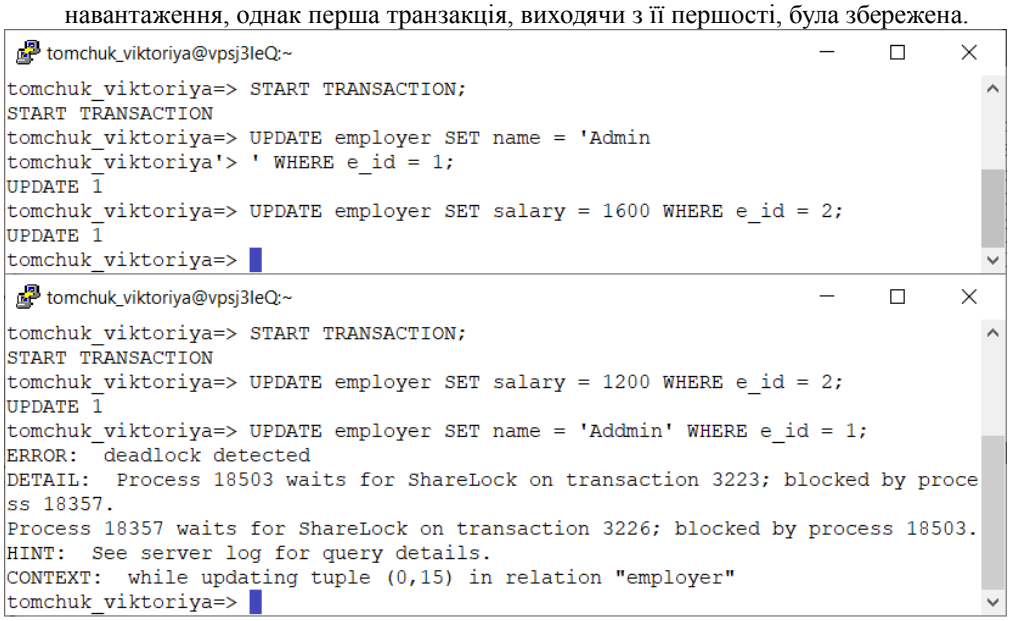
```
tomchuk_viktoriya@vpsj3leQ:~
tomchuk_viktoriya=> START TRANSACTION;
START TRANSACTION
tomchuk_viktoriya=> SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
SET
tomchuk_viktoriya=> SELECT * FROM employer;
 e_id |         name         | salary
------+----------------------+--------
    3 | Solo                 |   5600
    2 | NoAdmin              |    150
    1 | Admin                |   1600
(3 rows)

tomchuk_viktoriya=> UPDATE employer SET salary = 1600 WHERE e_id = 1;


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

Завдання 4.1. *Виконують модифікацію транзакцій так, щоб вони призводили до тупикової ситуації.*

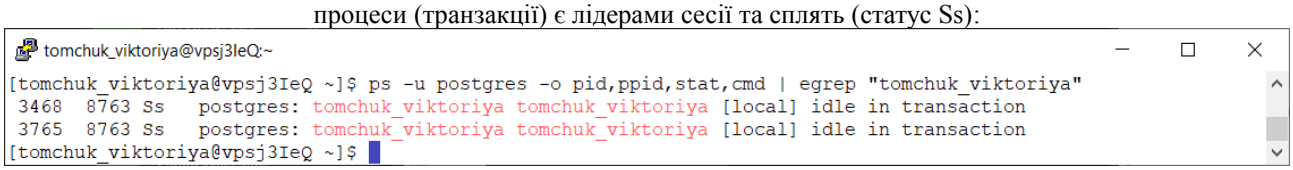
START TRANSACTION;	START TRANSACTION;
UPDATE employer SET name = 'Admin' WHERE e_id = 1;	UPDATE employer SET salary = 1200 WHERE e_id = 2;
UPDATE employer SET salary = 1600 WHERE e_id = 2;	UPDATE employer SET name = 'Addmin' WHERE e_id = 1;

Завдання 4.2. *Виконують дві модифіковані транзакції.*

Як можна помітити, виникає в обох транзакціях тупикова ситуація, яка закінчується скиданням навантаження, однак перша транзакція, виходячи з її першості, була збережена.



Для аналізу результатів роботи транзакції додатково переглядають таблицю процесів. Бачимо, що обидва процеси (транзакції) є лідерами сесії та сплять (статус Ss):



ВИСНОВОК: Вдалося дослідити поведінку процесів-транзакцій в базах даних та засоби керуванням ними через механізм блокування з використанням сучасних систем керування базами даних. Всі завдання було цікаво розбирати, використовуючи і вивчаючи процес роботи механізму брокіровок транзакцій.