

Лабораторна робота 5

Транзакції

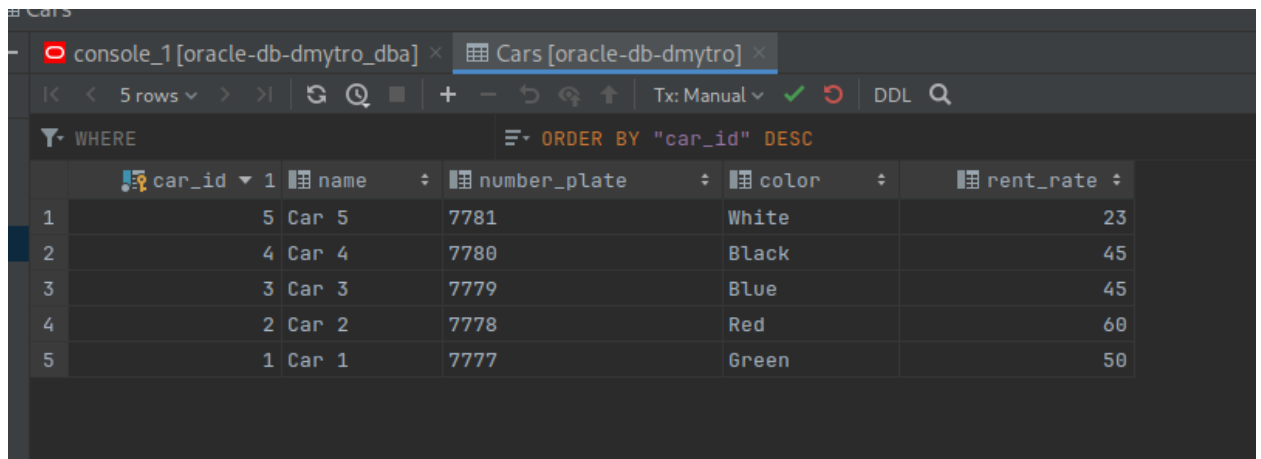
Завдання роботи

Завдання лабораторної роботи виконав на 74%

Завдання 1

1. Создайте таблицы согласно варианту вашей курсовой работы. Заполните одну из них 5 строками данных.

Таблиці до варіанту курсової я створював в одній з попередніх лабораторних робіт, тому просто перевірю наявність даних. Для цього завдання буду використовувати таблицю Cars:



The screenshot shows a database console window with two tabs: 'console_1 [oracle-db-dmytro_dba]' and 'Cars [oracle-db-dmytro]'. The 'Cars' tab is active, displaying a table with 5 rows of data. The table has columns: car_id, name, number_plate, color, and rent_rate. The data is ordered by car_id in descending order.

	car_id	name	number_plate	color	rent_rate
1	5	Car 5	7781	White	23
2	4	Car 4	7780	Black	45
3	3	Car 3	7779	Blue	45
4	2	Car 2	7778	Red	60
5	1	Car 1	7777	Green	50

2. Удалите из таблицы 2 любые строки.

Видалю 4 і 5 айді з таблиці:

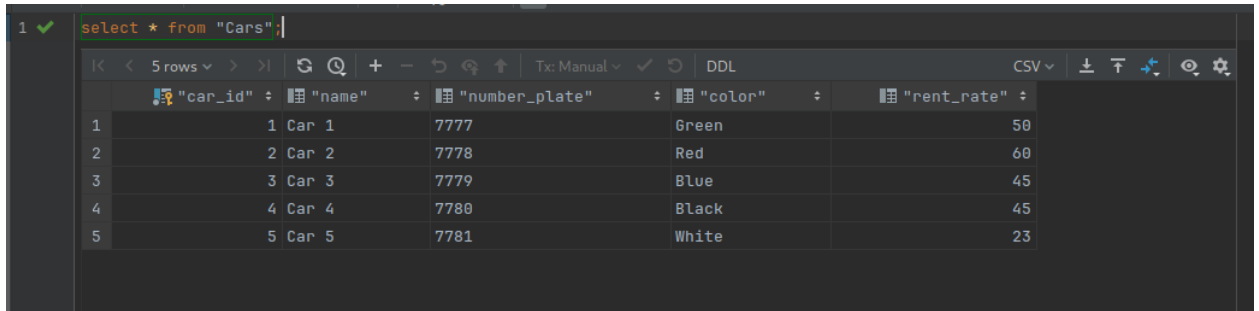
```
DMYTRO> delete from "Cars" where "car_id" in (5, 4)
[2023-03-30 21:37:13] 2 rows affected in 11 ms
```

3. Откатитесь к первоначальному состоянию и проверьте вернулись ли изменения.

Ролбек:

```
DMYTRO> delete from "Cars" where "car_id" in (5, 4)
[2023-03-30 21:37:13] 2 rows affected in 11 ms
DMYTRO> rollback
[2023-03-30 21:37:50] completed in 31 ms
```

Бачимо, що 4 та 5 автомобілі все ще присутні в таблиці:



The screenshot shows a database client interface with a SQL query `select * from "Cars";` executed. The result is a table with 5 rows and 5 columns: `car_id`, `name`, `number_plate`, `color`, and `rent_rate`.

	"car_id"	"name"	"number_plate"	"color"	"rent_rate"
1	1	Car 1	7777	Green	50
2	2	Car 2	7778	Red	60
3	3	Car 3	7779	Blue	45
4	4	Car 4	7780	Black	45
5	5	Car 5	7781	White	23

4. Создайте метку «START_POINT»

```
[2023-03-30 21:42:00] completed in 0 ms  
DMYTRO> create restore point start_point  
[2023-03-30 21:42:09] completed in 146 ms
```

5. Вставьте в таблицу 1 строку.

```
DMYTRO> insert into "Cars" values (6, 'Car 6', 'Red', 7782, 76)  
[2023-03-30 21:43:30] 1 row affected in 7 ms
```

6. Добавьте в таблицу индекс на любое из полей.

Додам індекс на поле «ціна ренти», оскільки ID це Primary Key і там індекс існує за замовчанням:

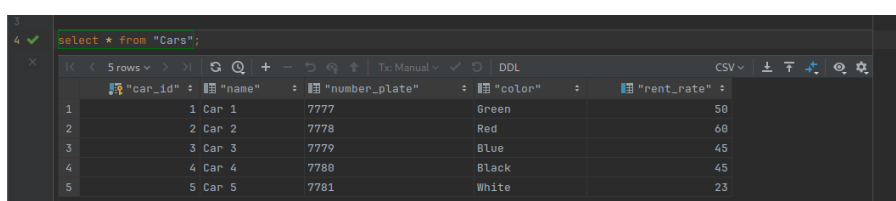
```
DMYTRO> create index rate_index on "Cars"("rent_rate")  
[2023-03-30 21:44:30] completed in 71 ms
```

7. Возможен ли откат к метке START_POINT? Проверьте.

Відкат до restore point:

```
DMYTRO> flashback table "Cars" to restore point start_point  
[2023-03-30 21:52:25] completed in 891 ms
```

Бачимо, що шостий запис зник:



The screenshot shows the same database client interface with the `select * from "Cars";` query. The result table now only contains 5 rows, with the 6th row (Car 6) removed.

	"car_id"	"name"	"number_plate"	"color"	"rent_rate"
1	1	Car 1	7777	Green	50
2	2	Car 2	7778	Red	60
3	3	Car 3	7779	Blue	45
4	4	Car 4	7780	Black	45
5	5	Car 5	7781	White	23

Задание 2. Откат к метке.

1. Создайте метку «START_POINT1»

```
[2023-03-30 22:04:03] completed in 888 ms  
DMYTRO> create restore point start_point1  
[2023-03-30 22:04:03] completed in 124 ms
```

2. Вставьте в таблицу 1 строку.

```
DMYTRO> insert into "Cars" values (6, 'Car 6', 7782, 'Red', 34)  
[2023-03-30 22:04:28] 1 row affected in 11 ms
```

3. Создайте метку «START_POINT2»

```
[2023-03-30 22:04:28] 1 row affected in 11 ms  
DMYTRO> create restore point start_point2  
[2023-03-30 22:04:44] completed in 121 ms
```

4. Вставьте в таблицу 1 строку.

```
DMYTRO> insert into "Cars" values (7, 'Car 7', 7783, 'Green', 67)  
[2023-03-30 22:06:29] 1 row affected in 123 ms
```

5. Создайте метку «START_POINT3»

```
DMYTRO> create restore point start_point3  
[2023-03-30 22:07:23] completed in 131 ms
```

6. Удалите из таблицы 3 строки.

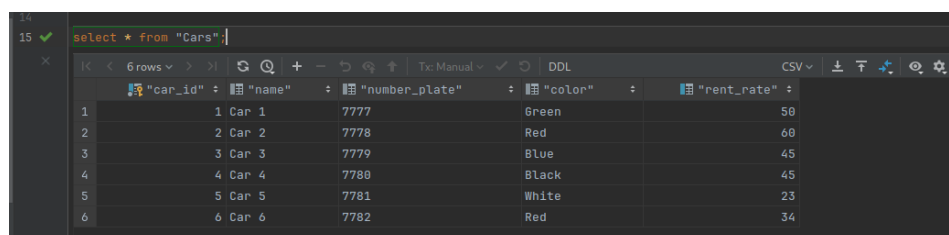
```
DMYTRO> delete from "Cars" where "car_id" in (5,6,7)  
[2023-03-30 22:08:39] 3 rows affected in 13 ms
```

7. Откатитесь к метке START_POINT2. Проверьте.

Відкат:

```
DMYTRO> flashback table "Cars" to restore point start_point2  
[2023-03-30 22:10:16] completed in 88 ms
```

Перевірка (пам'ятаємо, що на момент другого поінту був 6-й запис):



The screenshot shows a database query result for the 'Cars' table. The query is 'select * from "Cars";'. The result is displayed in a table with 6 rows and 5 columns: 'car_id', 'name', 'number_plate', 'color', and 'rent_rate'. The data is as follows:

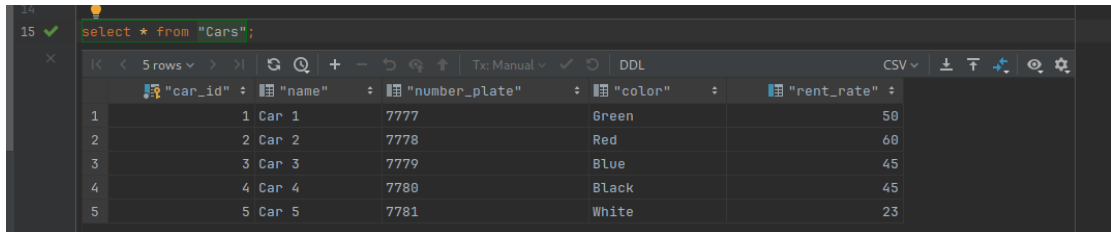
	car_id	name	number_plate	color	rent_rate
1	1	Car 1	7777	Green	50
2	2	Car 2	7778	Red	60
3	3	Car 3	7779	Blue	45
4	4	Car 4	7780	Black	45
5	5	Car 5	7781	White	23
6	6	Car 6	7782	Red	34

8. Откатитесь к метке START_POINT1. Проверьте.

Відкат:

```
DMYTRO> flashback table "Cars" to restore point start_point1
[2023-03-30 22:15:56] completed in 83 ms
```

Перевірка(повернулись у початковий стан):



	"car_id"	"name"	"number_plate"	"color"	"rent_rate"
1	1	Car 1	7777	Green	50
2	2	Car 2	7778	Red	60
3	3	Car 3	7779	Blue	45
4	4	Car 4	7780	Black	45
5	5	Car 5	7781	White	23

9. Удалите из таблицы 3 строки.

```
DMYTRO> delete from "Cars" where "car_id" in (3,4,5)
[2023-03-30 22:20:20] 3 rows affected in 9 s 757 ms
```

10. Создайте метку «START_POINT4»

```
DMYTRO> create restore point start_point4
[2023-03-30 22:20:53] completed in 121 ms
```

11. Вставьте в таблицу 1 строку.

```
DMYTRO> insert into "Cars" values (6, 'Car 6', 7782, 'Red', 34)
[2023-03-30 22:21:08] 1 row affected in 4 ms
```

12. Примените изменения.

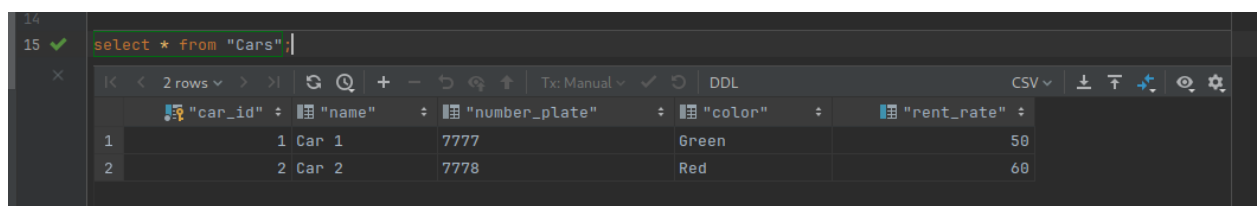
```
DMYTRO> commit
[2023-03-30 22:22:23] completed in 16 ms
```

13. Возможен ли откат к метке START_POINT4? Проверьте.

Відкат:

```
DMYTRO> flashback table "Cars" to restore point start_point4
[2023-03-30 22:22:41] completed in 89 ms
```

Перевірка(шостий ID зник):



	"car_id"	"name"	"number_plate"	"color"	"rent_rate"
1	1	Car 1	7777	Green	50
2	2	Car 2	7778	Red	60

3.Неявний COMMIT и ROLLBACK

Выберите в таблице по номеру Вашего варианта команду и определите, какой командой СУБД завершает их неявно: COMMIT / ROLLBACK / требуется явное завершения. Проверьте свое утверждение (приведите запросы, которые позволяют понять, что произошло).

Вариант 1 — drop table

Схоже що команда drop table неявно завершується комітом. Перевірити це можна дропнувши нашу таблицю й спробувати зробити rollback:

```
DMYTRO> drop table "Cars"
[2023-03-30 22:42:22] completed in 51 ms
DMYTRO> rollback
[2023-03-30 22:42:30] completed in 11 ms
DMYTRO> select * from "Cars"
[2023-03-30 22:42:35] [42000][942] ORA-00942: table or view does not exist
[2023-03-30 22:42:35] Position: 14
```

Бачимо, що зробити rollback до попереднього стану не вийшло.

Однак, згідно документації Oracle, під час дропання таблиць вони не видаляються безповоротно, а потрапляють в recycle bin (корзину). Неявно створюється restore point до якого можна повернутися після дропу.

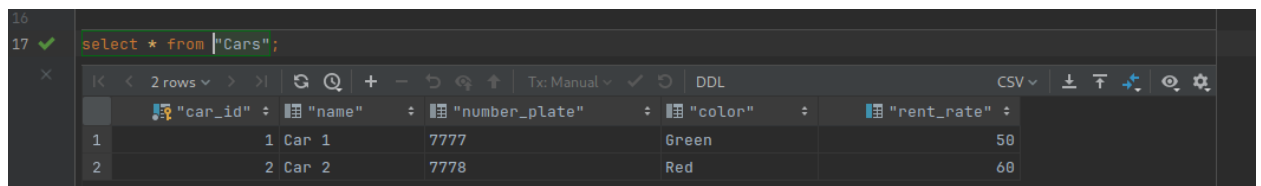
Перевіримо це на базі даних. Дропну таблицю Cars:

```
DMYTRO> drop table "Cars"
[2023-03-30 22:39:41] completed in 69 ms
```

Спробуємо її відновити:

```
DMYTRO> flashback table "Cars" to before drop
[2023-03-30 22:40:11] completed in 149 ms
```

Бачимо, що таблиця нікуди не дропнулась:

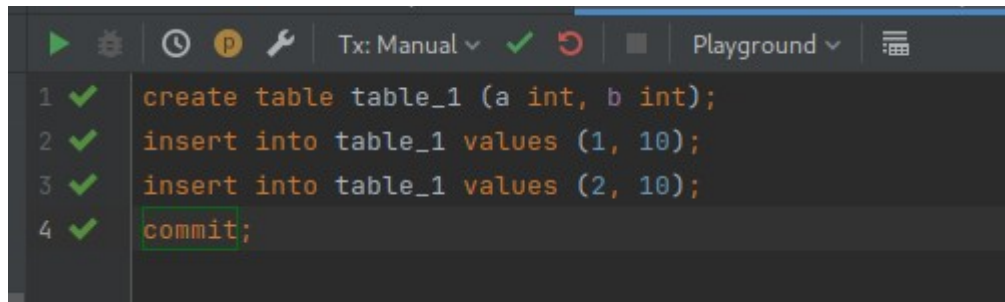


	"car_id"	"name"	"number_plate"	"color"	"rent_rate"
1	1	Car 1	7777	Green	50
2	2	Car 2	7778	Red	60

4. Deadlock

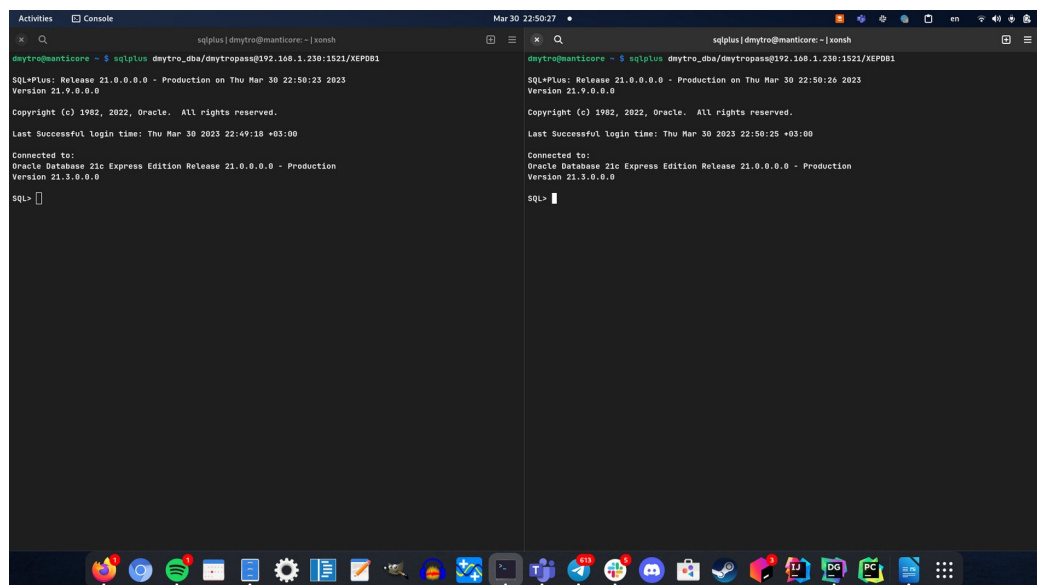
Реалізуйте ситуацію взаємної блокування. Приведіть скріншоти, які демонструють, як СУБД вирішила цю ситуацію.

Для початку створимо таблицю й додамо в неї два записи:



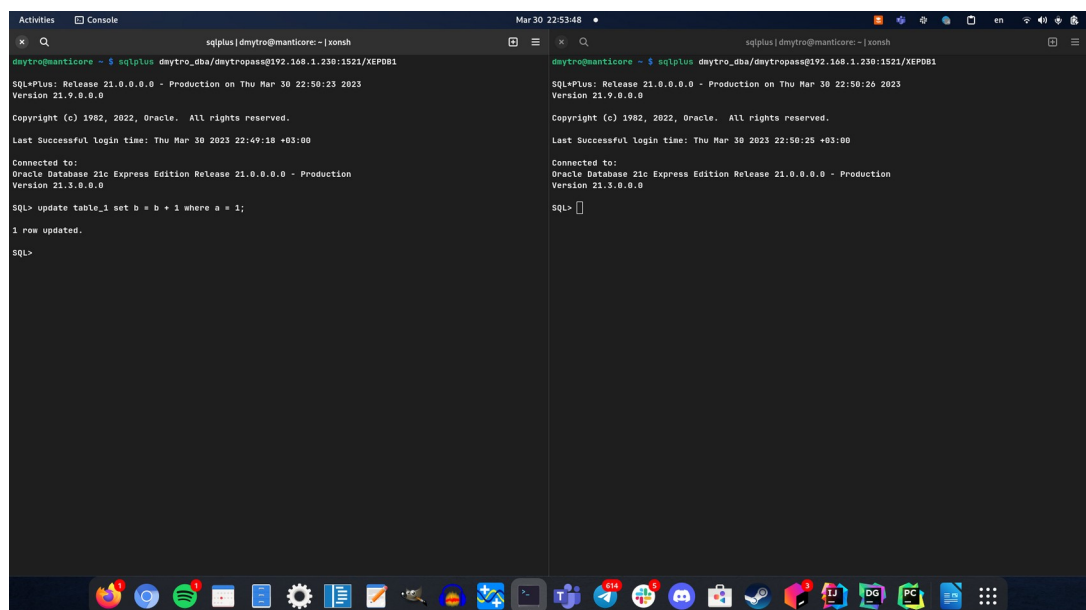
```
1 ✓ create table table_1 (a int, b int);
2 ✓ insert into table_1 values (1, 10);
3 ✓ insert into table_1 values (2, 10);
4 ✓ commit;
```

Далі відкриємо дві сесії за допомогою sqlplus



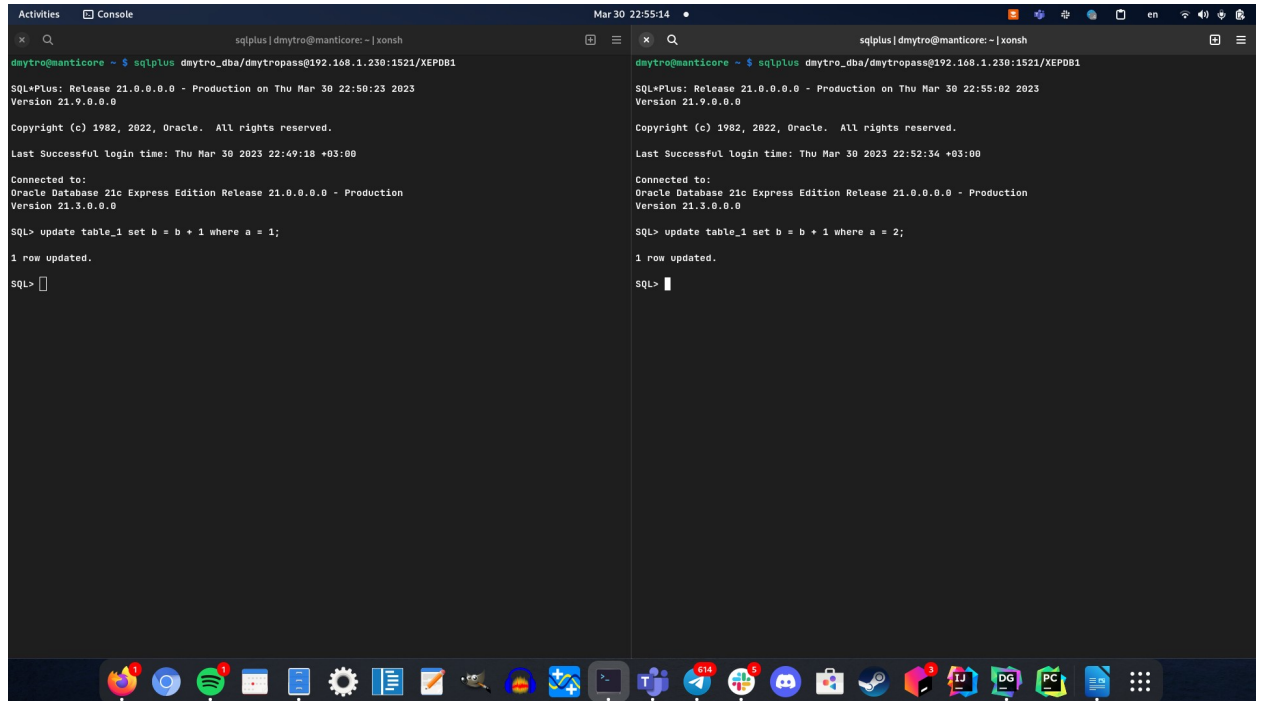
```
sqlplus dmytro@manticore:~|xonsh
dmytro@manticore ~ $ sqlplus dmytro_db/dmytrops@192.168.1.230:1521/XEPDB1
SQL*Plus: Release 21.0.0.0.0 - Production on Thu Mar 30 22:50:23 2023
Version 21.9.0.0.0
Copyright (c) 1982, 2022, Oracle. All rights reserved.
Last Successful login time: Thu Mar 30 2023 22:49:18 +03:00
Connected to:
Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0
SQL>
```

Після цього розпочнемо в першому терміналі транзакцію:



```
sqlplus dmytro@manticore:~|xonsh
dmytro@manticore ~ $ sqlplus dmytro_db/dmytrops@192.168.1.230:1521/XEPDB1
SQL*Plus: Release 21.0.0.0.0 - Production on Thu Mar 30 22:50:23 2023
Version 21.9.0.0.0
Copyright (c) 1982, 2022, Oracle. All rights reserved.
Last Successful login time: Thu Mar 30 2023 22:49:18 +03:00
Connected to:
Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0
SQL> update table_1 set b = b + 1 where a = 1;
1 row updated.
SQL>
```

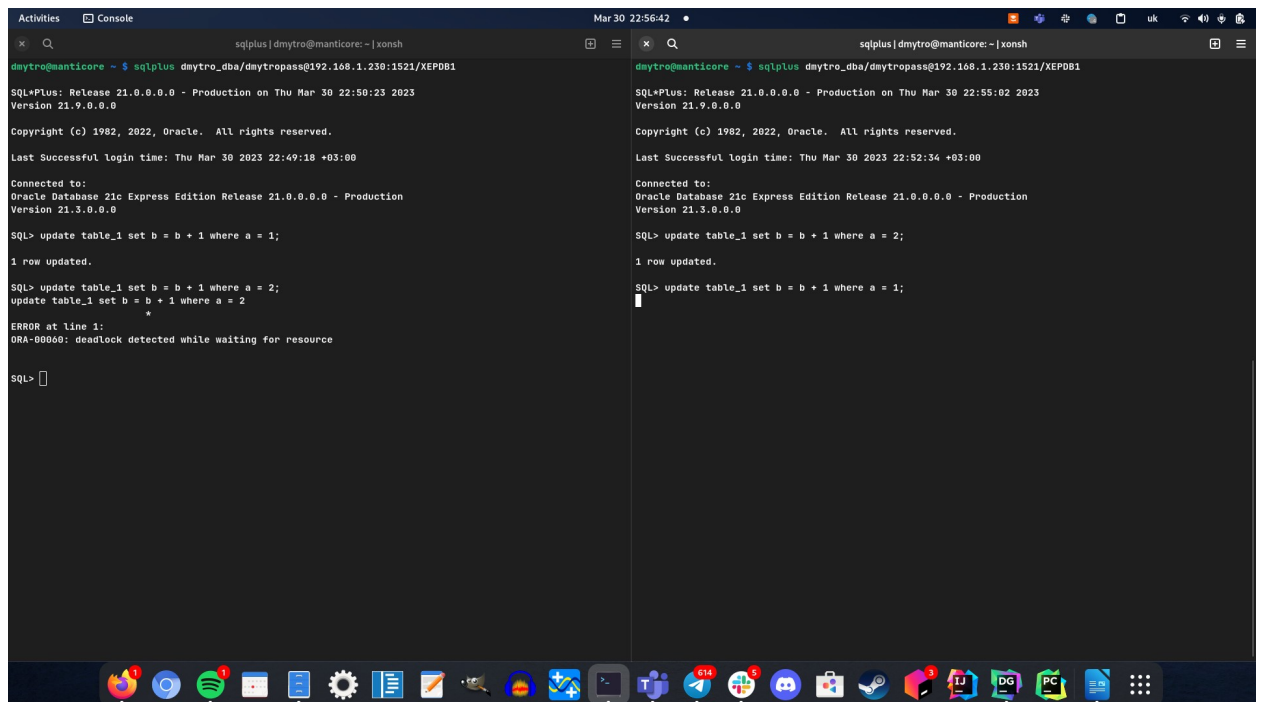

Далі розпочинаємо транзакцію в другому терміналі, редагуючи другий запис в БД:



```
Activities Console Mar 30 22:55:14
sqlplus | dmytro@manticore: ~ | xonsh
dmytro@manticore ~ $ sqlplus dmytro_dba/dmytypass@192.168.1.230:1521/XEP081
SQL*Plus: Release 21.0.0.0.0 - Production on Thu Mar 30 22:50:23 2023
Version 21.9.0.0.0
Copyright (c) 1982, 2022, Oracle. All rights reserved.
Last Successful login time: Thu Mar 30 2023 22:49:18 +03:00
Connected to:
Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0
SQL> update table_1 set b = b + 1 where a = 1;
1 row updated.
SQL>

sqlplus | dmytro@manticore: ~ | xonsh
dmytro@manticore ~ $ sqlplus dmytro_dba/dmytypass@192.168.1.230:1521/XEP081
SQL*Plus: Release 21.0.0.0.0 - Production on Thu Mar 30 22:55:02 2023
Version 21.9.0.0.0
Copyright (c) 1982, 2022, Oracle. All rights reserved.
Last Successful login time: Thu Mar 30 2023 22:52:34 +03:00
Connected to:
Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0
SQL> update table_1 set b = b + 1 where a = 2;
1 row updated.
SQL>
```

Далі з першого терміналу пробуємо стукати в другий запис, а з другого терміналу в перший:

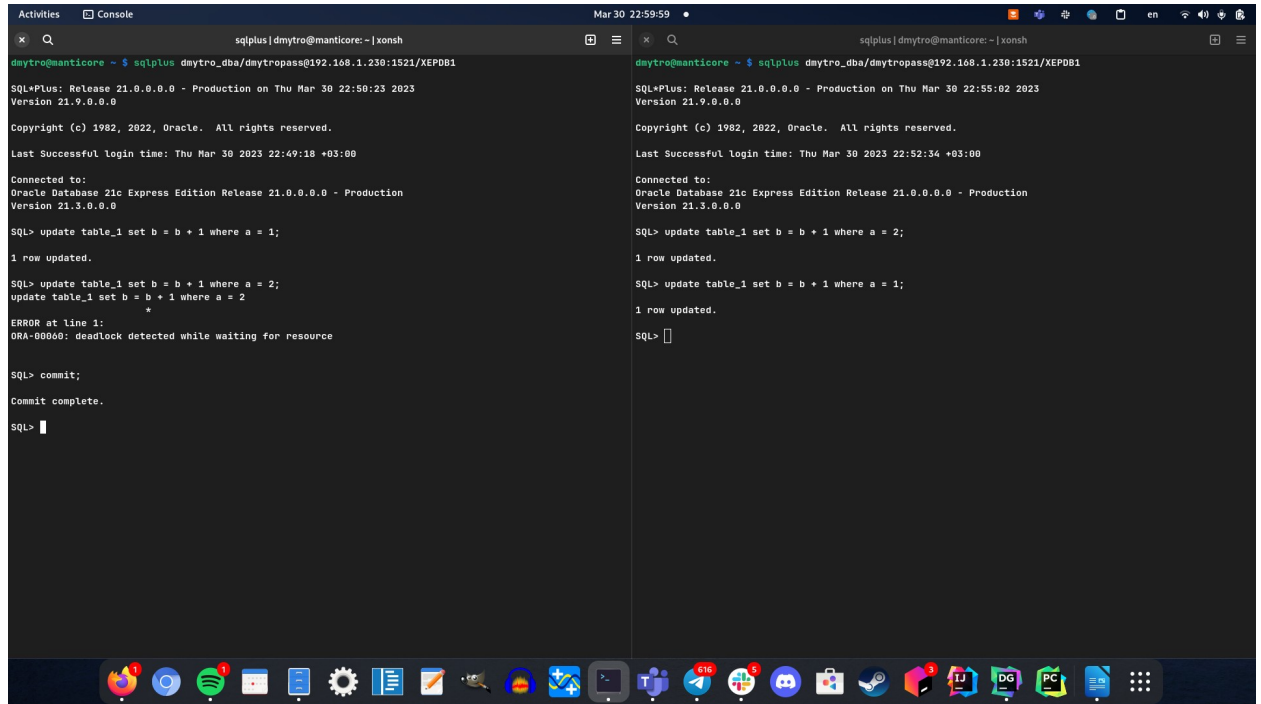


```
Activities Console Mar 30 22:56:42
sqlplus | dmytro@manticore: ~ | xonsh
dmytro@manticore ~ $ sqlplus dmytro_dba/dmytypass@192.168.1.230:1521/XEP081
SQL*Plus: Release 21.0.0.0.0 - Production on Thu Mar 30 22:50:23 2023
Version 21.9.0.0.0
Copyright (c) 1982, 2022, Oracle. All rights reserved.
Last Successful login time: Thu Mar 30 2023 22:49:18 +03:00
Connected to:
Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0
SQL> update table_1 set b = b + 1 where a = 1;
1 row updated.
SQL> update table_1 set b = b + 1 where a = 2;
update table_1 set b = b + 1 where a = 2
*
ERROR at line 1:
ORA-00060: deadlock detected while waiting for resource
SQL>

sqlplus | dmytro@manticore: ~ | xonsh
dmytro@manticore ~ $ sqlplus dmytro_dba/dmytypass@192.168.1.230:1521/XEP081
SQL*Plus: Release 21.0.0.0.0 - Production on Thu Mar 30 22:55:02 2023
Version 21.9.0.0.0
Copyright (c) 1982, 2022, Oracle. All rights reserved.
Last Successful login time: Thu Mar 30 2023 22:52:34 +03:00
Connected to:
Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0
SQL> update table_1 set b = b + 1 where a = 2;
1 row updated.
SQL> update table_1 set b = b + 1 where a = 1;
SQL>
```

В результаті отримуємо deadlock. База даних побачила, що є дві сесії, які мають відкриті транзакції, що не можуть завершитися через одна одну, тому кидає в першу сесію помилку-попередження про deadlock. Таким чином ми все-ще маємо змогу завершити першу транзакцію та у цей момент продовжиться друга. Перевіримо, зробимо коміт у першому терміналі.

Результат:



```
Activities Console Mar 30 22:59:59
sqlplus | dmytro@manticore: ~ | xonsh
dmytro@manticore ~ $ sqlplus dmytro_dba/dmytropass@192.168.1.230:1521/XEPDB1
SQL*Plus: Release 21.0.0.0.0 - Production on Thu Mar 30 22:50:23 2023
Version 21.9.0.0.0
Copyright (c) 1982, 2022, Oracle. All rights reserved.
Last Successful login time: Thu Mar 30 2023 22:49:18 +03:00
Connected to:
Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0
SQL> update table_1 set b = b + 1 where a = 1;
1 row updated.
SQL> update table_1 set b = b + 1 where a = 2;
update table_1 set b = b + 1 where a = 2
*
ERROR at line 1:
ORA-00060: deadlock detected while waiting for resource
SQL> commit;
Commit complete.
SQL>

sqlplus | dmytro@manticore: ~ | xonsh
dmytro@manticore ~ $ sqlplus dmytro_dba/dmytropass@192.168.1.230:1521/XEPDB1
SQL*Plus: Release 21.0.0.0.0 - Production on Thu Mar 30 22:55:02 2023
Version 21.9.0.0.0
Copyright (c) 1982, 2022, Oracle. All rights reserved.
Last Successful login time: Thu Mar 30 2023 22:52:34 +03:00
Connected to:
Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0
SQL> update table_1 set b = b + 1 where a = 2;
1 row updated.
SQL> update table_1 set b = b + 1 where a = 1;
1 row updated.
SQL>
```

Щойно я завершив транзакцію у першому терміналі, другий показав результат «1 row updated.», що свідчить про те, що транзакція продовжилась.