ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені ІВАНА ФРАНКА Факультет прикладної математики та інформатики

Бази даних та інформаційні системи

ЛАБОРАТОРНА РОБОТА №2 Транзакції в СКБД PostgreSQL

Виконала:

Ст. Кравець Ольга

ПМО-31

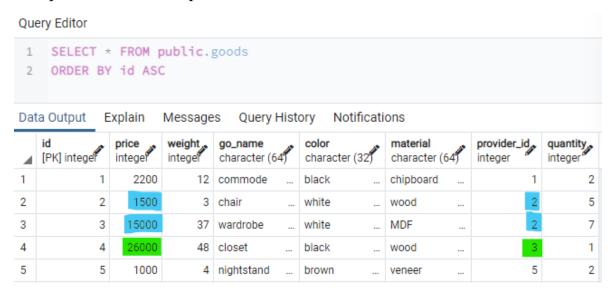
Tema: Вивчення понять транзакції та управління конкурентним доступом в СКБД PostgreSQL.

Мета роботи: Ознайомлення з використанням транзакцій, їх розробкою та застосуванням, рівнями ізоляцій та механізмом управління конкурентним доступом в СКБД PostgreSQL

Хід роботи

Опрацювала теоретичні відомості про транзакції, атомарні транзакції, команди, скасування змін, точки збереження, повернення до точок збереження, рівні ізоляції, явні блокування, перевірки цілісності даних, обмеження, блокування та індекси.

Написала елементарну транзакцію з трьома операціями, одна з яких скасована з допомогою SAVEPOINT'а та операції повернення до конкретного моменту ROLLBACK TO (Рівень ізоляції - Read Committed). Операції - збільшення ціни товарів для певних виробників:

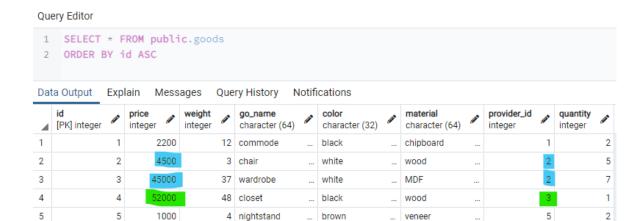


Query Editor

```
begin;
 1
    update goods set price = price*2 where provider_id = 3;
    savepoint sp_operation;
 3
    update goods set price = price*10 where provider_id = 1;
4
    rollback to sp_operation;
 5
 6
    update goods set price = price*3 where provider_id = 2;
    commit;
 7
                                            Notifications
Data Output
            Explain
                    Messages
                               Query History
```

COMMIT

Query returned successfully in 129 msec.



Написала транзакції, для демонстрації рівнів ізоляції Repeatable Read.

Dirty Read

Транзакція для демонстрації вибору товару з ID = 2:



Далі змінюю дані таблиці:





Query returned successfully in 49 msec.

Виконую ще раз select у першій транзакції, щоб поглянути чи щось змінилось:

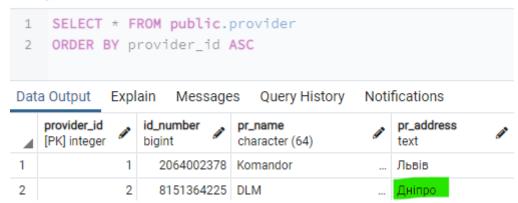


Дані не змінилися, оскільки транзакція бачить лише зміни, що виконані при першому читанні.

Nonrepeatable Read

Транзакція для демонстрації вибору постачальника з ID = 2:

Query Editor



Виконую 1 транзакцію:

Query Editor

```
1 -- Транзакція 1
 2 set transaction isolation level read committed;
 3 begin transaction;
4 --1
 5
    select * from provider where provider.provider_id = 2;
Data Output
            Explain Messages
                                 Query History
                                               Notifications
   provider_id ,
              id_number
                                                   pr_telephone
                           pr_name
                                        pr_address,
  [PK] integer
              bigint
                           character (64)
                                        text
                                                   character (15)
                8151364225 DLM
                                                   +(380)680571515
1
                                        Дніпро
```

Далі змінюю дані таблиці:

Query Editor

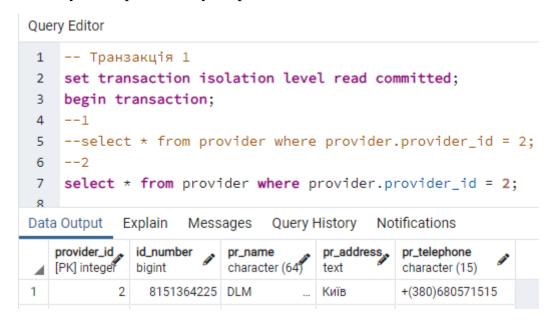
```
1 -- Транзакція 2
2 set transaction isolation level read committed;
3 update provider set "pr_address"='Київ' where provider.provider_id = 2;
4

Data Output Explain Messages Query History Notifications

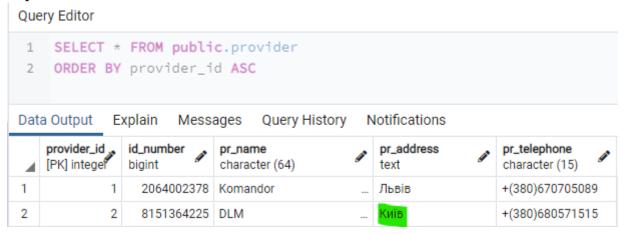
UPDATE 1
```

Query returned successfully in 48 msec.

Виконую ще раз select у 1 транзакції:



Дані змінилися, оскільки 1 транзакція читала дані, паралельно інша транзакція оновила дані і зафіксувала нові і тоді 1 транзакція перечитала ті самі дані і отримала інше значення:



Phantom Read

Транзакція для демонстрації вибору замовлення з сумою = 0:

Query Editor



Виконую 1 транзакцію:

Query Editor

1	1 Транзакція 1					
2	set transaction isolation level repeatable read;					
3	begin tran	begin transaction;				
4	1	1				
5	<pre>5 select * from orders where orders.sum = 0;</pre>					
Data Output Explain Messages Query History Notifications						
4	id [PK] integer	delivery_method character (25)	client_id integer	sum integer		
1	2	Self-pickup	1	0		
2	3	Post	4	0		

Далі додаю ще один рядок до таблиці зі значення суми = 0:

Query Editor

```
1 insert into orders (id, delivery_method, client_id, sum)
2 values (6, 'Post', 4, 0)

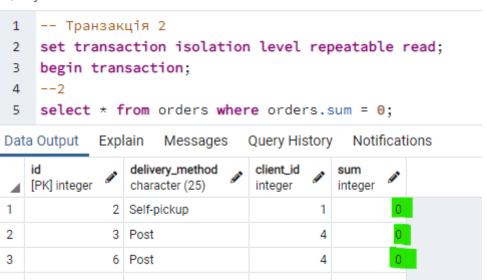
Data Output Explain Messages Query History Notifications

INSERT 0 1

Query returned successfully in 57 msec.
```

Виконую ще раз select у 1 транзакції:

Query Editor



Тепер стало 3 рядки з даними, де сума=0, оскільки 1 транзакція читала рядки, в яких була сума=0, а тоді додався рядок, який відповідав критерію, що сума=0 і 1 транзакція повторно виконалася і вже отримала інший набір рядків.

Написала процедуру, яка змінює назву товару з заданим ID. Якщо товар з вказаним ID знайдено, то викликається ROLLBACK (скасовуються зміни) та повертається EXCEPTION.

Query Editor

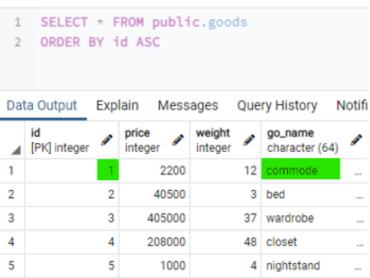
```
create procedure change_goods_name(g_id int, new_name text)
1
    language plpgsql
3 as $$
4 declare
5 ▼ begin
   update goods
6
    set "go_name"=new_name
7
    where goods.id = g_id;
9 v if not found then
10
            rollback;
            raise exception 'Goods with this ID is not found.';
11
12
        else
13
            commit;
    end if;
14
15
    end;
    $$
16
Data Output
           Explain
                   Messages
                              Query History
                                           Notifications
```

CREATE PROCEDURE

Query returned successfully in 94 msec.

До виклику:

Query Editor



Виклик:



```
1 call change_goods_name(1, 'chair');

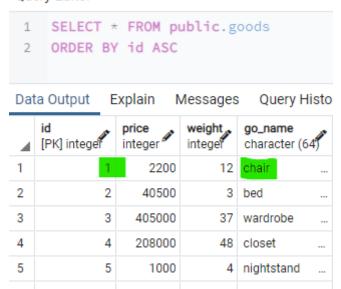
Data Output Explain Messages Query History Notifications

CALL

Query returned successfully in 73 msec.
```

Після виклику:

Query Editor



Виклик процедури з неіснуючим товаром:

Query Editor

```
1 call change_goods_name(7, 'chair');

Data Output Explain Messages Query History Notifications

ERROR: ПОМИЛКА: Goods with this ID is not found.

CONTEXT: Функція PL/pgSQL change_goods_name(integer,text) рядок 9 в RAISE

SQL state: P0001
```

Повертається помилка, що не існує товару з вказаним ID.

Висновок: на цій лабораторній роботі я ознайомилася з поняттям транзакцій, команд, скасування змін, рівнів ізоляції, перевірки цілісності даних.