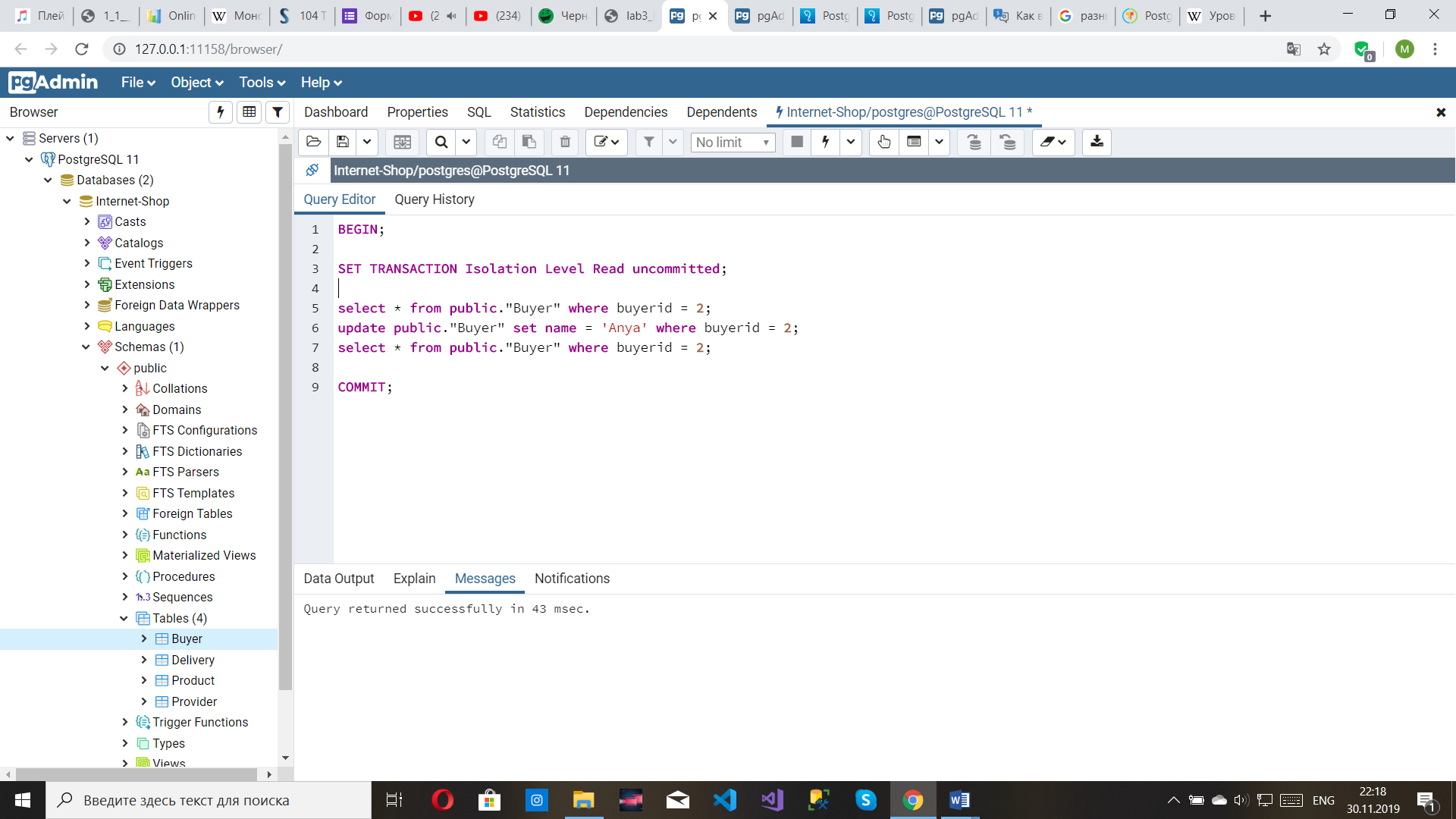
***Завдання 4:***

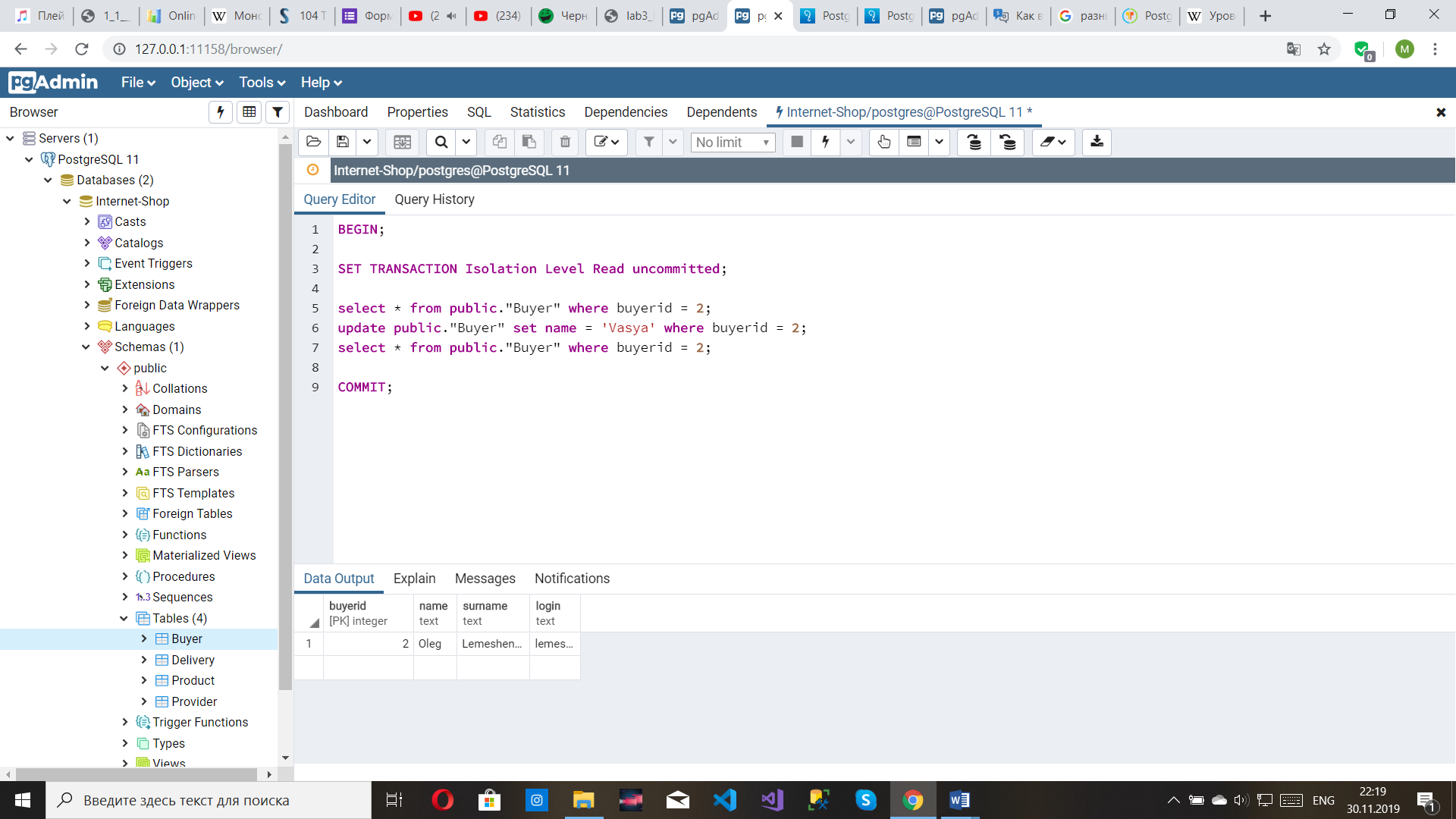
1. *Read Uncommitted*

Якщо кілька паралельних транзакцій намагаються змінювати один і той же рядок таблиці, то в остаточному варіанті рядок буде мати значення, визначене останньою успішно виконаною транзакцією - ***Втрачене оновлення (Lost Update)***

1 транзакція



2 транзакція



При послідовному виконанні, після виконання update в 1й і виклику update в 2й, 2 видає таке вікно (Рис 1.1) і після виконання обох транзакцій результатом є рядок з 2-ї транзакції

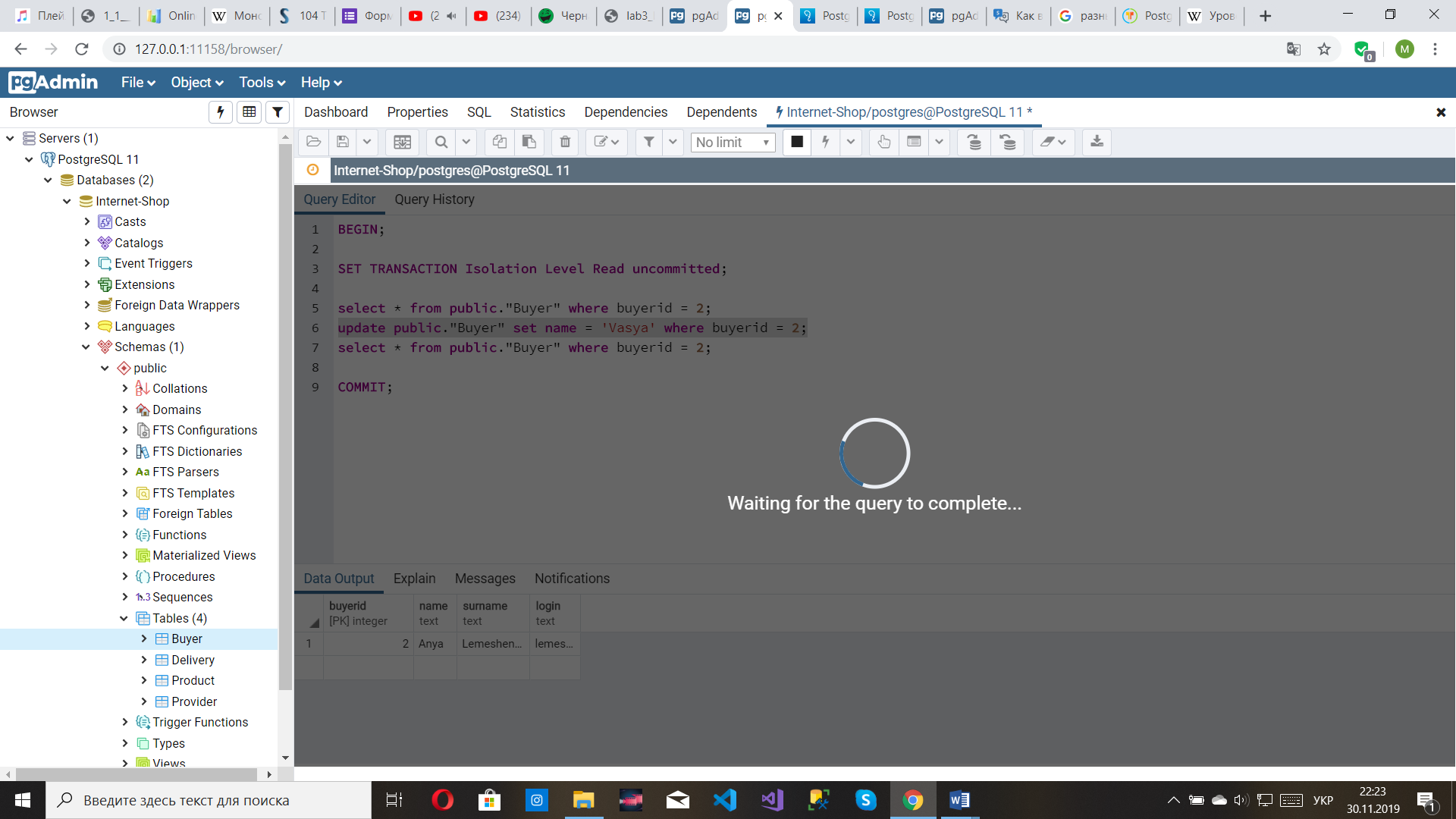


Рис. 1.1

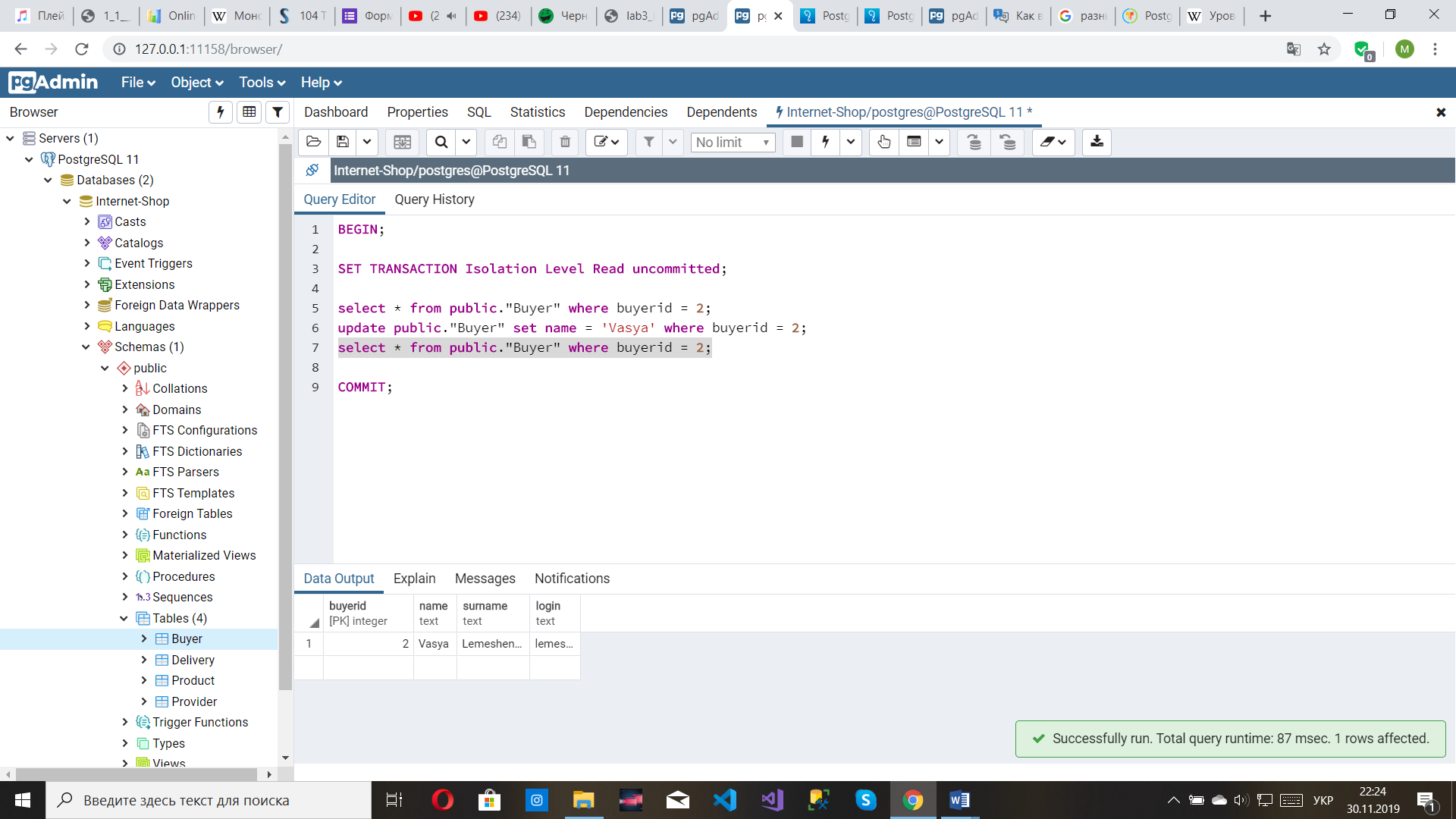


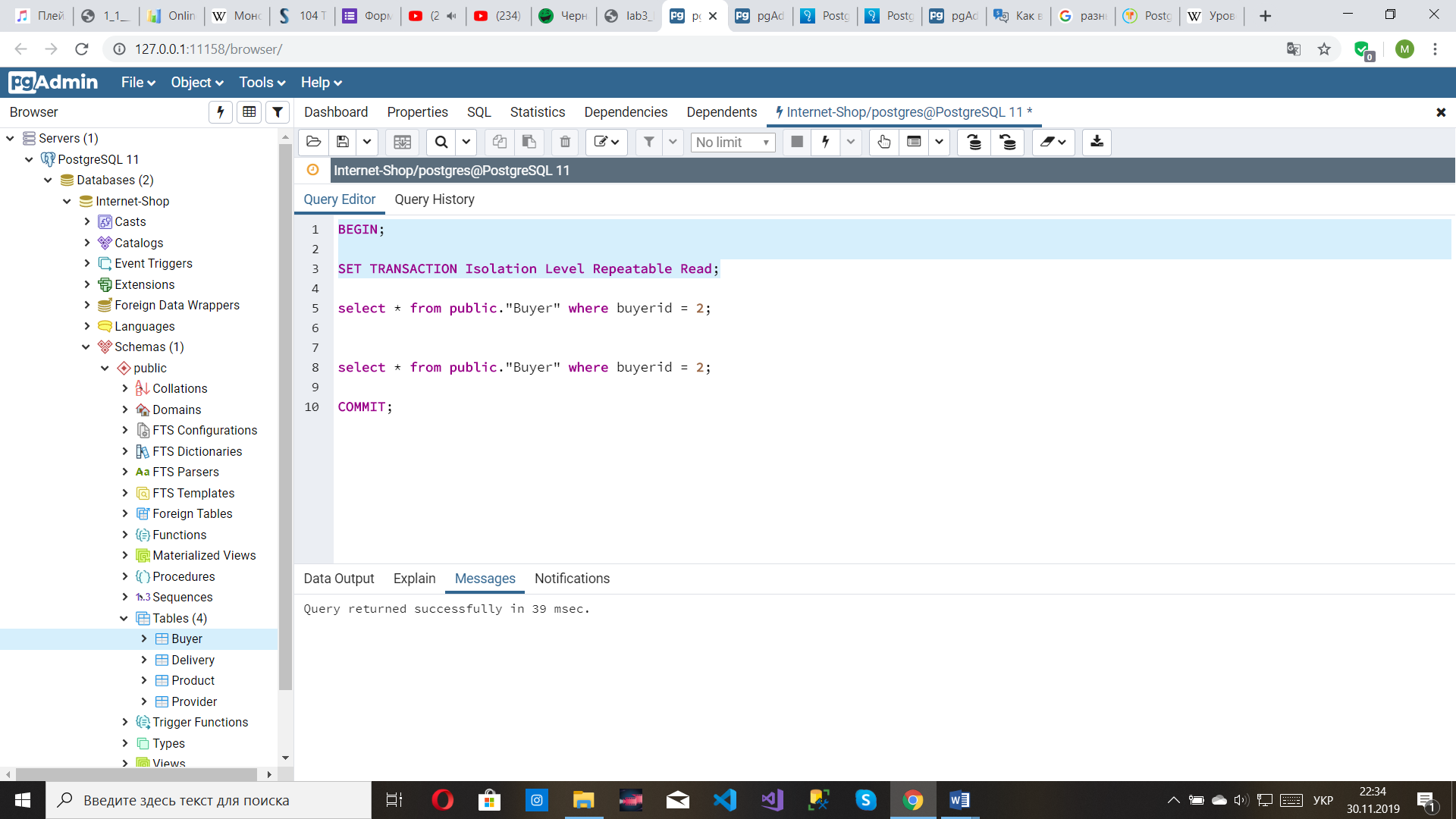
Рис. 1.2 Результат виконання 2 транзакцій

1. *Repeatable Read*

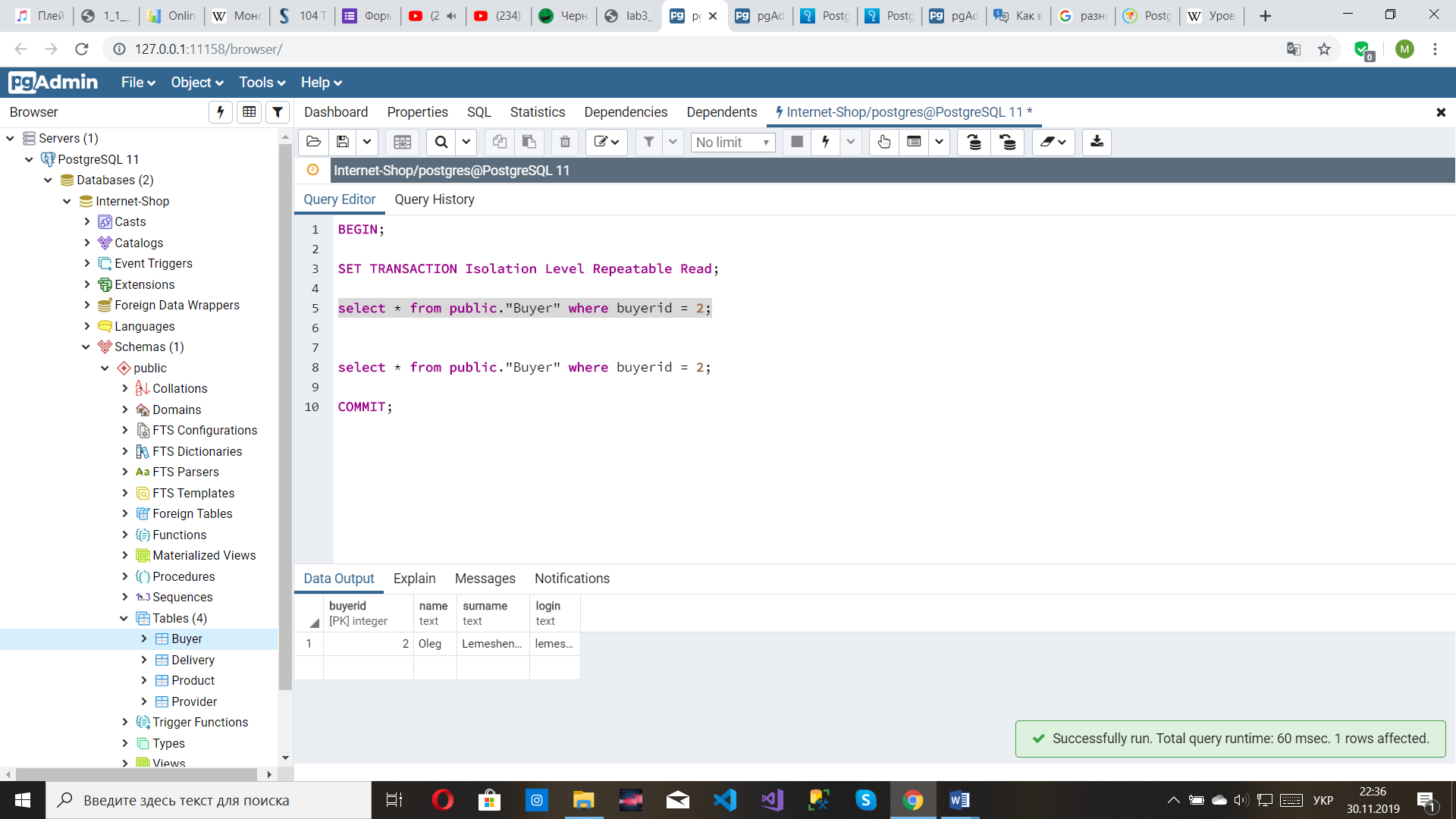
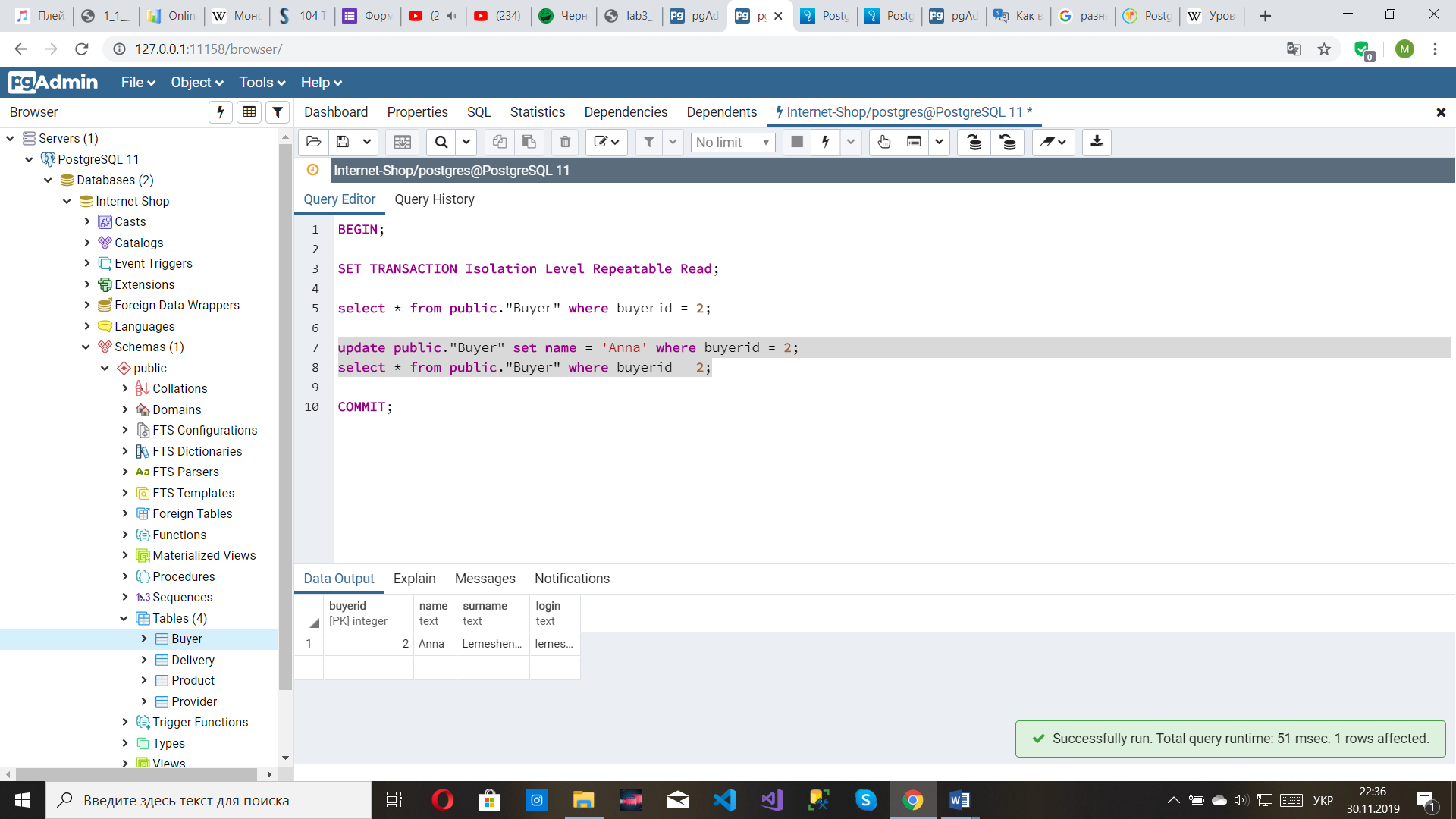
Всі оператори поточної транзакції бачать тільки ті рядки, які були зафіксовані перед першим запитом на вибірку або зміну даних, виконаним в цій транзакції.

Виконаємо дві наступні транзакції, і побачимо, що по завершенню 2-ї, перша все ще не «бачитиме» змін, що відбулись.

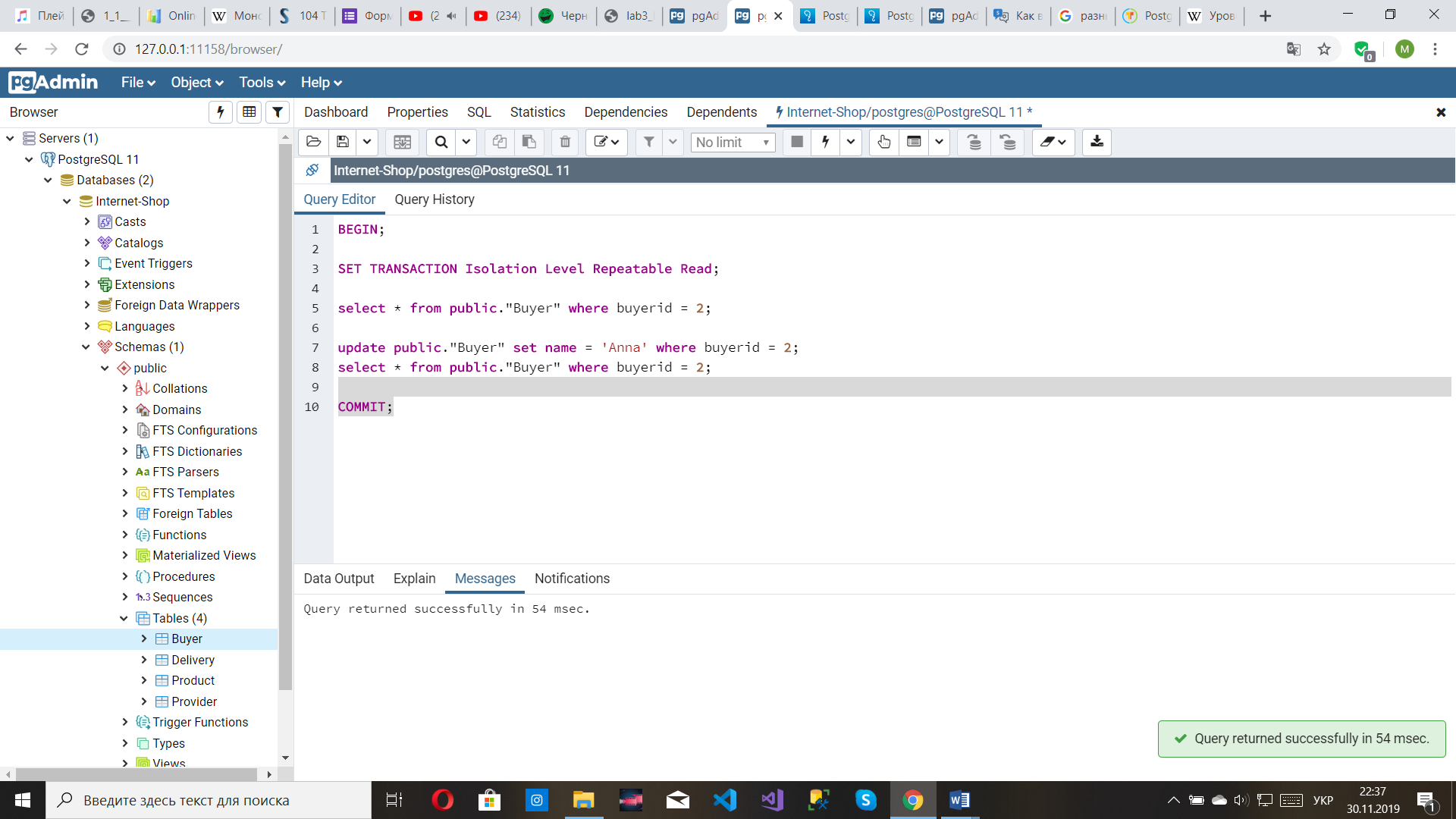
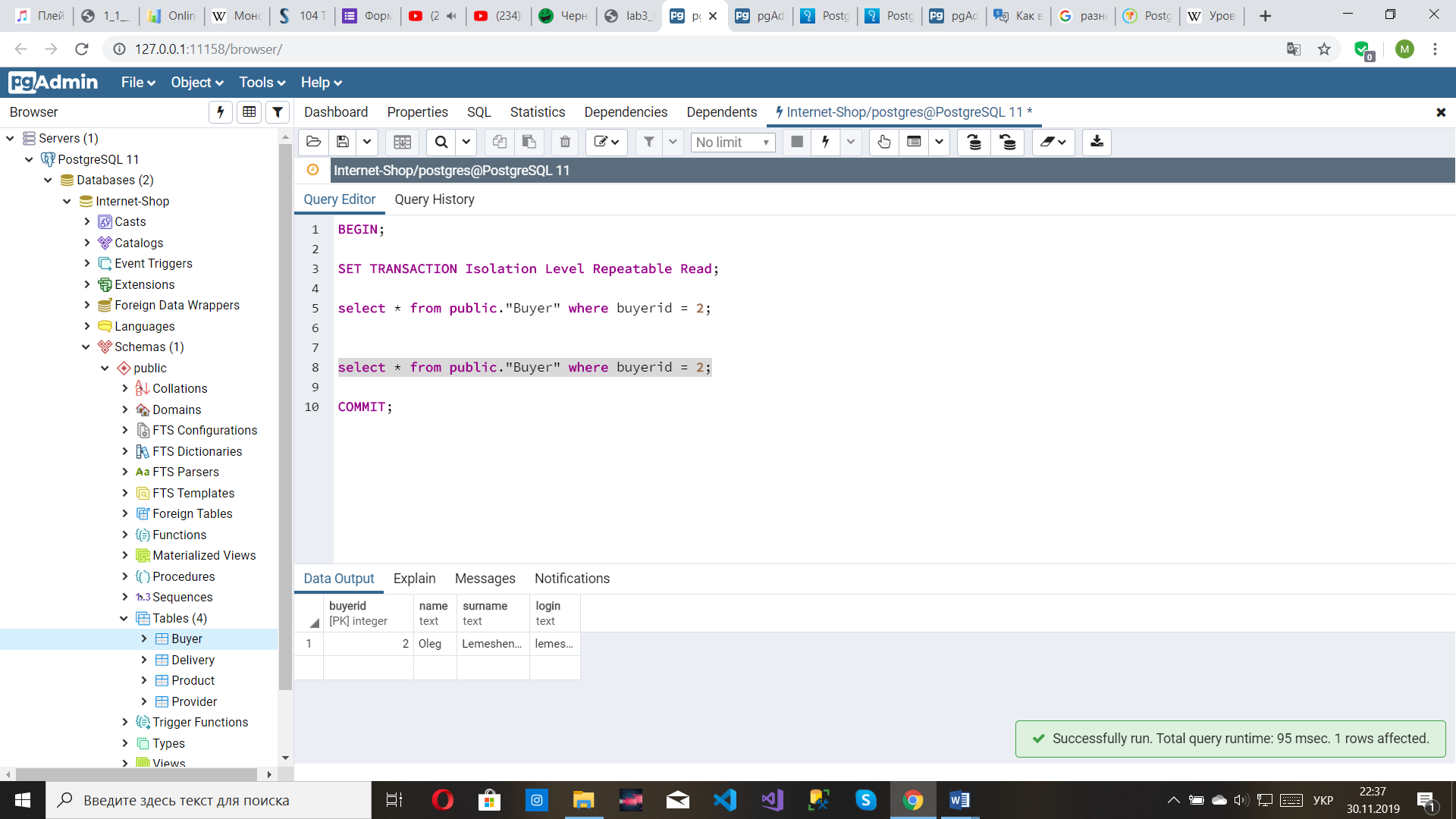
**1 крок :**

**2 крок:**

**3 крок:**

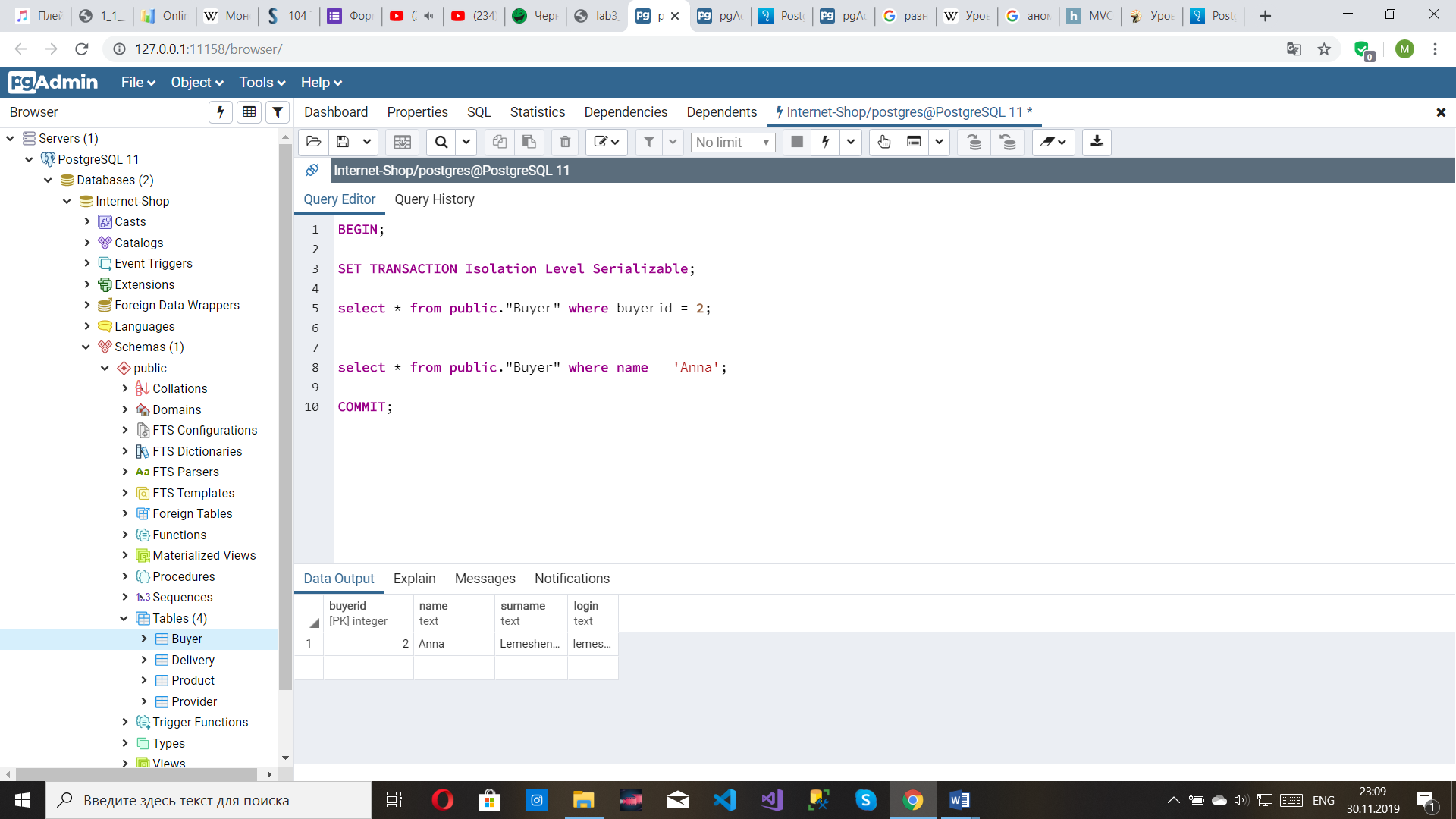
 

Тобто 1-ша транзакція не бачить змін, які відбулись у 2-й

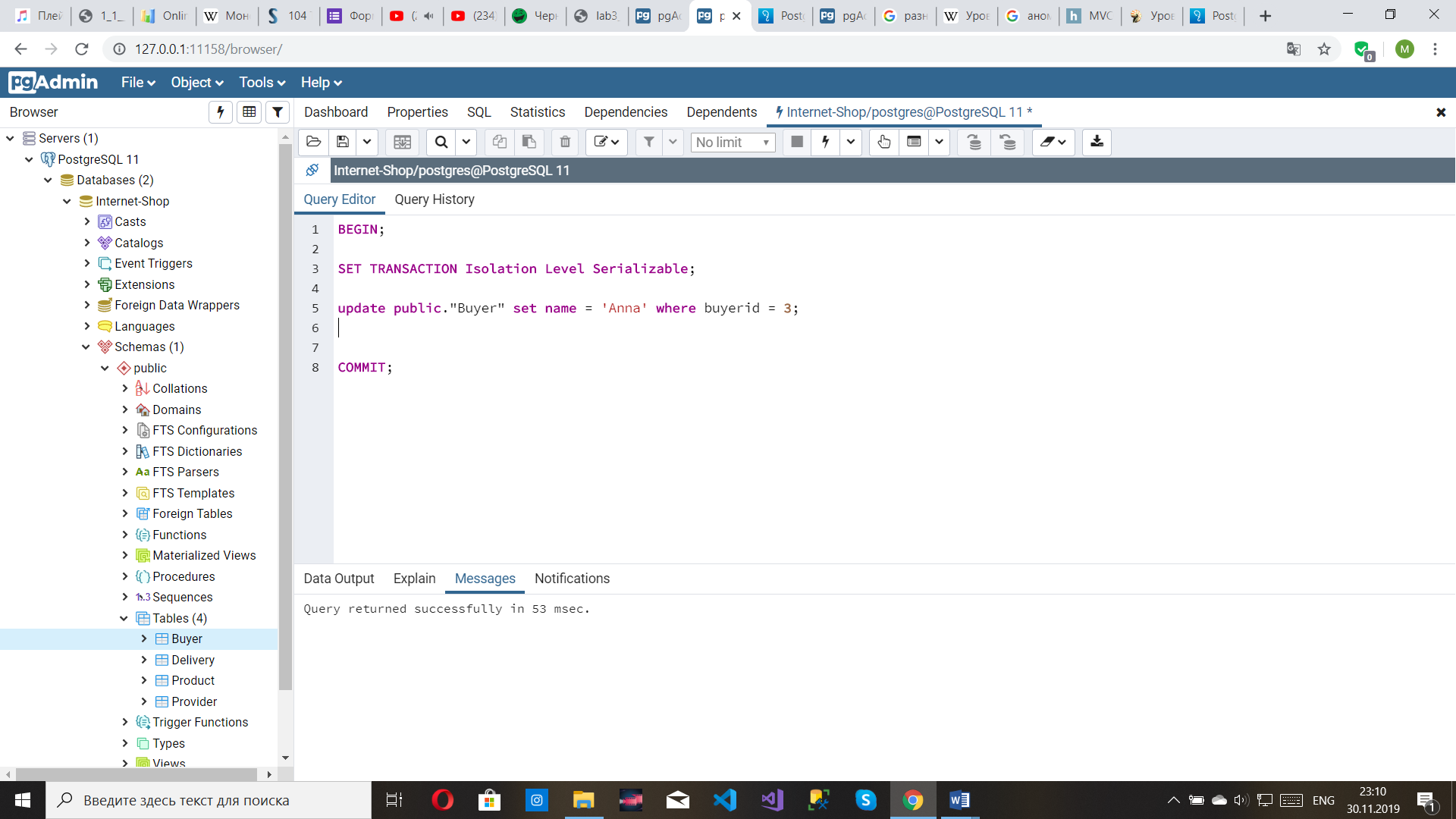
1. *Serializable*

***Фантомне читання (Phantom Reads)*** - одна транзакція в ході свого виконання кілька разів вибирає декілька рядків по одним і тим же критеріям. Інша транзакція в інтервалах між цими вибірками додає або видаляє рядки або змінює поля деяких рядків, які використовуються в умовах вибірки першої транзакції, і успішно закінчується. В результаті вийде, що одні й ті ж вибірки в першій транза-кції дають різну кількість рядків.

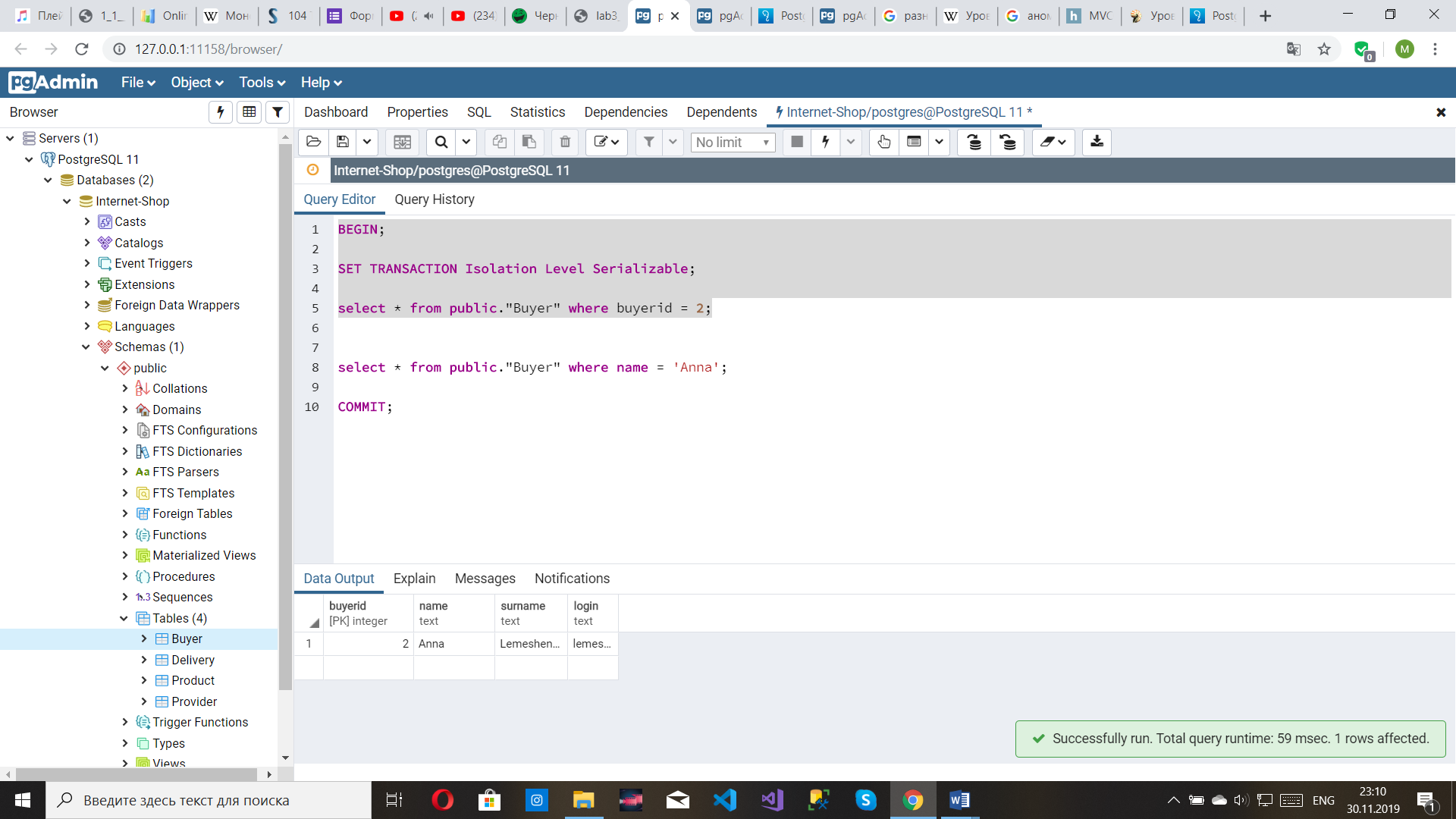
Перевіримо фантомне читання, маємо 2 транзакції :



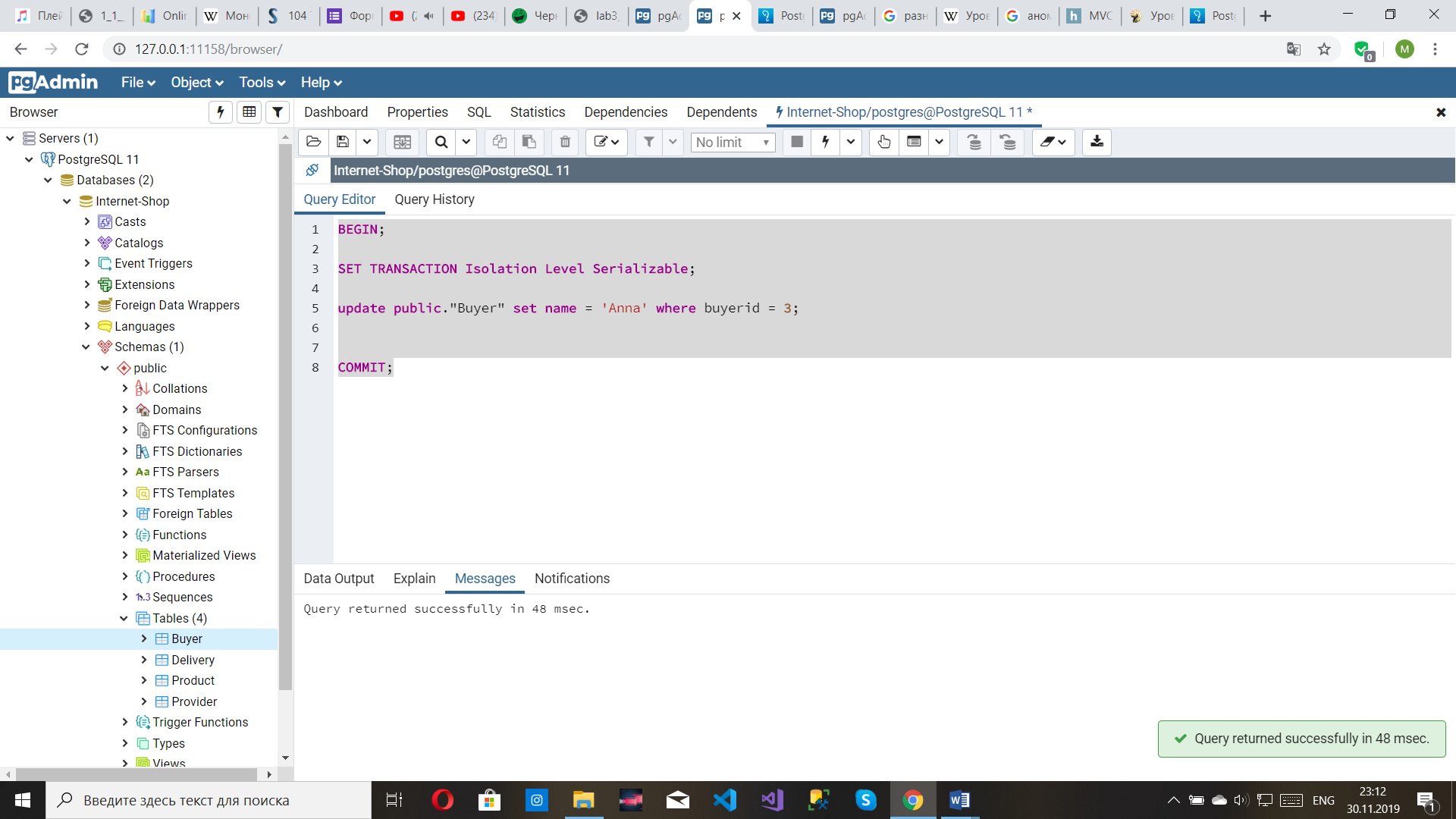
1 виконує «фантомне» читання, 2-га виконує оновлення.



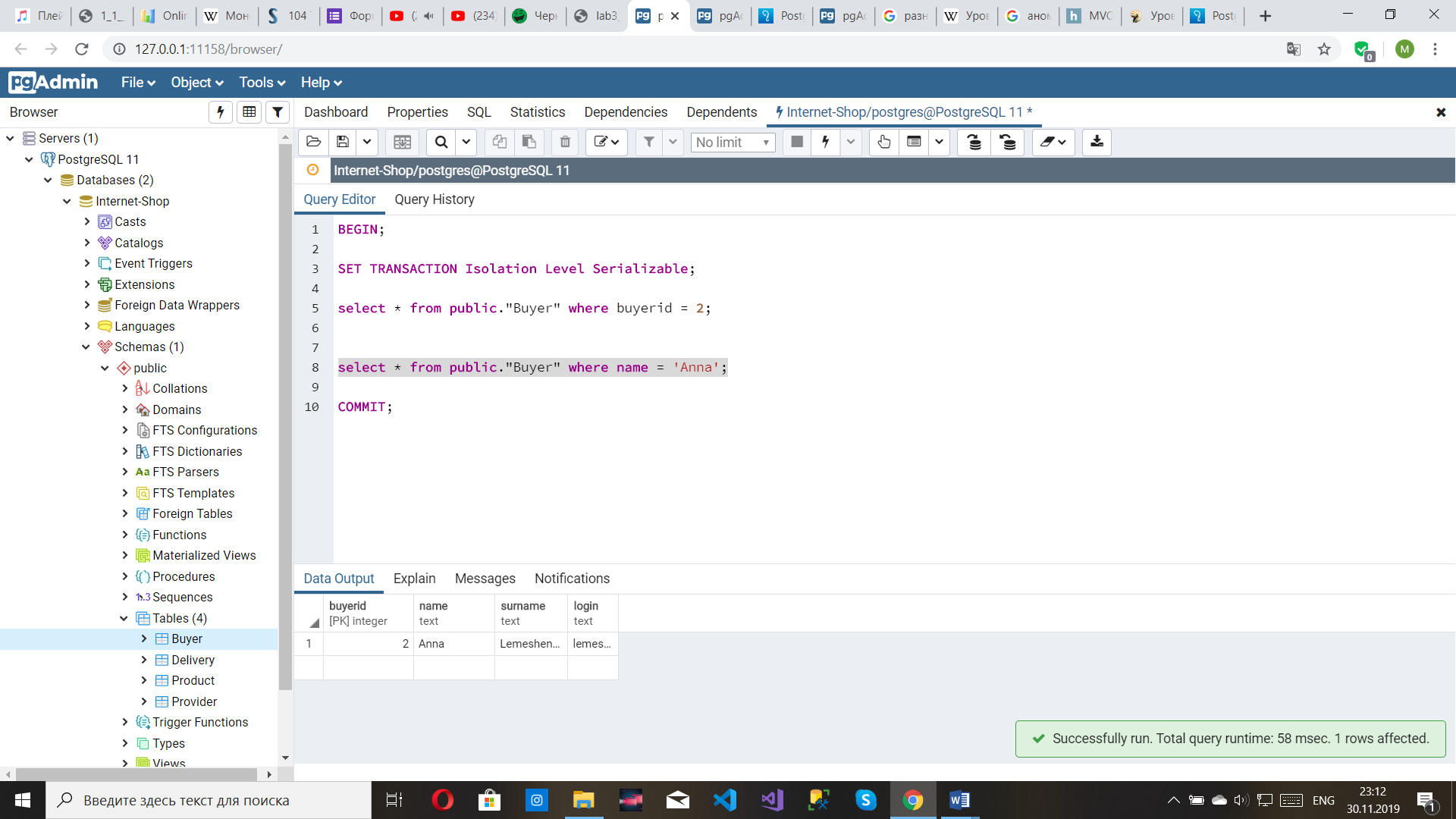
1 читання в 1й транзакції :



Виконання оновлення і комміту 2-ї транзакції :



Спроба отримати «фантомне читання» :



Як бачимо, у данному випадку цей феномен відсутній.

***Завдання 3:***

*BEFORE UPDATE:*

Створено тригер та тригерну функцію, з використанням умовних операторів та обробкою виключних ситуацій. На Рис. 1.3 зображено виконання виключної ситуації.

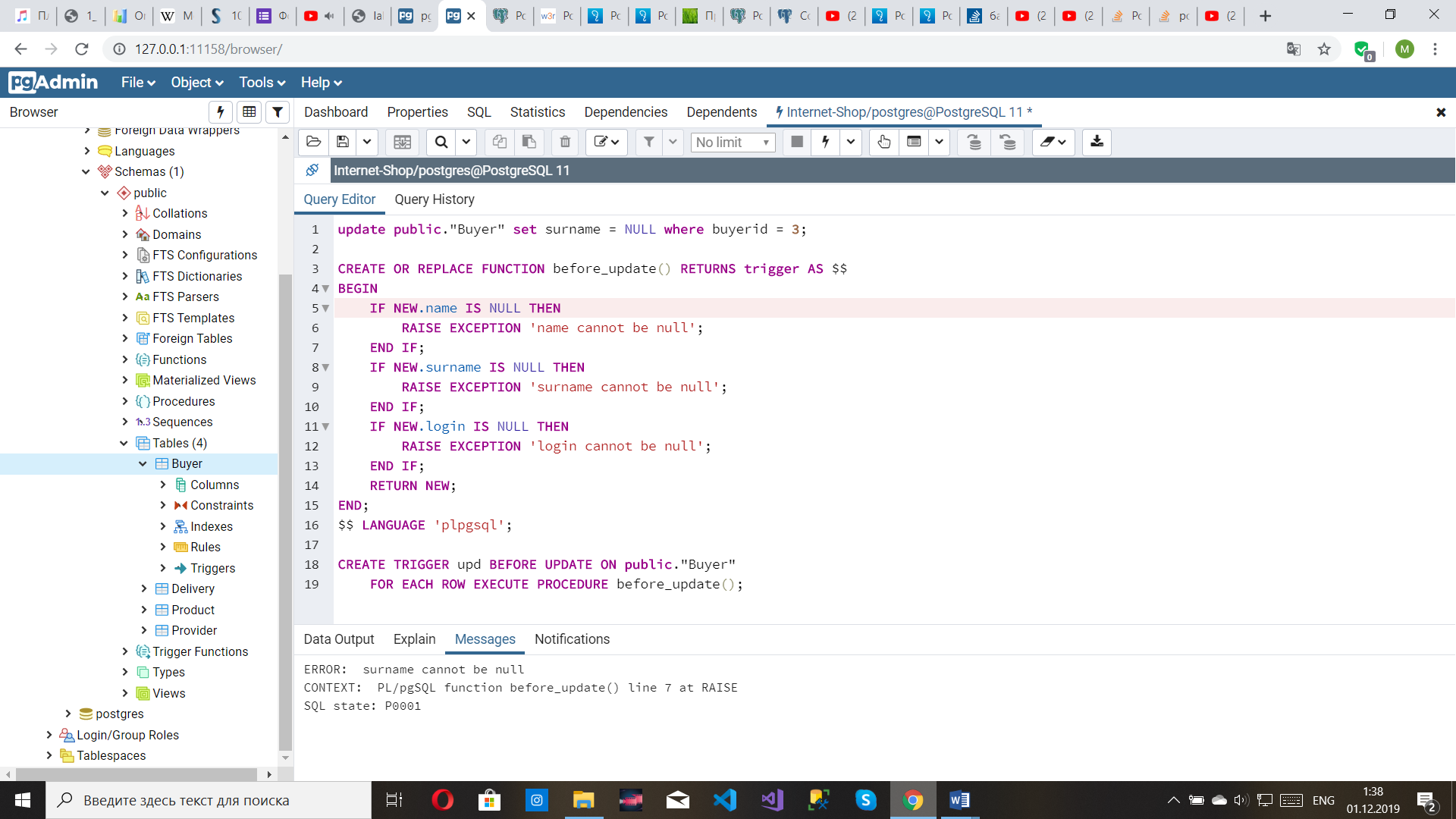
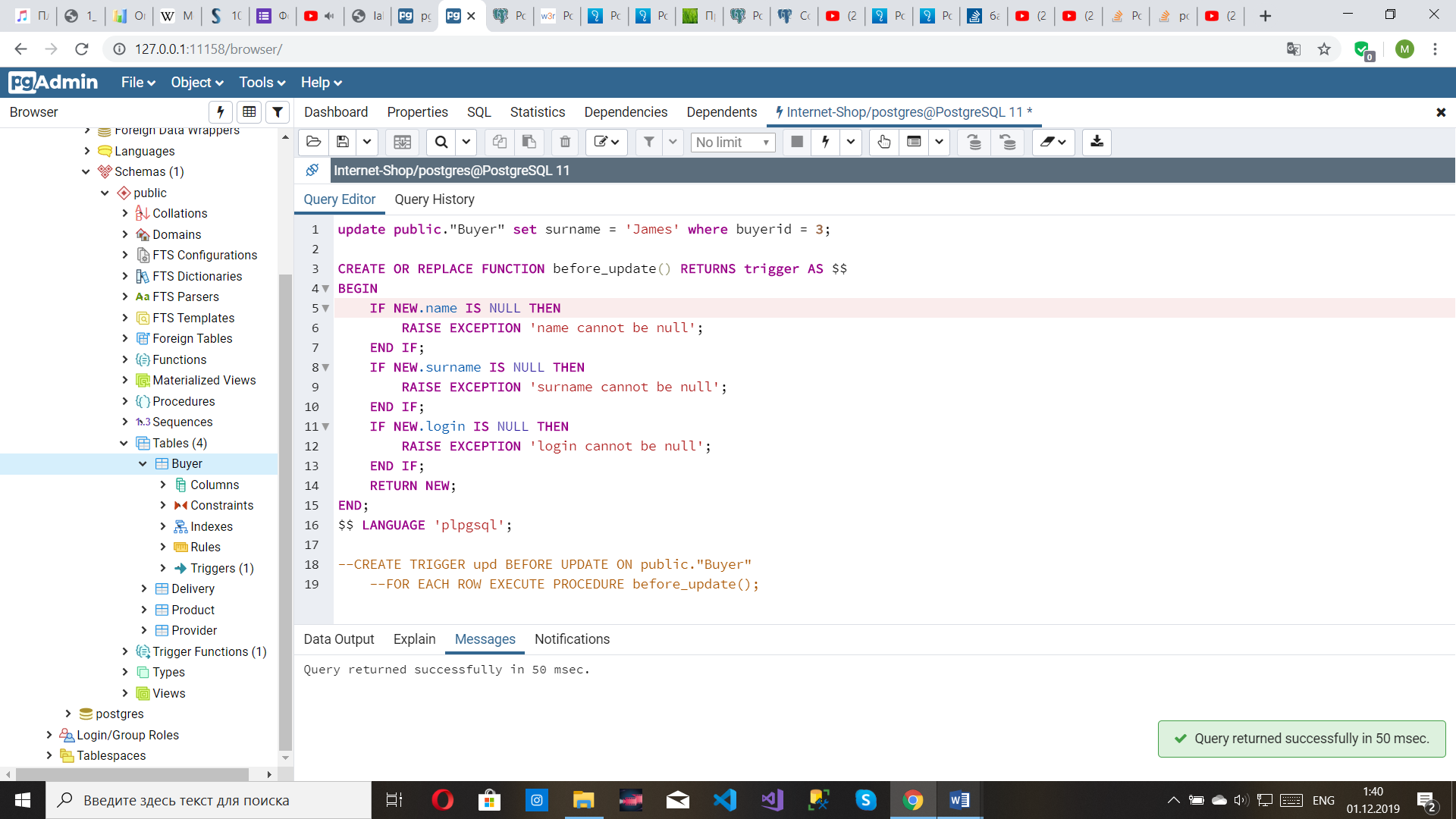


Рис. 1.3

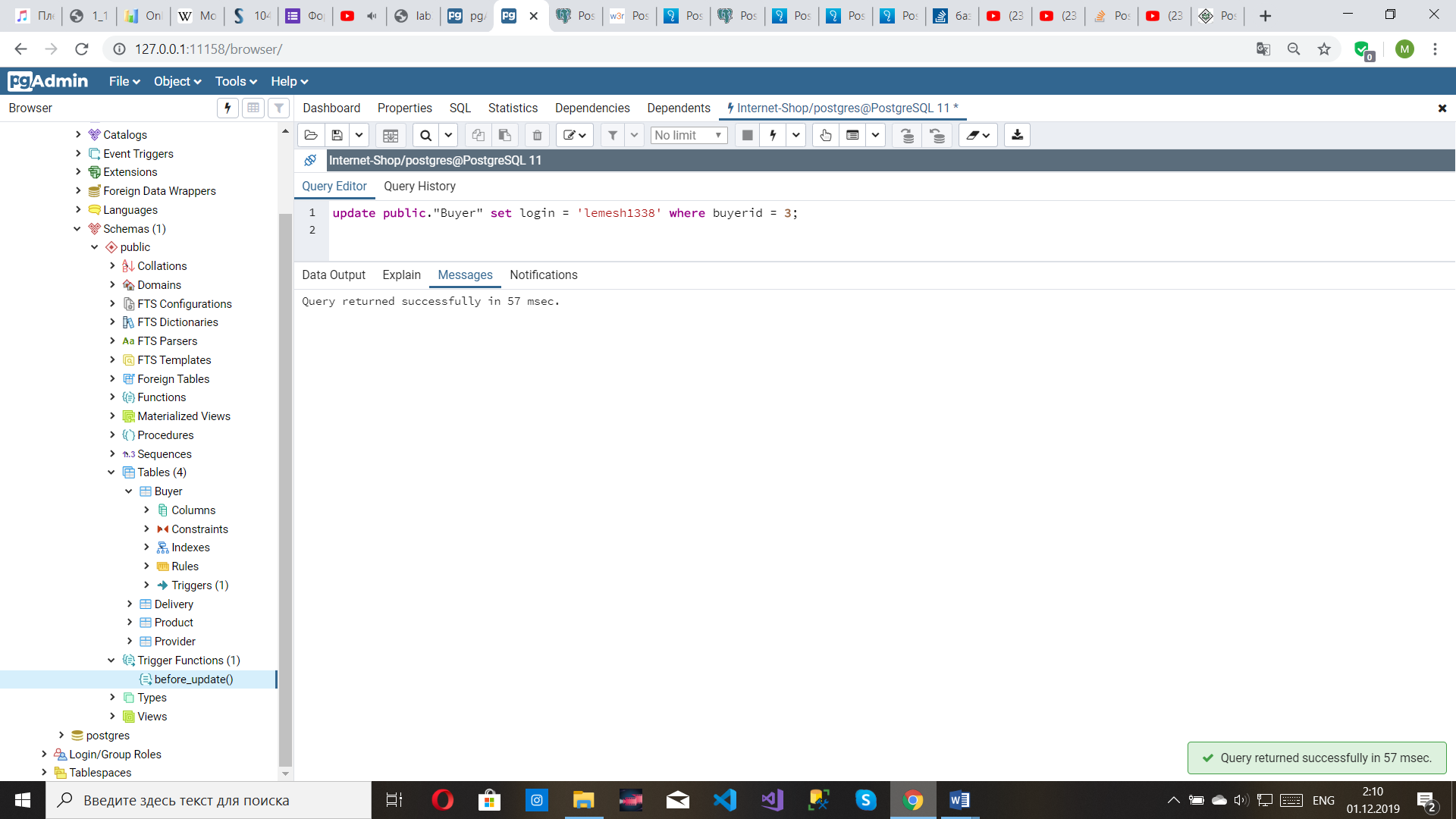
Успішне виконання операції:



Приклад запиту з використанням курсору і намаганням оновлення логіну, за умови, що такий логін існує в системі:

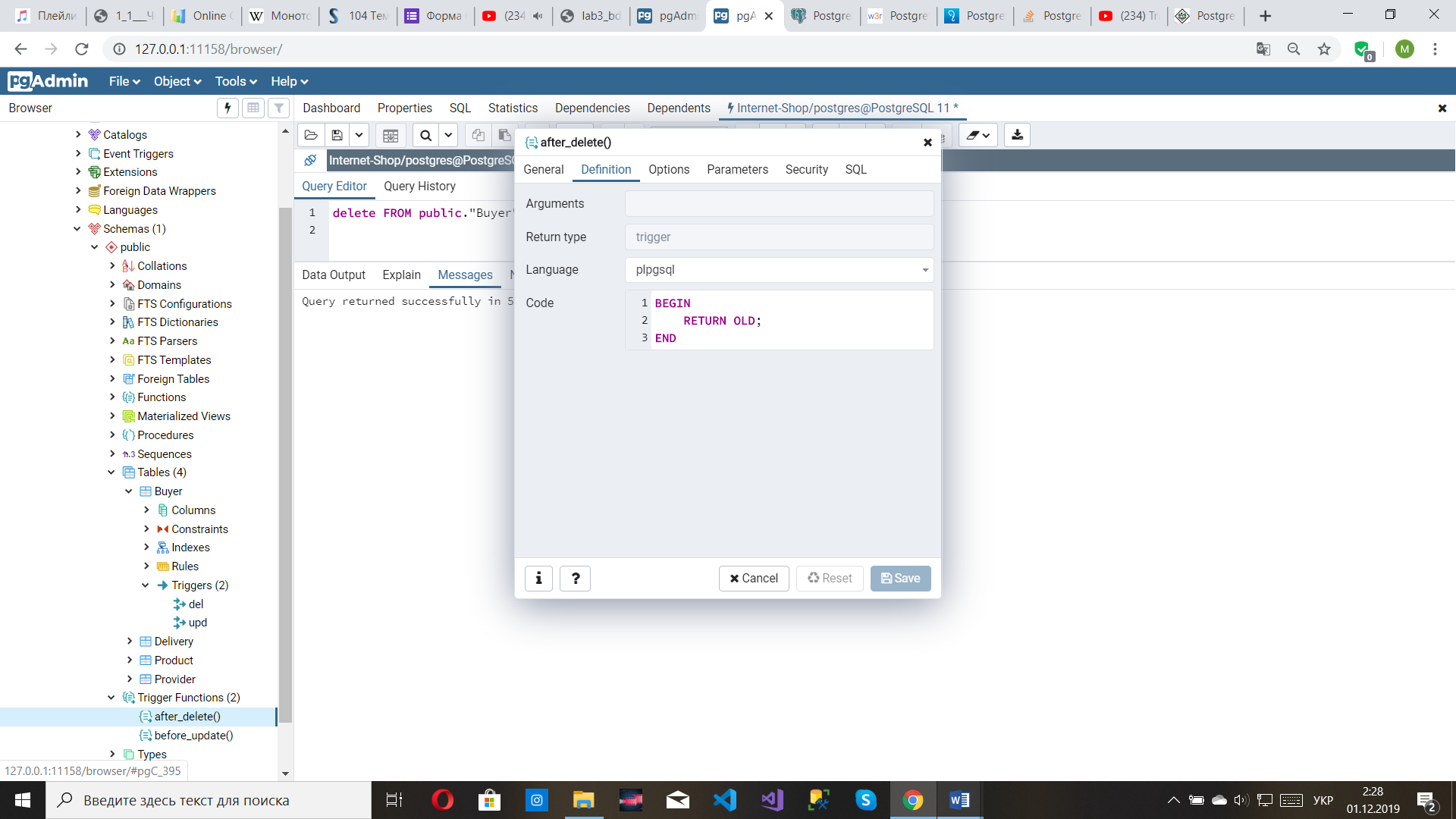


Вставка з неіснуючим логіном:

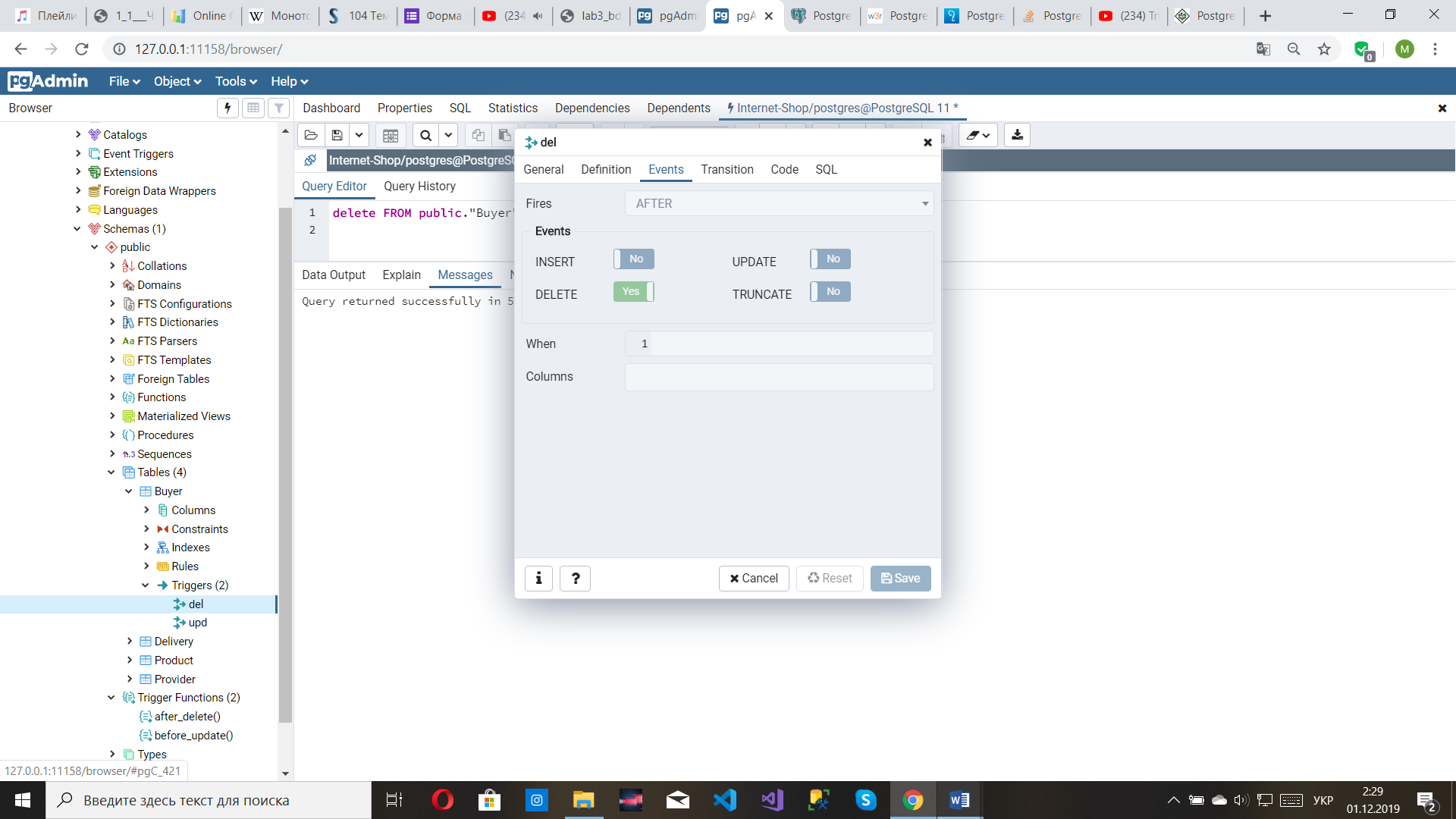


*DELETE*

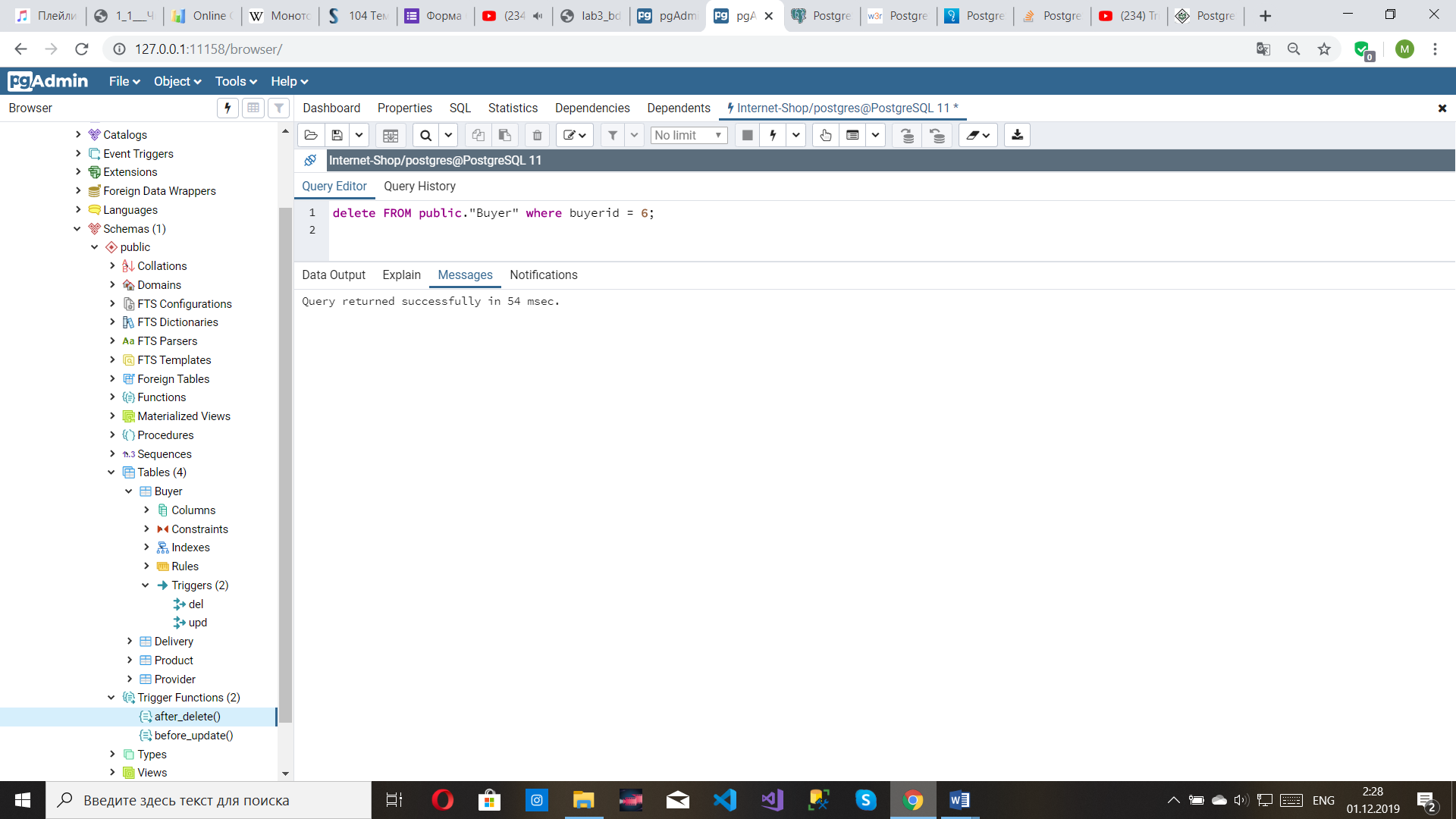
Функція:



Тригер:

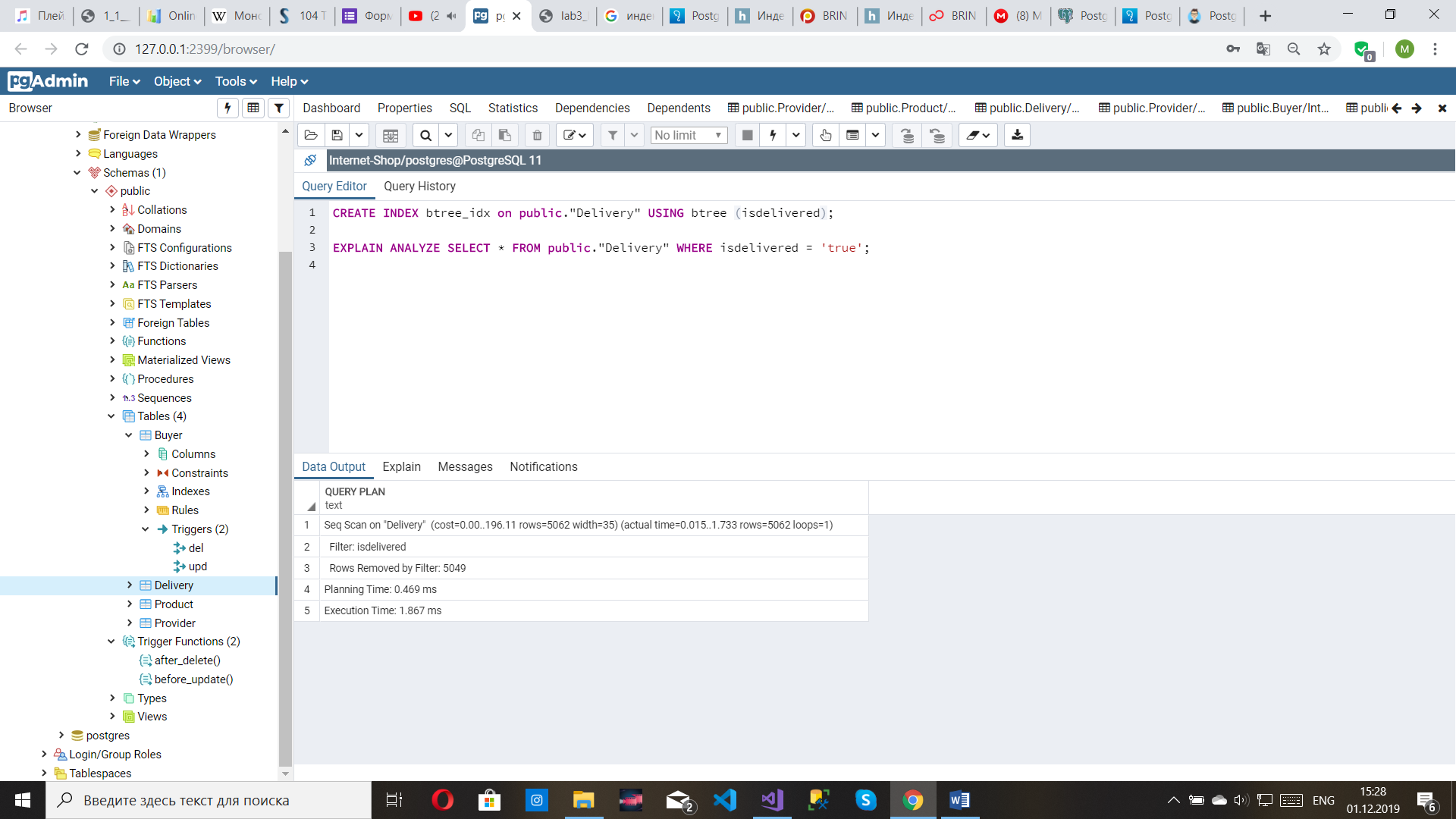


Результати виконання:

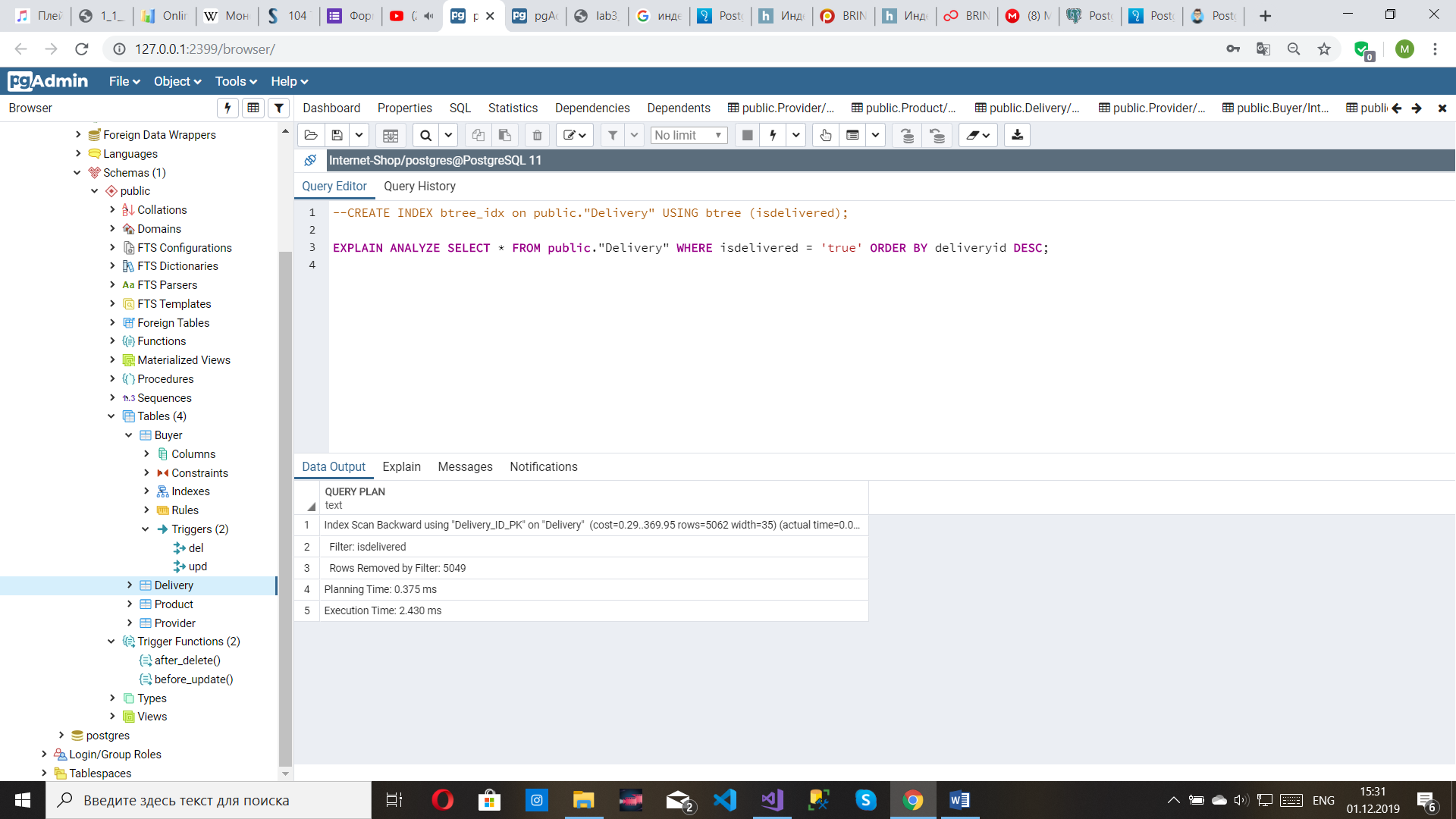


***Завдання 2:***

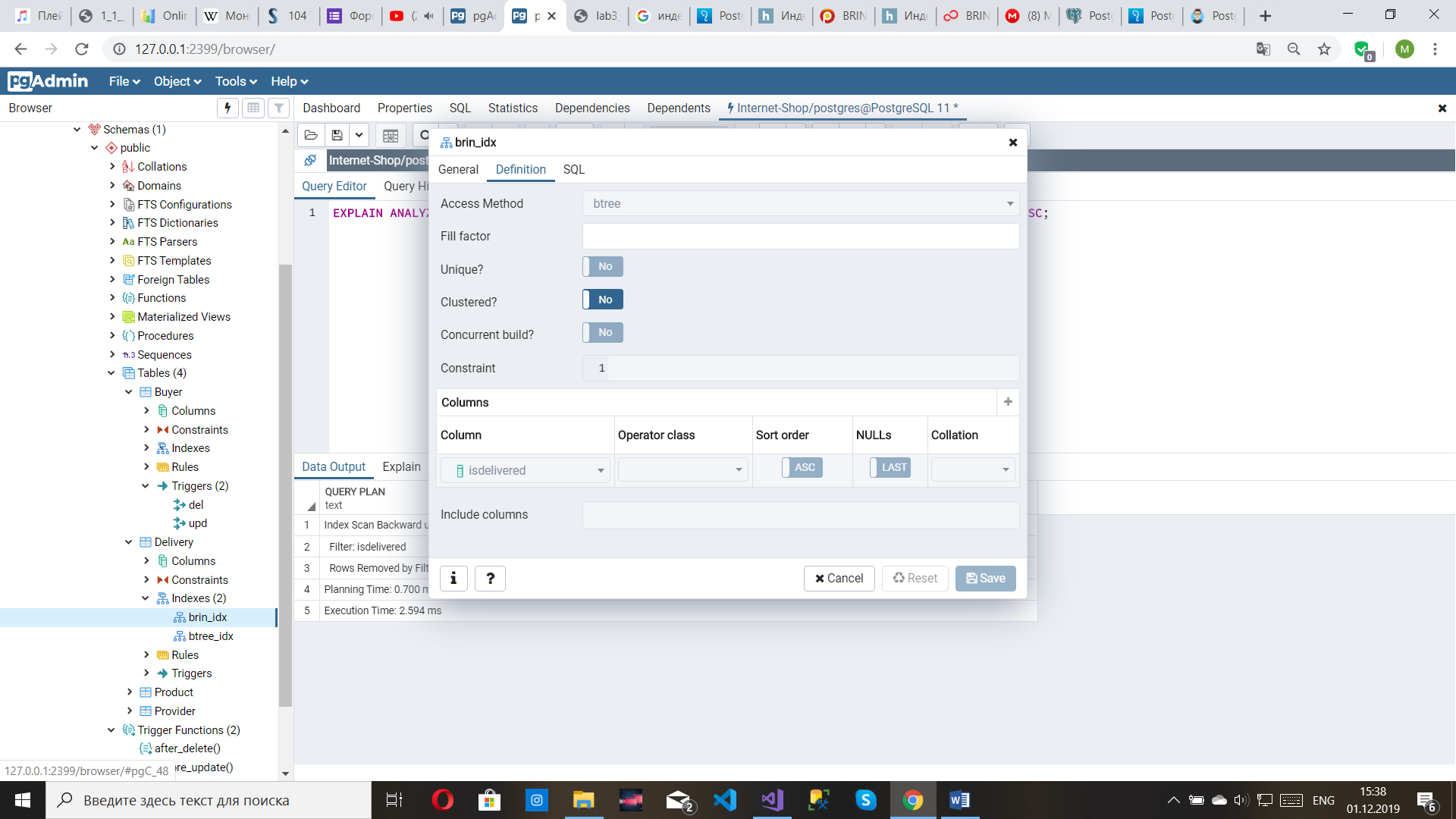
Створення *b-tree* індексу і виконання запиту *Select* з фільтрацією.



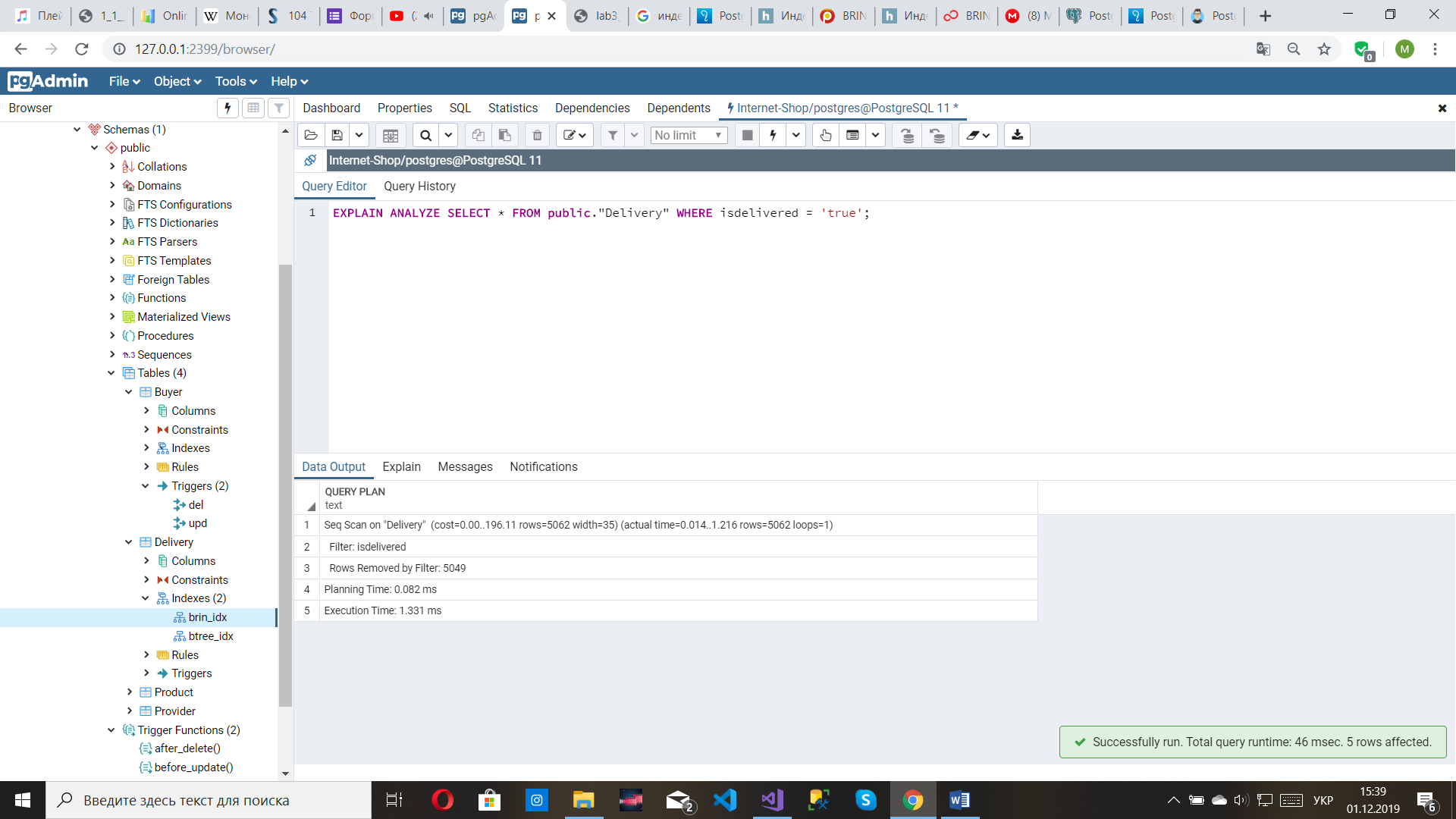
Створення *b-tree* індексу і виконання запиту *Select* з фільтрацією і сортуванням.



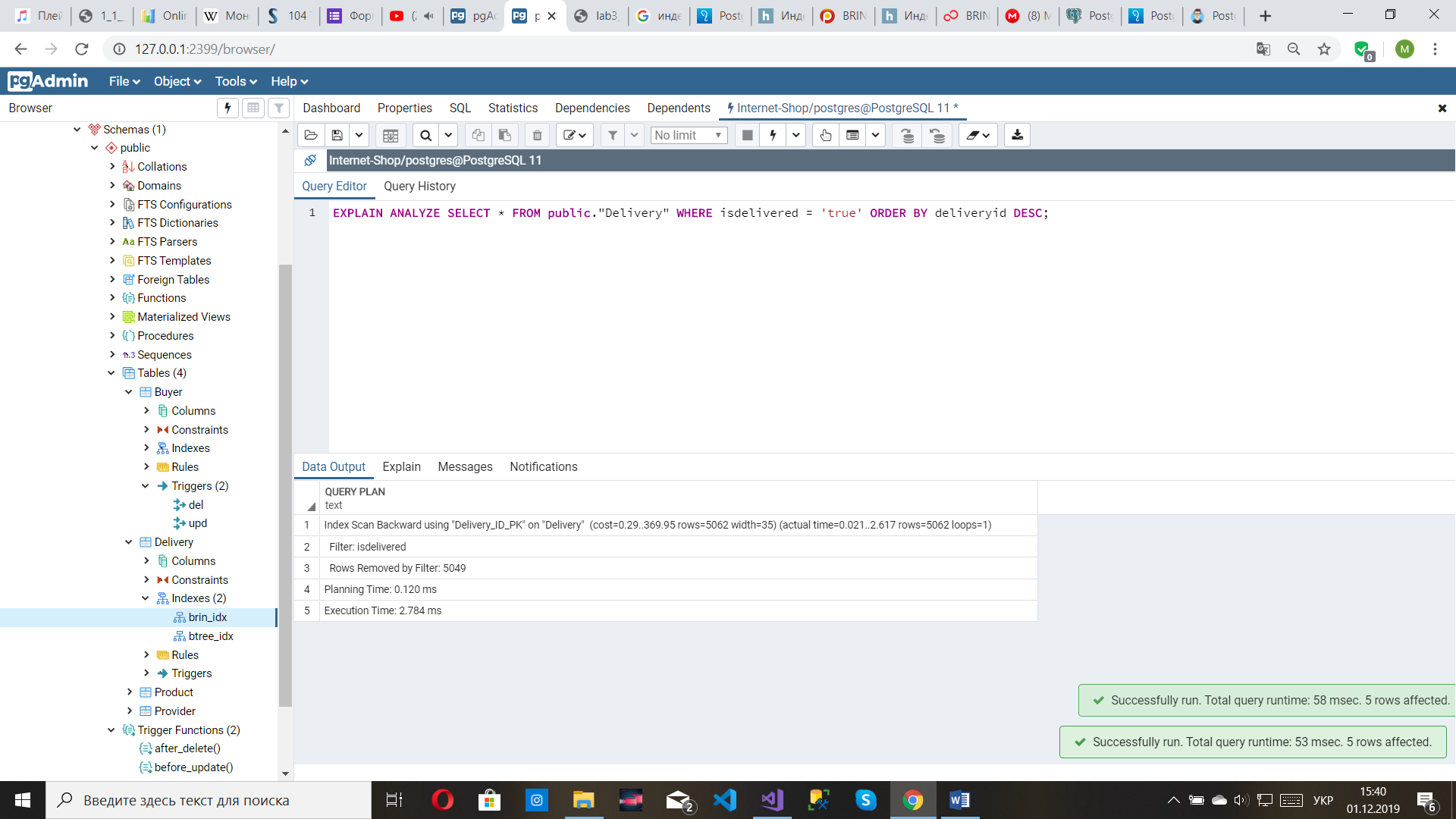
Створення BRIN індексу:



Виконання запиту для BRIN індексу:



Виконання другого запиту з сортуванням:



Як бачимо, очікувано, що час виконання при використанні BRIN індексу значно менший за час виконання з використанням Btree індексу.