

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
Факультет прикладної математики
Кафедра прикладної математики

**Звіт
лабораторної роботи №2**

із дисципліни «Бази даних» на тему
«Реалізація каунтера з використанням PostgreSQL»

Виконала:
студентка групи КМ-21
Челюскіна Ю.Ф.

Керівник:
асистент Дмитренко О. А.

Зміст

1	Мета	3
2	Постановка задачі	3
3	Особливості методів	3
4	Хід роботи.....	4
4.1	Тест для одного користувача	4
4.2	Тест для 100 000 користувачів.....	5

1 МЕТА

Метою цієї роботи є реалізація різних способів оновлення значення лічильника (counter) в базі даних PostgreSQL для одного або декількох користувачів, а також оцінка часу виконання кожного з варіантів. Це дозволяє порівняти ефективність різних методів оновлення в умовах багатозадачності та високої конкуренції.

2 ПОСТАНОВКА ЗАДАЧІ

Необхідно реалізувати різні методи оновлення значення лічильника в PostgreSQL (Lost Update, In-place Update, Row-level Locking, Optimistic Concurrency Control). Порівняти час виконання кожного методу при паралельному доступі, а також в оцінці їх ефективності та коректності роботи в умовах високої конкуренції між потоками.

3 ОСОБЛИВОСТІ МЕТОДІВ

Lost-update Два потоки одночасно зчитують значення лічильника, потім обидва оновлюють його, що призводить до втрати одного з оновлень.

In-place update Лічильник оновлюється без попереднього зчитування значення — просто збільшується на 1 в SQL-запиті для конкретного користувача.

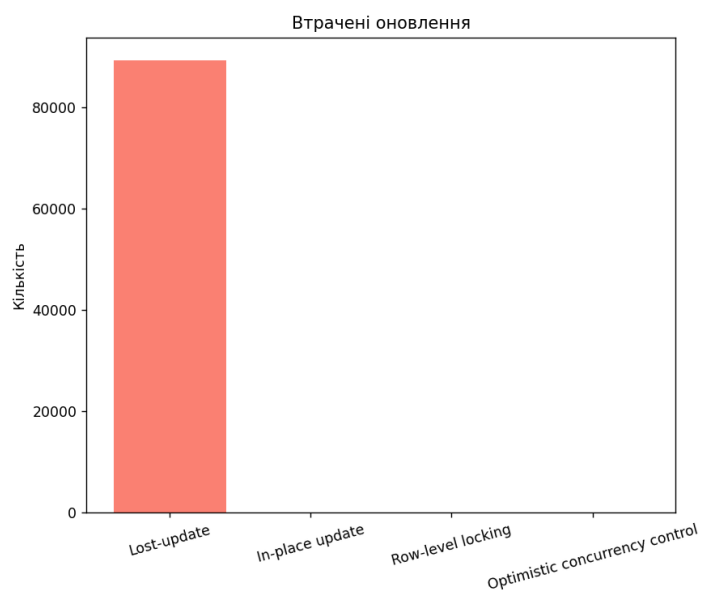
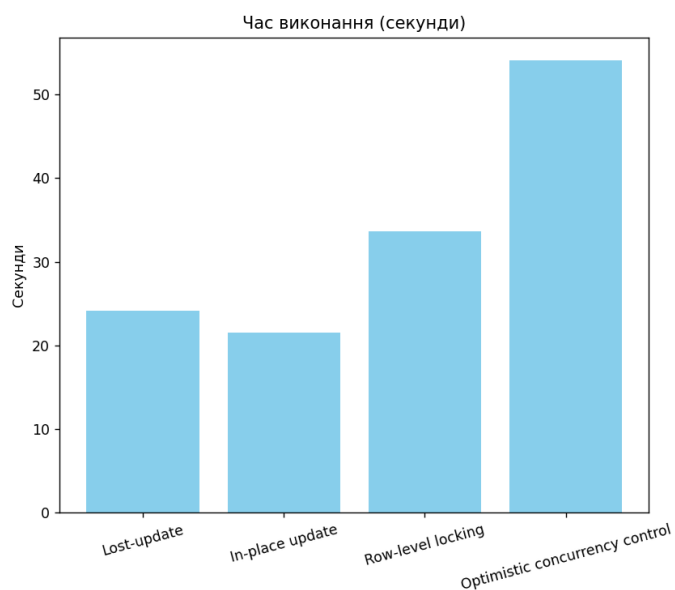
Row-level locking Використовує блокування на рівні рядка через SELECT ... FOR UPDATE, щоб забезпечити виключний доступ до рядка під час оновлення. Блокування може суттєво знизити продуктивність при великій кількості запитів, оскільки інші потоки або клієнти повинні чекати на звільнення рядка, що може призвести до затримок. (у фінансових системах, банках, сховищах товарів, де важлива 100% точність оновлень)

Optimistic concurrency control Включає перевірку версії даних, щоб уникнути конфліктів. Кожен потік намагається оновити лічильник, перевіряючи, чи не змінили його інші потоки з моменту останнього зчитування. Потрібно додатково управляти версіями, що ускладнює реалізацію.

4 ХІД РОБОТИ

4.1 ТЕСТ ДЛЯ ОДНОГО КОРИСТУВАЧА

```
-----  
Lost-update is running  
Time: 0:00:34.666456  
Counter value: 10696  
-----  
In-place update is running  
Time: 0:00:20.978037  
Counter value: 100000  
-----  
Row-level locking is running  
Time: 0:00:33.759259  
Counter value: 100000  
-----  
Optimistic concurrency control is running  
Time: 0:00:46.115436  
Counter value: 100000
```



4.2 ТЕСТ ДЛЯ 100 000 КОРИСТУВАЧІВ

```
-----  
Lost-update is running  
Time: 0:00:14.761401  
Counter value: 99995  
-----  
In-place update is running  
Time: 0:00:10.380728  
Counter value: 100000  
-----  
Row-level locking is running  
Time: 0:00:14.979569  
Counter value: 100000  
-----  
Optimistic concurrency control is running  
Time: 0:00:14.948783  
Counter value: 100000  
-----
```

