Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Навчально-науковий інститут атомної та теплової енергетики

Кафедра цифрових технологій в енергетиці

**Звіт**

**до лабораторної роботи № 5**

**з дисципліни «Проектування та використання баз даних»**

**Тема : «Робота з блокуванням та транзакцією»**

Виконав:

студент групи ТР-23

Ровний Григорій

*Перевірив :*

асистент

Головакін Микита Андрійович

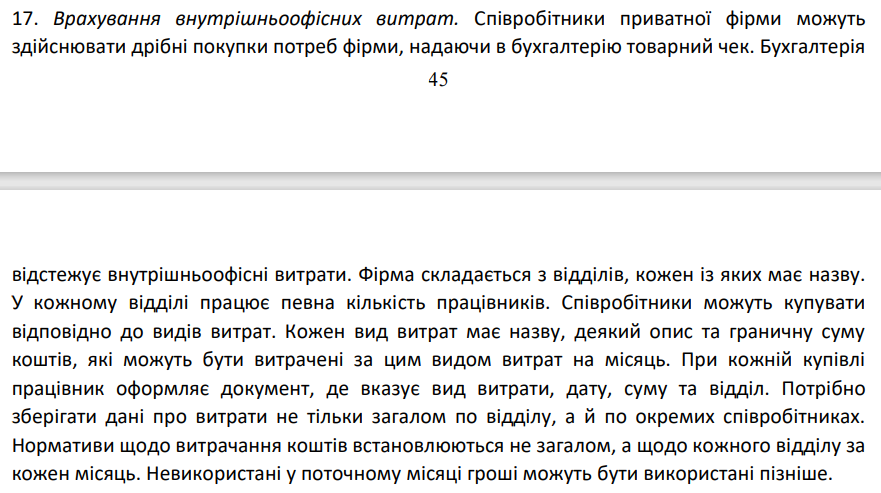
КИЇВ – 2024

**Мета лабораторної роботи :** Проаналізувати рівні ізоляції даних при роботі з транзакціями.

**Завдання:**

1. Cтворити двох корстувачів та продемонструвати їх паралельну роботу.
2. Написати транзакції та здійснити блокування для власної бази даних; 1 транзакція на оновлення даних в таблиці та її повний відкат; 1 транзакція із точкою відкату та субтранзакціями після неї та відкат до цієї точки; 1 блокування таблиці на читання, спроба перегляду даних та розблокування таблиці.
3. Зміна рівня ізоляції транзакцій на READ UNCOMMITED. 1 транзакція на оновлення даних в таблиці без явного режиму транзакцій. Перегляд внесених змін. Відкат цієї транзакції.

**Варіант**



**Результати виконання завдань:**

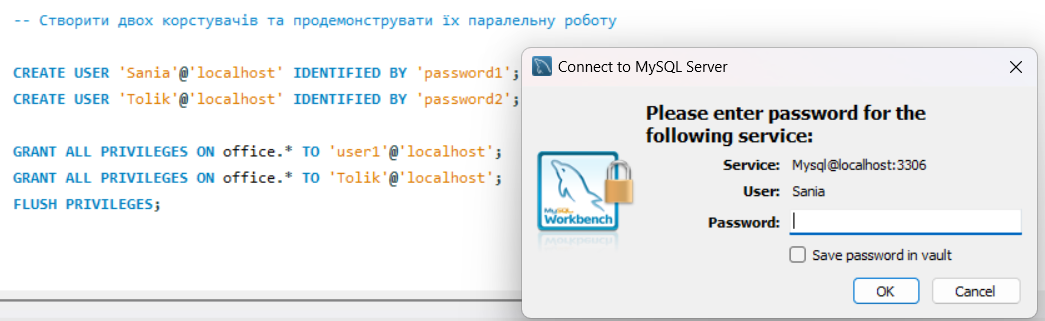
**Завдання 1:**

Створимо двох юзерів Sania і Tolik через команди:

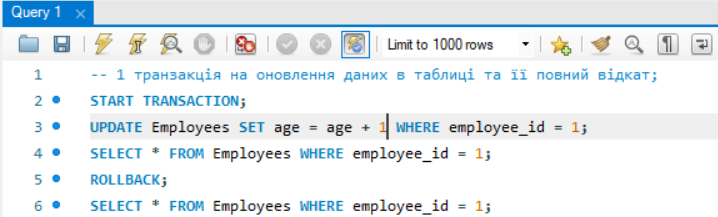
**CREATE USER 'Sania'@'localhost' IDENTIFIED BY 'password1';**

**CREATE USER 'Tolik'@'localhost' IDENTIFIED BY 'password2';**

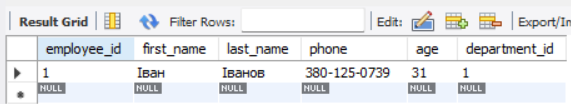
А також надамо їм права доступу. Після чого можна виконати авторизацію через Connect to Database. У кожного користувача буде своя сесія, що дозволить виконувати незалежні транзакції.



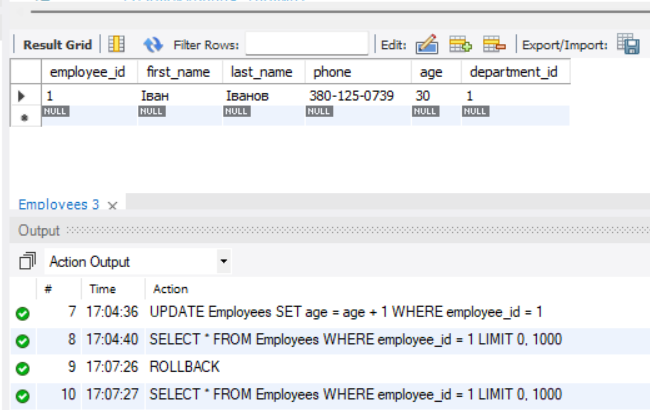
Створимо просту транзакцію, яка буде мати повний відкат.



Запустимо транзакцію, виконаємо **UPDATE** і побачимо, що в таблиці вік змінився на 31.



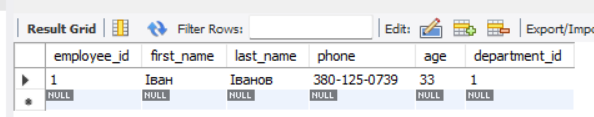
Виконаємо **ROLLBACK;** Знову переглянемо користувача в таблиці і побачимо що дані «відкотилися» назад.



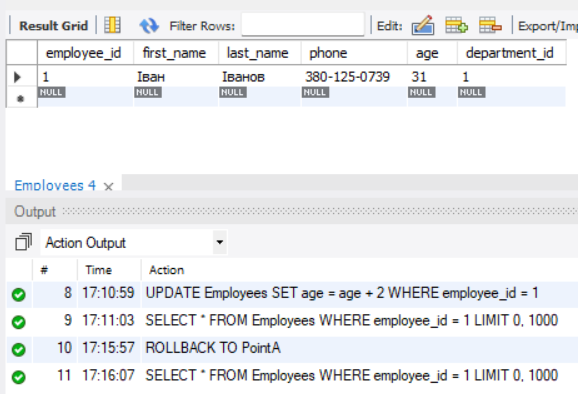
Далі зробимо транзакцію, яка буде мати точки збереження.



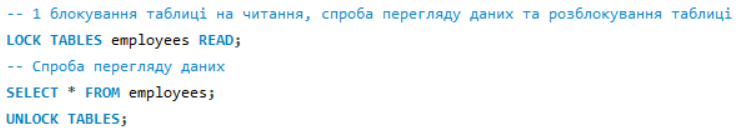
Запустимо транзакію. Зробимо такий самий апдейт і додамо точку відкату (**SAVEPOINT PointA**). Зробимо другий апдейт. Перевіримо дані в таблиці:



Після зробимо ролбек то точки А. І отримаємо ті зміни які були зроблені до першого SAVEPONT.



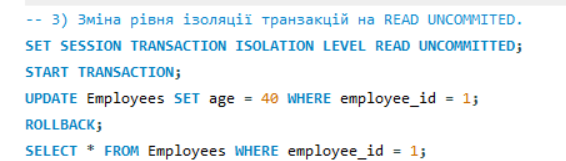
Зробимо блокування таблиці на читання через використання команди **LOCK TABLES employees READ;**:



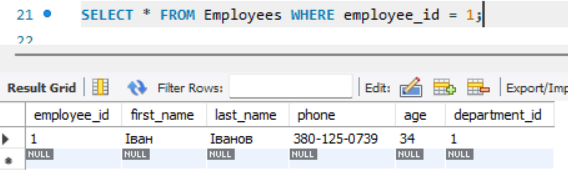
Якщо спробуємо вибрати дані з таблиці employees через **SELECT,** то отримаємо помилку:



**Зробимо зміну рівня ізоляції транзакцій на READ UNCOMMITED.**



Для початку зафіксуємо дані в таблиці employees для користувача 1. Зробимо вибірку:



Тепер встановимо **SET SESSION TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;** для зміни рівня ізоляції, запускаємо транзакіцю і знову робимо зміну віку на 40.

Переходимо до другого користувача. Встановимо йому такий самий рівень ізоляції і спробуємо зробити вирірку даних з таблиці для робітника з id = 1. В результаті ми можемо переглядати незакомічені дані, що і повинно відбуватись при такому рівні ізоляції, оскільки **READ UNCOMMITTED** дозволяє виконувати брудне читання. Для відкату просто скористаємось **ROLLBACK**;

