Міністерство освіти та науки України

Харківський національний університет радіоелектроніки

Кафедра Інформатики

Звіт

з лабораторної роботи №3

з дисципліни «Системи управління базами даних»

Виконав: Перевірив:

Ст.гр. ІТІНФ-20-1 доц. кафедри Інформатики

Самченко С. О. Кириченко І. Ю.

Харків 2023

**Мета роботи:** навчитися створювати та використовувати тригери рівня DML у Oracle.

**Завдання.**

1. Написати попередній тригер на зміну даних, що вставляються/видаляються/оновлюються в цій же таблиці.
2. Написати завершальний тригер, що призводить до змін в іншій таблиці.
3. Написати тригер із використанням автономної транзакції
4. Здійснити дії в таблицях, що призводять до запуску тригерів.

Все це має бути виконане ТІЛЬКИ засобами PL/SQL.

**Хід роботи:**

**База Даних:**

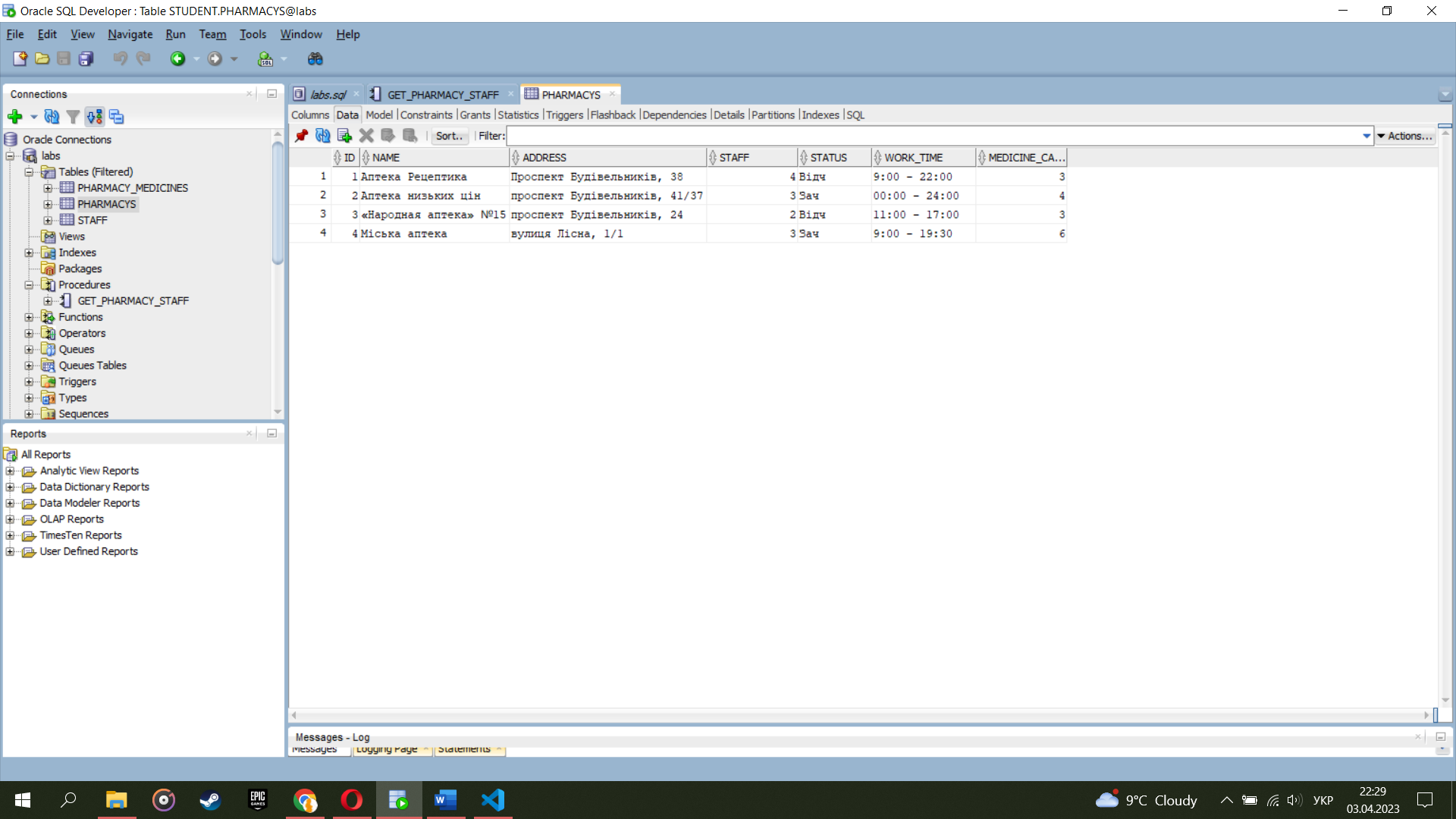


Рис. 1 – таблиця аптек

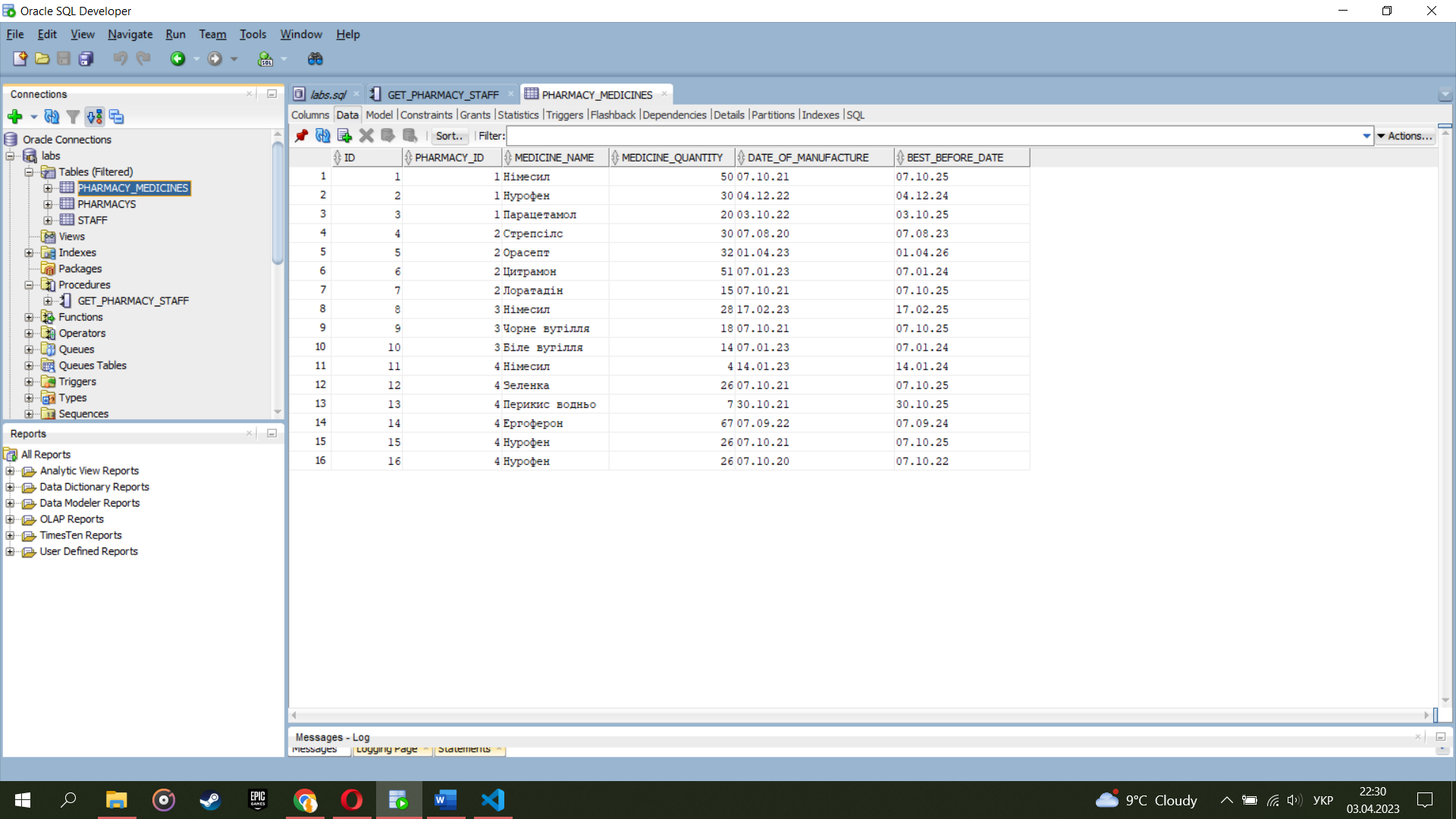


Рис. 2 – таблиця медикаментів

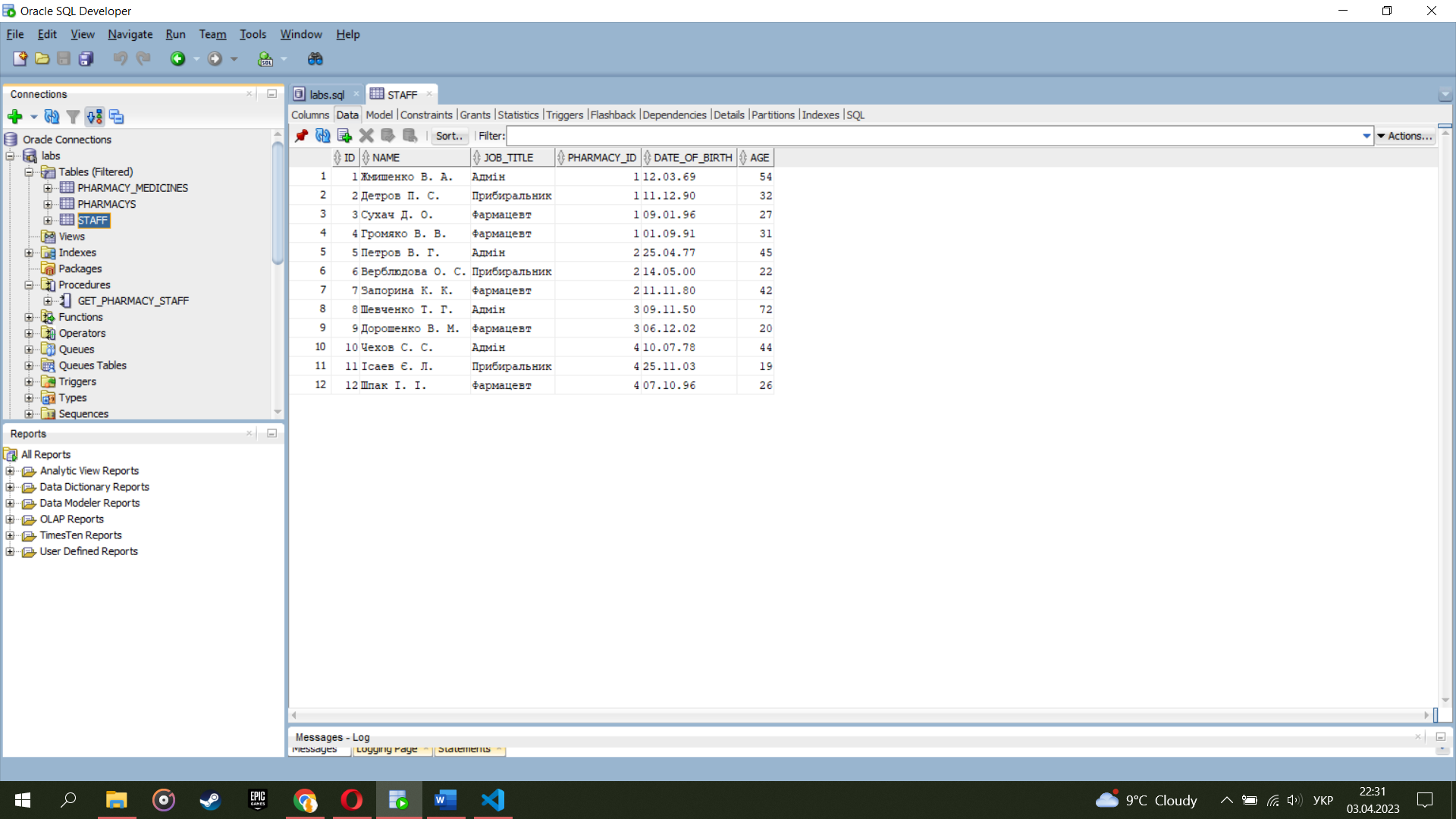


Рис. 3 – таблиця персоналу

**Попередній тригер на зміну даних, що вставляються/видаляються/оновлюються в цій же таблиці.**

CREATE OR REPLACE TRIGGER check\_medicine\_quantity

BEFORE INSERT OR UPDATE ON pharmacy\_medicines

FOR EACH ROW

BEGIN

  IF :NEW.medicine\_quantity <= 0 THEN

    RAISE\_APPLICATION\_ERROR(-20001, 'Количество медикаментов должно быть положительным.');

  END IF;

  IF :NEW.best\_before\_date < TRUNC(SYSDATE) THEN

    RAISE\_APPLICATION\_ERROR(-20002, 'Дата истечения срока годности должна быть больше или равна текущей дате.');

  END IF;

END;

/

Цей тригер називається "check\_medicine\_quantity" та виконується перед операціями INSERT або UPDATE на таблиці "pharmacy\_medicines". Він перевіряє дві умови:

1. Якщо значення "medicine\_quantity" в новому записі менше або рівні нулю, то тригер викликає помилку користувача з кодом -20001 і повідомленням "Кількість медикаментів повинна бути невід'ємною."
2. Якщо значення "best\_before\_date" у новому записі менше значення поточної дати, то тригер викликає помилку користувача з кодом -20002 і супутнім повідомленням.

Таким чином, цей тригер гарантує, що до таблиці "pharmacy\_medicines" додаватимуться тільки записи з невід'ємною кількістю медикаментів та правильними датами закінчення терміну придатності.

Результат:

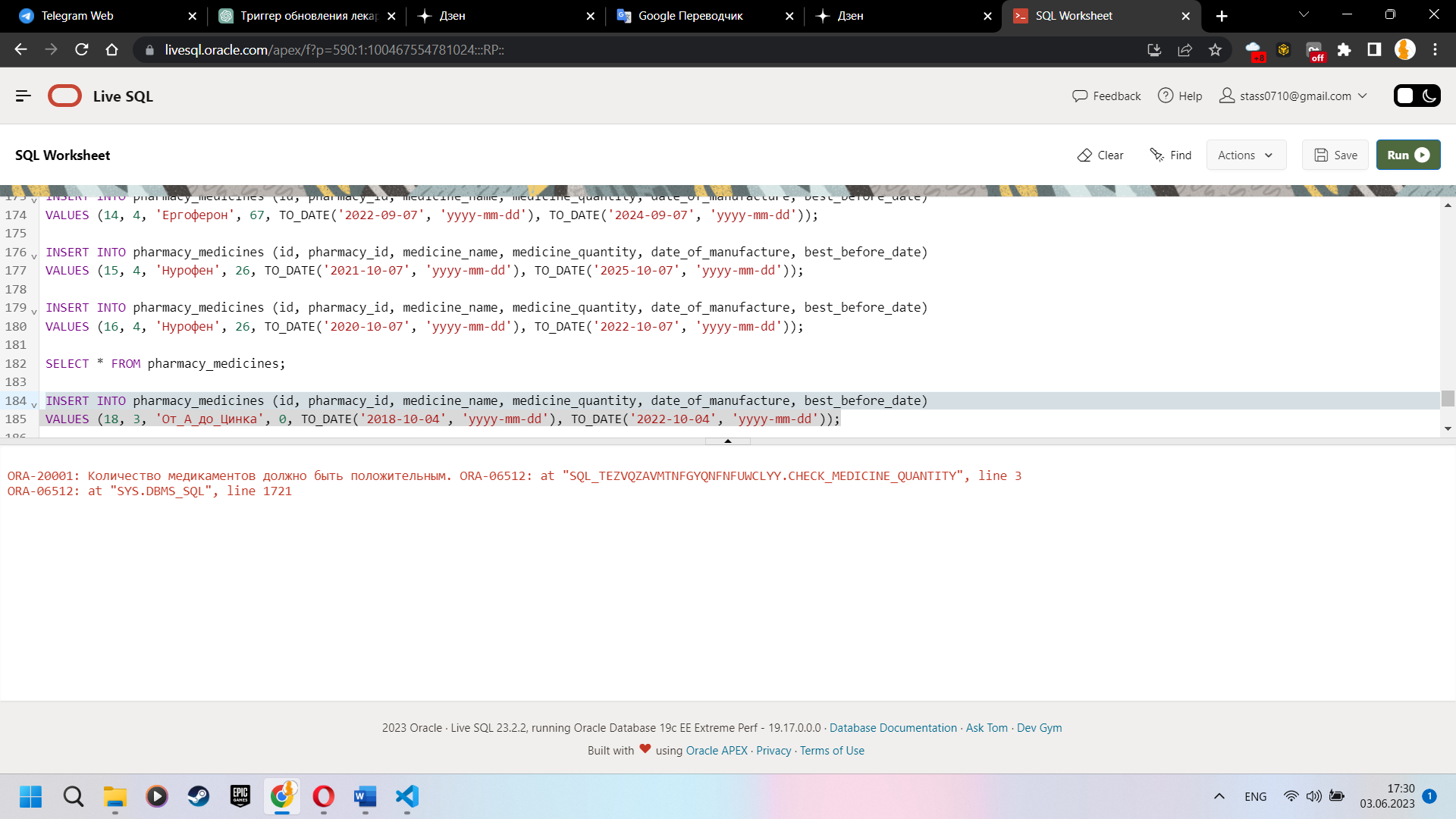


Рис. 4 – перевірка першої умови

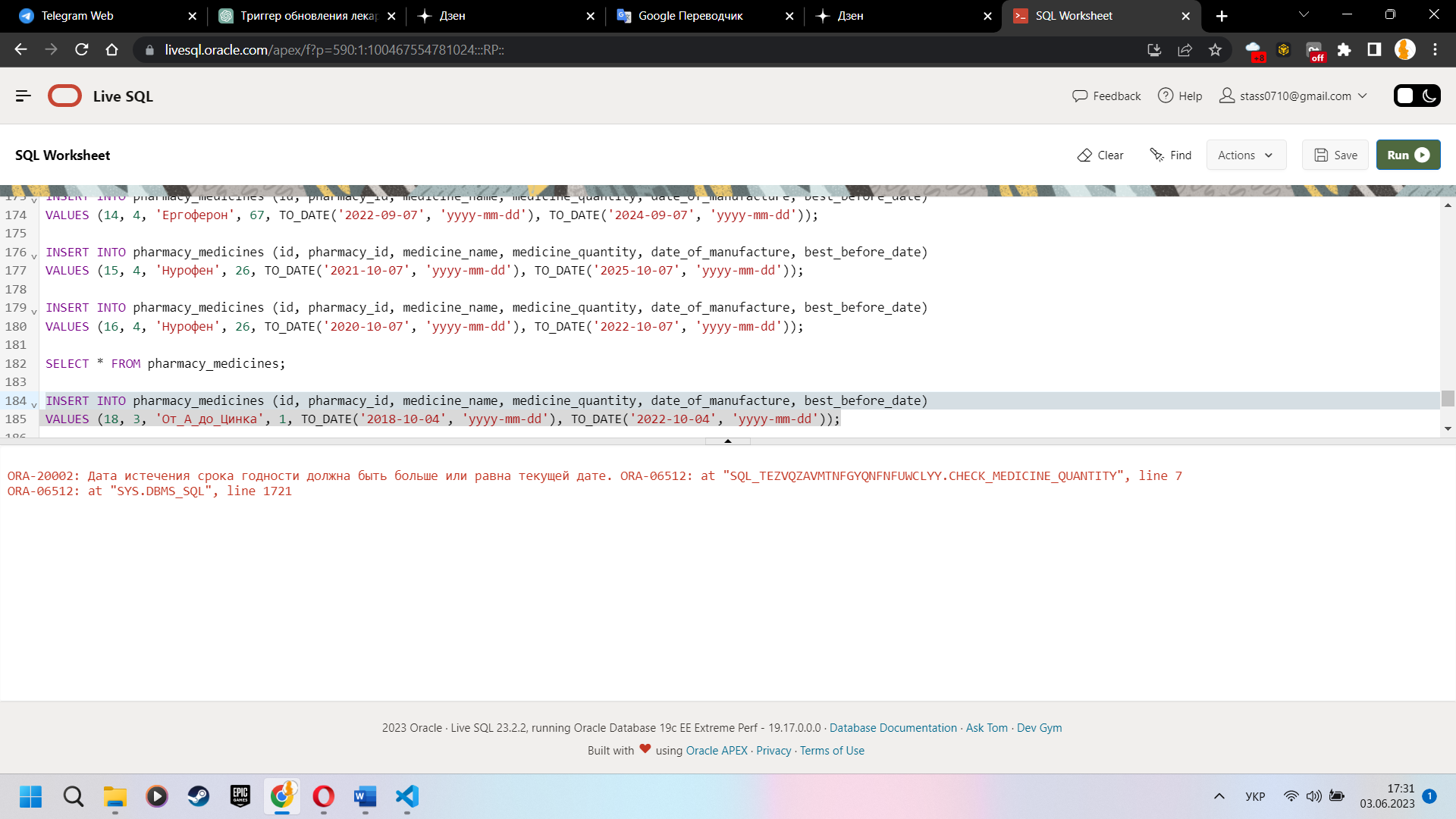


Рис. 5 – перевірка другої умови

**Написати завершальний тригер, що призводить до змін в іншій таблиці.**

CREATE OR REPLACE TRIGGER update\_pharmacy\_staff\_count

AFTER INSERT OR DELETE ON staff

FOR EACH ROW

BEGIN

  IF INSERTING THEN

    UPDATE pharmacys

    SET staff = staff + 1

    WHERE id = :NEW.pharmacy\_id;

  ELSIF DELETING THEN

    UPDATE pharmacys

    SET staff = staff - 1

    WHERE id = :OLD.pharmacy\_id;

  END IF;

END;

/

У цьому прикладі тригер спрацьовує після кожної операції додавання (INSERT) або видалення (DELETE) записів таблиці staff. Якщо виконується операція додавання (INSERTING), кількість співробітників (staff) у відповідній аптеці збільшується на 1. Якщо виконується операція видалення (DELETING), кількість співробітників в аптеці зменшується на 1.

Таким чином, при додаванні нового співробітника в таблицю staff кількість співробітників у відповідній аптеці автоматично збільшується. А при видаленні співробітника з таблиці staff кількість співробітників в аптеці зменшується. Це оновлення відбувається автоматично завдяки тригеру.

Результат:

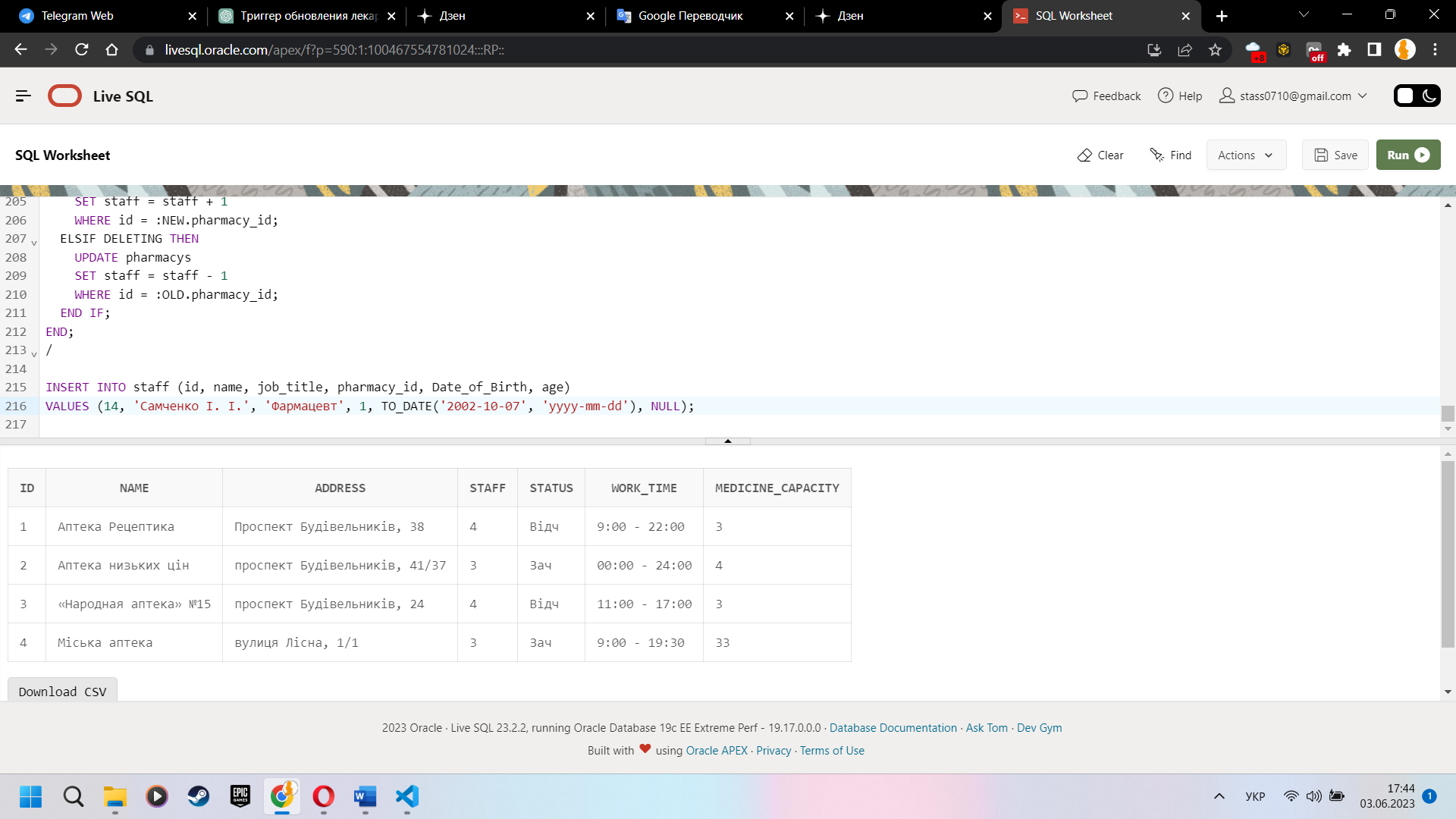


Рис. 6 – початкова таблиця

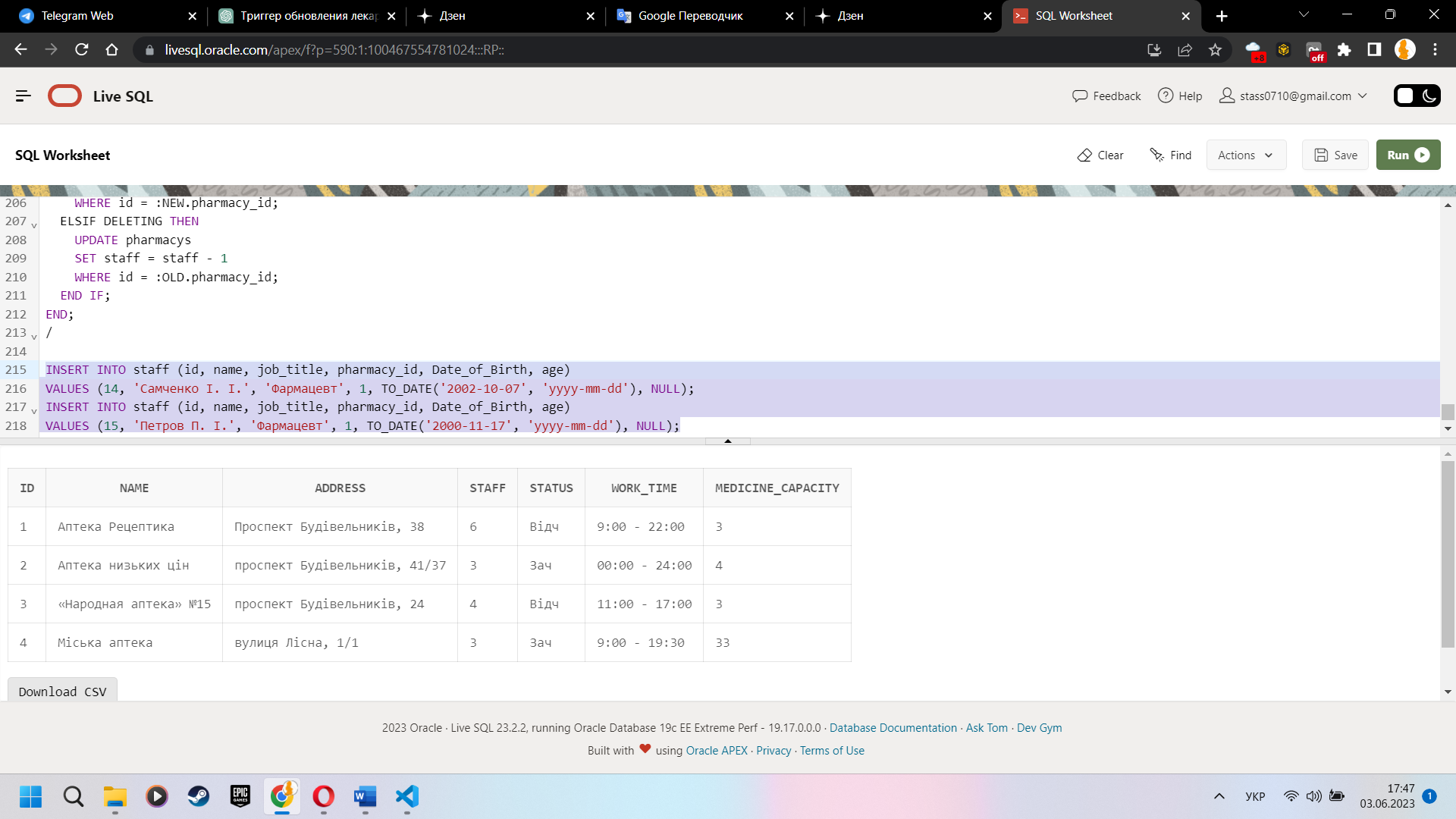


Рис. 7 – таблиця після виконання процедури вставки нових співробітників

**Написати тригер із використанням автономної транзакції**

CREATE OR REPLACE TRIGGER update\_medicine\_quantity

AFTER UPDATE ON pharmacy\_medicines

FOR EACH ROW

DECLARE

   PRAGMA AUTONOMOUS\_TRANSACTION;

BEGIN

   UPDATE pharmacys

   SET medicine\_capacity = (

      SELECT SUM(medicine\_quantity)

      FROM pharmacy\_medicines

      WHERE pharmacy\_id = :NEW.pharmacy\_id

   )

   WHERE id = :NEW.pharmacy\_id;

   COMMIT;

EXCEPTION

   WHEN OTHERS THEN

      ROLLBACK;

      RAISE;

END;

/

Цей тригер буде викликатись після кожного оновлення таблиці pharmacy\_medicines. Він автоматично оновить загальну кількість медикаментів у відповідній аптеці (medicine\_capacity) на основі зміненої кількості медикаментів (medicine\_quantity) у таблиці pharmacy\_medicines. Використовується автономна транзакція (PRAGMA AUTONOMOUS\_TRANSACTION), щоб фіксувати зміни окремої транзакції.

Результат:

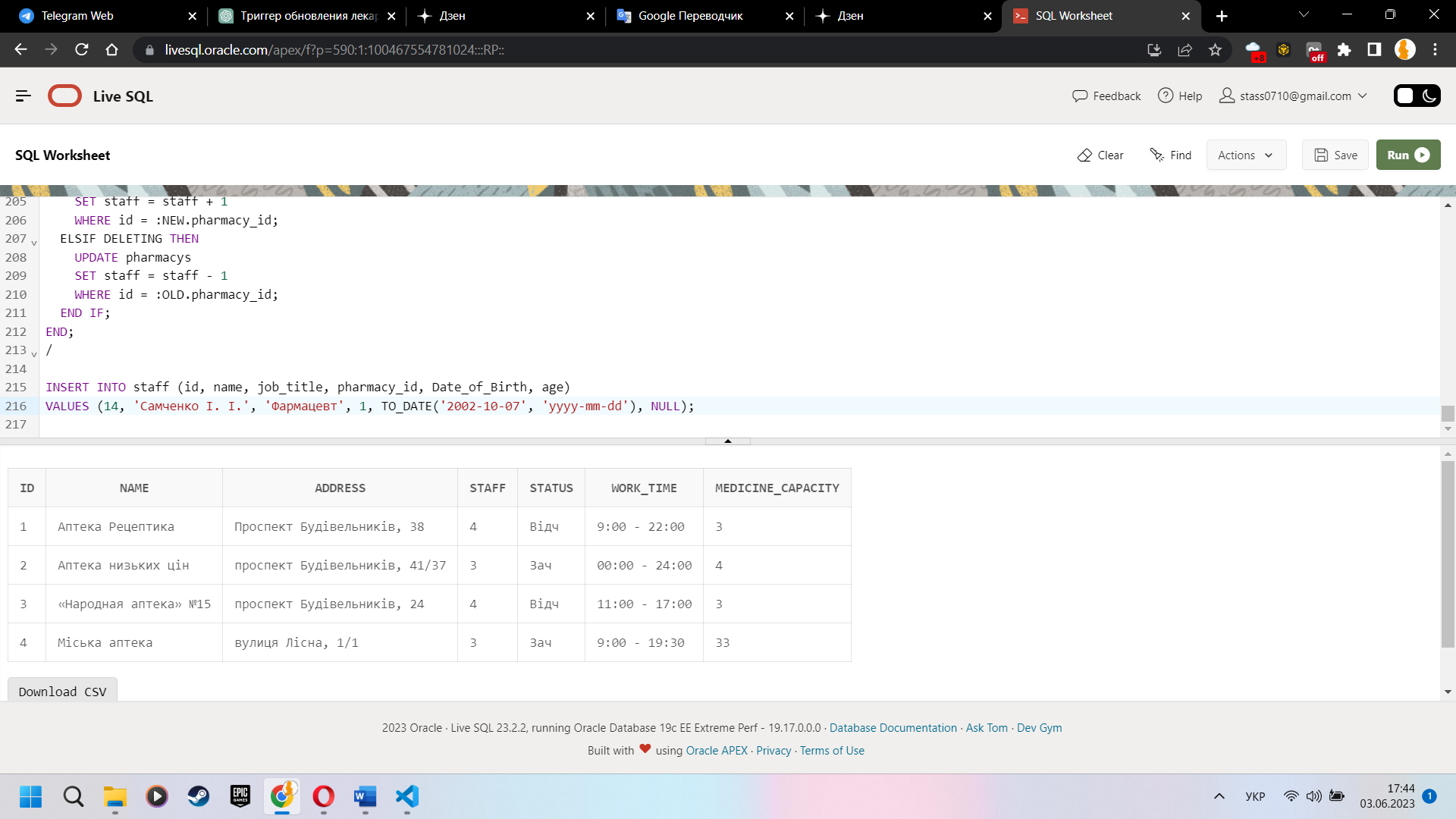


Рис. 8 – початкова таблиця

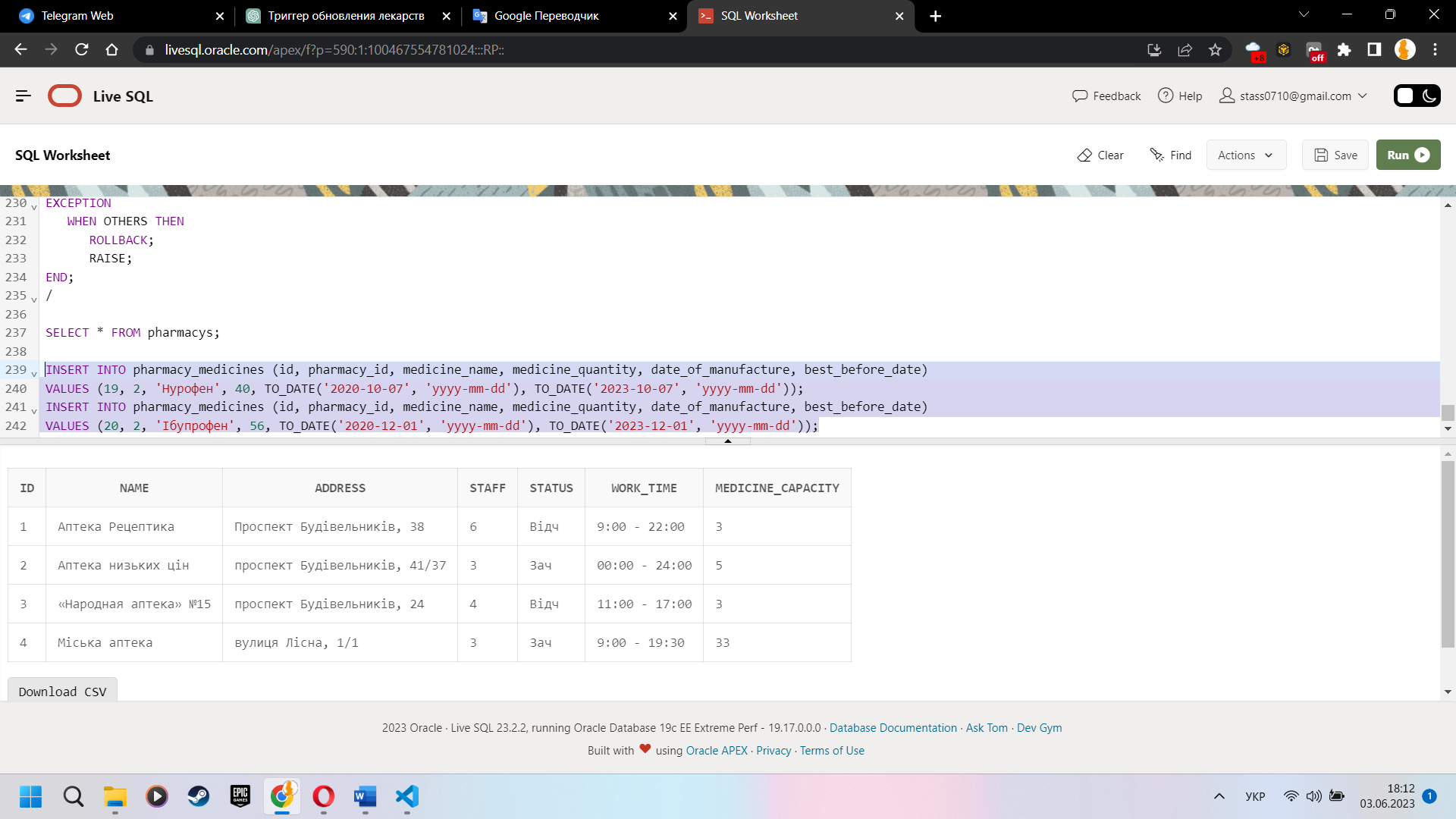


Рис. 9 – таблиця після виконання процедури вставки нових медикаментів

**Висновок:** у рамках даної роботи були здобуті практичні навички створювання та використання тригерів рівня DML у Oracle.