Міністерство освіти та науки України

Харківський національний університет радіоелектроніки

Кафедра Інформатики

Звіт

з лабораторної роботи №2

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 функції, процедури, пакети засобами PL/SQL, а також використовувати явні курсори.

**Завдання:**

1. Написати і виконати 10 складних запитів до своєї БД з використанням вбудованих функцій (5 з них повинні використовувати функції для роботи з датами)
2. Створити процедуру, що містить явний курсор.
3. Створити функцію на будь-яке із завдань бізнес-логіки.
4. Об'єднати процедуру та функцію в пакет.
5. Написати скрипт для виклику процедури та функції.

**База Даних:**

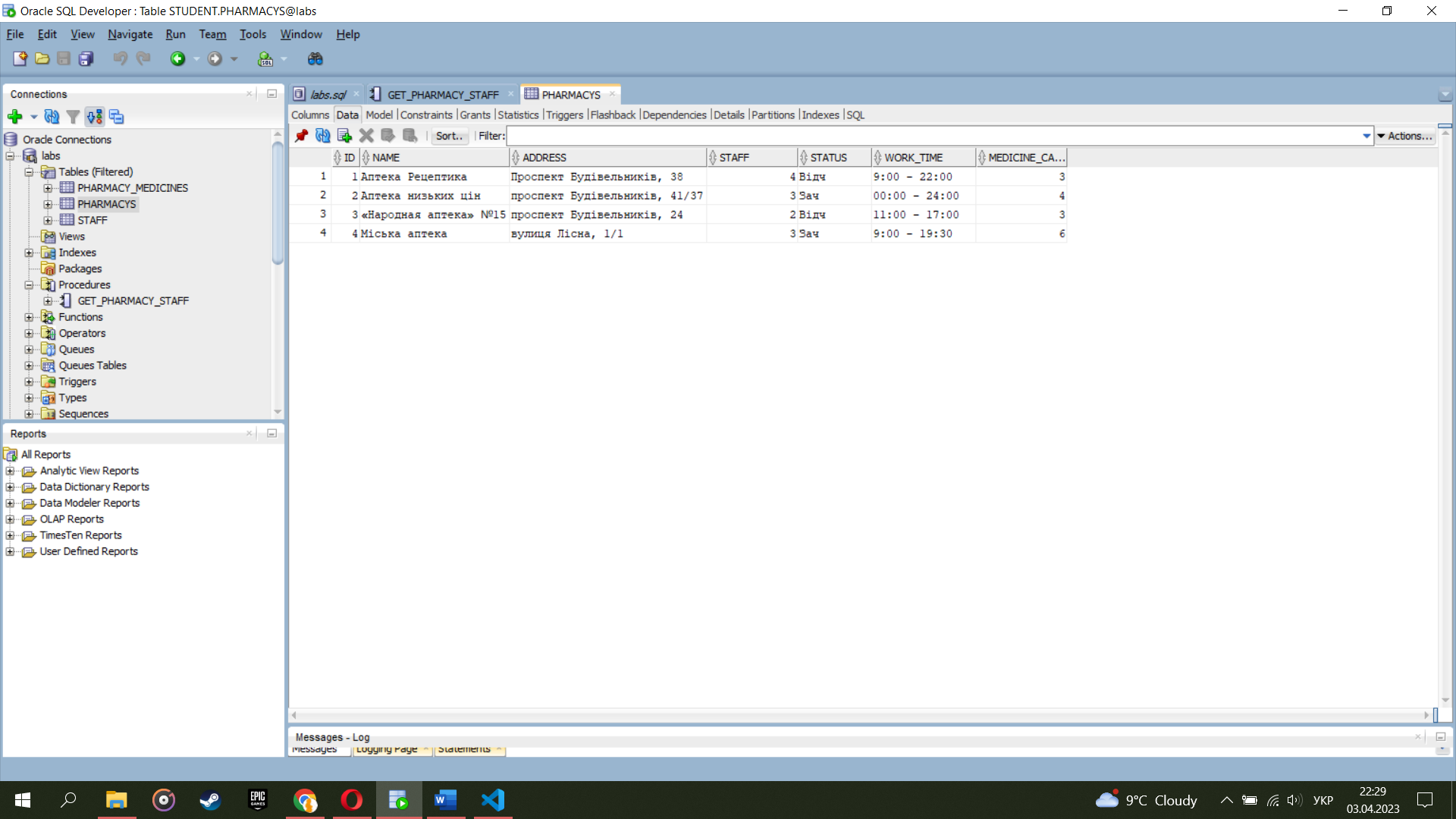


Рис. 1 – таблиця аптек

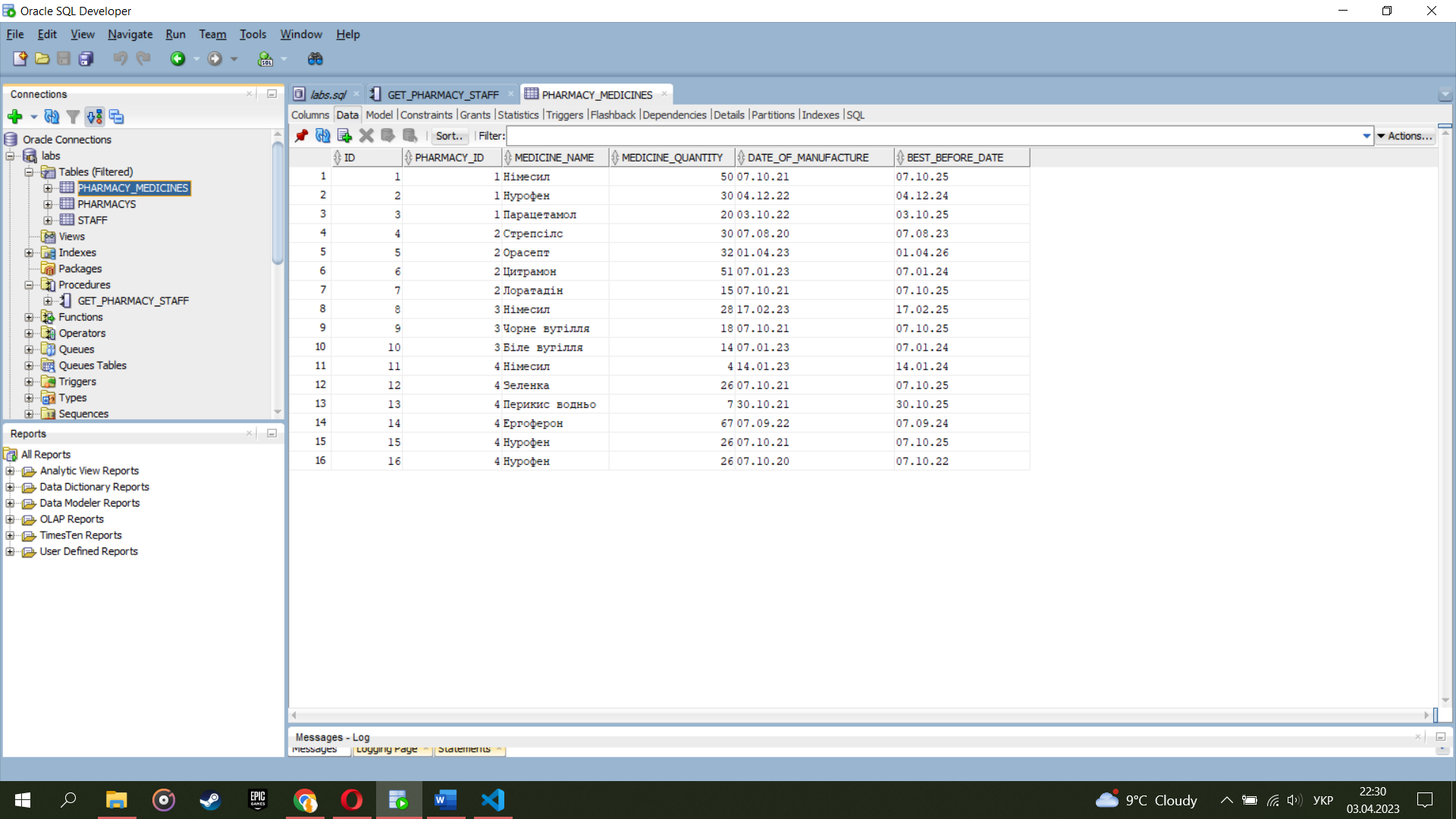


Рис. 2 – таблиця медикаментів

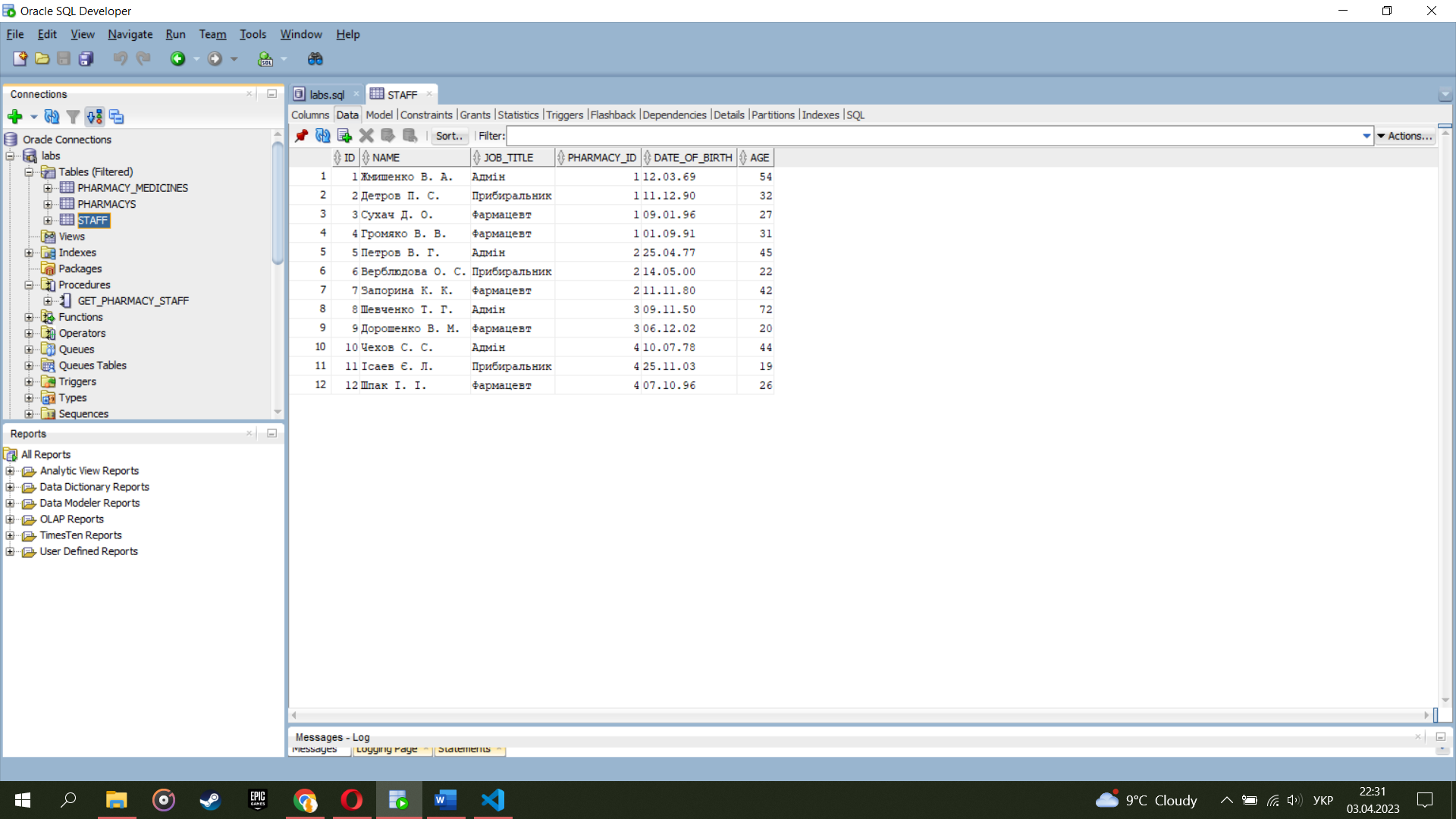


Рис. 3 – таблиця персоналу

**Скрипт процедури з курсором, функції, пакета, скрипт для їх запуску зі скріншотами результатів:**

* *Створити процедуру, що містить явний курсор:*

CREATE OR REPLACE PROCEDURE get\_pharmacy\_staff(

p\_pharmacy\_id IN pharmacys.id%TYPE,

p\_min\_age IN staff.age%TYPE

)

IS

CURSOR staff\_cursor IS

SELECT id, name, job\_title, age

FROM staff

WHERE pharmacy\_id = p\_pharmacy\_id AND age >= p\_min\_age;

v\_staff\_id staff.id%TYPE;

v\_staff\_name staff.name%TYPE;

v\_job\_title staff.job\_title%TYPE;

v\_age staff.age%TYPE;

BEGIN

OPEN staff\_cursor;

LOOP

FETCH staff\_cursor INTO v\_staff\_id, v\_staff\_name, v\_job\_title, v\_age;

EXIT WHEN staff\_cursor%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('Staff ID: ' || v\_staff\_id || ', Name: ' || v\_staff\_name || ', Job Title: ' || v\_job\_title || ', Age: ' || v\_age);

END LOOP;

CLOSE staff\_cursor;

END;

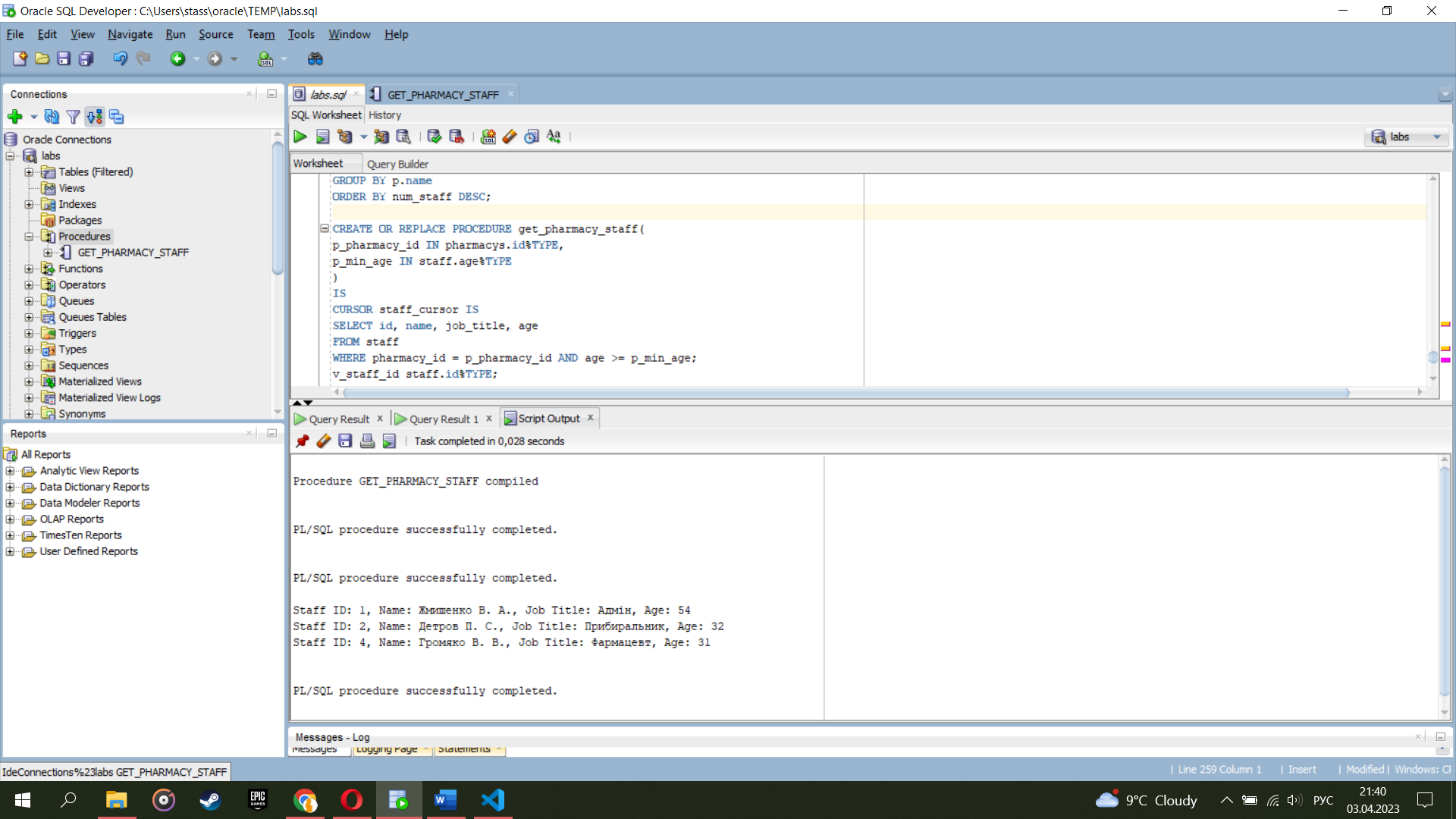
BEGIN

get\_pharmacy\_staff(1, 30);

END;

Процедура get\_pharmacy\_staff використовує явний курсор staff\_cursor для отримання інформації про співробітників аптеки із заданим ідентифікатором та віком, рівним або більшим за заданий мінімальний вік. Результати запиту зберігаються в курсорі staff\_cursor, потім перебираються в циклі. Для кожного рядка відображається інформація про співробітника за допомогою функції DBMS\_OUTPUT.PUT\_LINE. Курсор потім закривається.

Результат:



* *Створити функцію на будь-яке із завдань бізнес-логіки:*

CREATE OR REPLACE FUNCTION get\_oldest\_medicine

RETURN pharmacy\_medicines.medicine\_name%TYPE

IS

v\_oldest\_medicine pharmacy\_medicines.medicine\_name%TYPE;

BEGIN

SELECT medicine\_name INTO v\_oldest\_medicine

FROM pharmacy\_medicines

WHERE date\_of\_manufacture = (SELECT MIN(date\_of\_manufacture) FROM pharmacy\_medicines);

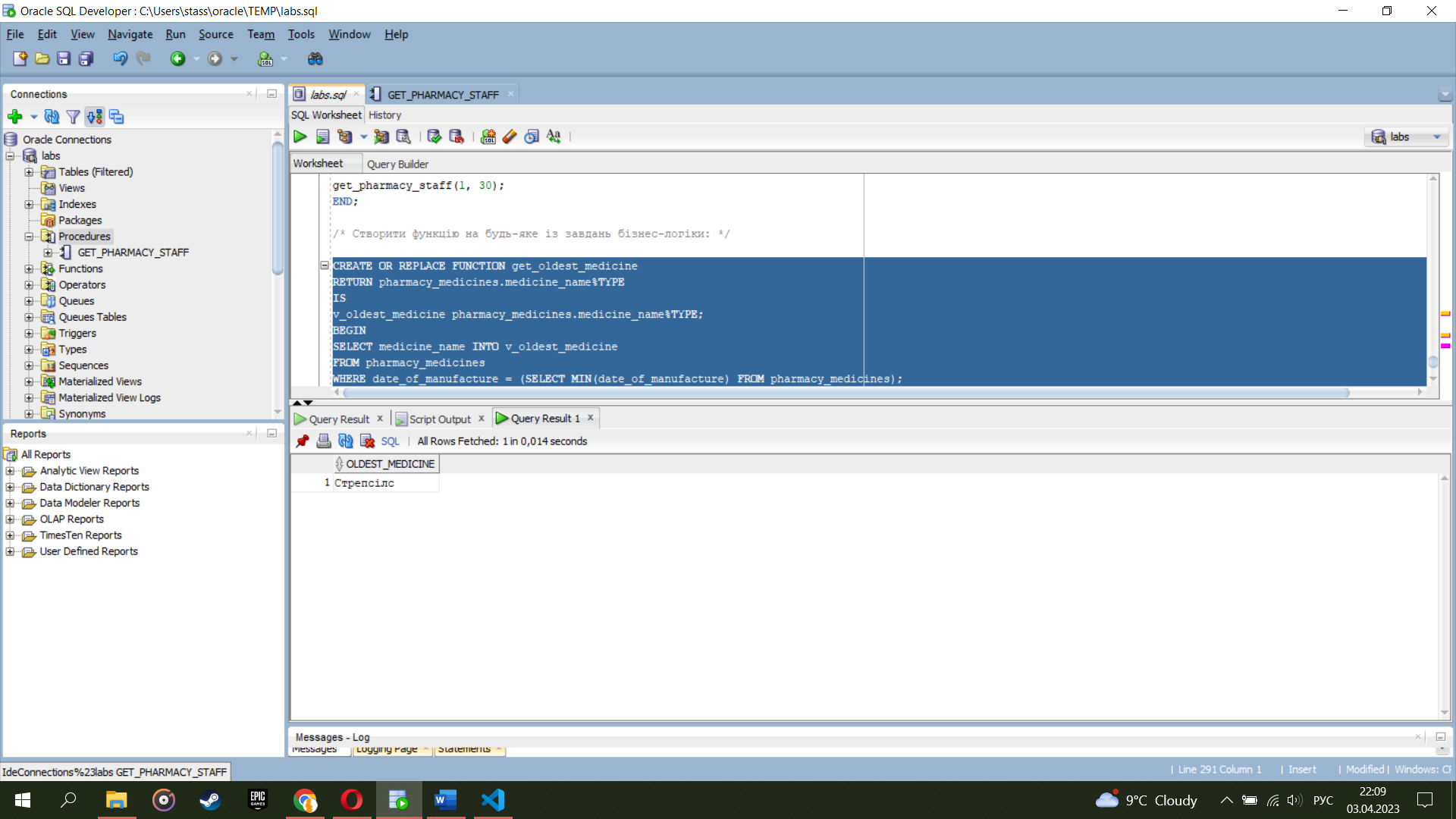
RETURN v\_oldest\_medicine;

END;

SELECT get\_oldest\_medicine() as oldest\_medicine FROM dual;

Цей скрипт створює функцію get\_oldest\_medicine, яка повертає найменування найстаріших ліків у таблиці pharmacy\_medicines.

Результат:



* *Об'єднати процедуру та функцію в пакет:*

CREATE OR REPLACE PACKAGE pharmacy\_pkg AS

PROCEDURE get\_staff\_age(

p\_staff\_id IN staff.id%TYPE

);

FUNCTION get\_oldest\_medicine

RETURN pharmacy\_medicines.medicine\_name%TYPE;

END pharmacy\_pkg;

/

CREATE OR REPLACE PACKAGE BODY pharmacy\_pkg AS

PROCEDURE get\_staff\_age(

p\_staff\_id IN staff.id%TYPE

)

IS

v\_staff\_age staff.age%TYPE;

BEGIN

SELECT age INTO v\_staff\_age

FROM staff

WHERE id = p\_staff\_id;

DBMS\_OUTPUT.PUT\_LINE('Staff ID ' || p\_staff\_id || ' is ' || v\_staff\_age || ' years old.');

END;

FUNCTION get\_oldest\_medicine

RETURN pharmacy\_medicines.medicine\_name%TYPE

IS

v\_oldest\_medicine pharmacy\_medicines.medicine\_name%TYPE;

BEGIN

SELECT medicine\_name INTO v\_oldest\_medicine

FROM pharmacy\_medicines

WHERE date\_of\_manufacture = (SELECT MIN(date\_of\_manufacture) FROM pharmacy\_medicines);

RETURN v\_oldest\_medicine;

END;

END pharmacy\_pkg;

/

DECLARE

oldest\_medicine pharmacy\_medicines.medicine\_name%TYPE;

BEGIN

oldest\_medicine := pharmacy\_pkg.get\_oldest\_medicine();

DBMS\_OUTPUT.PUT\_LINE('The oldest medicine is ' || oldest\_medicine);

pharmacy\_pkg.get\_staff\_age(1);

END;

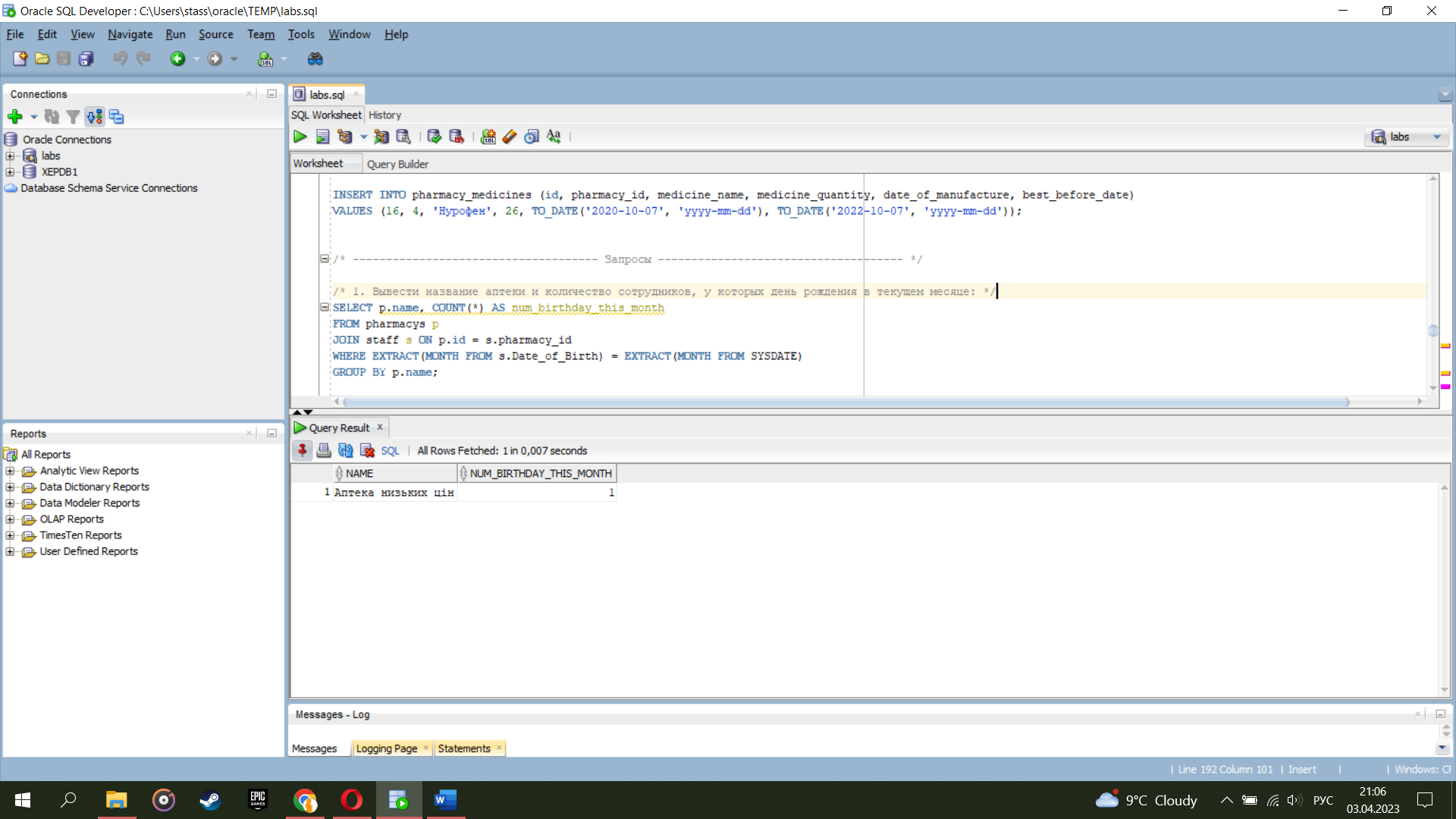
У цьому прикладі пакет "pharmacy\_pkg" містить процедуру "get\_staff\_age", яка приймає ID співробітника (staff\_id) як вхідний параметр і виводить вік цього співробітника. Також пакет містить функцію "get\_oldest\_medicine", яка повертає назву найстаріших ліків у таблиці "pharmacy\_medicines".

Результат:

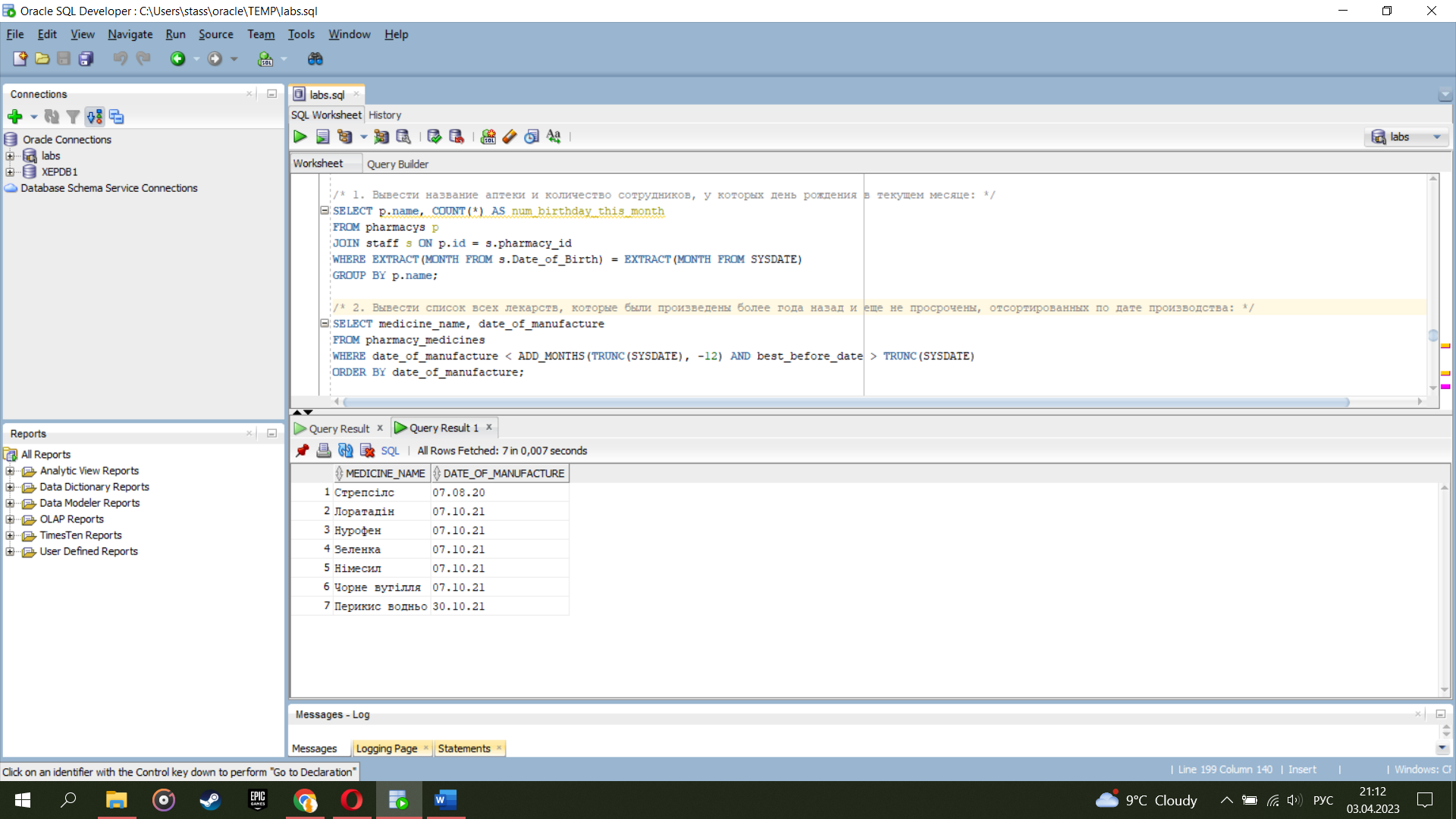


**10 складних запитів із застосуванням вбудованих функцій зі скріншотами результатів:**

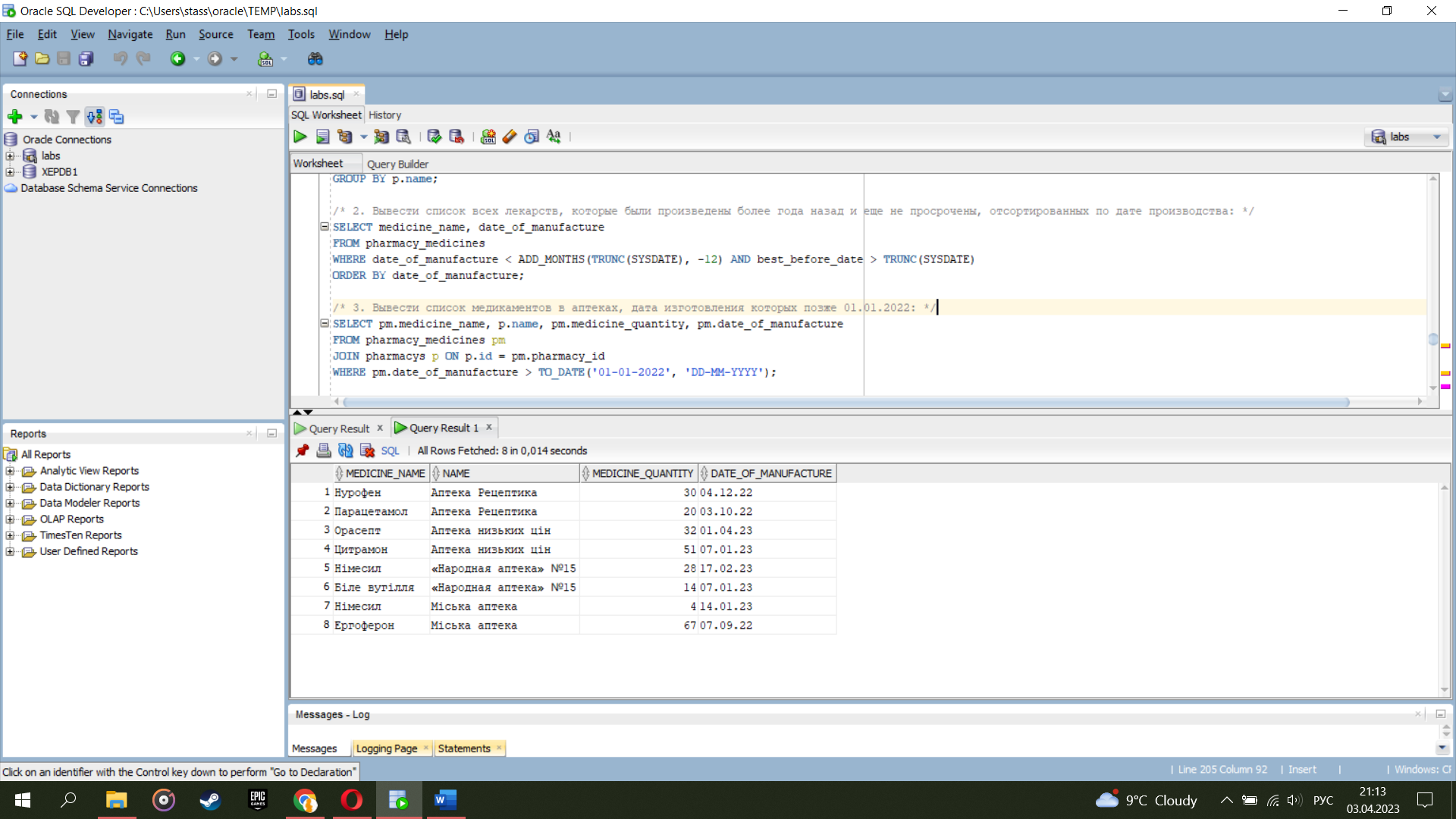
1. Вивести назву аптеки та кількість співробітників, у яких день народження у поточному місяці:



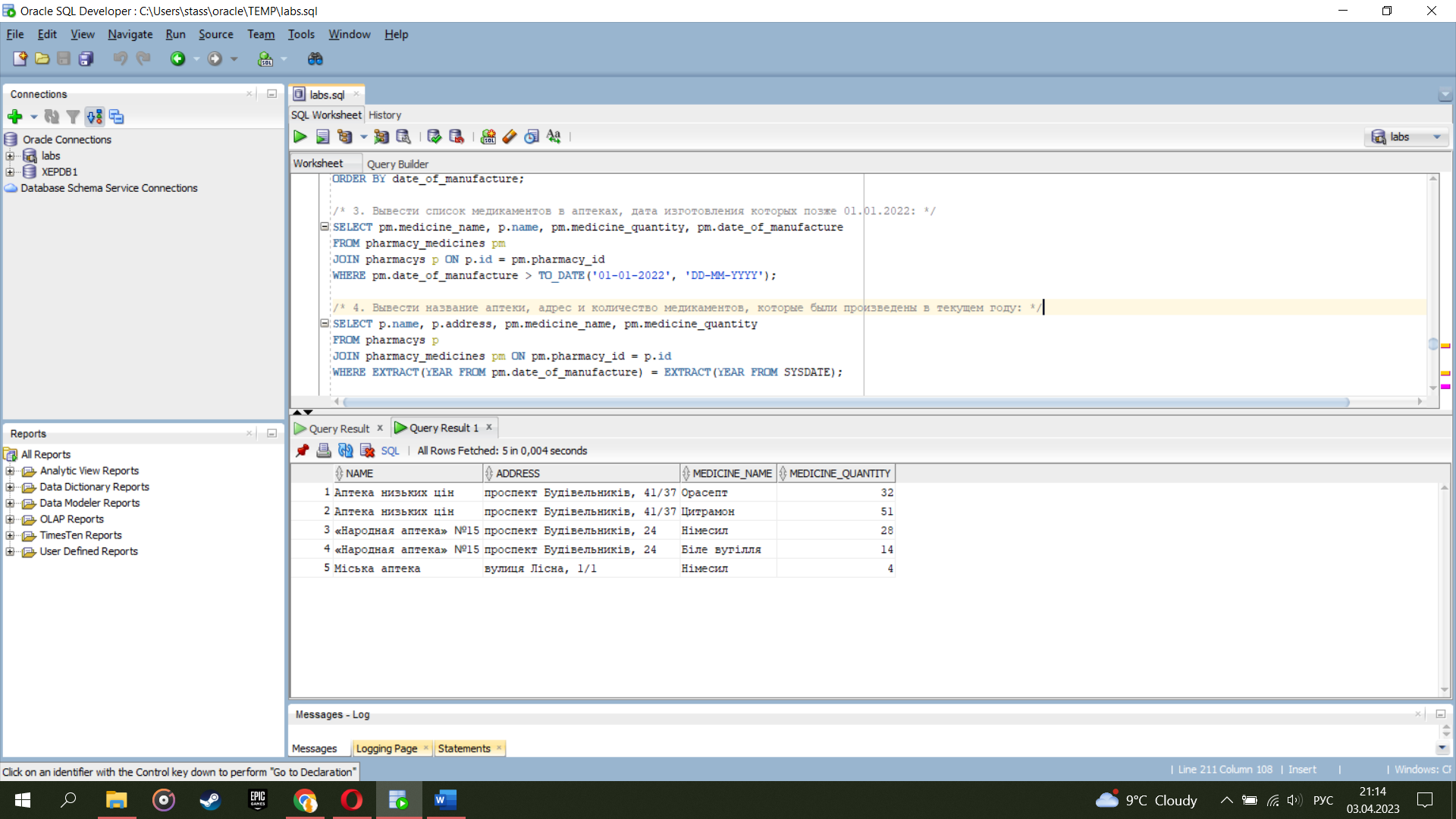
1. Вивести список усіх ліків, які були зроблені понад рік тому та ще не прострочені, відсортовані за датою виробництва:



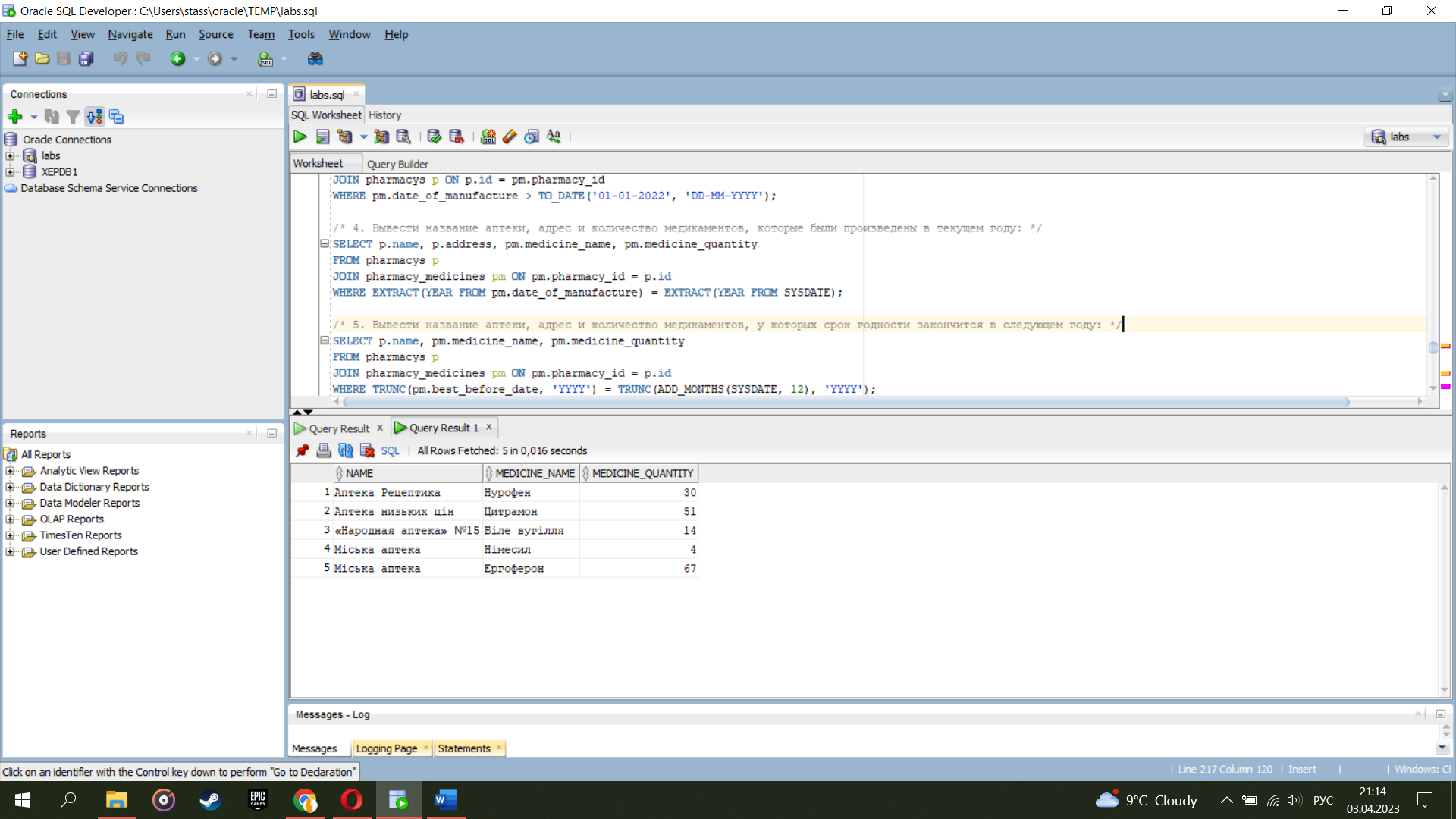
1. Вивести список медикаментів в аптеках, дата виготовлення яких пізніше 01.01.2022:



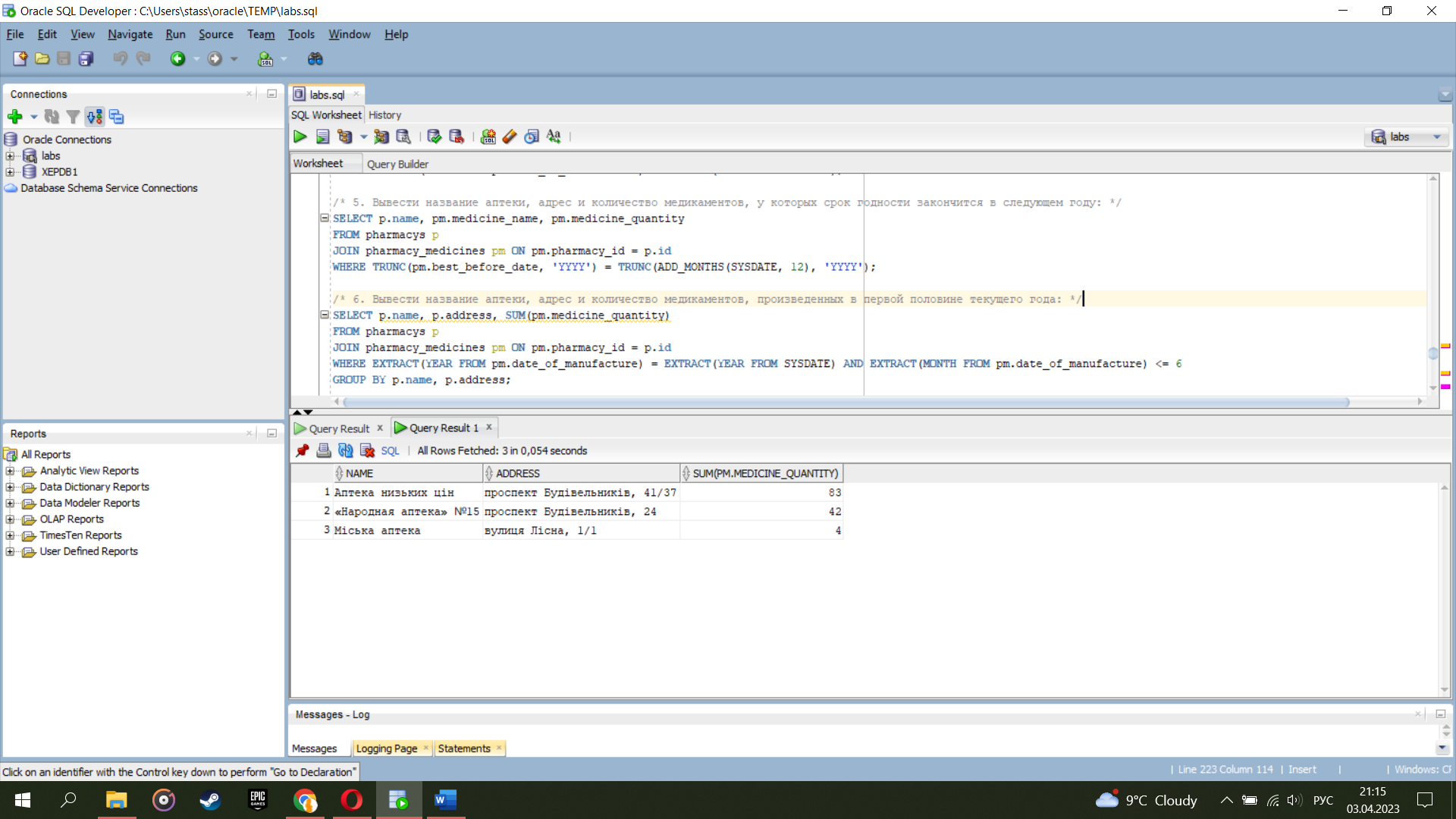
1. Вивести назву аптеки, адресу та кількість медикаментів, які були зроблені у поточному році:



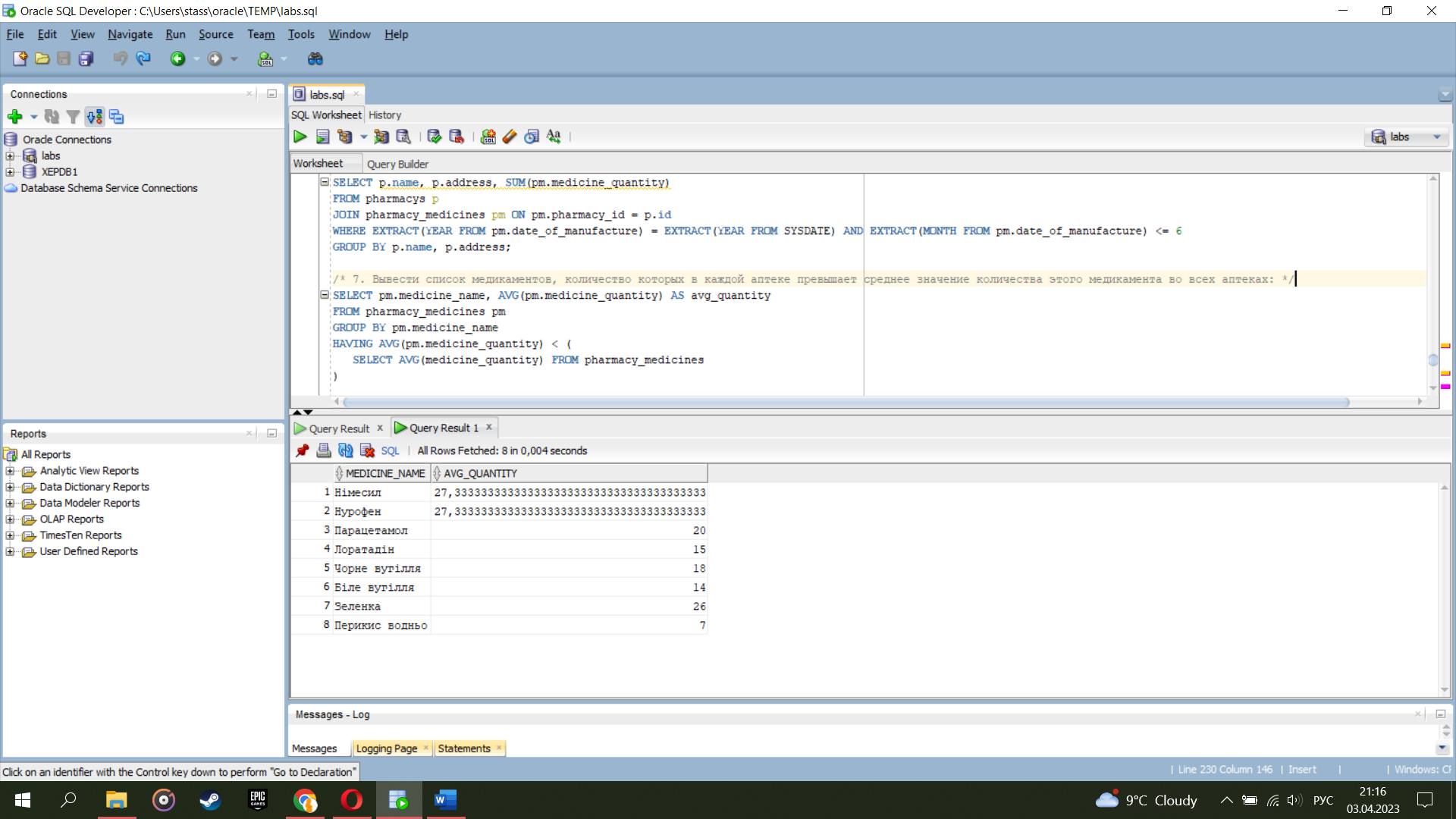
1. Вивести назву аптеки, адресу та кількість медикаментів, у яких термін придатності закінчиться наступного року:



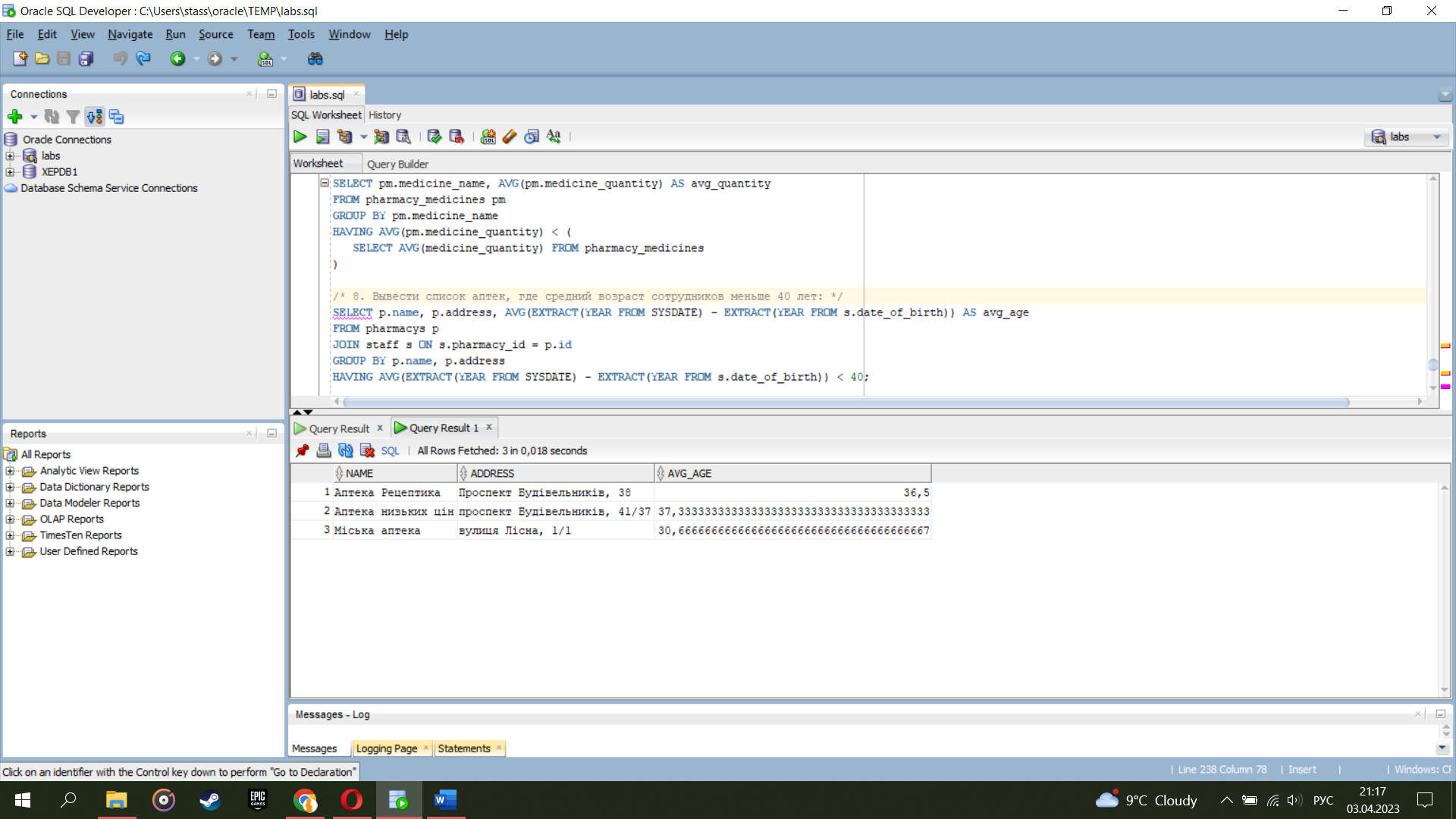
1. Вивести назву аптеки, адресу та кількість медикаментів, вироблених у першій половині поточного року:



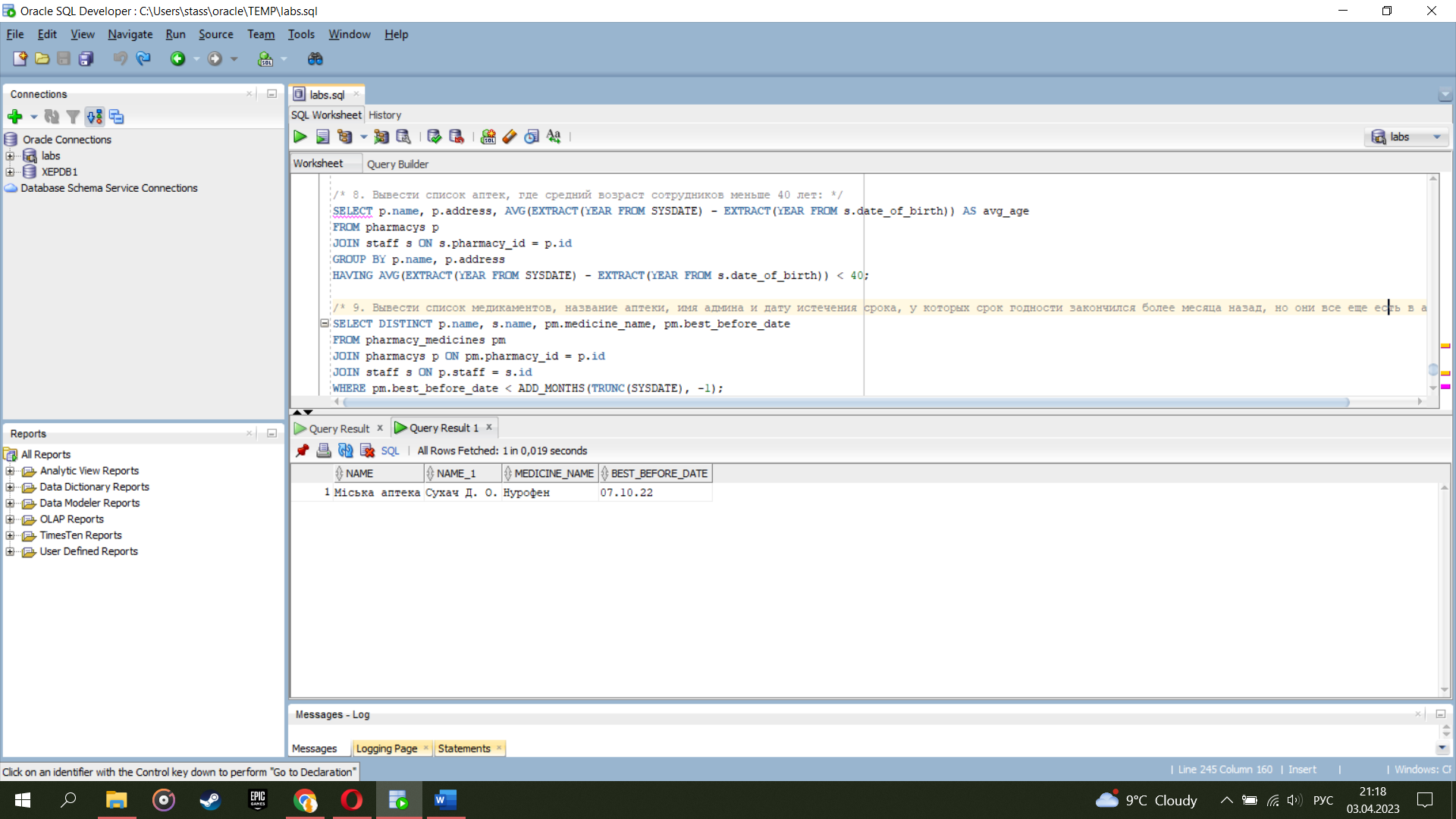
1. Вивести список медикаментів, кількість яких у кожній аптеці перевищує середнє значення кількості цього медикаменту у всіх аптеках:



1. Вивести список аптек, де середній вік співробітників менше 40 років:



1. Вивести список медикаментів, назву аптеки, ім'я адміна та дату закінчення терміну, у яких термін придатності закінчився більше місяця тому, але вони все ще є в аптеках:



1. Вивести список аптек, де працює хоча б один співробітник старше 50 років, відсортований за кількістю співробітників в аптеці в порядку зменшення:



*Скрипт запитів:*

/\* 1. Вывести название аптеки и количество сотрудников, у которых день рождения в текущем месяце: \*/

SELECT p.name, COUNT(\*) AS num\_birthday\_this\_month

FROM pharmacys p

JOIN staff s ON p.id = s.pharmacy\_id

WHERE EXTRACT(MONTH FROM s.Date\_of\_Birth) = EXTRACT(MONTH FROM SYSDATE)

GROUP BY p.name;

/\* 2. Вывести список всех лекарств, которые были произведены более года назад и еще не просрочены, отсортированных по дате производства: \*/

SELECT medicine\_name, date\_of\_manufacture

FROM pharmacy\_medicines

WHERE date\_of\_manufacture < ADD\_MONTHS(TRUNC(SYSDATE), -12) AND best\_before\_date > TRUNC(SYSDATE)

ORDER BY date\_of\_manufacture;

/\* 3. Вывести список медикаментов в аптеках, дата изготовления которых позже 01.01.2022: \*/

SELECT pm.medicine\_name, p.name, pm.medicine\_quantity, pm.date\_of\_manufacture

FROM pharmacy\_medicines pm

JOIN pharmacys p ON p.id = pm.pharmacy\_id

WHERE pm.date\_of\_manufacture > TO\_DATE('01-01-2022', 'DD-MM-YYYY');

/\* 4. Вывести название аптеки, адрес и количество медикаментов, которые были произведены в текущем году: \*/

SELECT p.name, p.address, pm.medicine\_name, pm.medicine\_quantity

FROM pharmacys p

JOIN pharmacy\_medicines pm ON pm.pharmacy\_id = p.id

WHERE EXTRACT(YEAR FROM pm.date\_of\_manufacture) = EXTRACT(YEAR FROM SYSDATE);

/\* 5. Вывести название аптеки, адрес и количество медикаментов, у которых срок годности закончится в следующем году: \*/

SELECT p.name, pm.medicine\_name, pm.medicine\_quantity

FROM pharmacys p

JOIN pharmacy\_medicines pm ON pm.pharmacy\_id = p.id

WHERE TRUNC(pm.best\_before\_date, 'YYYY') = TRUNC(ADD\_MONTHS(SYSDATE, 12), 'YYYY');

/\* 6. Вывести название аптеки, адрес и количество медикаментов, произведенных в первой половине текущего года: \*/

SELECT p.name, p.address, SUM(pm.medicine\_quantity)

FROM pharmacys p

JOIN pharmacy\_medicines pm ON pm.pharmacy\_id = p.id

WHERE EXTRACT(YEAR FROM pm.date\_of\_manufacture) = EXTRACT(YEAR FROM SYSDATE) AND EXTRACT(MONTH FROM pm.date\_of\_manufacture) <= 6

GROUP BY p.name, p.address;

/\* 7. Вывести список медикаментов, количество которых в каждой аптеке превышает среднее значение количества этого медикамента во всех аптеках: \*/

SELECT pm.medicine\_name, AVG(pm.medicine\_quantity) AS avg\_quantity

FROM pharmacy\_medicines pm

GROUP BY pm.medicine\_name

HAVING AVG(pm.medicine\_quantity) < (

   SELECT AVG(medicine\_quantity) FROM pharmacy\_medicines

)

/\* 8. Вывести список аптек, где средний возраст сотрудников меньше 40 лет: \*/

SELECT p.name, p.address, AVG(EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM s.date\_of\_birth)) AS avg\_age

FROM pharmacys p

JOIN staff s ON s.pharmacy\_id = p.id

GROUP BY p.name, p.address

HAVING AVG(EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM s.date\_of\_birth)) < 40;

/\* 9. Вывести список медикаментов, название аптеки, имя админа и дату истечения срока, у которых срок годности закончился более месяца назад, но они все еще есть в аптеках: \*/

SELECT DISTINCT p.name, s.name, pm.medicine\_name, pm.best\_before\_date

FROM pharmacy\_medicines pm

JOIN pharmacys p ON pm.pharmacy\_id = p.id

JOIN staff s ON p.staff = s.id

WHERE pm.best\_before\_date < ADD\_MONTHS(TRUNC(SYSDATE), -1);

/\* 10. Вывести список аптек, где работает хотя бы один сотрудник старше 50 лет, отсортированный по количеству сотрудников в аптеке в порядке убывания: \*/

SELECT p.name, COUNT(s.id) as num\_staff

FROM pharmacys p

JOIN staff s ON s.pharmacy\_id = p.id

WHERE s.age > 50

GROUP BY p.name

ORDER BY num\_staff DESC;

**Висновок:** у рамках даної лабораторної роботи були здобуті навички створювати функції, процедури, пакети засобами PL/SQL, а також використовувати явні курсори.