Table of Contents

[1. How many patients are there in the patient table? 2](#_Toc163930486)

[2. Write a query to determine if there are more males than females in the patient table. 2](#_Toc163930487)

[3. Retrieve the names of patients who have 'Lee' in their name and 'follow-up' in the appointment notes. 3](#_Toc163930488)

[4. Retrieve the medications that have been prescribed for patients whose names ending with 'Smith'. 4](#_Toc163930489)

[5. List the doctors who have not been assigned any appointments using a subquery. 4](#_Toc163930490)

[6. Retrieve the doctors who have prescribed medications containing 'pain' in their description more than 2 times. 5](#_Toc163930491)

[7. Write a query to find patients who have appointments scheduled in the month of June 2023. 6](#_Toc163930492)

[8. Write a query to find the doctor with the highest number of appointments scheduled in a single day. 6](#_Toc163930493)

[9. List the total number of appointments based on the day of the week. 7](#_Toc163930494)

[10. Display the patients if they are sharing the same birth month as another patient. 7](#_Toc163930495)

[11. How many patients have never had an appointment or scheduled any appointments so far? 8](#_Toc163930496)

[12. Show medications that have never been prescribed to by any doctor using a JOIN of your choice. 9](#_Toc163930497)

[13. List the doctors who have treated at least one patient from ‘WA’. 10](#_Toc163930498)

[14. Which medicine has the second-highest overall prescription rate? 10](#_Toc163930499)

[15. Retrieve the patients who have never cancelled an appointment. 11](#_Toc163930500)

[16. Retrieve the youngest and oddest female patient. 12](#_Toc163930501)

[17. Find medications that have appeared only once in any prescription. 13](#_Toc163930502)

[18. Find medications that have appeared only once in any prescription along with the doctors who prescribed them. 13](#_Toc163930503)

[19. Retrieve the patients who have had appointments with at least three different doctors. 14](#_Toc163930504)

[20. In not more than 200 words, comment on whether the tables are in 3NF. Justify your argument with relevant examples, and then explain at least two ways to improve this database based on. 15](#_Toc163930505)

[3NF Analysis: 15](#_Toc163930506)

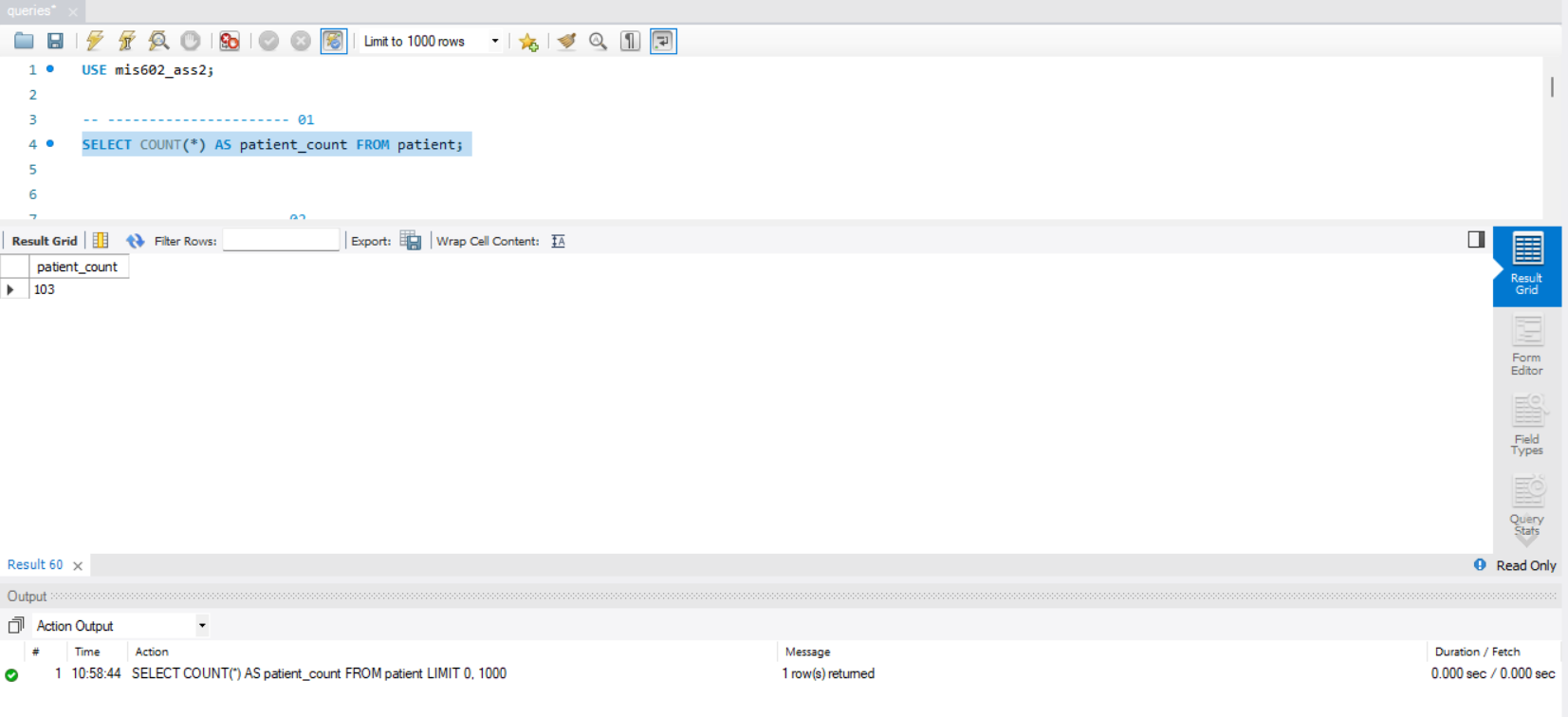
# How many patients are there in the patient table?

SELECT

COUNT(\*) AS patient\_count

FROM

patient;



# Write a query to determine if there are more males than females in the patient table.

SELECT

CASE

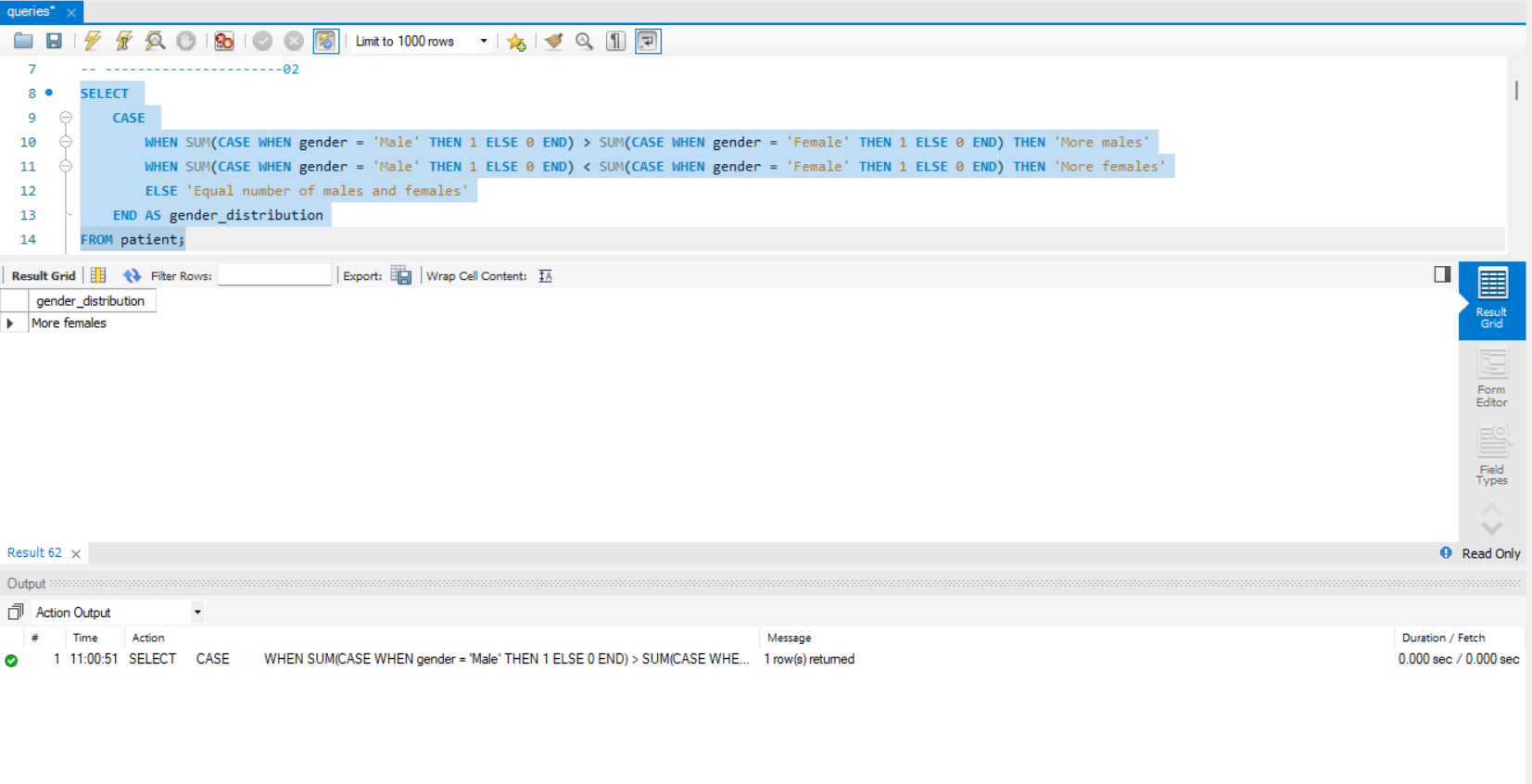
WHEN SUM(CASE WHEN gender = 'Male' THEN 1 ELSE 0 END) > SUM(CASE WHEN gender = 'Female' THEN 1 ELSE 0 END) THEN 'More males'

WHEN SUM(CASE WHEN gender = 'Male' THEN 1 ELSE 0 END) < SUM(CASE WHEN gender = 'Female' THEN 1 ELSE 0 END) THEN 'More females'

ELSE 'Equal number of males and females'

END AS gender\_distribution

FROM patient;



# Retrieve the names of patients who have 'Lee' in their name and 'follow-up' in the appointment notes.

SELECT p.name

FROM patient p

JOIN appointment a ON p.patient\_id = a.patient\_id

WHERE p.name LIKE '%Lee%' AND a.notes LIKE '%follow-up%';

A screenshot of a computer

Description automatically generated

# Retrieve the medications that have been prescribed for patients whose names ending with 'Smith'.

SELECT m.name AS medication\_name

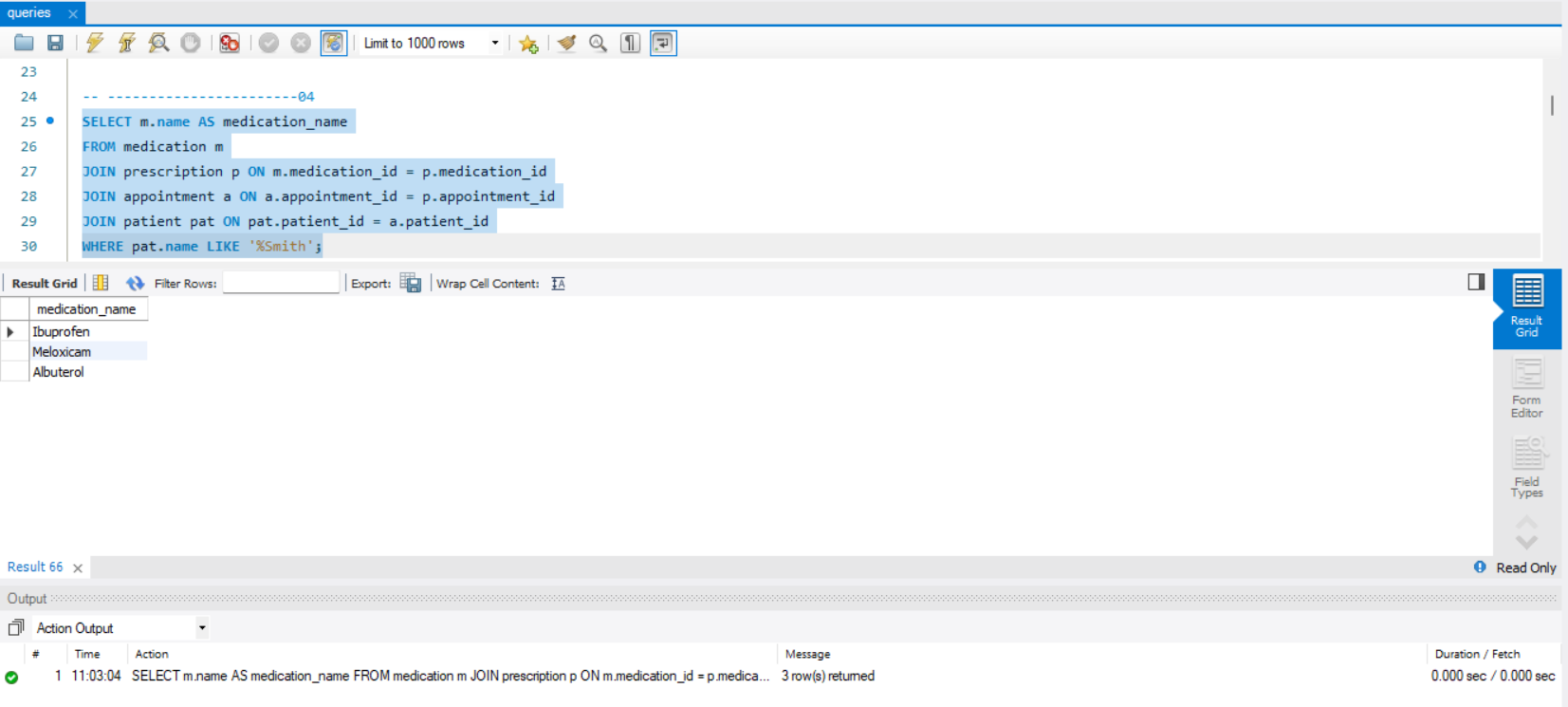
FROM medication m

JOIN prescription p ON m.medication\_id = p.medication\_id

JOIN appointment a ON a.appointment\_id = p.appointment\_id

JOIN patient pat ON pat.patient\_id = a.patient\_id

WHERE pat.name LIKE '%Smith';



# List the doctors who have not been assigned any appointments using a subquery.

SELECT \*

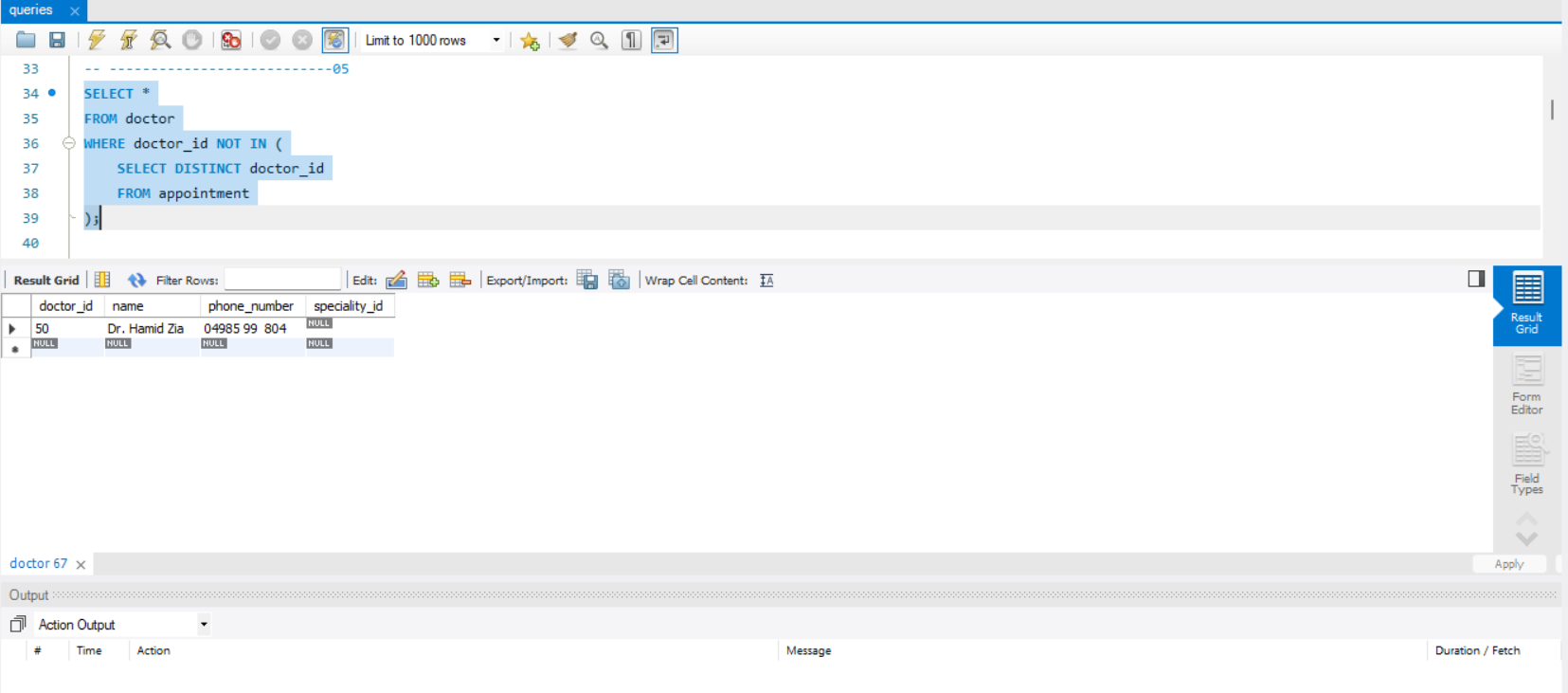
FROM doctor

WHERE doctor\_id NOT IN (

SELECT DISTINCT doctor\_id

FROM appointment

);



# Retrieve the doctors who have prescribed medications containing 'pain' in their description more than 2 times.

SELECT d.\*

FROM doctor d

JOIN appointment a ON d.doctor\_id = a.doctor\_id

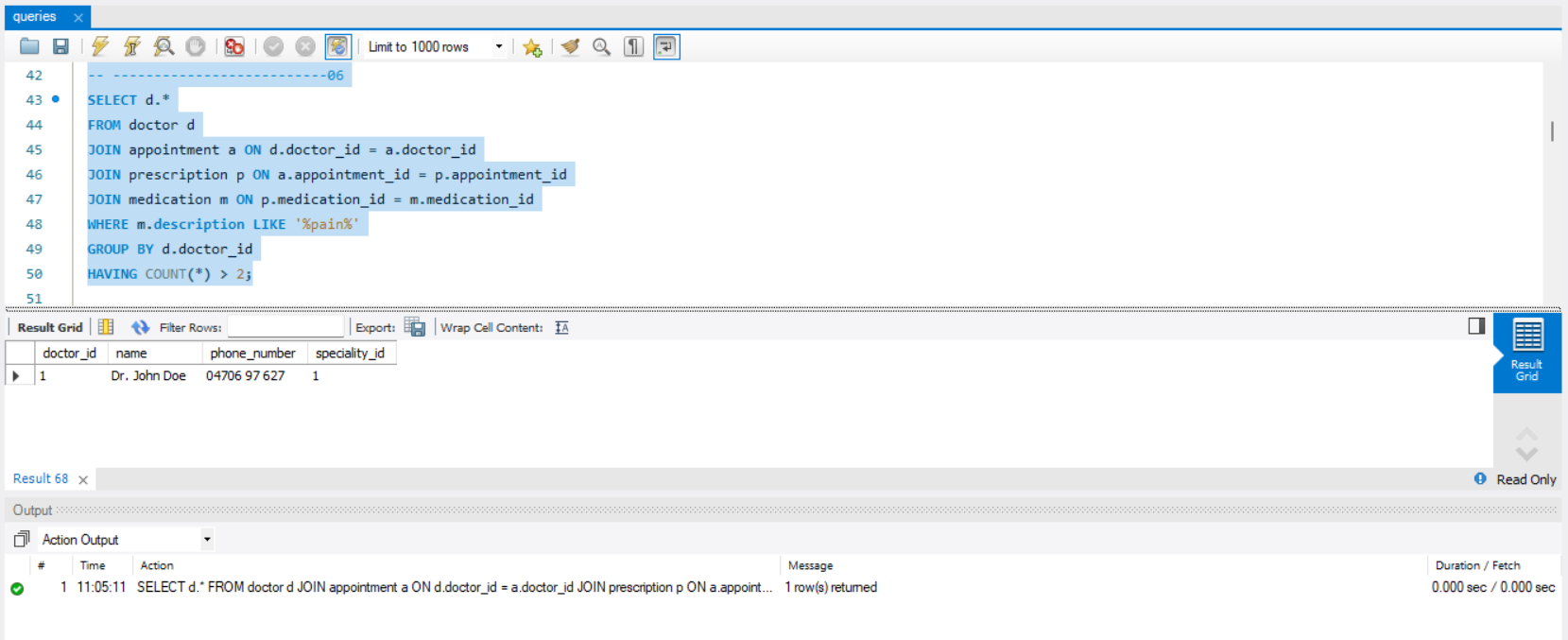
JOIN prescription p ON a.appointment\_id = p.appointment\_id

JOIN medication m ON p.medication\_id = m.medication\_id

WHERE m.description LIKE '%pain%'

GROUP BY d.doctor\_id

HAVING COUNT(\*) > 2;



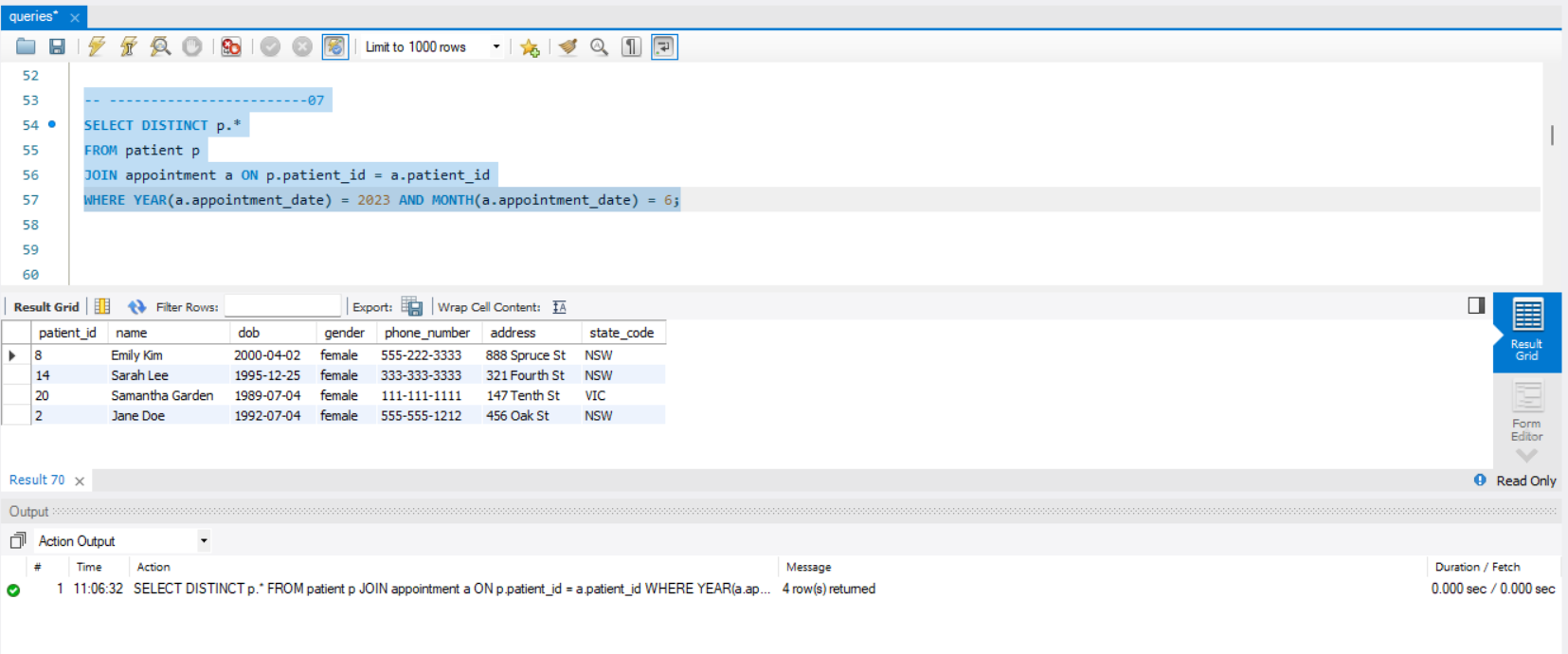
# Write a query to find patients who have appointments scheduled in the month of June 2023.

SELECT DISTINCT p.\*

FROM patient p

JOIN appointment a ON p.patient\_id = a.patient\_id

WHERE YEAR(a.appointment\_date) = 2023 AND MONTH(a.appointment\_date) = 6;



# Write a query to find the doctor with the highest number of appointments scheduled in a single day.

SELECT d.doctor\_id, d.name AS doctor\_name, COUNT(\*) AS num\_appointments

FROM appointment a

JOIN doctor d ON a.doctor\_id = d.doctor\_id

GROUP BY d.doctor\_id, DATE(a.appointment\_date)

ORDER BY num\_appointments DESC

LIMIT 1;

A screenshot of a computer

Description automatically generated

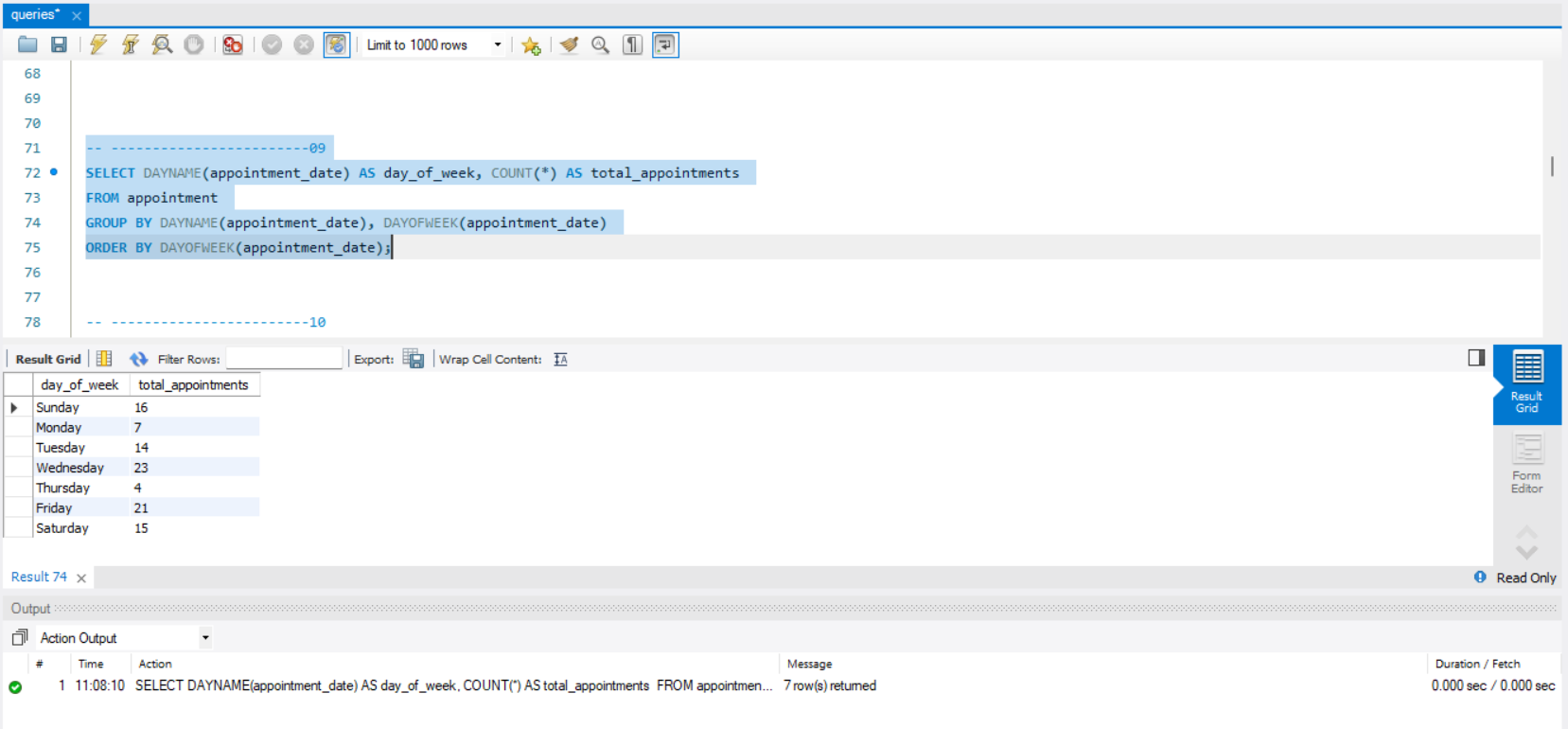
# List the total number of appointments based on the day of the week.

SELECT DAYNAME(appointment\_date) AS day\_of\_week, COUNT(\*) AS total\_appointments

FROM appointment

GROUP BY DAYNAME(appointment\_date), DAYOFWEEK(appointment\_date)

ORDER BY DAYOFWEEK(appointment\_date);



# Display the patients if they are sharing the same birth month as another patient.

SELECT DISTINCT

LEAST(p1.name, p2.name) AS patient1\_name,

GREATEST(p1.name, p2.name) AS patient2\_name,

MONTH(p1.dob) AS birth\_month

FROM

patient p1

JOIN

patient p2 ON p1.patient\_id < p2.patient\_id

WHERE

MONTH(p1.dob) = MONTH(p2.dob)

ORDER BY

birth\_month;

A screenshot of a computer

Description automatically generated

# How many patients have never had an appointment or scheduled any appointments so far?

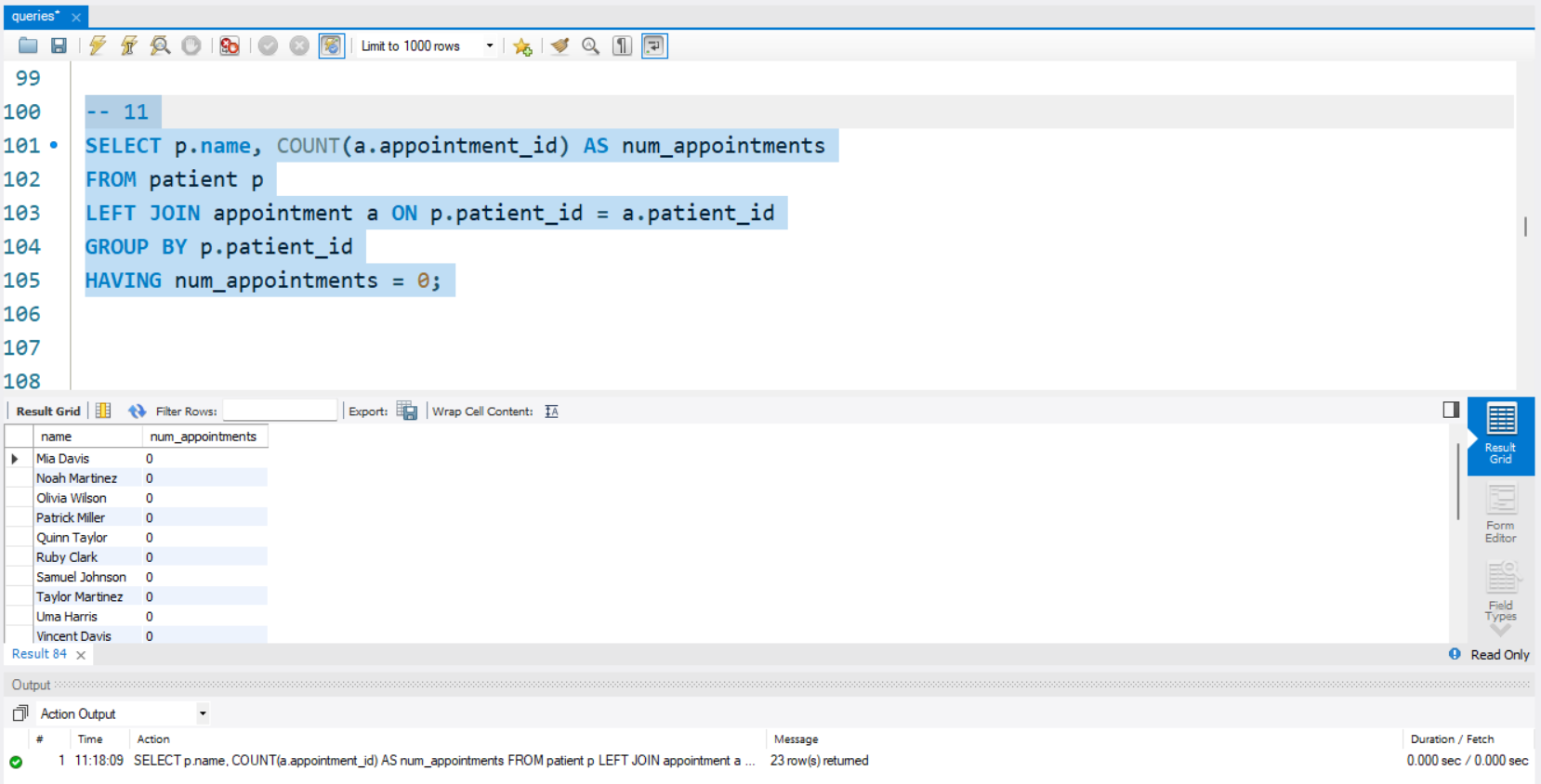
SELECT p.name, COUNT(a.appointment\_id) AS num\_appointments

FROM patient p

LEFT JOIN appointment a ON p.patient\_id = a.patient\_id

GROUP BY p.patient\_id

HAVING num\_appointments = 0;



# Show medications that have never been prescribed to by any doctor using a JOIN of your choice.

SELECT m.medication\_id, m.name

FROM medication m

LEFT JOIN prescription p ON m.medication\_id = p.medication\_id

WHERE p.prescription\_id IS NULL;

A screenshot of a computer

Description automatically generated

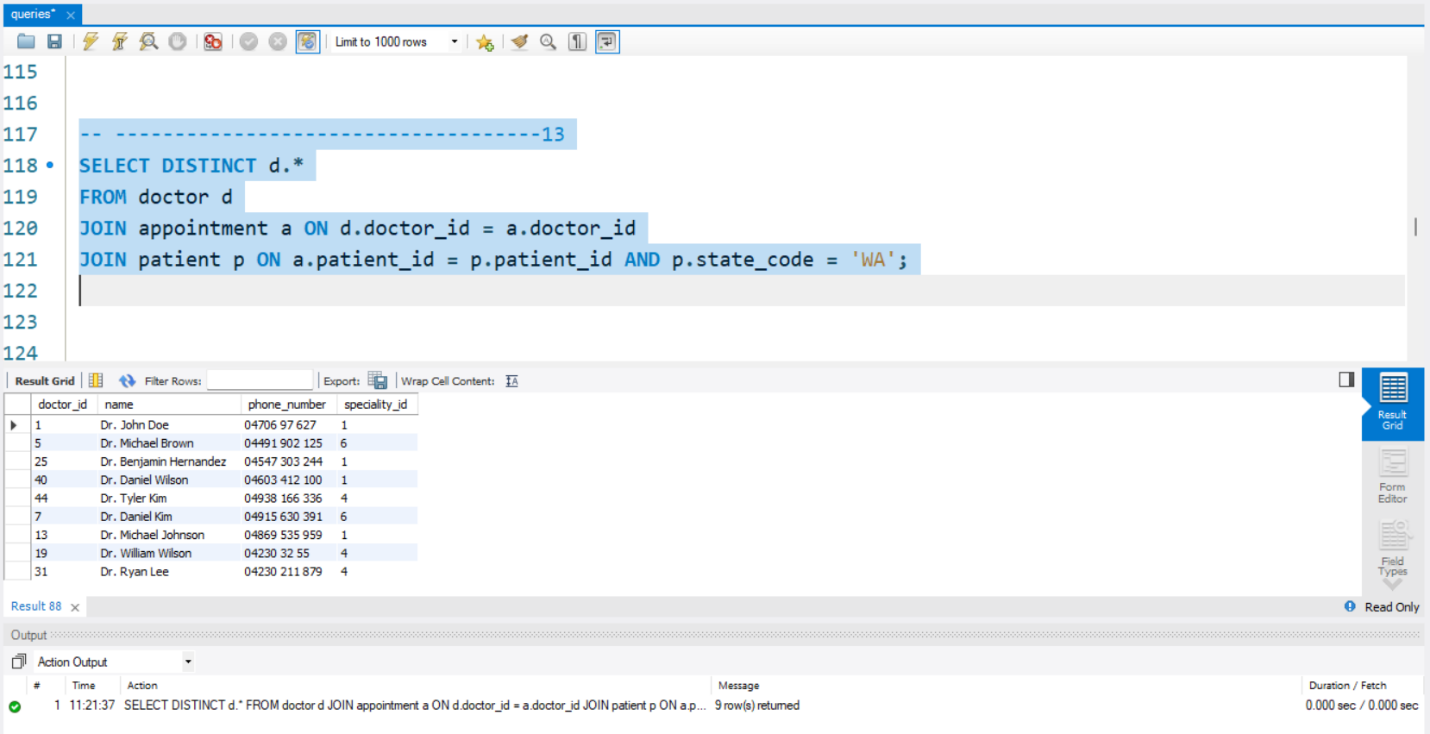
# List the doctors who have treated at least one patient from ‘WA’.

SELECT DISTINCT d.\*

FROM doctor d

JOIN appointment a ON d.doctor\_id = a.doctor\_id

JOIN patient p ON a.patient\_id = p.patient\_id AND p.state\_code = 'WA';



# Which medicine has the second-highest overall prescription rate?

SELECT name AS medication\_name, COUNT(\*) AS prescription\_count

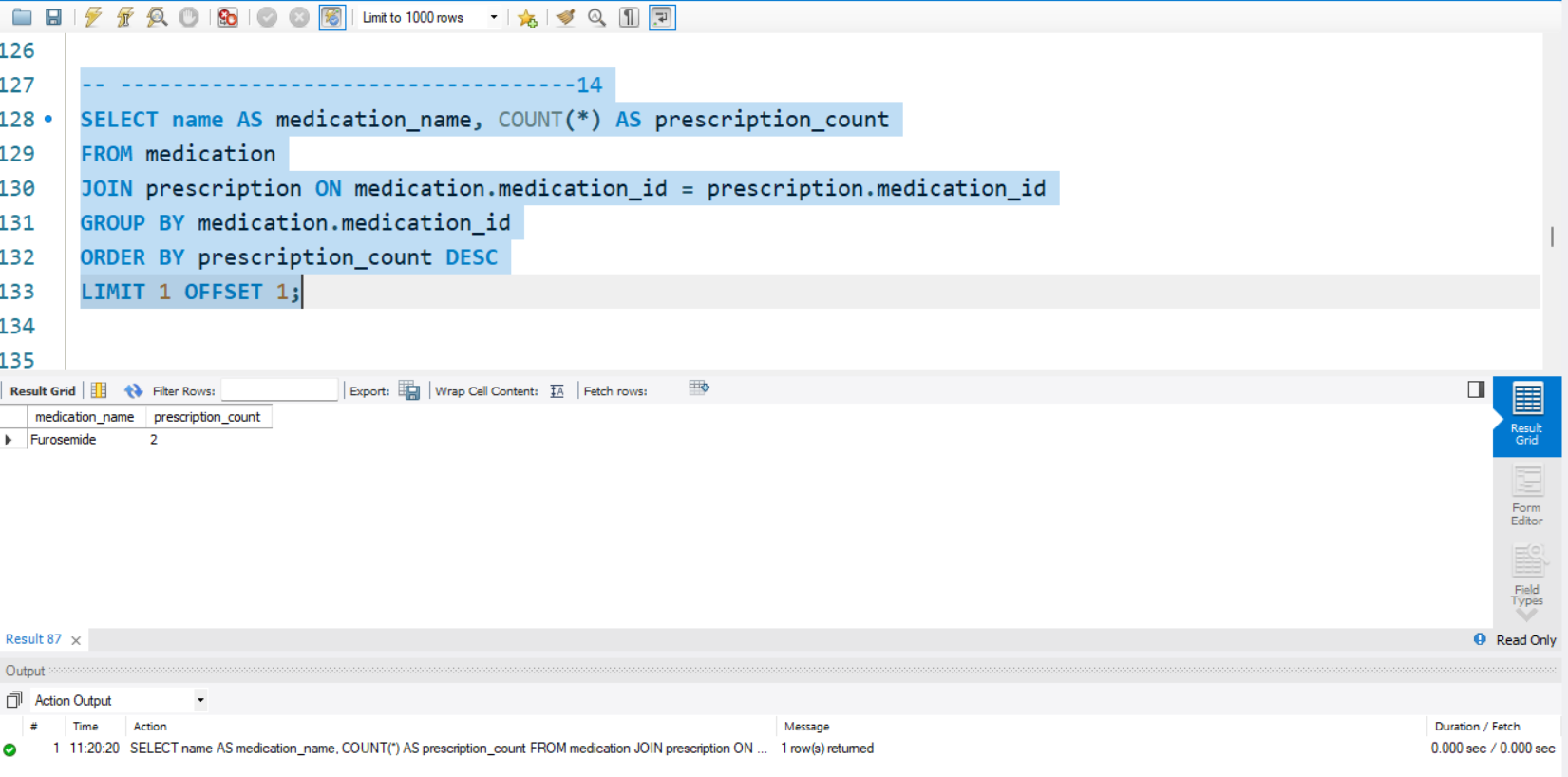
FROM medication

JOIN prescription ON medication.medication\_id = prescription.medication\_id

GROUP BY medication.medication\_id

ORDER BY prescription\_count DESC

LIMIT 1 OFFSET 1;



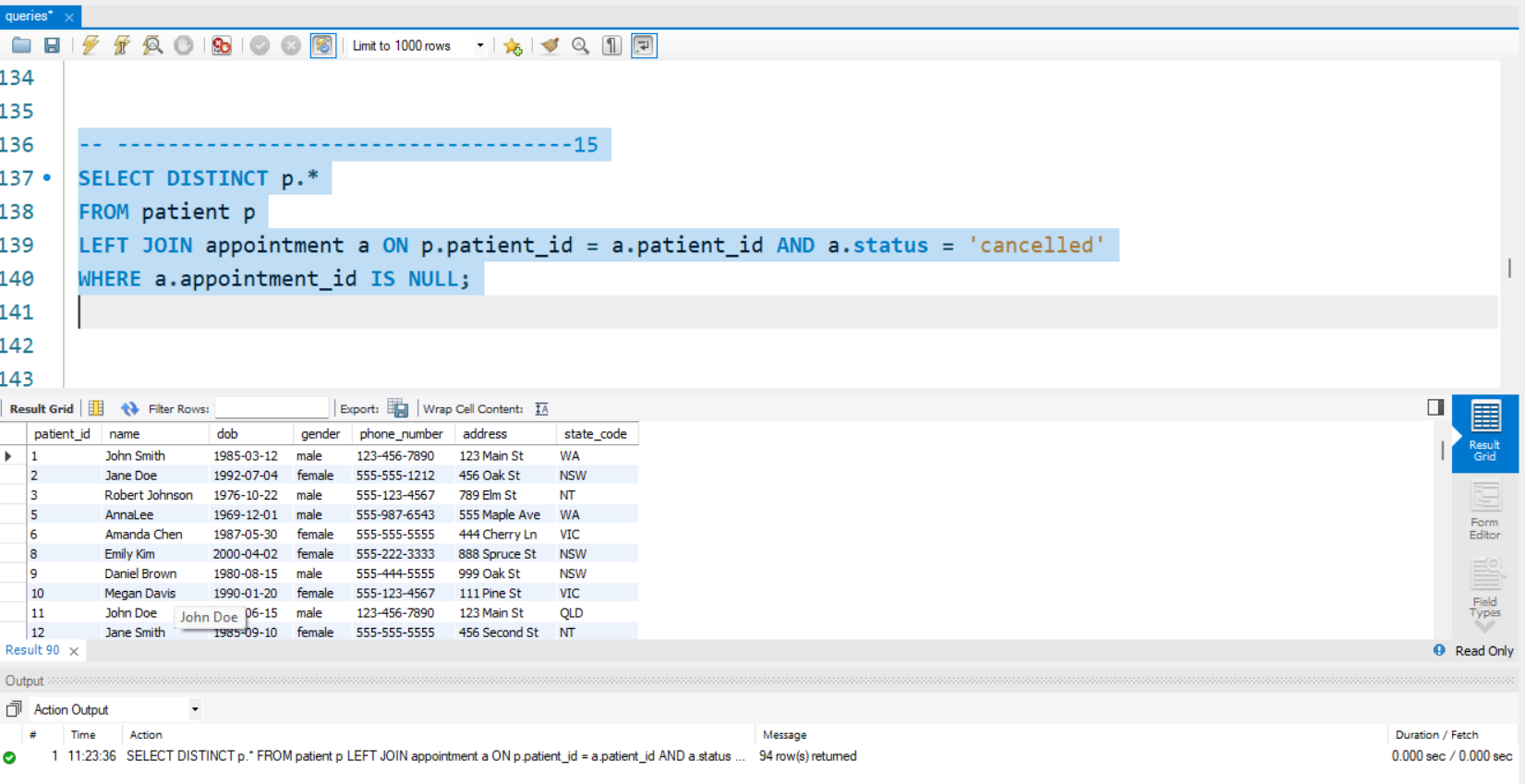
# Retrieve the patients who have never cancelled an appointment.

SELECT DISTINCT p.\*

FROM patient p

LEFT JOIN appointment a ON p.patient\_id = a.patient\_id AND a.status = 'cancelled'

WHERE a.appointment\_id IS NULL;



# Retrieve the youngest and oddest female patient.

SELECT \*

FROM (

SELECT \*

FROM patient

WHERE gender = 'female'

ORDER BY dob ASC

LIMIT 1

) AS youngest\_female

UNION

SELECT \*

FROM (

SELECT \*

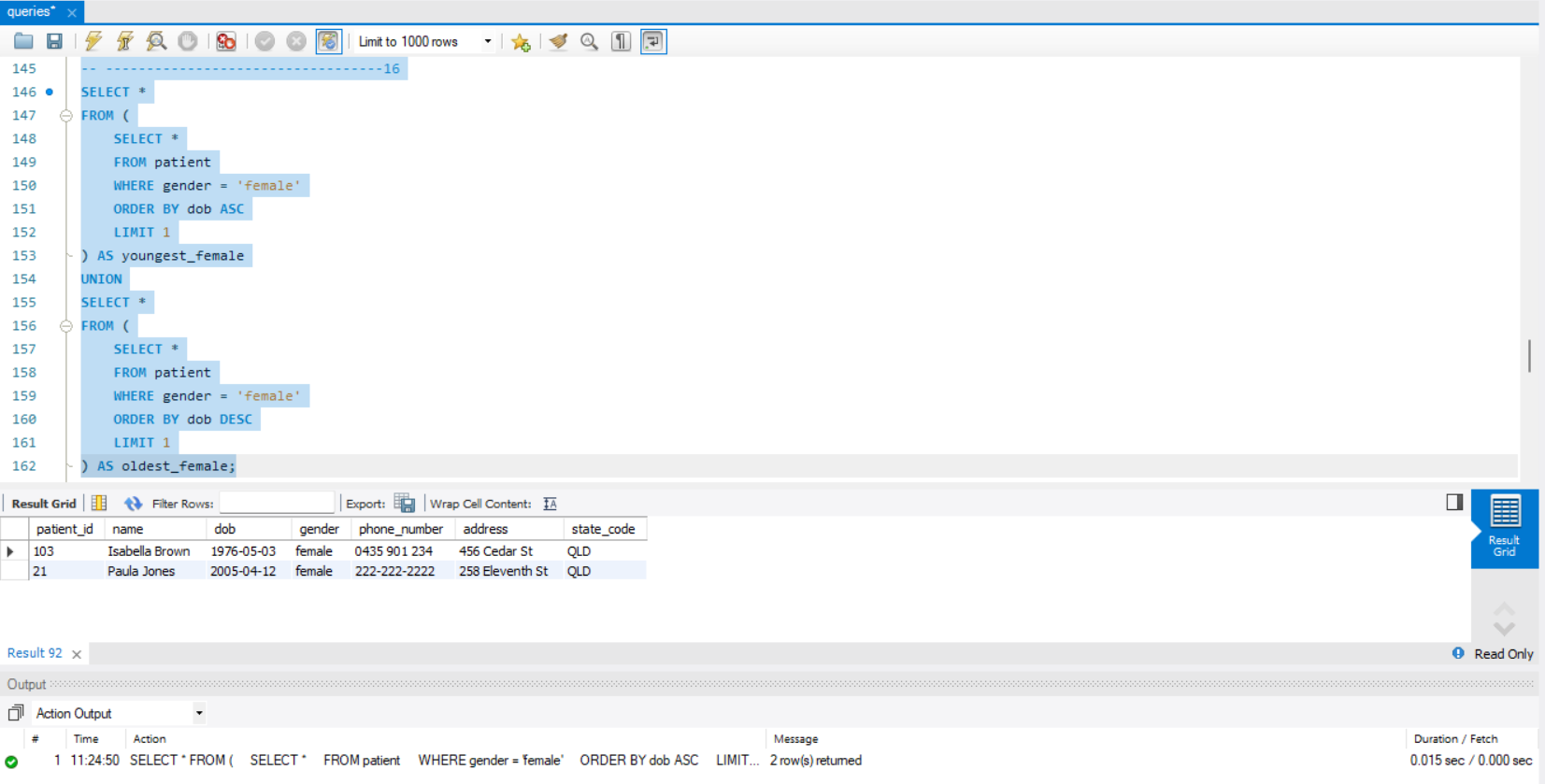
FROM patient

WHERE gender = 'female'

ORDER BY dob DESC

LIMIT 1

) AS oldest\_female;



# Find medications that have appeared only once in any prescription.

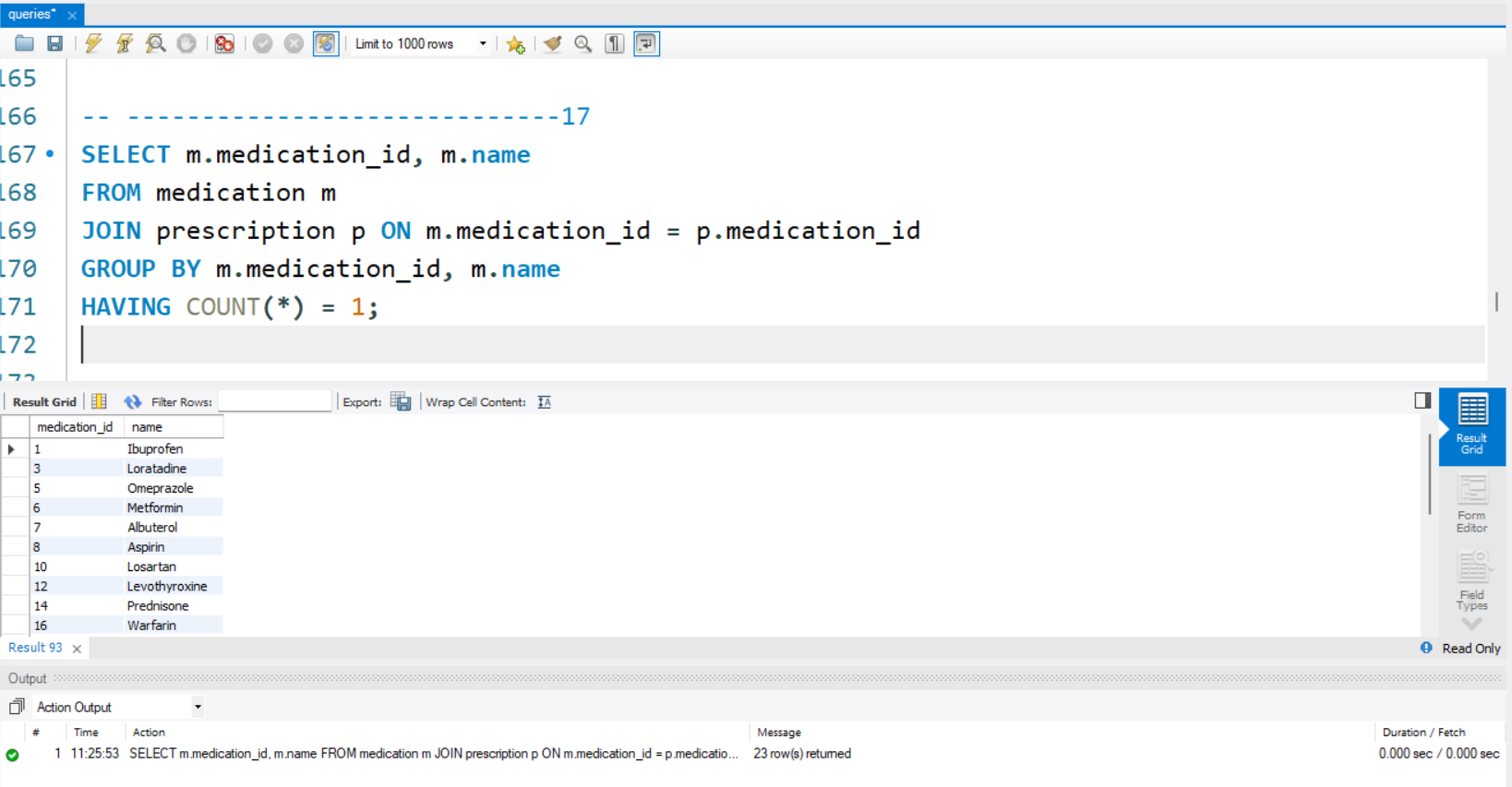
SELECT m.medication\_id, m.name

FROM medication m

JOIN prescription p ON m.medication\_id = p.medication\_id

GROUP BY m.medication\_id, m.name

HAVING COUNT(\*) = 1;



# Find medications that have appeared only once in any prescription along with the doctors who prescribed them.

SELECT m.name AS medication\_name, d.name AS doctor\_name

FROM medication m

JOIN prescription p ON m.medication\_id = p.medication\_id

JOIN appointment a ON p.appointment\_id = a.appointment\_id

JOIN doctor d ON a.doctor\_id = d.doctor\_id

GROUP BY m.medication\_id, m.name, d.doctor\_id, d.name

HAVING COUNT(\*) = 1;

A screenshot of a computer

Description automatically generated

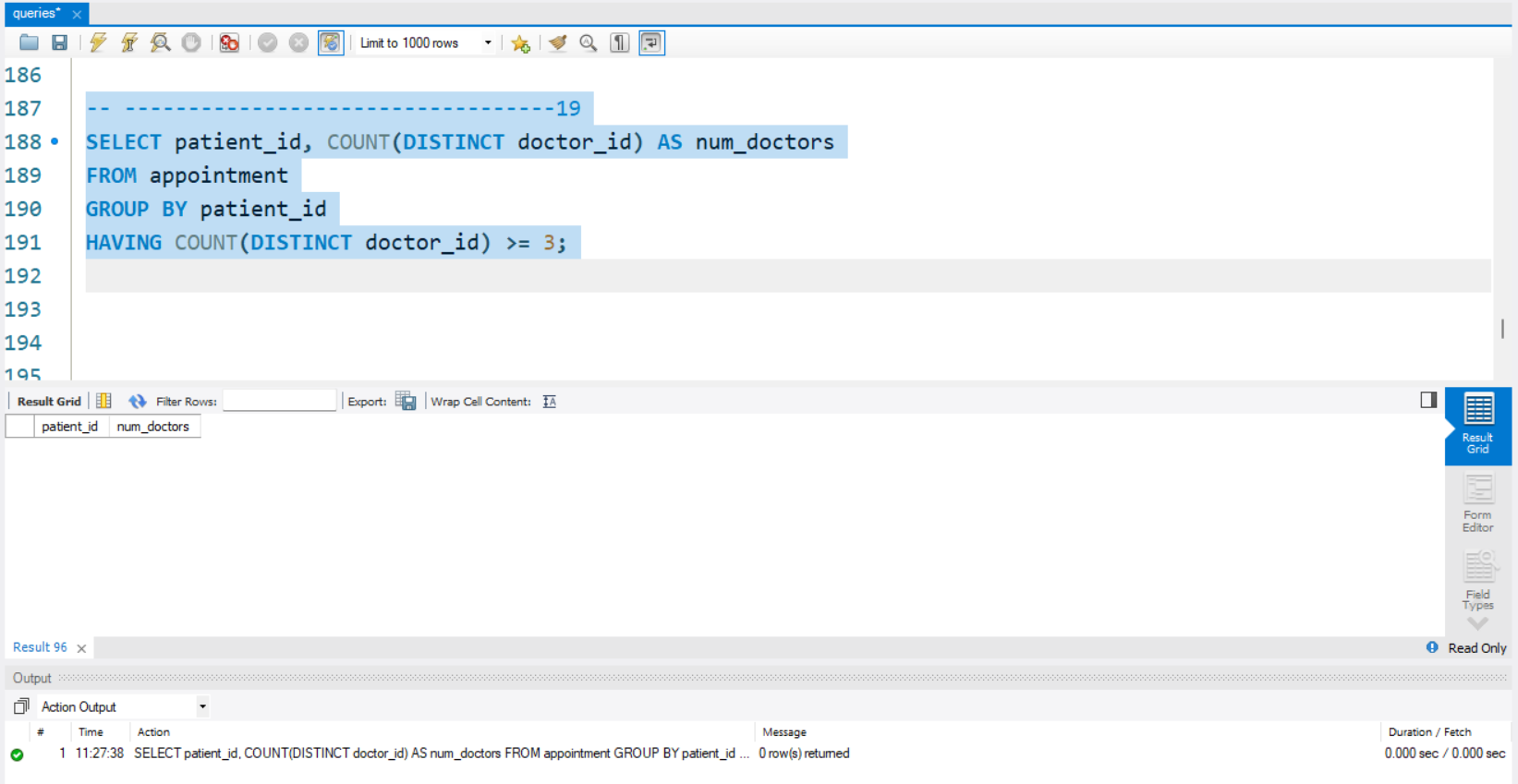
# Retrieve the patients who have had appointments with at least three different doctors.

SELECT patient\_id, COUNT(DISTINCT doctor\_id) AS num\_doctors

FROM appointment

GROUP BY patient\_id

HAVING COUNT(DISTINCT doctor\_id) >= 3;



# In not more than 200 words, comment on whether the tables are in 3NF. Justify your argument with relevant examples, and then explain at least two ways to improve this database based on.

## 3NF Analysis:

Justification:

The tables seem to be in Third Normal Form (3NF). Here's why:

1. **Atomic Values**: Each table contains just atomic qualities. For instance, in the patient table, the phone\_number attribute holds a solitary worth as opposed to a rundown of phone numbers.
2. **No Transitive Dependencies**: There are no transitive dependencies in the tables. For example, in the doctor table, the name and phone\_number attributes rely just upon the doctor\_id, which is the primary key.
3. **Separate Tables**: Each table represents a single entity type, such as patients, doctors, medications, etc., and the attributes are logically grouped under each entity type.

Possible Improvements:

1. **Normalization**: While the tables appear to be in 3NF, further normalization could be advantageous. For instance, the appointment table might actually be parted into two tables: one for appointments and one more for appointment status/history, to keep away from overt repetitiveness and better coordinate the information.
2. **Indexing**: Including suitable records much of the time questioned columns can further develop question execution, particularly in huge datasets. For example, indexing the gender column in the patient table may be gainful for questions including gender-based examination.