**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНТЕЛЕКТУАЛЬНИХ**

**ТЕХНОЛОГІЙ І ЗВ’ЯЗКУ**

**Звіт**

**з дисципліни Основи Баз Даних та Знань**

**Лабораторна робота №4**

**на тему: «Представление. Подзапросы. Аналитические функции»**

Виконав: студент 3 курсу, групи ІПЗ-3.04 спеціальності

121 Інженерія програмного забезпечення

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Бухта М.М.

Перевірив\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Глазунова Л.В.

**Одеса  2024**

**ЗАВДАННЯ**

1. Составить представление, отображающее список болезней у максимального количества пациентов каждого врача.

2.Аналит. ф-ции  
Получить рейтинг каждого врача при ранжировании по: – количеству принятых пациентов; – количеству выписанных анализов; – количеству назначенных препаратов за последний квартал.

3. Аналит. ф-ции Составить представление для отображения тенденции количества посещений врачей (по датам посещений).

**ВИКОНАННЯ**

**Task 1**

/\* Descripiton:

\* Create a view displaying the list of diseases for the maximum number of patients of each doctor.

\*/

CREATE VIEW doctor\_patient\_diseases AS

WITH max\_patients\_per\_doctor AS (

SELECT

bd.doctor\_id,

COUNT(DISTINCT bd.patient\_id) AS patient\_count

FROM

booked\_doctors bd

GROUP BY

bd.doctor\_id

HAVING

COUNT(DISTINCT bd.patient\_id) = (

SELECT

MAX(patient\_count)

FROM (

SELECT

COUNT(DISTINCT patient\_id) AS patient\_count

FROM

booked\_doctors

GROUP BY

doctor\_id

) AS subquery

)

)

SELECT

d.name AS doctor\_name,

d.surname AS doctor\_surname,

d.middle\_name AS doctor\_middle\_name,

p.name AS patient\_name,

p.surname AS patient\_surname,

p.middle\_name AS patient\_middle\_name,

da.complaint AS disease

FROM

max\_patients\_per\_doctor mpd

JOIN

users d ON mpd.doctor\_id = d.id

JOIN

booked\_doctors bd ON mpd.doctor\_id = bd.doctor\_id

JOIN

doctor\_appointments da ON bd.id = da.booked\_doctor\_id

JOIN

users p ON bd.patient\_id = p.id

ORDER BY

d.id, p.id;

**Task 2**

/\* Description:

\* Get the ranking of each doctor based on:

\* - the number of patients seen;

\* - the number of analyses prescribed;

\* - the number of drugs prescribed in the last quarter.

\*/

WITH patient\_count AS (

SELECT

bd.doctor\_id,

COUNT(DISTINCT bd.patient\_id) AS patient\_count

FROM

booked\_doctors bd

WHERE

bd.appointment\_time >= date\_trunc('quarter', CURRENT\_DATE) - interval '3 month'

GROUP BY

bd.doctor\_id

),

analyse\_count AS (

SELECT

bd.doctor\_id,

COUNT(aa.id) AS analyse\_count

FROM

analyse\_appointments aa

JOIN

doctor\_appointments da ON aa.doctor\_appointment\_id = da.id

JOIN

booked\_doctors bd ON da.booked\_doctor\_id = bd.id

WHERE

aa.is\_completed = TRUE

AND bd.appointment\_time >= date\_trunc('quarter', CURRENT\_DATE) - interval '3 month'

GROUP BY

bd.doctor\_id

),

drug\_count AS (

SELECT

bd.doctor\_id,

COUNT(pd.id) AS drug\_count

FROM

prescription\_drugs pd

JOIN

doctor\_appointments da ON pd.doctor\_appointment\_id = da.id

JOIN

booked\_doctors bd ON da.booked\_doctor\_id = bd.id

WHERE

bd.appointment\_time >= date\_trunc('quarter', CURRENT\_DATE) - interval '3 month'

GROUP BY

bd.doctor\_id

),

combined AS (

SELECT

u.id AS doctor\_id,

u.name AS doctor\_name,

u.surname AS doctor\_surname,

u.middle\_name AS doctor\_middle\_name,

COALESCE(pc.patient\_count, 0) AS patient\_count,

COALESCE(ac.analyse\_count, 0) AS analyse\_count,

COALESCE(dc.drug\_count, 0) AS drug\_count

FROM

users u

LEFT JOIN

patient\_count pc ON u.id = pc.doctor\_id

LEFT JOIN

analyse\_count ac ON u.id = ac.doctor\_id

LEFT JOIN

drug\_count dc ON u.id = dc.doctor\_id

WHERE

EXISTS (SELECT 1 FROM users\_roles ur WHERE ur.user\_id = u.id AND ur.role\_id = 2)

)

SELECT

doctor\_id,

doctor\_name,

doctor\_surname,

doctor\_middle\_name,

patient\_count,

RANK() OVER (ORDER BY patient\_count DESC) AS patient\_rank,

analyse\_count,

RANK() OVER (ORDER BY analyse\_count DESC) AS analyse\_rank,

drug\_count,

RANK() OVER (ORDER BY drug\_count DESC) AS drug\_rank

FROM

combined

ORDER BY

doctor\_id;

**Task 3**

/\* Description:

\* Create a view to display the trend in the number of doctor visits (by visit dates).

\*/

CREATE VIEW doctor\_visit\_trend AS

WITH daily\_visits AS (

SELECT

bd.doctor\_id,

date\_trunc('day', bd.appointment\_time) AS visit\_date,

COUNT(\*) AS daily\_visit\_count

FROM

booked\_doctors bd

GROUP BY

bd.doctor\_id, date\_trunc('day', bd.appointment\_time)

)

SELECT

u.id AS doctor\_id,

u.name AS doctor\_name,

u.surname AS doctor\_surname,

u.middle\_name AS doctor\_middle\_name,

dv.visit\_date,

dv.daily\_visit\_count,

SUM(dv.daily\_visit\_count) OVER (PARTITION BY dv.doctor\_id ORDER BY dv.visit\_date) AS cumulative\_visits

FROM

daily\_visits dv

JOIN

users u ON dv.doctor\_id = u.id

ORDER BY

u.id, dv.visit\_date;

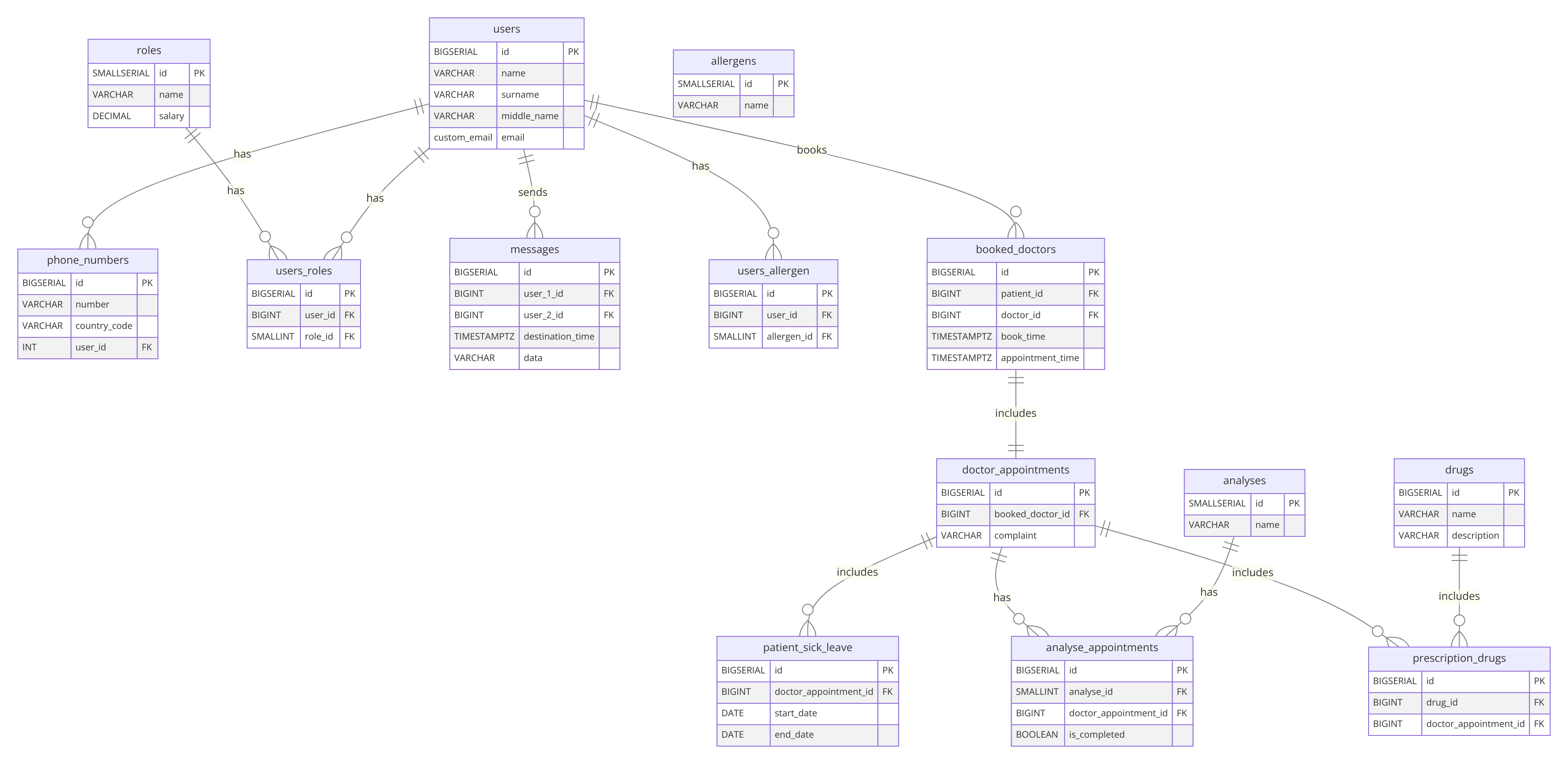
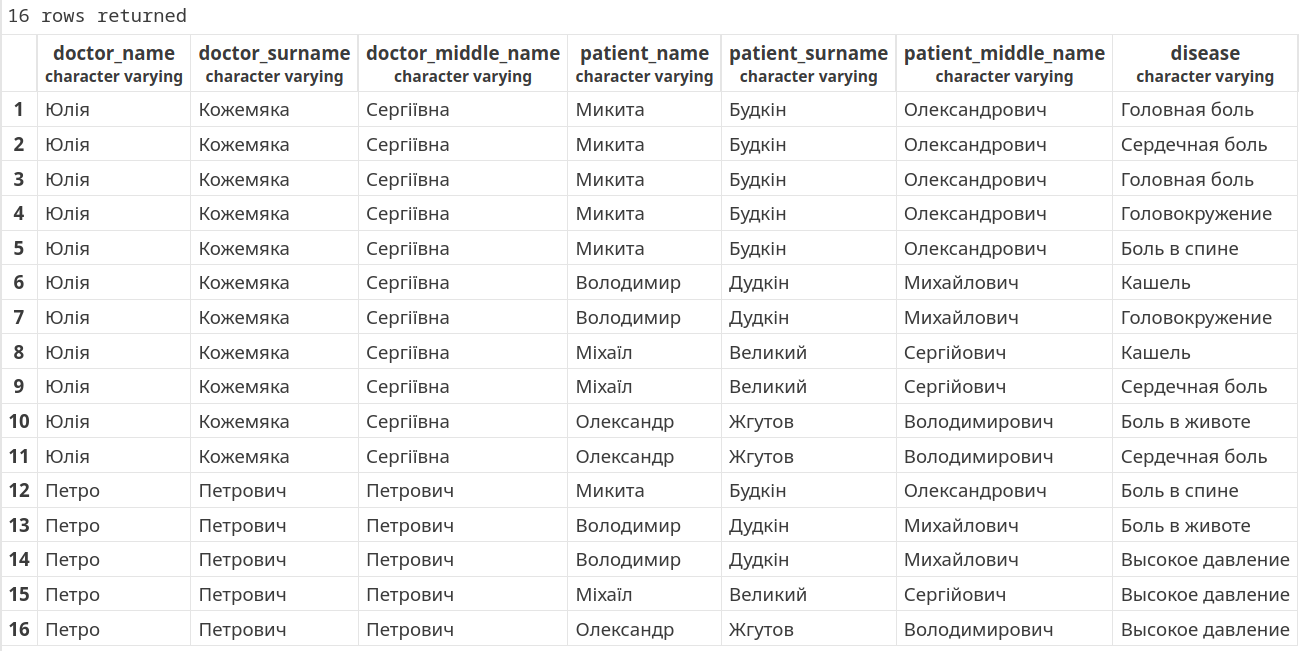
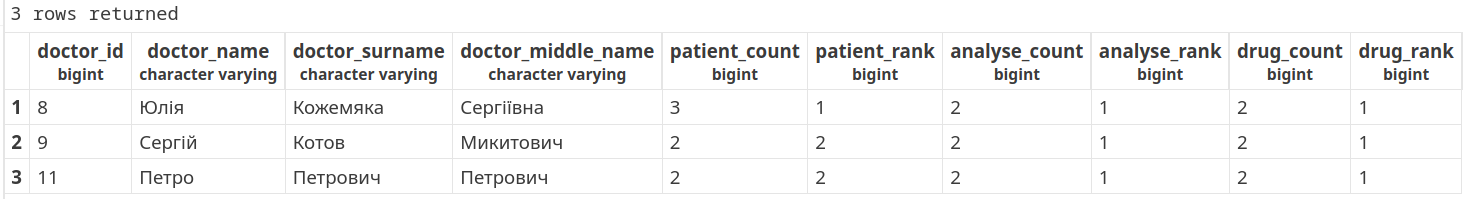
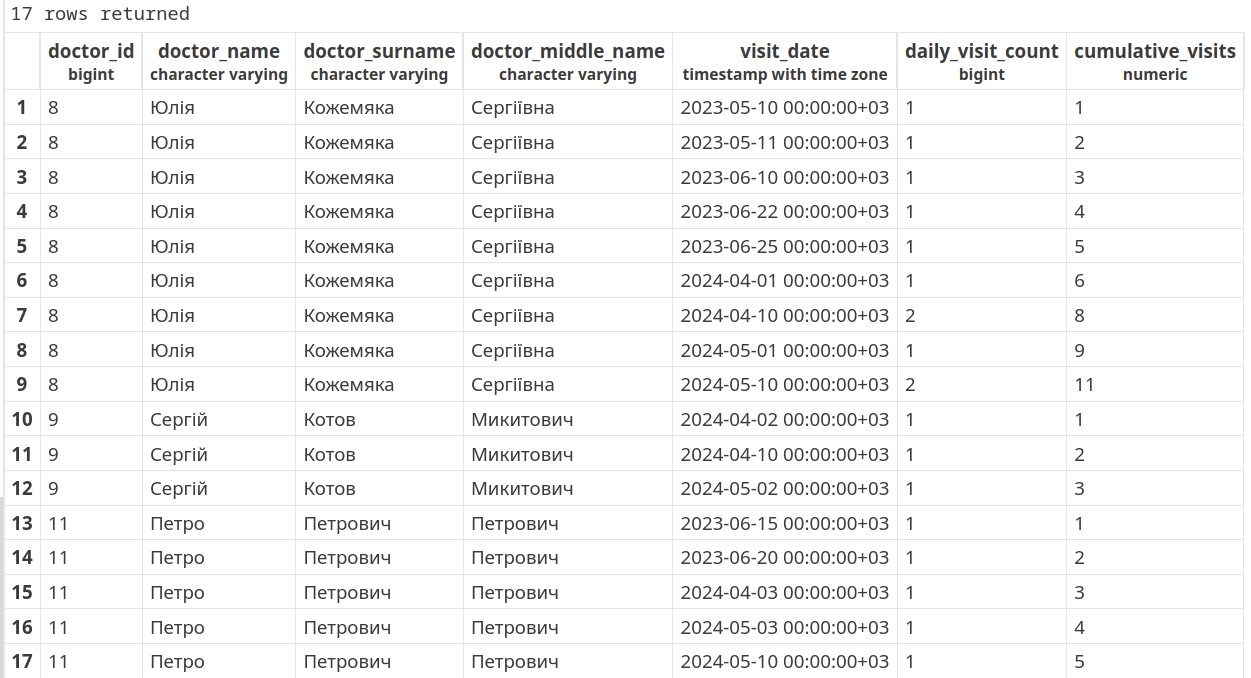


Рисунок 1 – ER діаграмма

**РЕЗУЛЬТАТ ВИКОНАННЯ**

Рисунок 2 – результат виконання завдання 1

Рисунок 3 – результат виконання завдання 2

Рисунок 4 – результат виконання завдання 3