**PROIECT – BAZE DE DATE**

**DEPARTAMENT SPITAL**

**Coordonator**: Diaconu Diana **Cursant**: Recheșan Lidia

**ATELIERELE ILBAH**

**Cuprins:**

1. Introducere………………………………………………….…….4
2. Tema lucrării…………..……………………………….…………4
3. Obiectivele/scopul lucrării……………………………….……….5
4. Crearea bazei de date…………………………………….……….5
5. Crearea unui login și a unui user………………………….……....6
6. Crearea tabelelor – fundament al bazei de date…………………...6

6.1 Crearea tabelului RespDepartment……………………...……7

6.2 Crearea tabelului Employees……………........................……7

Inserturi în tabelul Employees……………………………………..8

6.3 Crearea tabelului InPatients…………………………..………9

6.4 Crearea tabelului PatientFile……………………..…………...9

Inserturi table PatientFile……………………………………..…..10

6.5 Crearea tabelului Farmacy……………………………………10

6.6 Crearea tabelului Laboratory…………………………………..11

6.7 Crearea tabelului Treatment…………………………..……….11

7. Diagrama UML……………………………………………………12

8. VIEW……………………………………………………………..13

9. Sub-interogare……………………………………………………..14

10. INNER JOIN……………………………………………………..15

11. FULL JOIN……………………………………………………....15

12. UNION ALL……………………………………………………..16

13.Funcția COUNT…………………………………………………..17

14. Funcția MIN……………………………………………………...18

15. Funcția MAX………………………………………………….....19

16, EXISTS………………………………………………………..…20

17. GROUP BY………………………………………………………21

18. AVG……………………………………………………………22

19. STRING………………………………………………………. 23

20. SYSDATETIME……………………………………………… 23

21. LEFT()………………………………………………………… 24

22. ASCII………………………………………………………… 24

23. SUBSTRING………………………………………………… 24

24. UPDATE în tranzacție…………………………………………25

25. STORAGE PROCEDURE…………………………………… .26

26. Crearea unui Back up………………………………………… 27

27.Bibliografie…………………………………………………… 28

**Introducere**

Bazele de date reprezintă colecții de informații logic organizate astfel încât să ușureze și să eficientizeze diverse acțiuni. O bază de date este controlată de regulă de un sistem de management al bazelor de date (DBSM).

Cumulat, datele, DBSM, și aplicațiile associate reprezintă un sistem de baze de date denumit prescurtat bază de date.

Ce introducem într-o bază de date sunt date, iar ce extragem dintr-o bază de date sunt informații.

Modelul relațional are ca idei urmatoarele:

\*simplu de înțeles;

\* util în modelarea unor situații/entități din lumea reală;

\*conduce la interogări de o complexitate redusă;

-utilizează matematica în descrierea/reprezentarea înregistrărilor și a colecțiilor de înregistrări.

**Tema proiectului** este realizarea unei baze de date pentru un departament dintr -un spital de stat, care să aibă diverse proprietati uzuale, poprietatea de gestionare a informatiilor, proprietatea de adaugare a unor noi informatii, de vizualizare, stergere, etc.

Am ales această temă, deoarece, având o bază de date creată, se poate evidenția/monitoriza activitatea zilnică din departament cu detalii.

Baza de date clarifică și ușurează mult documentația cadrelor medicale minimalizând riscurile de a confunda pacientul, diagnosticul, tratamentul, investigațiile.

**Scopul** bazei de date:

-ținerea unei evidențe clare a pacienților care se internează și se externează;

-înregistrarea tuturor informațiilor legate de date personale, diagnostic, tratament, investigații.

-ușurează procesul de înregistrare si păstrare a informației,

-creează posibilitatea actualizării datelor fiecărui pacient internat,

-gestionează cheltuielile cu salariile angajaților, materiale, medicamente, reactivi pentru analizele de laborator.

Baza de date creată, memorează informații despre angajatii secției si despre pacienții secției RESPIRATOR.

Internarea se face pe baza diagnosticului pus de medicul specialist. La internare sunt înregistrate datele personale ale pacientului.

Baza de date trebuie să asigure gestionarea corespunzatoare a tuturor datelor referitoare la pacient: data internării, motivele internării, departament, diagnostic, tratament, investigații, data externării.

Se creează anumite legături între entități:

Fiecare secție/departament trebuie să aibă unul sau mai multi medici;

Fiecare medic trebuie să aparțină unui singur departament;

Fiecare pacient trebuie să aibă o singură fișă/file;

Fiecare fișă are prescris unul sau mai multe medicamente;

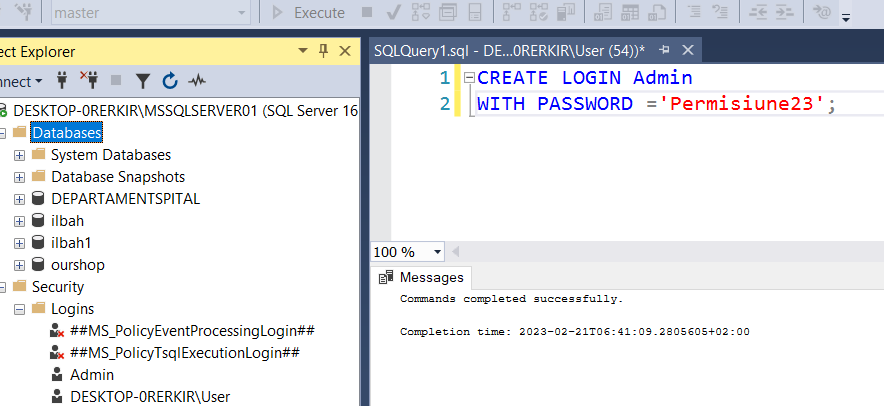
Fiecare investigație trebuie să aparțină unei singure fișe,

Fiecare medicament eliberat de farmacie apartine unui singur pacient.

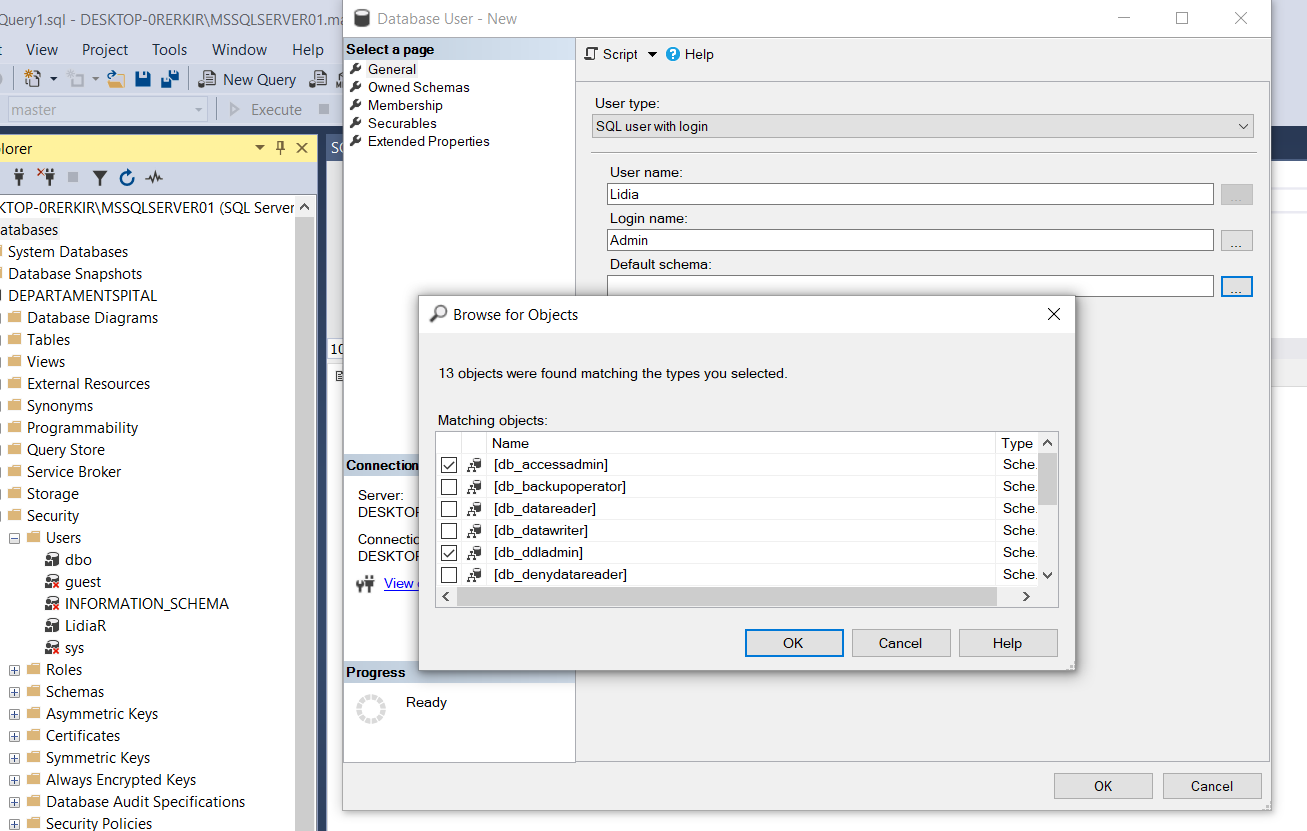
Prin modelarea entităților în SQL Developer, acestea au devenit tabele cu atribute și restricții.

Pentru a crea o bază de date nouă folosind SSMS, este necesară autentificarea pe server, din partea stangă se alege numele serverului, se extinde fie prin dublu-click, fie prin click pe simbolul ”+”. Prin executarea unui click dreapta pe folderol **Databases**, după care se alege **New Databases**. Numele bazei de date create **DEPARTAMENT SPITAL** apare automat în partea stângă, în secțiunea de **Object Explorer.**

--Folosind comanda SQL se poate crea un **Login** cu parolă:

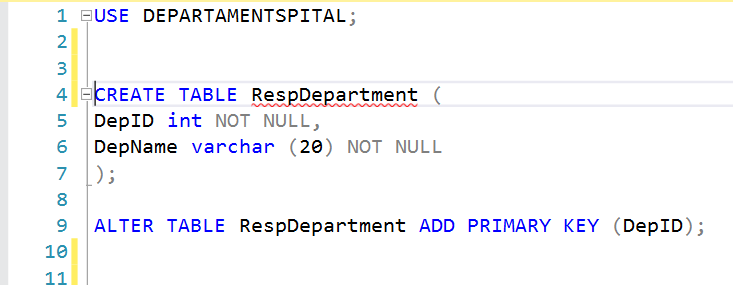


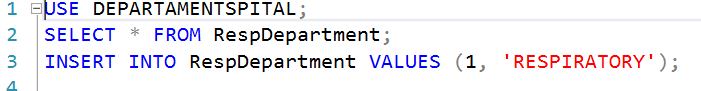
--Adăugarea unui **utilizator** pe baza de date DEPARTAMENTSPITAL cu numele Lidia, se face și cu click dreapta pe folderul Security, New User. Se pot atribui rolurile dorite (click dreapta – Properties).



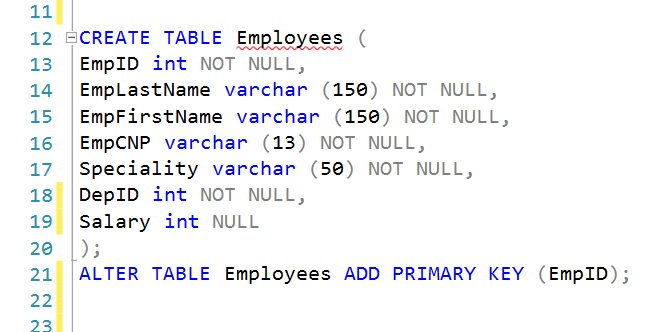
Un **tabel** este format din rânduri și coloane, având o structură ce poate păstra în interiorul acestuia toate detaliile.

--Crearea primului tabel **RespDepartment**, adăugarea cheii primare DepID și insertul se pot vedea în captura de mai jos:

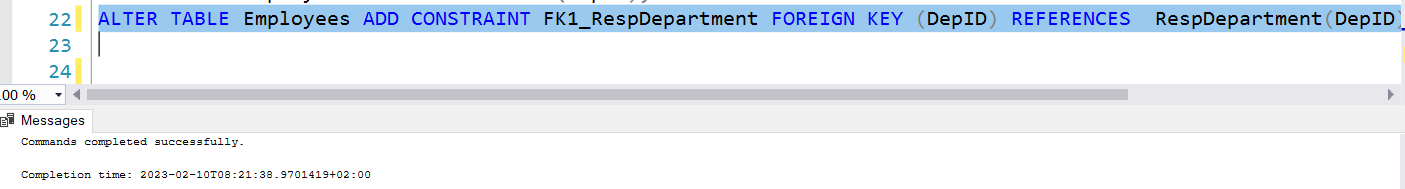




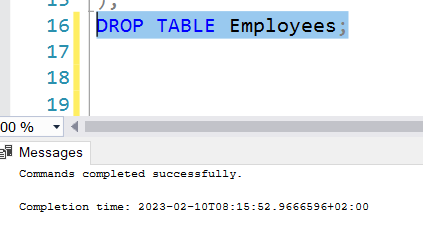
--Crearea tabelului **Employees** format din rânduri și coloane și adăugarea cheii primare (EmpID):



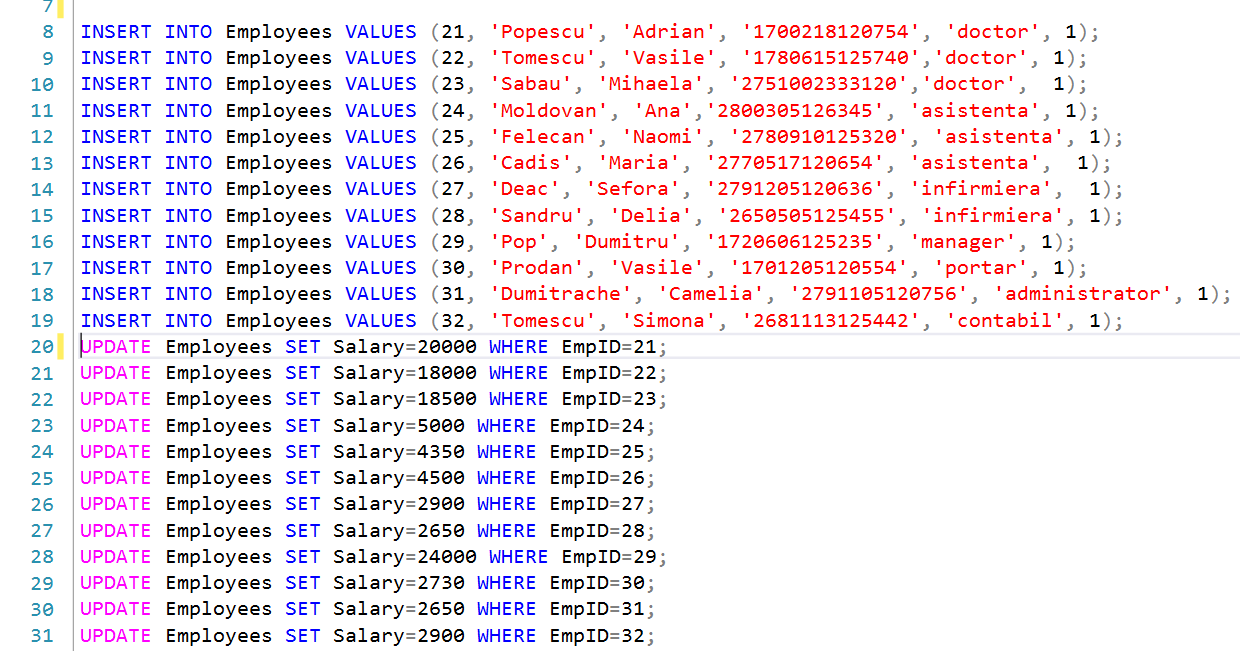
--Adăugarea **cheii străine** pe tabelul Employees:



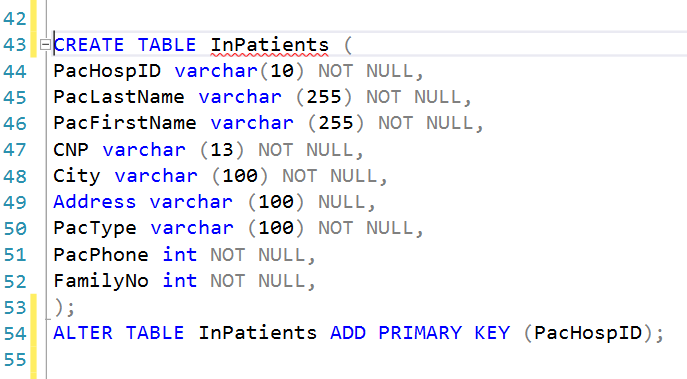
--Captură **stergere tabel :** aplicarea comenzii **DROP TABLE** pe tabelul Employees, a șters tabelul care a trebuit recreat cu comanda CREATE TABLE.



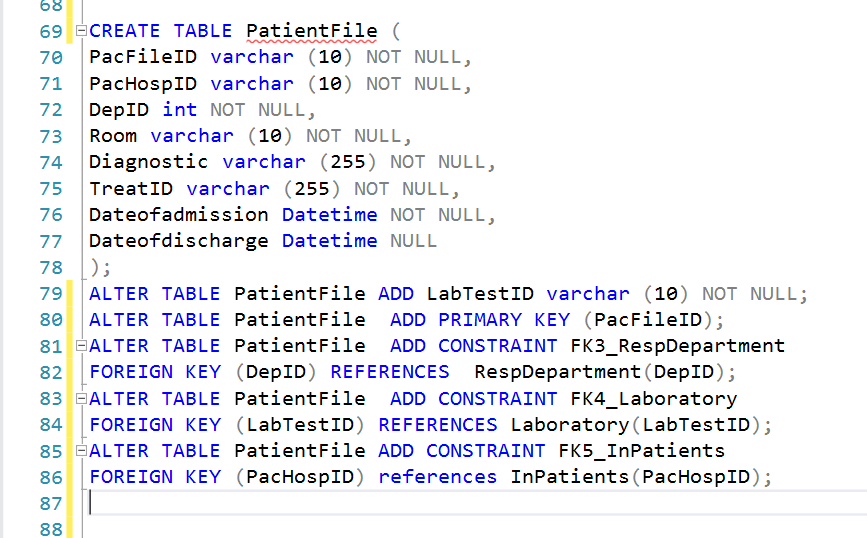
--**Inserturile** din tabelul Employees se pot vedea în captura de mai jos. Coloana Salary a fost adaugată ulterior, motiv pentru care s-a facut UPDATE pentru fiecare înregistrare.



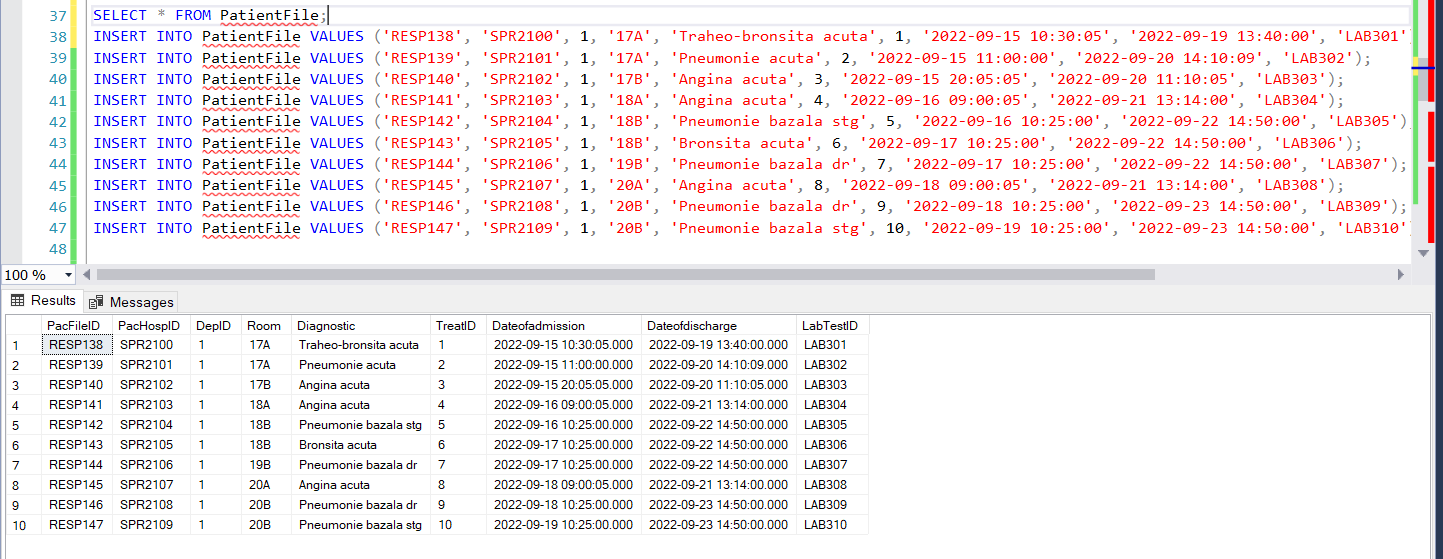
--Crearea tabelului **InPatients**:



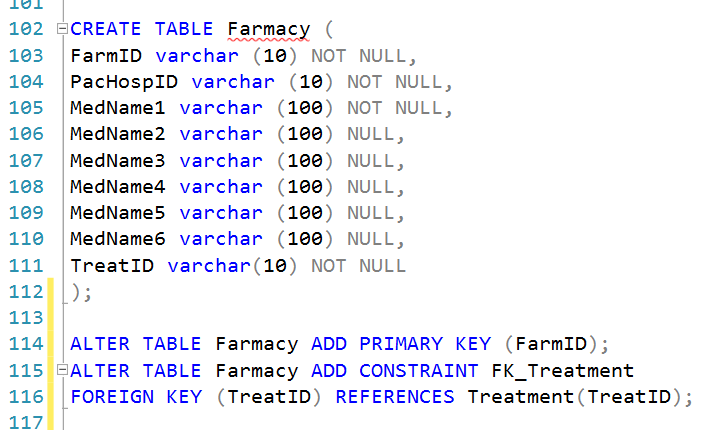
--Crearea tabelul **PatientFile** cu adăugarea coloanei LabTestID, a cheii primare și a cheilor străine:



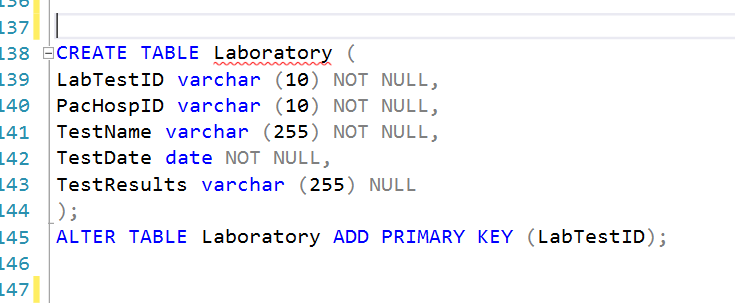
--Captură **inserturi** în tabelul PatientFile**+SELECT**



--Crearea tablului **Farmacy** cu adaugarea cheii primare și a cheii străine:

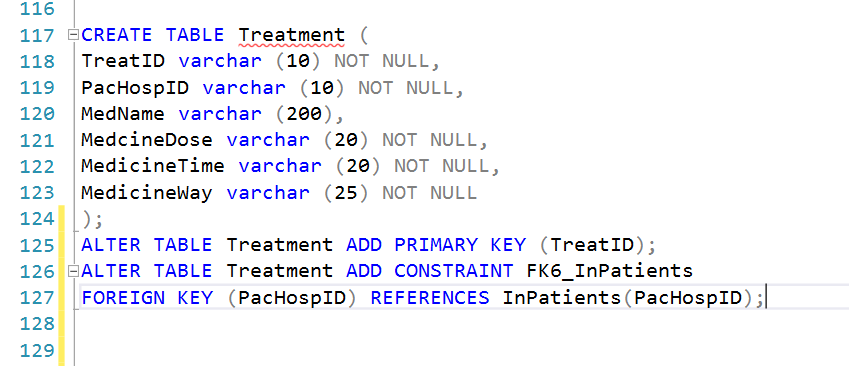


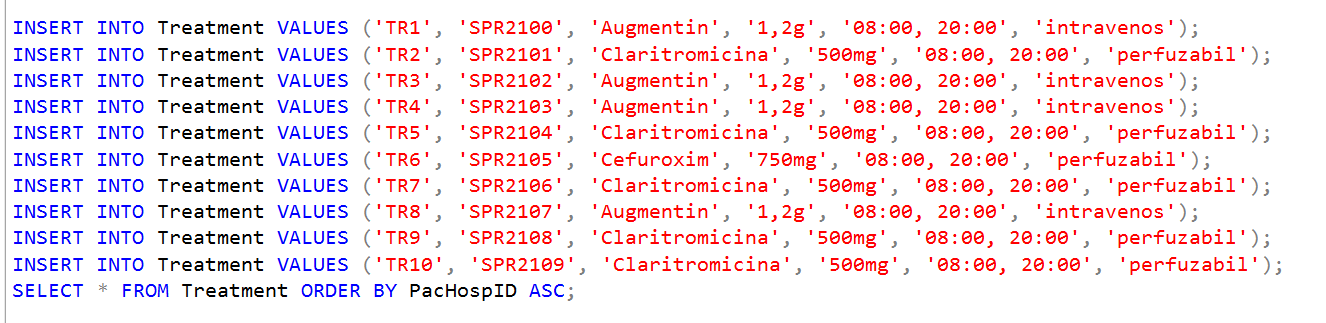
--Crearea tabelului **Laboratory** cu adăugarea cheii primare:



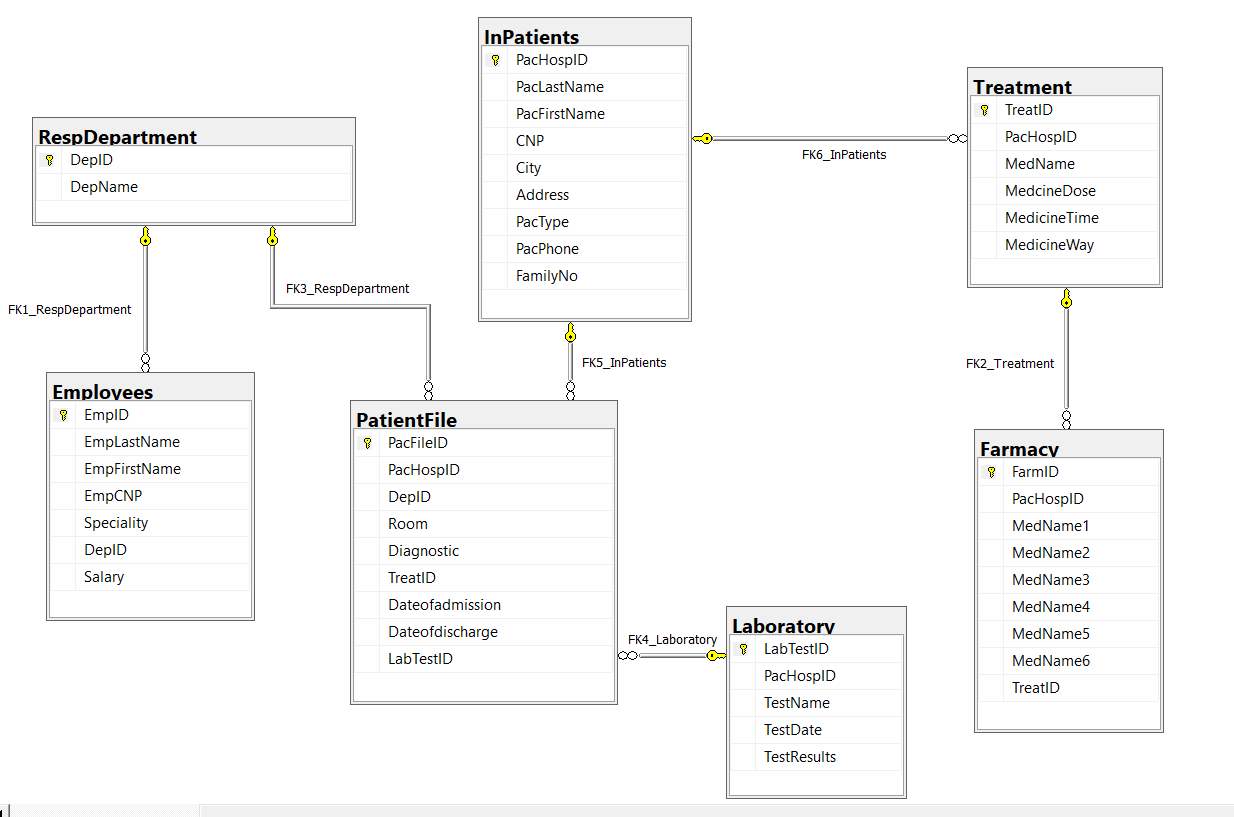
--Crearea tabelului **Treatment** cu adăugarea cheii primare și străine+inserturi.

--Pentru că, coloanna TreatID nu apărea în ordine crescătoare, am folosit **ORDER BY ASC**, așa cum se vede în captură:

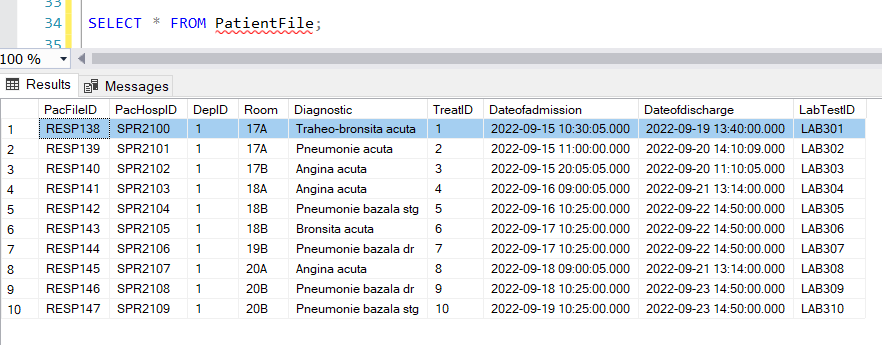




**--Diagrama UML** întocmește structura bazei de date DEPARTAMENTSPITAL, afișează cheile primare și cheile străine care leagă tabelele:

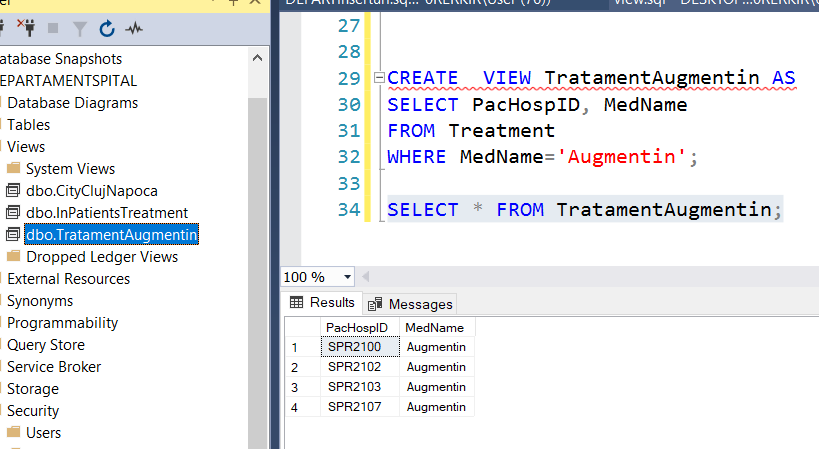


-- **SELECT** \* din tabelul PatientFile a returnat toate înregistrările din tabel.



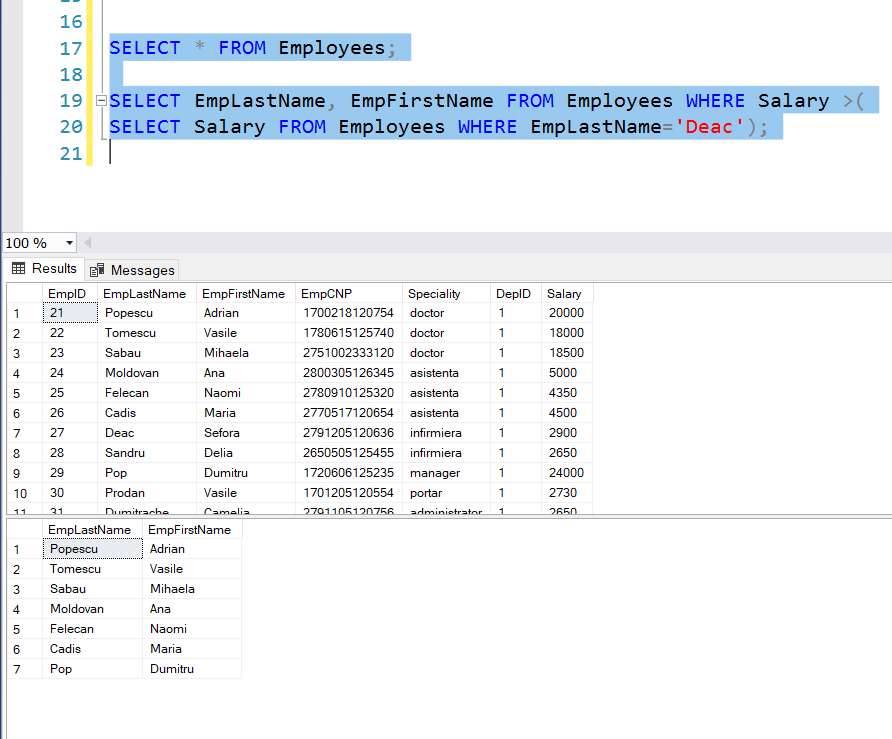
--**VIEW** – ul atașat, referențiază date din tabelul Treatment afisând PacHospID=Pacientii care sunt cu Augmentin.

O vizualizare conține la fel ca și orice tabel rânduri și coloane pentru afișarea datelor, date care sunt preluate și referite din tabelele din care fac parte. Setul de date returnat să fie păstrat pentru mai târziu care să se execute atunci când avem nevoie.

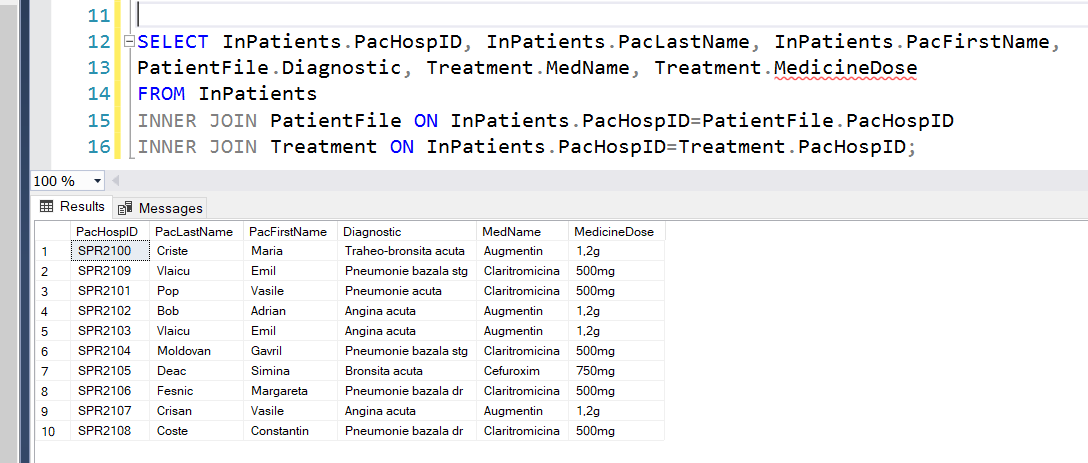


**Sub-interogare**: reprezentată de posibilitatea de a aplica o interogare peste rezultatul unei alte interogări, într-o singură comandă SQL.

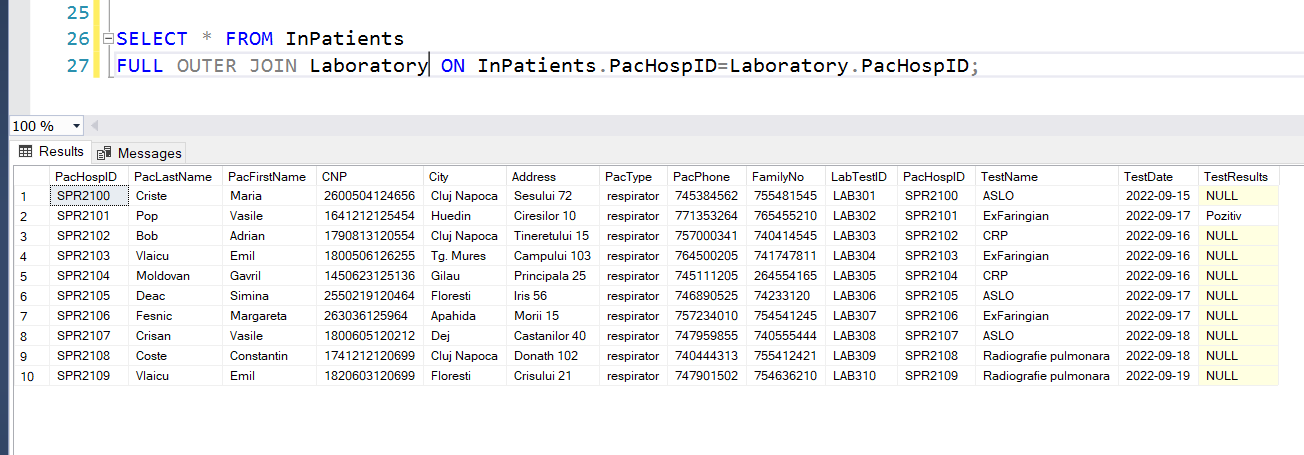
--Rezultatul interogării returnează salariile angajaților care au salar mai mare decat angajatul ”Deac” (2900)



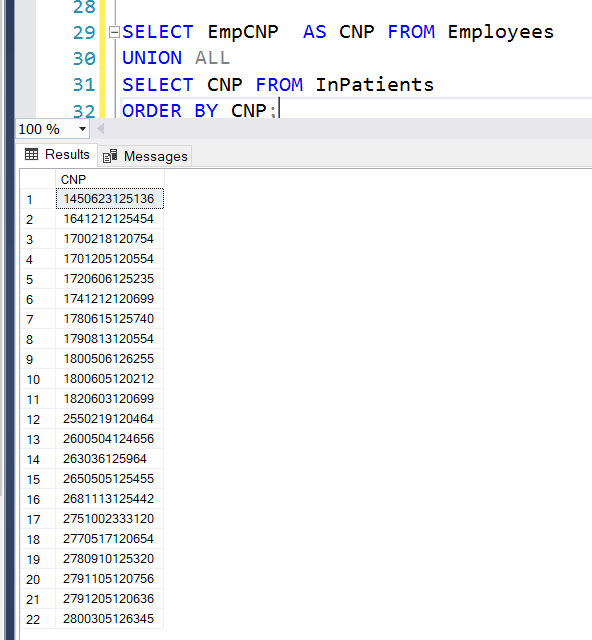
--Folosind în interogarea de mai jos **INNER JOIN**, a returnat rezultatul dorit– fiecare pacient ce diagnostic are și ce tratament i se administrează.



--**FULL JOIN** a returnat toate înregistrările din tabelul InPatients și toate înregistrările din tabelul Laoratory:



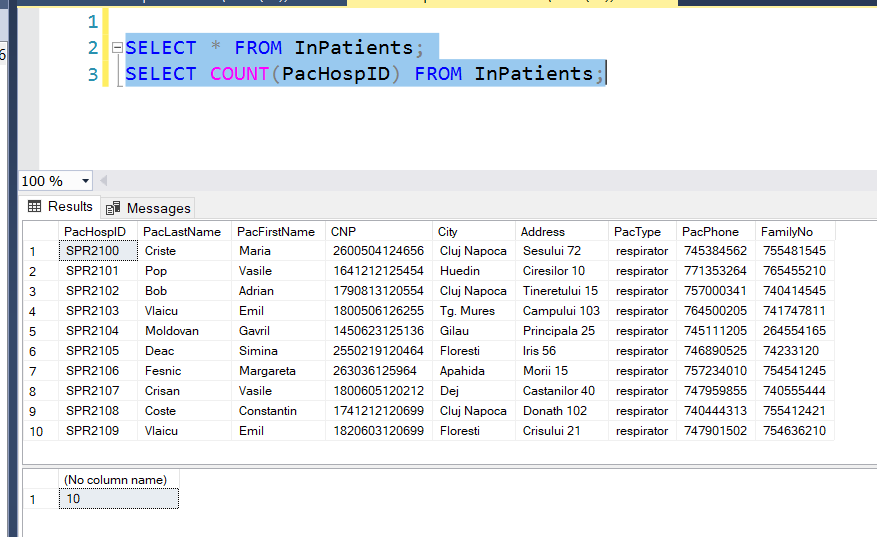
--Utilizarea comenzii **UNION ALL** și **ALLIAS** a returnat toate înregistrările (păstrând duplicatele dacă există), combinându-le într-un singur set de date.



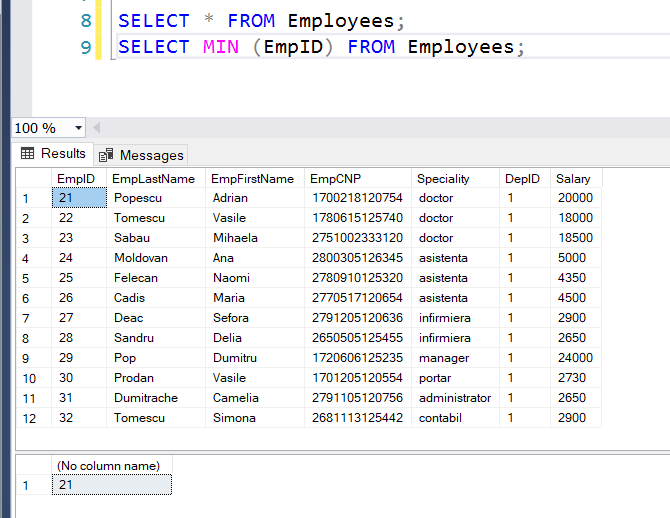
**Funcțiile** sunt blocuri de instrucțiuni care primesc anumite date de intrare și produc anumite date de ieșire.

Funcțiile in SQL vor returna întotdeauna o valoare și sunt în general folosite pentru a efectua calcule rapide, eficiente și repetitive, pentru a modifica și altera date din tabele, pentru a converti diferite tipuri de date și pentru a afișa date si numere.

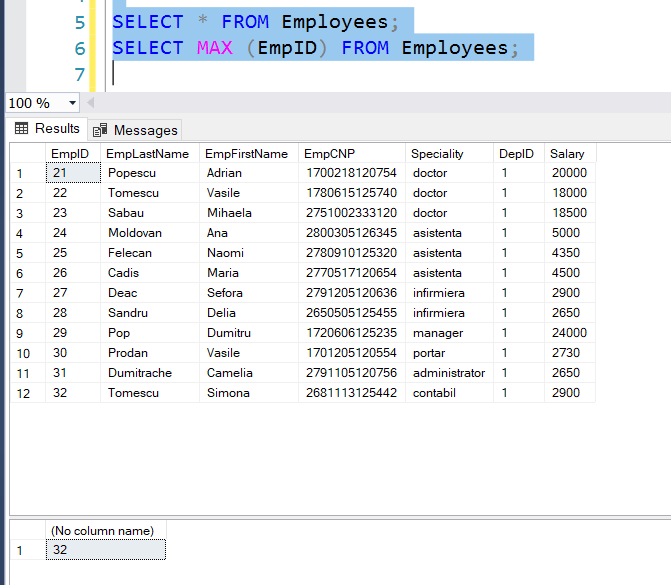
-- Funcția **COUNT** aplicată pe tabelul InPatients a returnat numărul de pacienți internați.



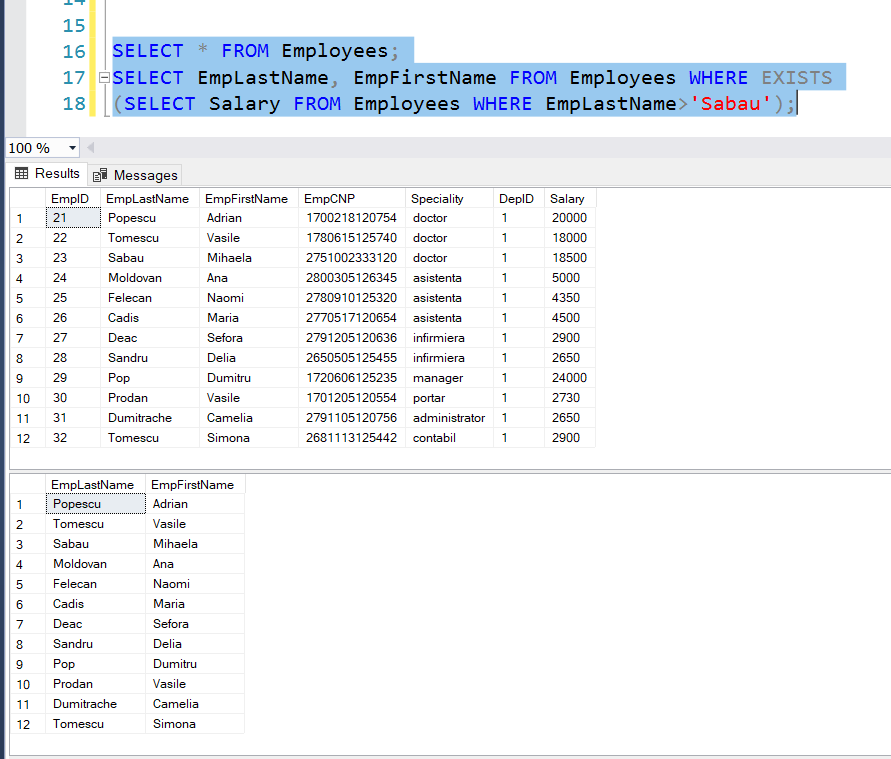
--Funcția **MIN** returnează valoarea minimă a coloanei EmpID din tabelul Employees care este 21.



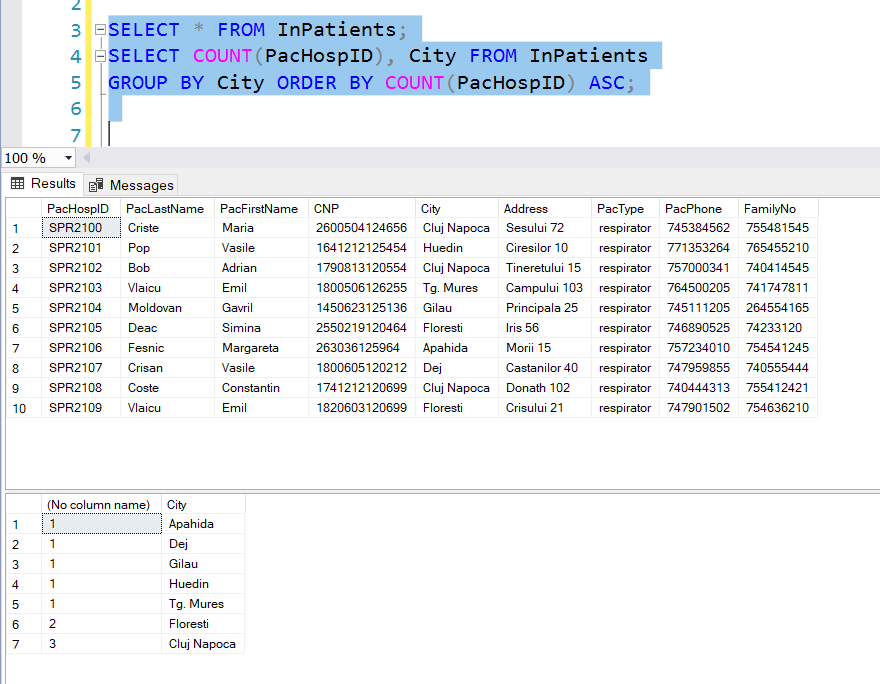
--Funcția **MAX** aplicată pe tabelul Employees returnează cea mai mare valoare a coloanei EmpID care este 32.



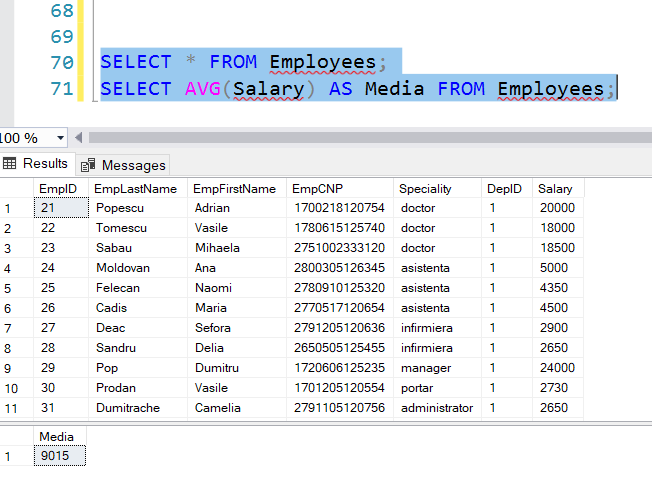
--Funcționalitatea **EXISTS** a returnat valoarea TRUE:



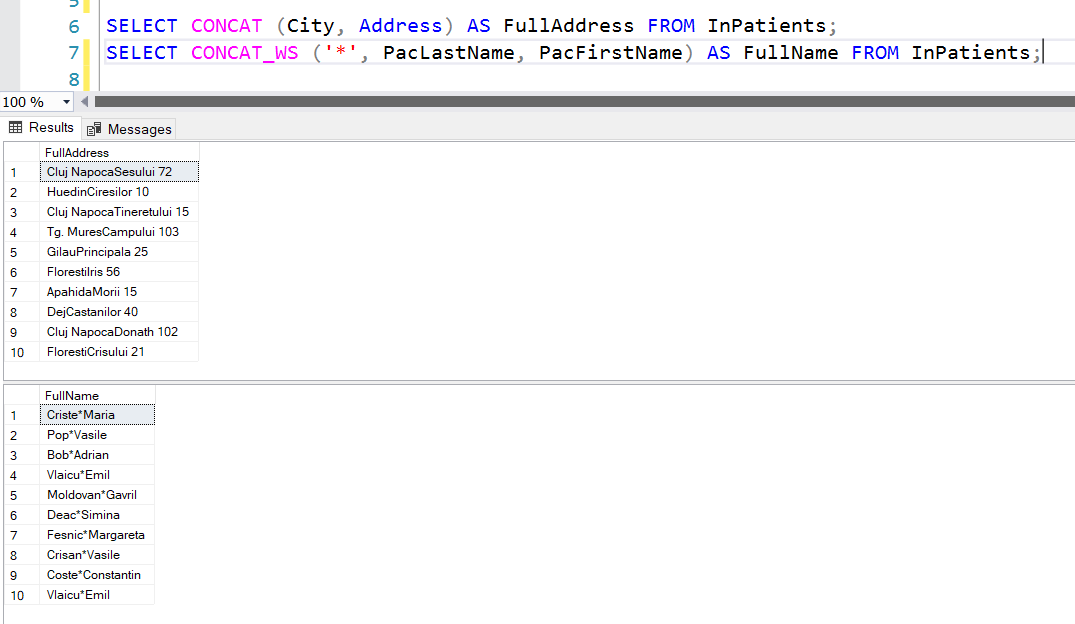
**--GROUP BY** – în SELECT-ul de mai jos a calculat și a grupat localitățile identice, afisându-le în ordine crescătoare:



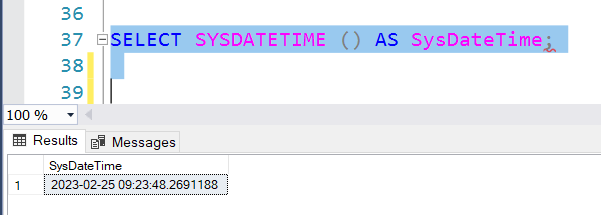
--Funcția **AVG** aplicată pe coloana Salary din tabelul Employees a returnat valoarea medie a valorilor din coloană, acesta reprezentând media salarului.



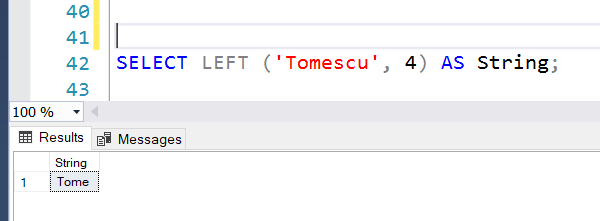
--Funcții **STRING:** au unit în primul select City cu Address, iar în al doiliea exemplu a folosit \* pentru a delimita PacLastName de PacFirstName.



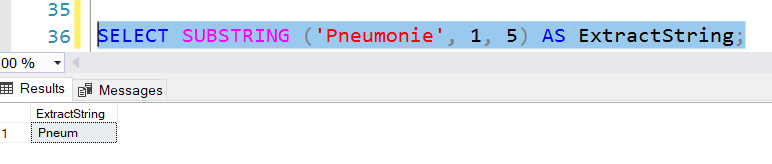
--**SYSDATETIME** a returnat data și ora serverului pe care rulează baza de date:



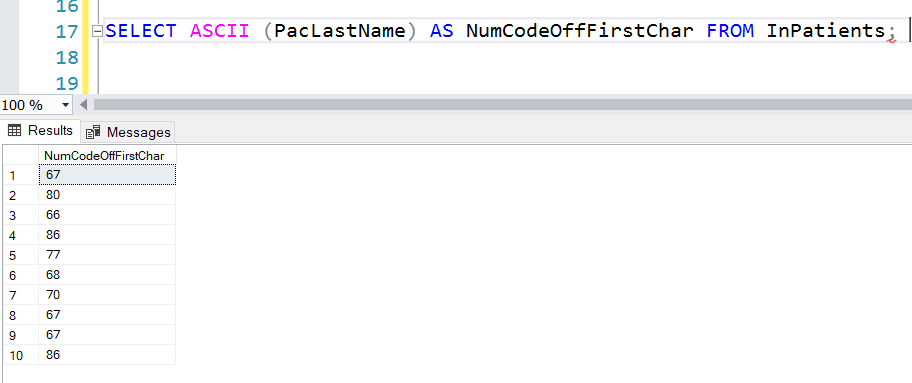
--Funcția **LEFT ()** a extras 4 caractere din șir începând cu partea stângă.



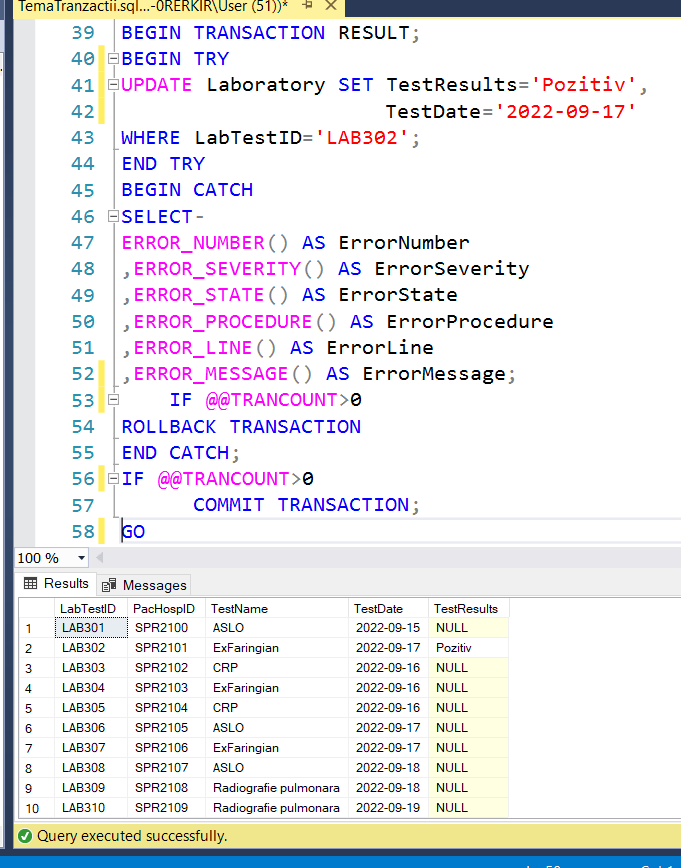
--Funcția **SUBSTRING** a returnat un nou string extras din altul.



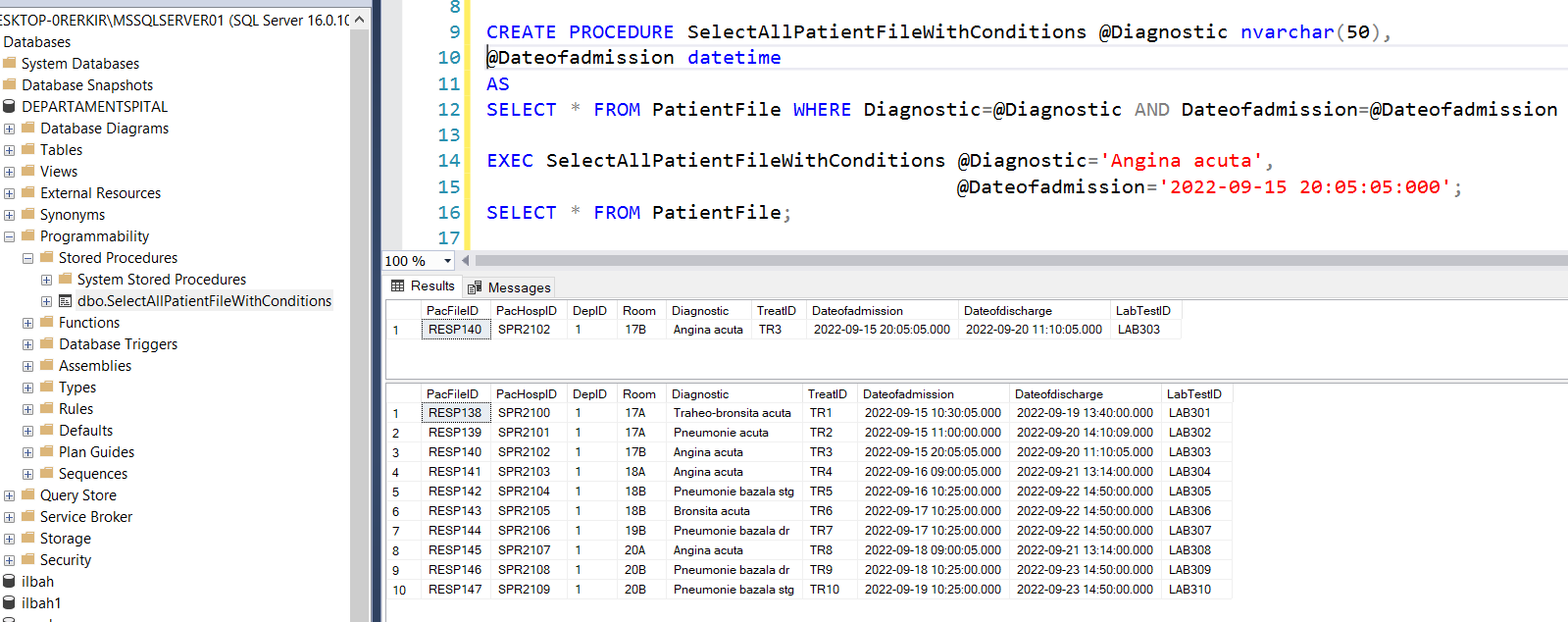
--Funcția **ASCII** a returnat valoarea ASCII a primului character din coloana PacFirstName



--Rezultatul unei investigații a fost introdus în tabel folosind **UPDATE într-o tranzacție**.

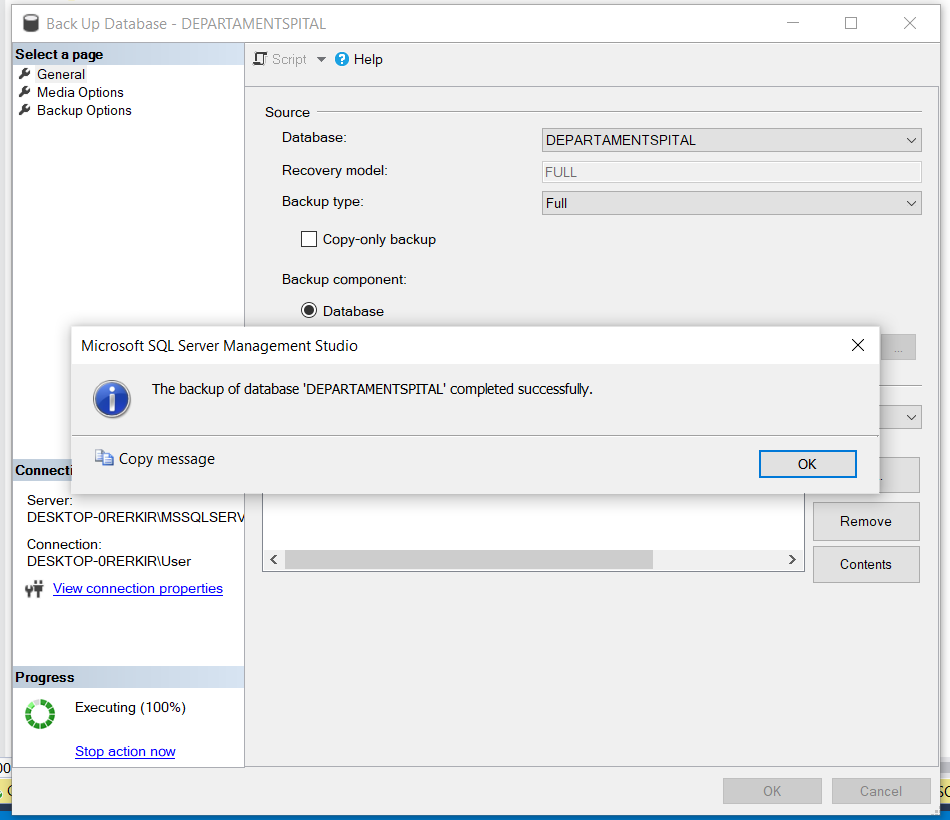


--Captură **Stored Procedure** folosind două variabile a returnat pacientii cu Angină acută internați în data de 15.09.2022. Aceasta informație/procedură stocată poate fi reutilizată ori de câte ori este nevoie.



--Pentru a crea **back up**-ul bazei de date DEPARTAMENTSPITAL am folosit SSMS, click dreapta pe baza de date, apoi am ales Tasks și Back Up…

Back-up-ul este o copie de rezervă a informațiilor, datelor, fișierelor care se doresc a fi păstrate într-un meiu de stocare separat, pentru a nu se pierde.



**Bibliografie:**

**-** Arhitectura bazelor de date-Mediul SQLde Dona Fusaru;

- Baze de date curs UBB Cluj, Dan Mircea;

- SQL tutorial – Full Database Course by Mitchell Pearson;

-https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16;

-https://docs.microsoft.com/en-us/sql/t-sql/language-elements/try-catch-transact-sql?view=sql-server-ver16;

-https://support.microsoft.com/en-us/topic/how-to-schedule-a-database-backup-operation-by-using-sql-server-management-studio-in-sql-server-

2005-58dc6e60-6c2a-7982-870d-bf34dbafcf7c;