**1) Descrierea modelului real, a utilității acestuia și a regulilor de funcționare**

În acest proiect am implementat o bază de date pentru un spital. La baza acestuia se află pacienții, întrucât scopul spitalului este de a rezolva problemele acestora prin intermediul angajaților. Angajații pot avea diverse posturi, cum ar fi doctor consultant, chirurg, asistent medical, medic de gardă, fiecare având un rol și o ocupație diferită în cadrul unui spital, dar scopul este același: tratarea și rezolvarea problemelor pacienților. Odată ce un pacient este consultat, există două posibile rezultate: acesta nu are probleme grave și îi sunt prescrise anumite medicamente pe care le poate cumpăra din cadrul spitalului sau acesta este într-o stare gravă, poate chiar inconștient (de obicei aceștia fiind aduși de ambulanța condusă de angajații spitalului), și este necesar să fie plasat în secția potrivită pentru el, spitalul având mai multe secții precum cea de pediatrie, de maternitate, de vârstnici, psihiatrie etc. Spitalul le asigură tuturor pacienților echipamentele necesare tratării lor în secțiile respective. De asemenea, le oferă și posibilitatea de a cumpăra medicamentele cel mai bine potrivite stării și afecțiunilor lor. Pentru cazuri dificile spitalul este dotat și cu săli de operație unde pacientul urmează a fi tratat de medicul chirurg.

**2) Descrierea constrângerilor impuse asupra modelului.**

În acest model, fiecare pacient trebuie să primească tratament de la angajat. Toți angajații ocupă un post în cadrul spitalului. Toate secțiile au o anumită locație și categorie stabilită în spital. O ambulanță funcționează doar cu ajutorul angajaților (spre exemplu un șofer) și este necesar să aibă o destinație. Angajatul lucrează în cadrul spitalului doar dacă îi este alocat un salariu. Salariul crește în funcție de vechimea și postul ocupat de acesta. Echipamentele pot avea doar o stare funcțională sau nefuncțională. Costurile unei operații pot fi zero. Pretul unui medicament poate fi, de asemenea, zero.

**3)** **Descrierea entităților și a cheilor primare.**

În această bază de date am implementat 9 entități pentru diagrama entitate-relație. Totul pornește de la entitatea **Spital** (cheie primară: id\_spital) în jurul căruia se învârt și celelalte entități. Acesta este împărțit în mai multe **Secții** (cheie primară: id\_sectie), fiecare fiind de o anumită categorie. În cadrul secțiilor avem **Echipamente** (cheie primară: id\_echipamente) cu rolul de a susține și a ajuta pacienții, unele echipamente fiind chiar vitale în a-i ține în viață. În acest spital avem mai mulți **Angajați** (cheie primară: id\_angajati) care lucrează, fiecare o anumită vechime și, de asemenea, ocupă un post diferit, cum ar fi medic consultant, chirurg, în funcție de abilitățile fiecăruia. Ocupația principală a angajaților este de a trata **Pacienții** (cheie primară: id\_pacient), care suferă de diverse boli, mai mult sau mai puțin grave și doresc să se vindece. Relația dintre angajat și pacient este cea de **Tratament** (cheie primară: id\_tratament) întrucât angajatul oferă tratamentul iar pacientul îl primește. Din nefericire, se poate ajunge și în momente mai dificile in care unii pacienți au nevoie de una sau mai multe **Operații** (cheie primară: id\_operatie) unde este nevoie de intervenția angajaților, a celor mai buni medici. Pe de altă parte, pot fi și cazuri mai puțin grave unde problemele pacienților sunt rezolvate prin consumul **Medicamentelor** (cheie primară: id\_medicamente). Un alt element important, prezent în spital, este **Ambulanța** (cheie primară: id\_ambulanta), care faciliteaza în multe cazuri transportul dinspre și către spital, întrucât pacienții pot fi în stare gravă.

**4) Descrierea relațiilor și a cardinalității acestora**

Relația **SPITAL - SECTIE**

În cadrul unul spital **se află** mai multe secții, în funcție de tipul de boală a pacientului (1-M).

Relația **SECTIE - ECHIPAMENT**

O secție **folosește** mai multe echipamente (1-M).

Relația **SPITAL - ANGAJAT**

Într-un spital **lucrează** mai mulți angajați, fiecare având un anumit rol (1-M).

Relația **ANGAJAT – TRATAMENT**

Mai mulți angajați **oferă** un tratament (1-M).

Relația **TRATAMENT** – **PACIENT**

Mai mulți pacienți **primesc** un tratament (1-M).

Relația **ANGAJAT** – **OPERATIE**

În cazul în care boala este într-un stadiu avansat avem mai mulți angajați care **aplică** o operație unui pacient (1-M).

Relația **PACIENT** – **MEDICAMENTE**

Un pacient **consumă** mai multe medicamente, în funcție de boală (1-M).

Relația **PACIENT** – **OPERATIE**

Un pacient **suferă** mai multe operații, dacă boala sa este într-un stadiu foarte avansat (1-M).

Relația **ANGAJAT** - **AMBULANTA**

Un angajat **conduce** o singură ambulanță (1-1).

**5) Descrierea atributelor, a tipurilor de date, eventuale constrângeri, a valorilor implicite sau posibile ale atributelor**

Spital:

* id\_spital int primary key,
* id\_sectie int foreign key;
* id\_angajat int foreign key;
* adresa\_spital varchar(50) not null,
* nume\_spital varchar(50) not null,
* categorie\_spital varchar(20) not null.

Sectie:

* id\_sectie int primary key,
* id\_echipamente int foreign key;
* nume\_sectie varchar(30) not null,
* categorie\_sectie varchar(20) not null,
* locatie\_sectie varchar(20) not null.

Echipamente:

* id\_echipamente int primary key,
* nume\_echipament varchar(50) not null,
* stare varchar(20) not null, **valori** posibile=”functional” sau ”nefunctional”,
* cantitate varchar(30) not null.

Angajat:

* id\_angajat int primary key,
* nume\_angajat varchar(30) not null,
* prenume\_angajat varchar(30) not null,
* data\_nasterii varchar(20) not null,
* salariu varchar(10) not null,
* post varchar(30) not null,
* vechime varchar(10) not null.

Ambulanta:

* id\_ambulanta int primary key,
* id\_angajat int foreign key;
* destinatie varchar(30) not null.

Tratament:

* id\_tratament int primary key,
* id\_angajat int foreign key;
* id\_pacient int foreign key;
* data\_inceput varchar(30) not null.

Operatie:

* id\_operatie int primary key,
* id\_angajat int foreign key;
* id\_pacient int foreign key;
* costuri varchar(10) not null,
* tip\_operatie varchar(20) not null,
* sala varchar(10) not null,
* data\_operatie varchar(20) not null.

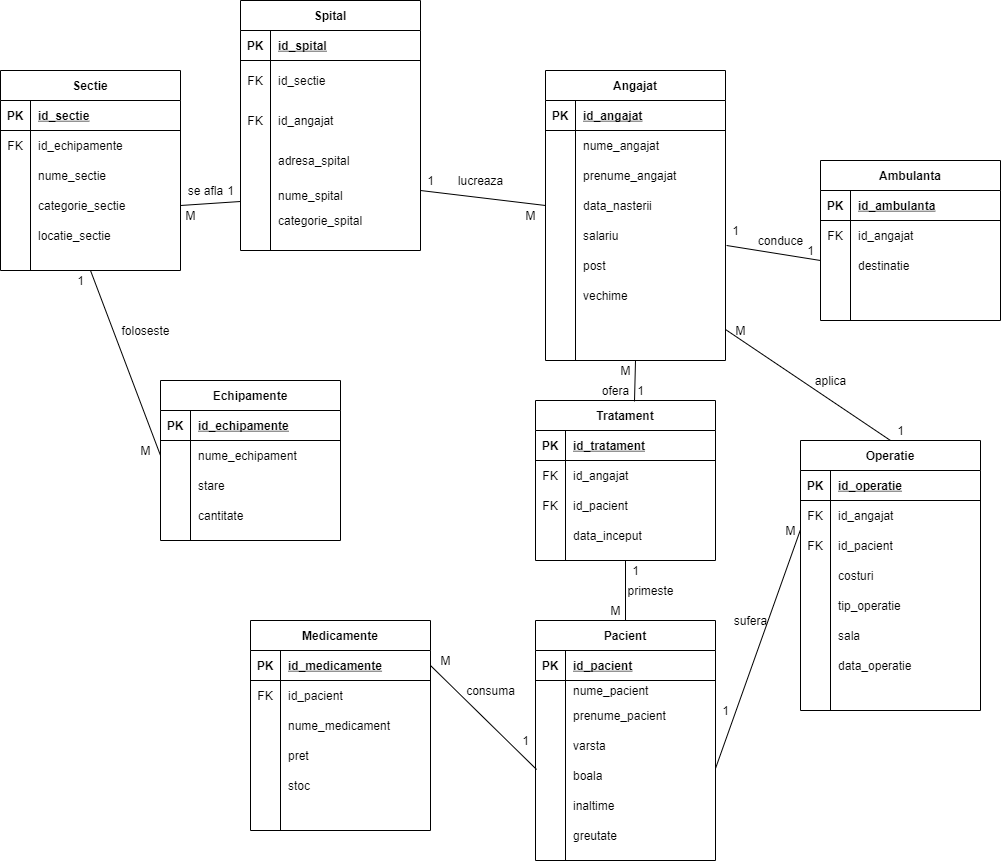
Pacient:

* id\_pacient int primary key,
* nume\_pacient varchar(30) not null,
* prenume\_pacient varchar(30) not null,
* varsta varchar(10) not null,
* boala varchar(30) not null,
* inaltime varchar(10) not null,
* greutate varchar(10) not null.

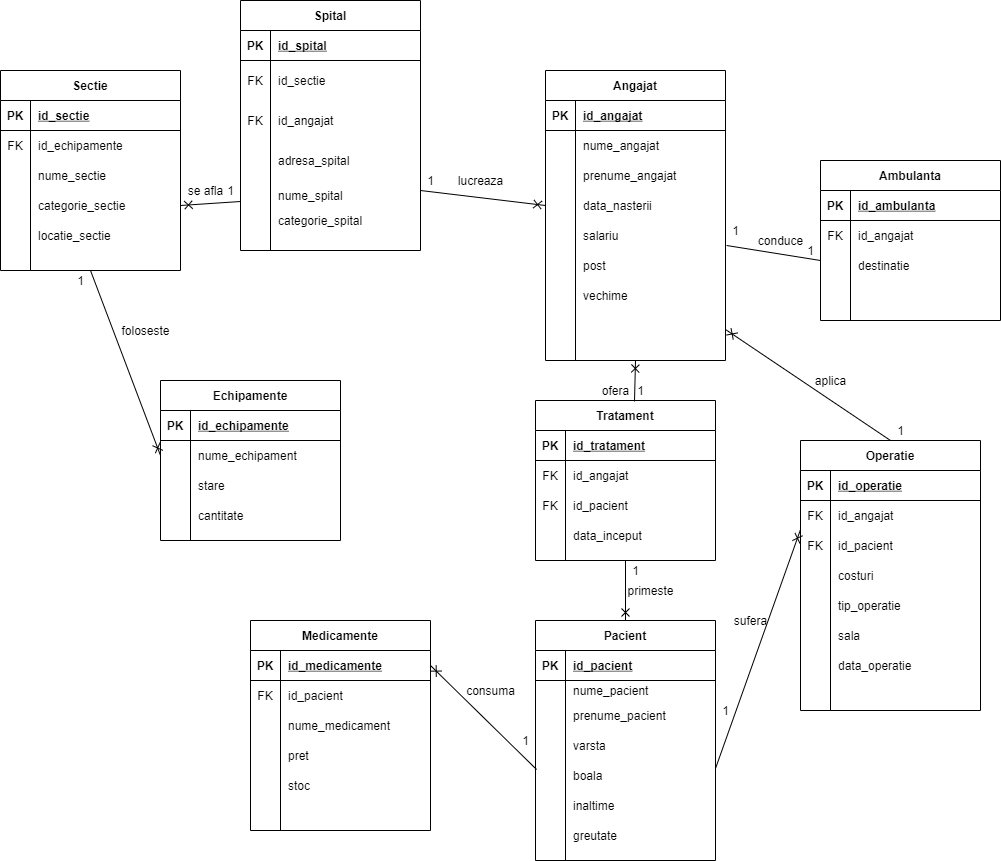
Medicamente:

* id\_medicamente int primary key,
* id\_pacient int foreign key;
* nume\_medicament varchar(30) not null,
* pret varchar(10),
* stoc varchar(10) not null.

**6) Diagrama Entitate-Relatie**



**7) Diagrama conceptuală, inclusiv tabelul asociativ**



**8) Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale.**

Spital (#id\_spital, id\_sectie, id\_angajat, adresa\_spital, nume\_spital, categorie\_spital)

Sectie (#id\_sectie, id\_echipamente, nume\_sectie, categorie\_sectie, locatie\_sectie)

Echipamente (#id\_echipamente, nume\_echipament, stare, cantitate)

Angajat (#id\_angajat, nume\_angajat, prenume\_angajat, data\_nasterii, salariu, post, vechime)

Ambulanta (#id\_ambulanta, id\_angajat, destinatie)

Tratament(#id\_tratament, id\_angajat, id\_pacient, data\_inceput)

Pacient (#id\_pacient, nume\_pacient, prenume\_pacient, varsta, boala, inaltime, greutate)

Operatie (#id\_operatie, id\_angajat, id\_pacient, costuri, tip\_operatie, sala, data\_operatie)

Medicamente (#id\_medicamente, id\_pacient, nume\_medicament, pret, stoc)

**9) Realizarea normalizării până la forma normala 3 (FN1-FN3).**

Non-FN1

Angajat(#id\_angajat, nume\_angajat, prenume\_angajat, data\_nasterii, salariu, post, vechime)

FN1

Angajat(#id\_angajat, salariu, post)

Non-FN2

angajat(#id\_angajat, nume\_angajat, prenume\_angajat, data\_nasterii, salariu, post, vechime)

FN2

Chirurg(#id\_chirurg, nume\_chirurg, prenume\_chirurg, data\_nasterii, salariu, post, vechime)

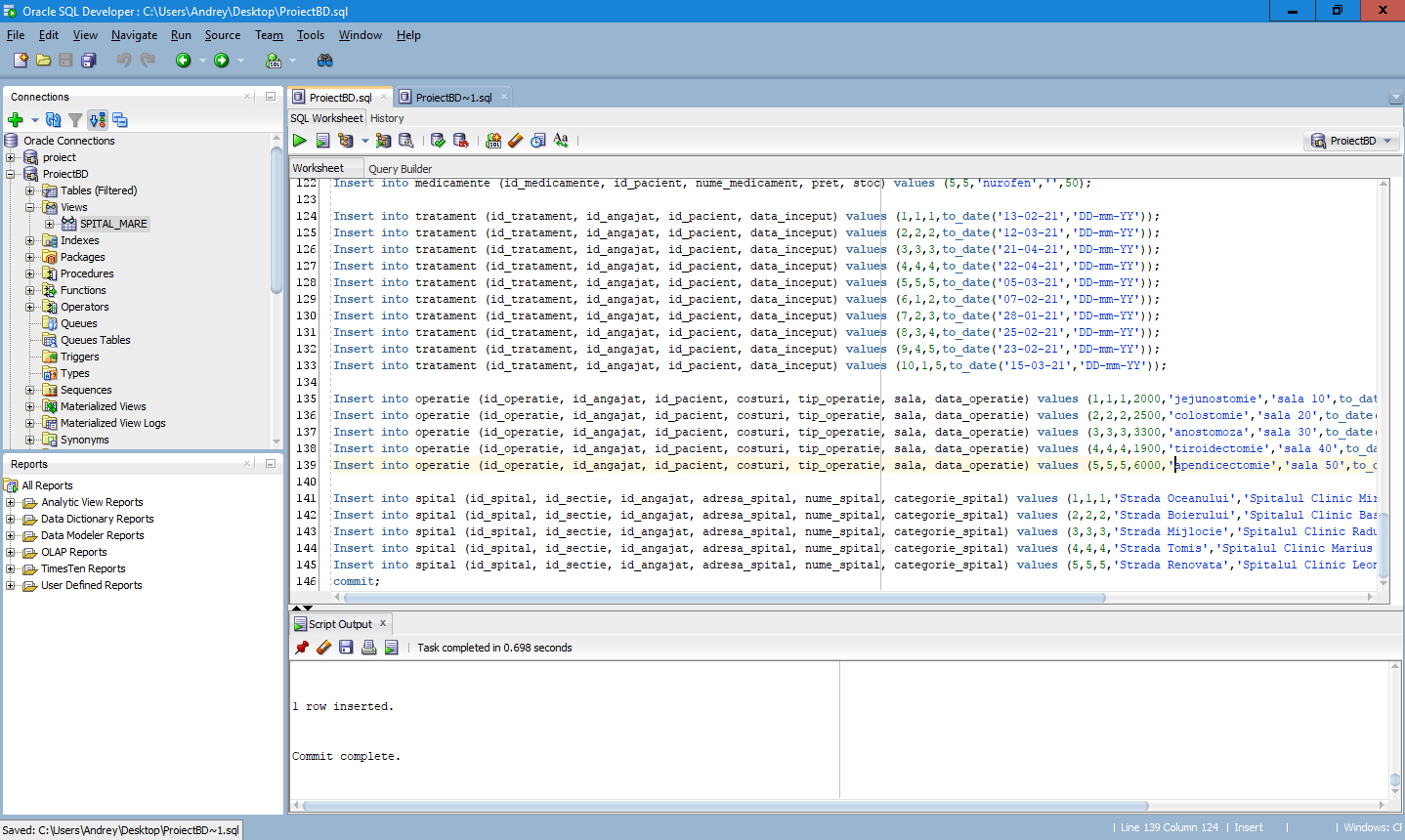
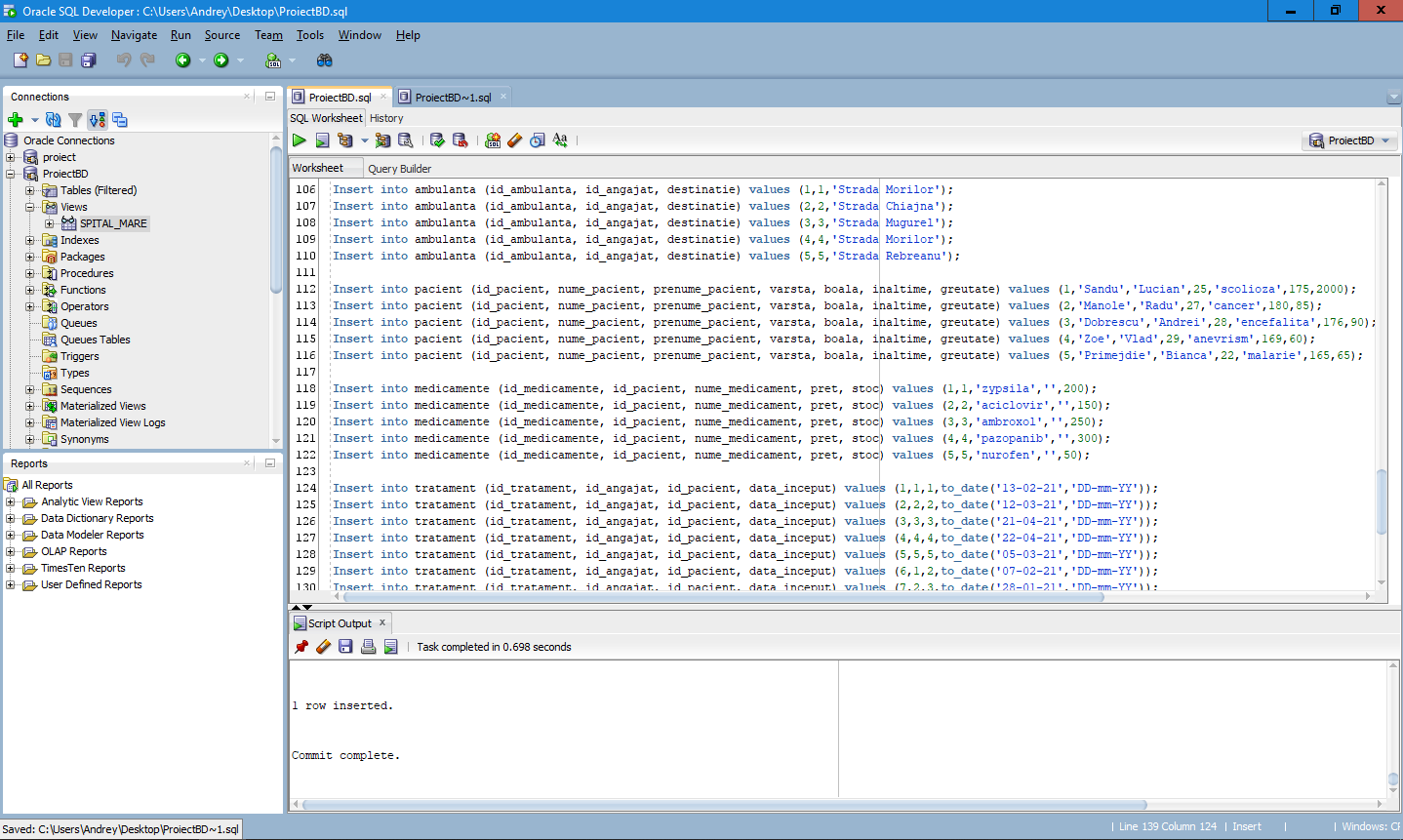
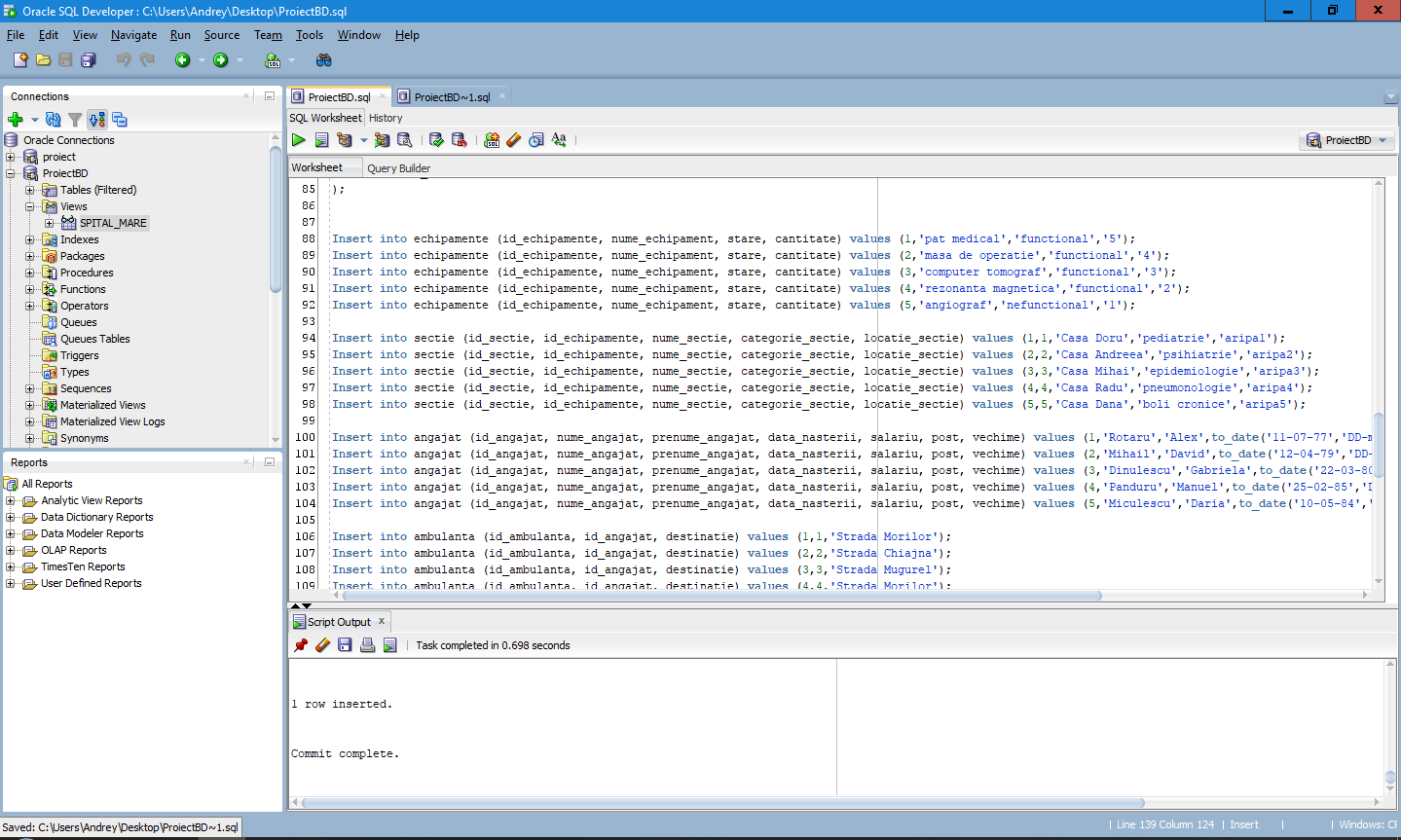
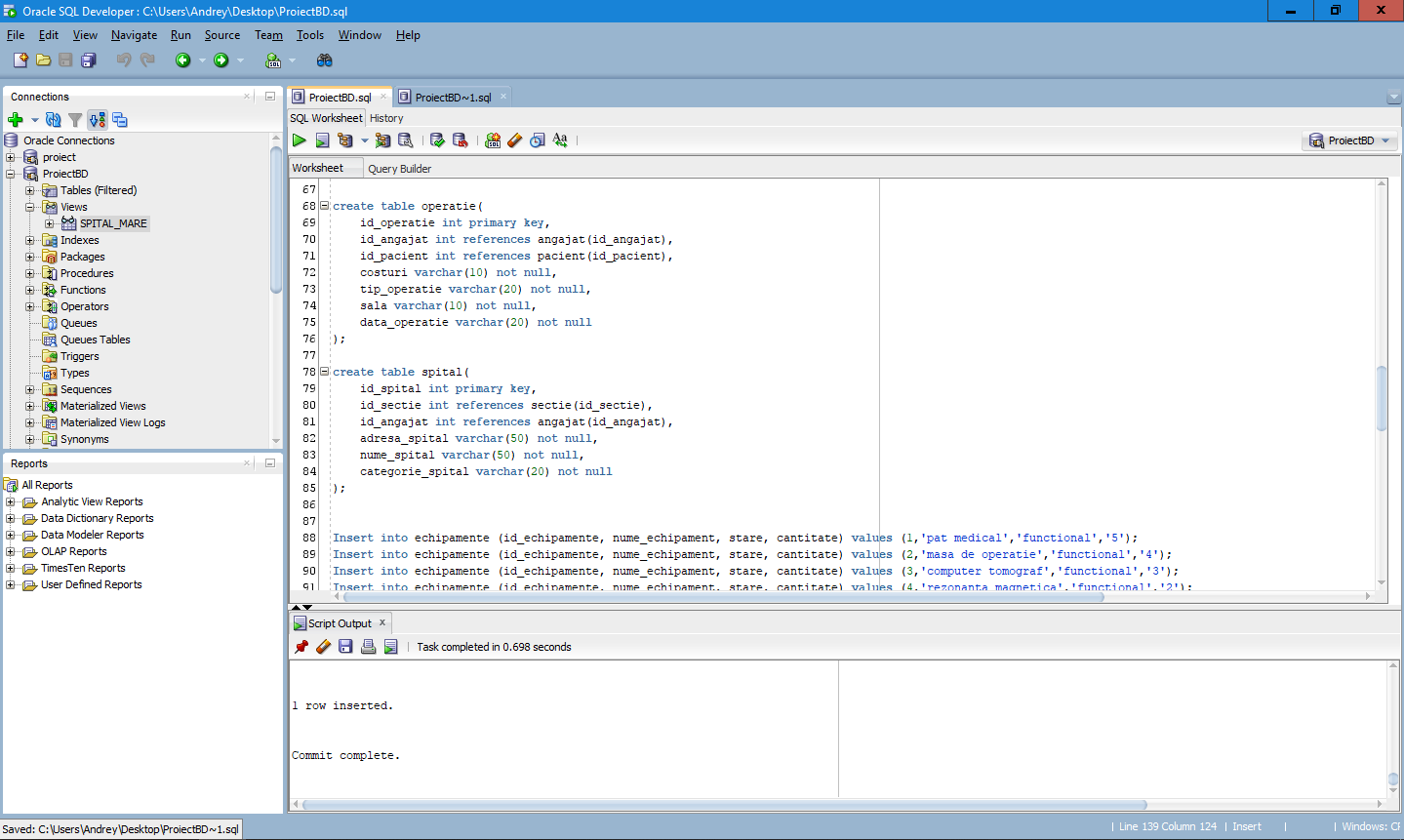
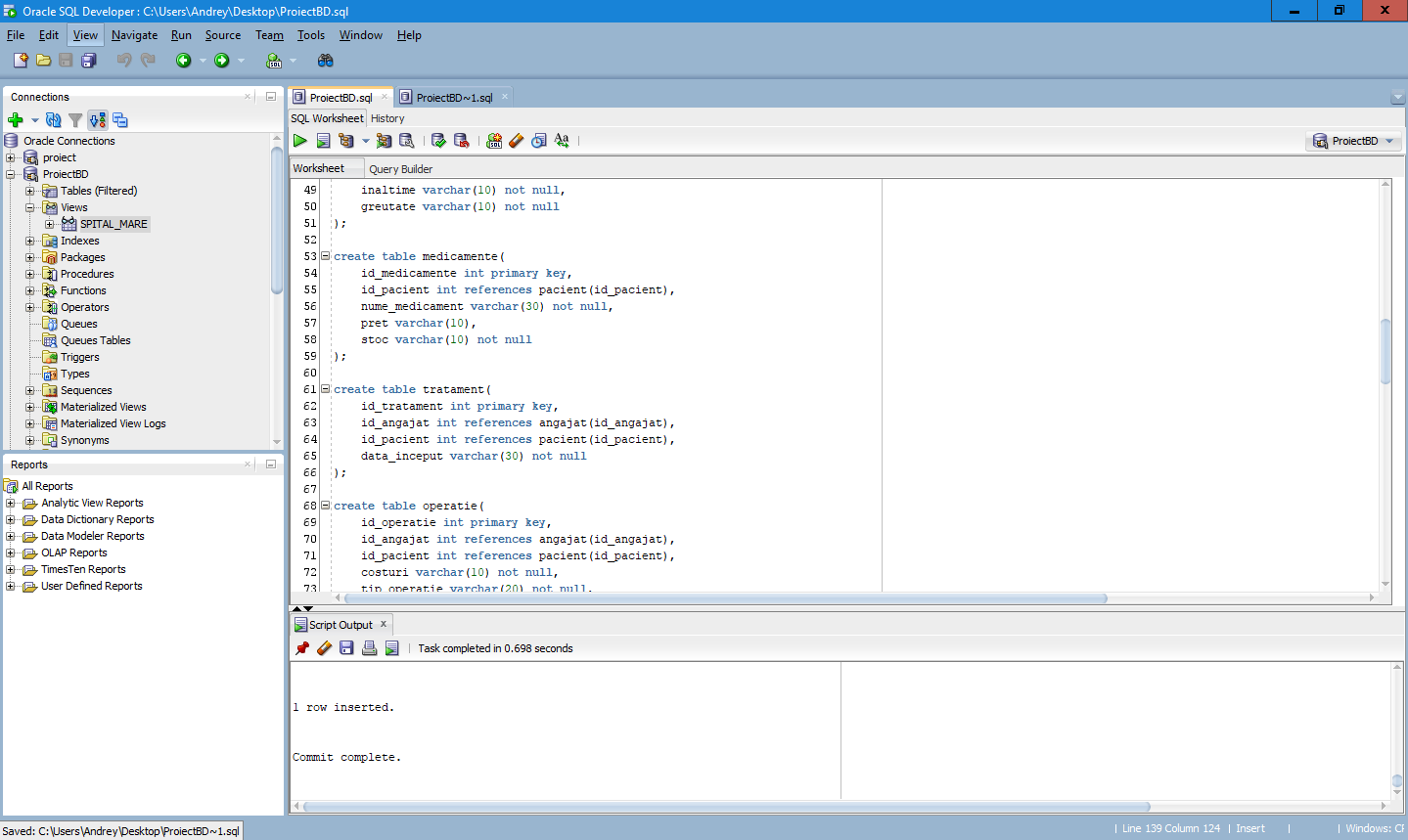
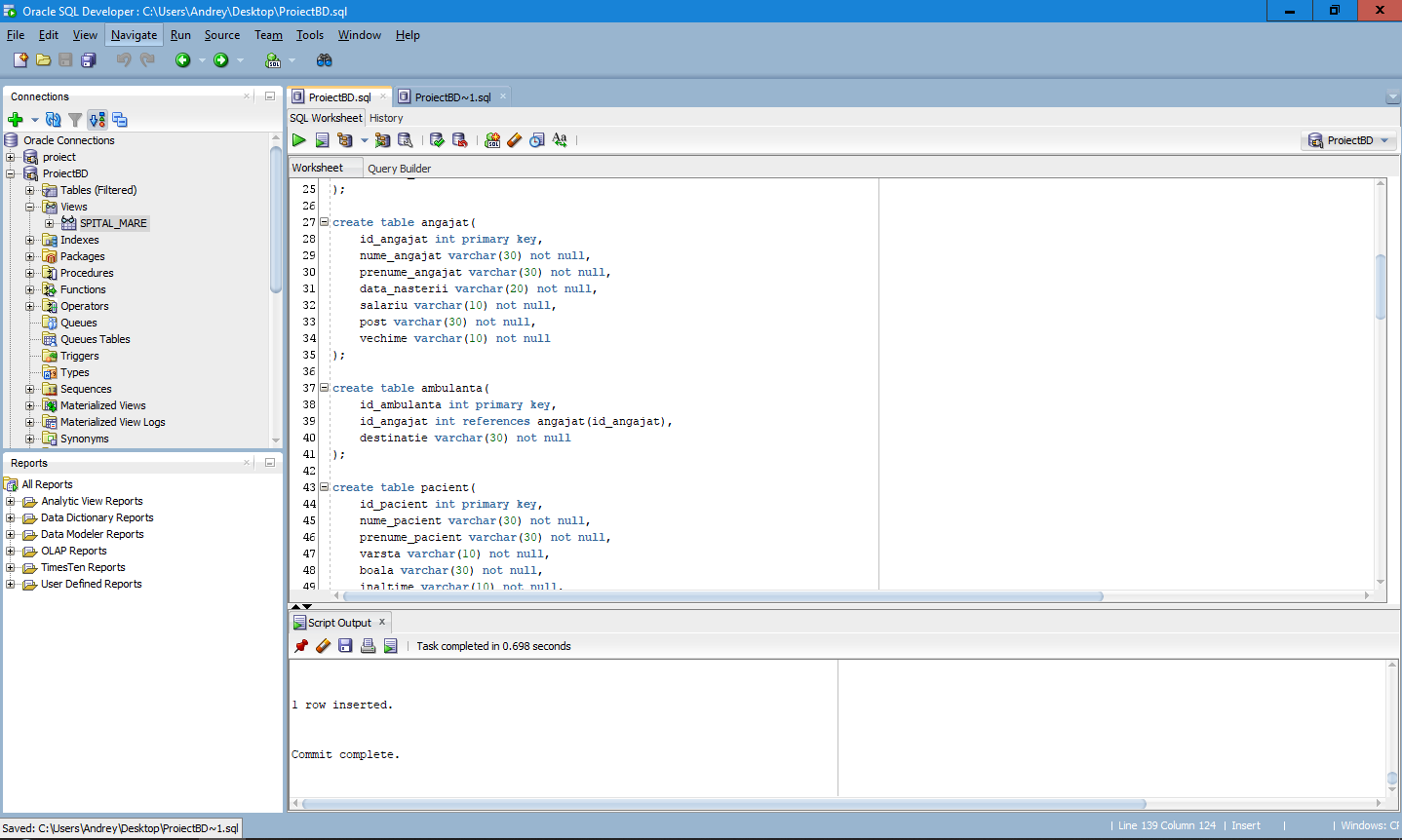
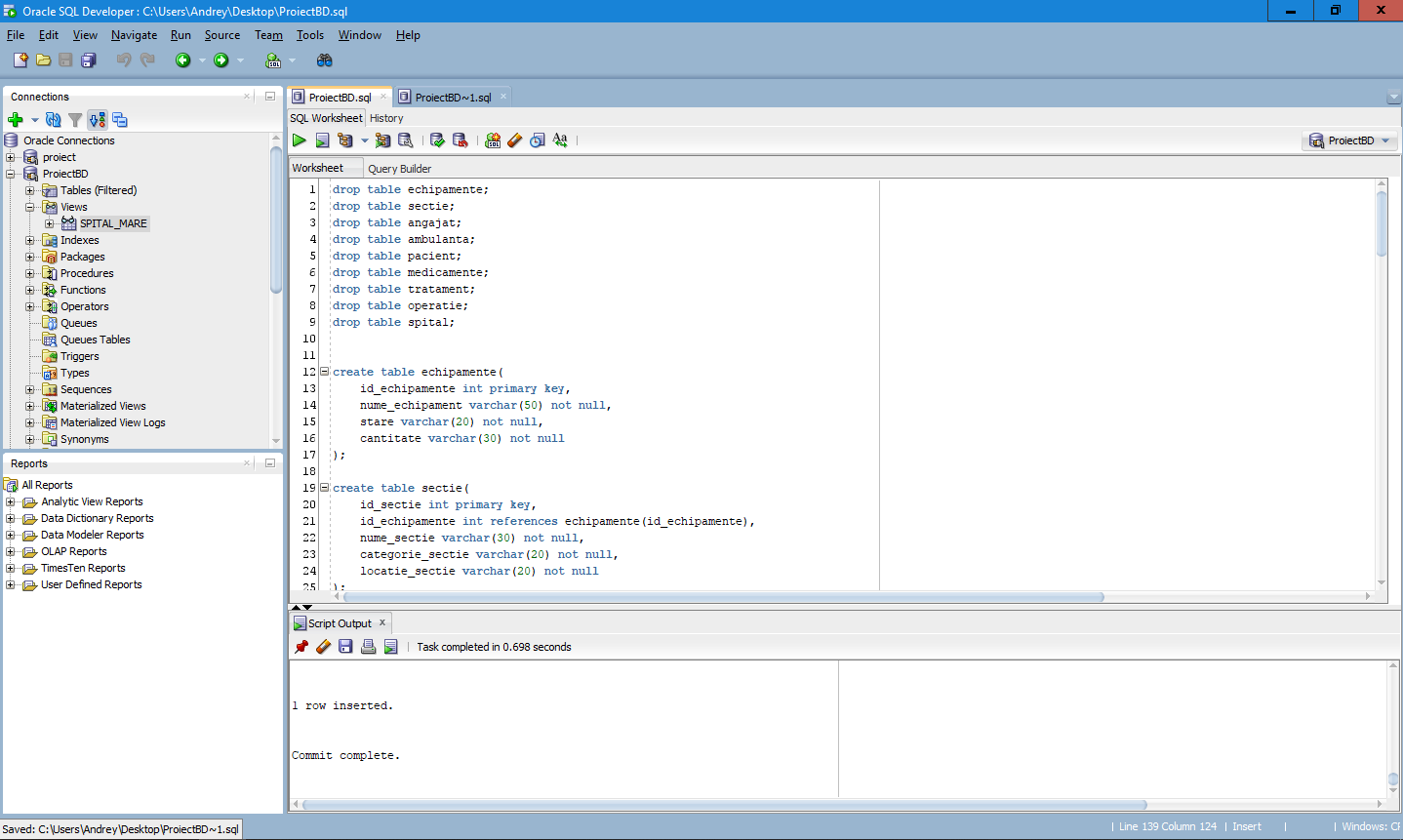
Asistent\_medical(id\_asistent\_medical, nume\_asistent\_medical, prenume\_asistent\_medical, data\_nasterii, salariu, post, vechime)

Non-FN3

Angajat(id\_angajat, nume\_angajat, prenume\_angajat, id\_pacient, nume\_pacient, prenume\_pacient)

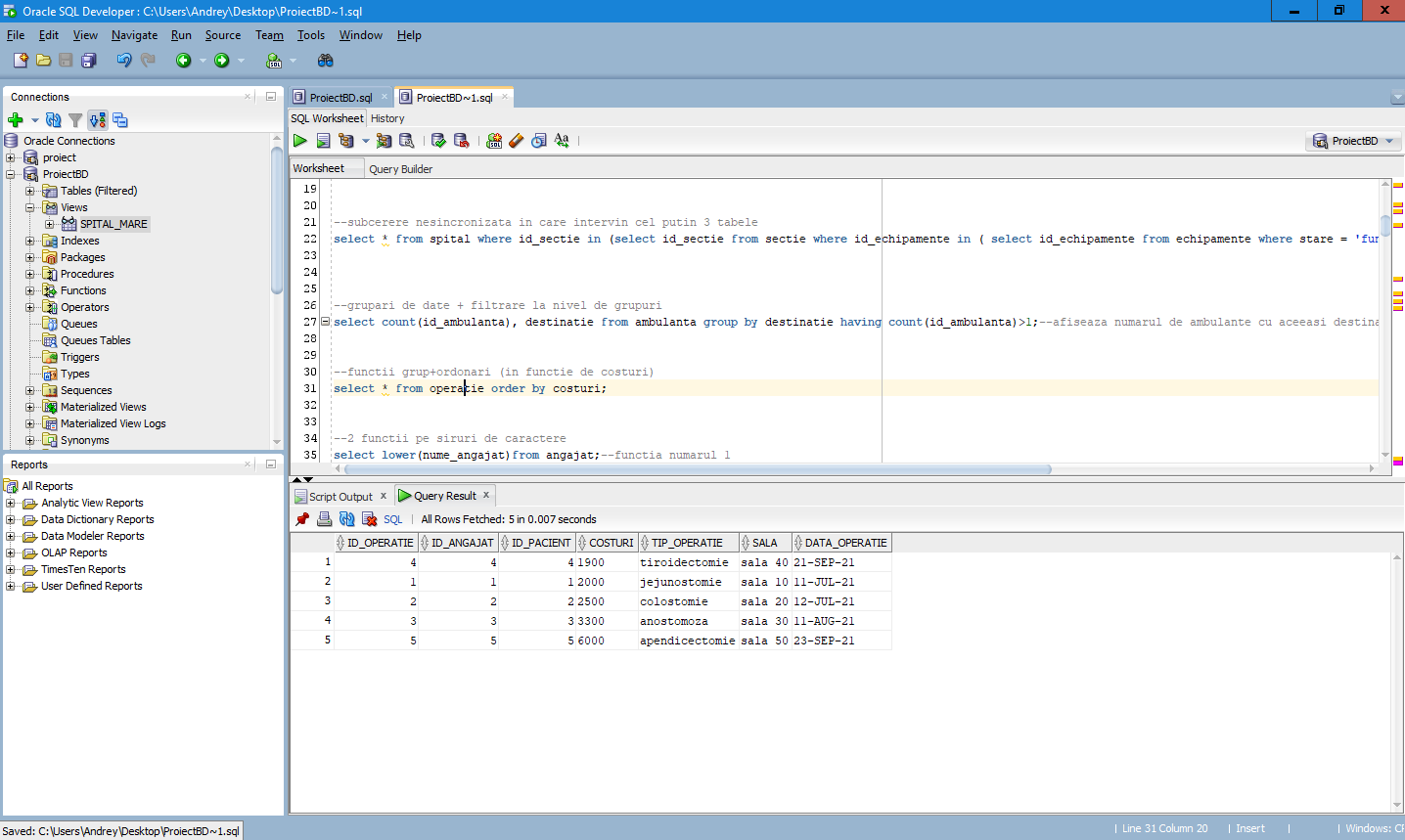
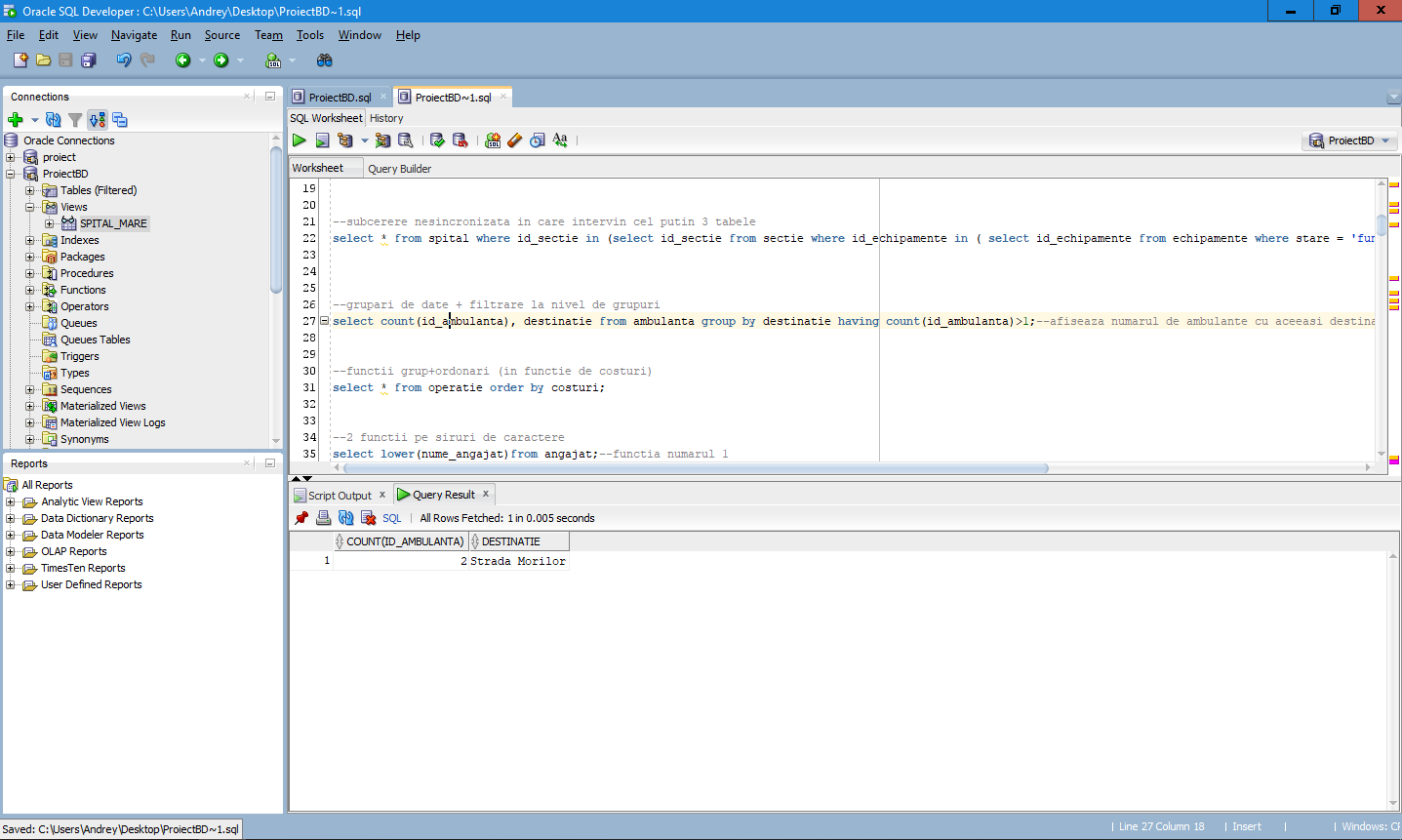
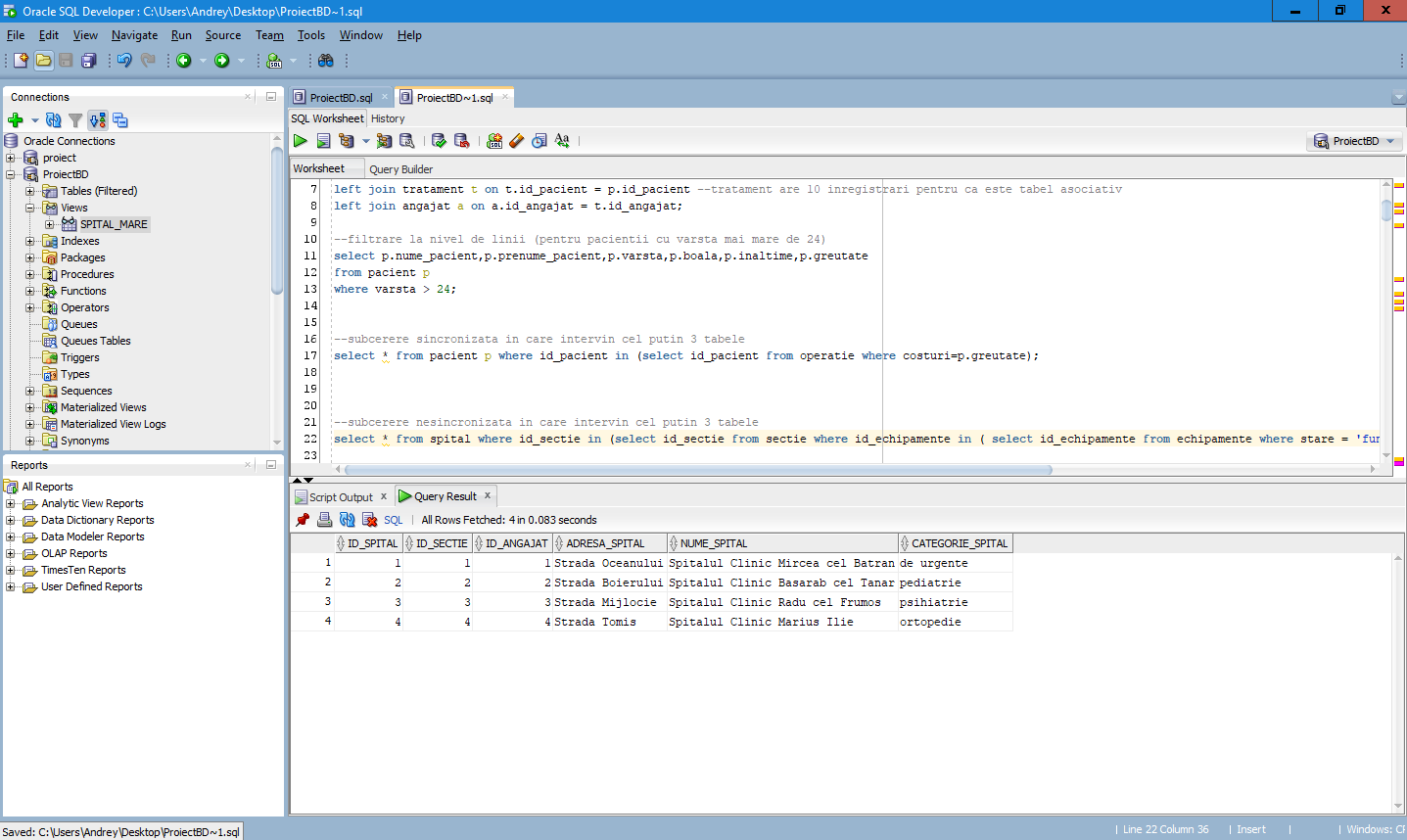
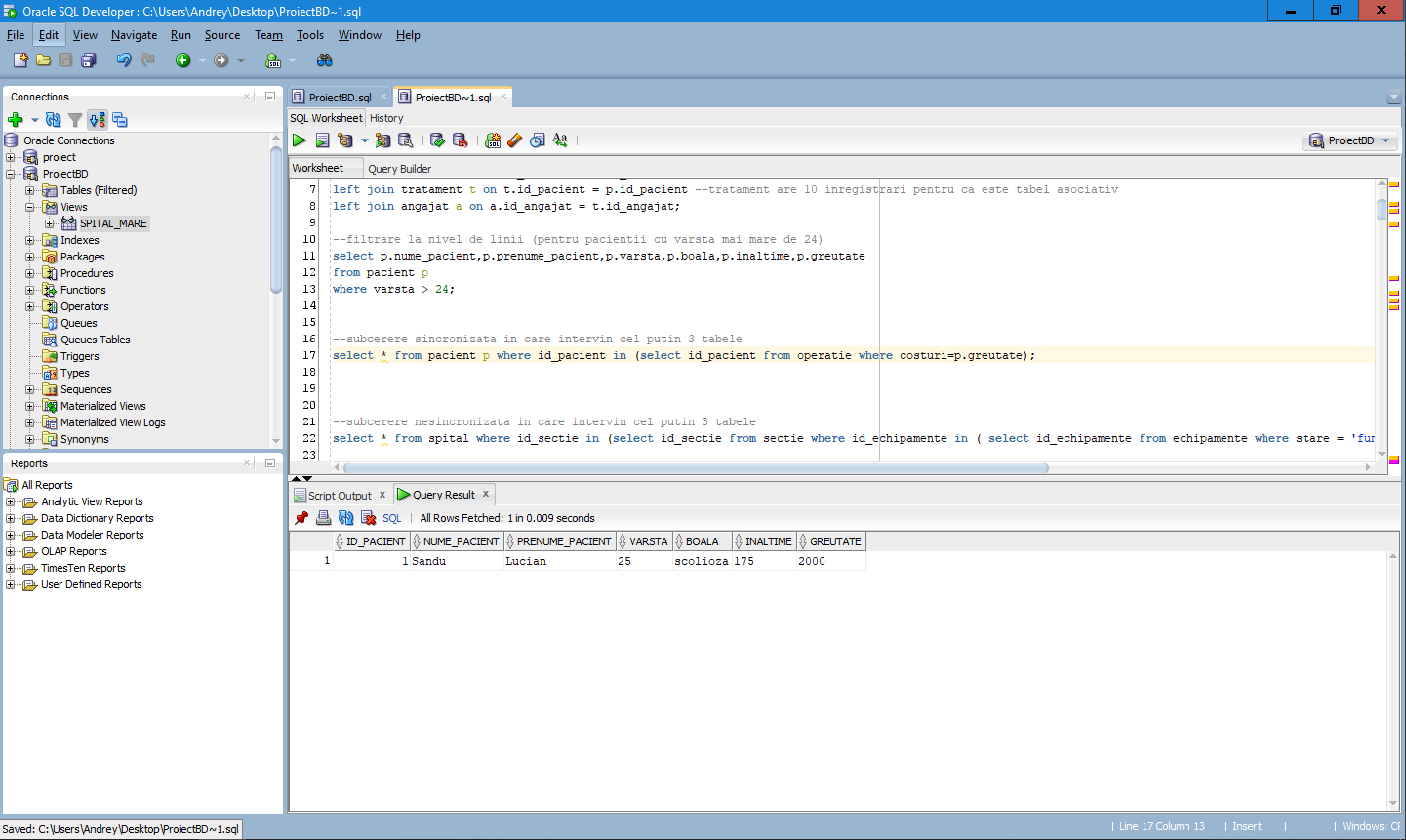
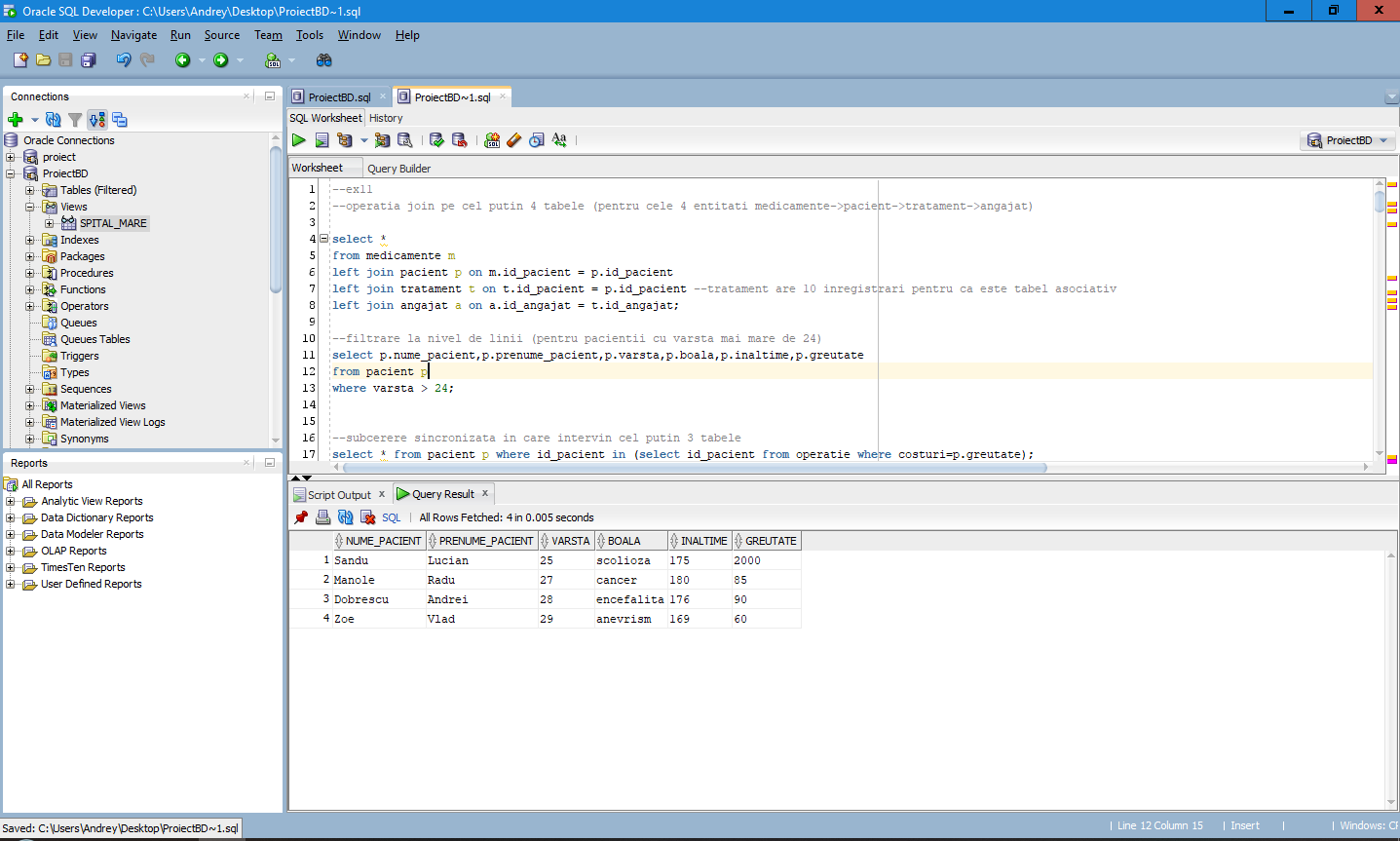
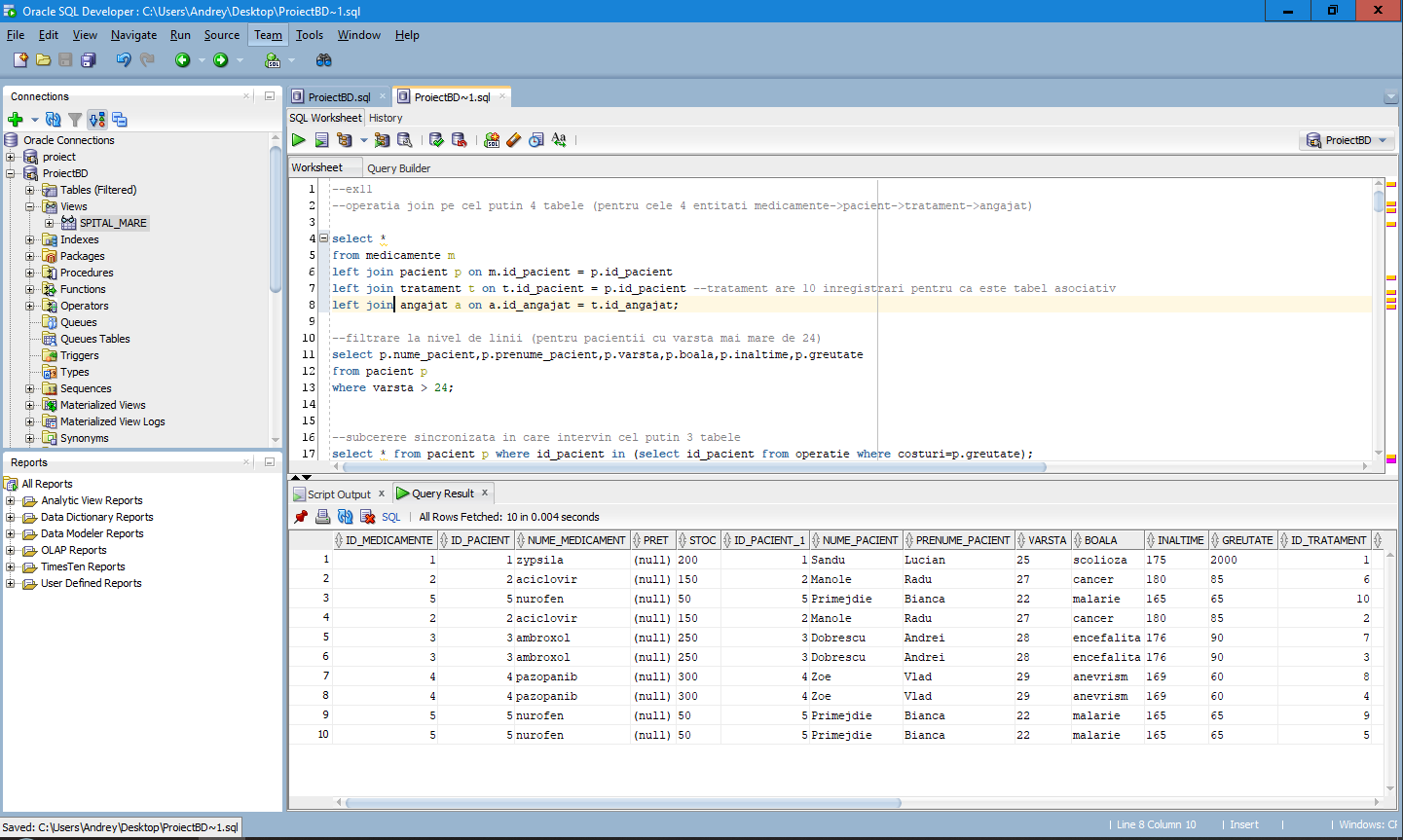
FN3(nume\_angajat, prenume\_angajat, nume\_pacient, prenume\_pacient)

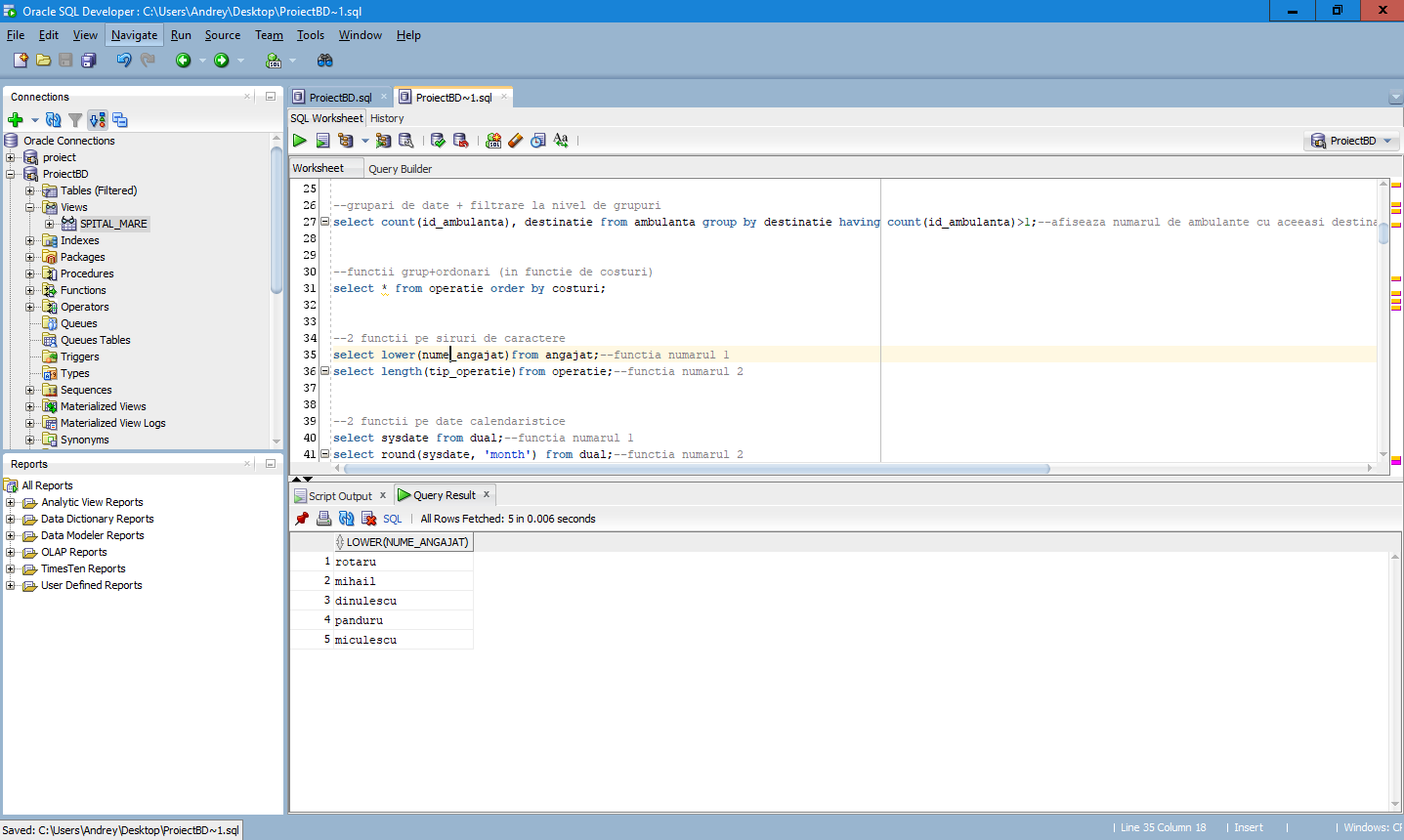
**10) Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea.**

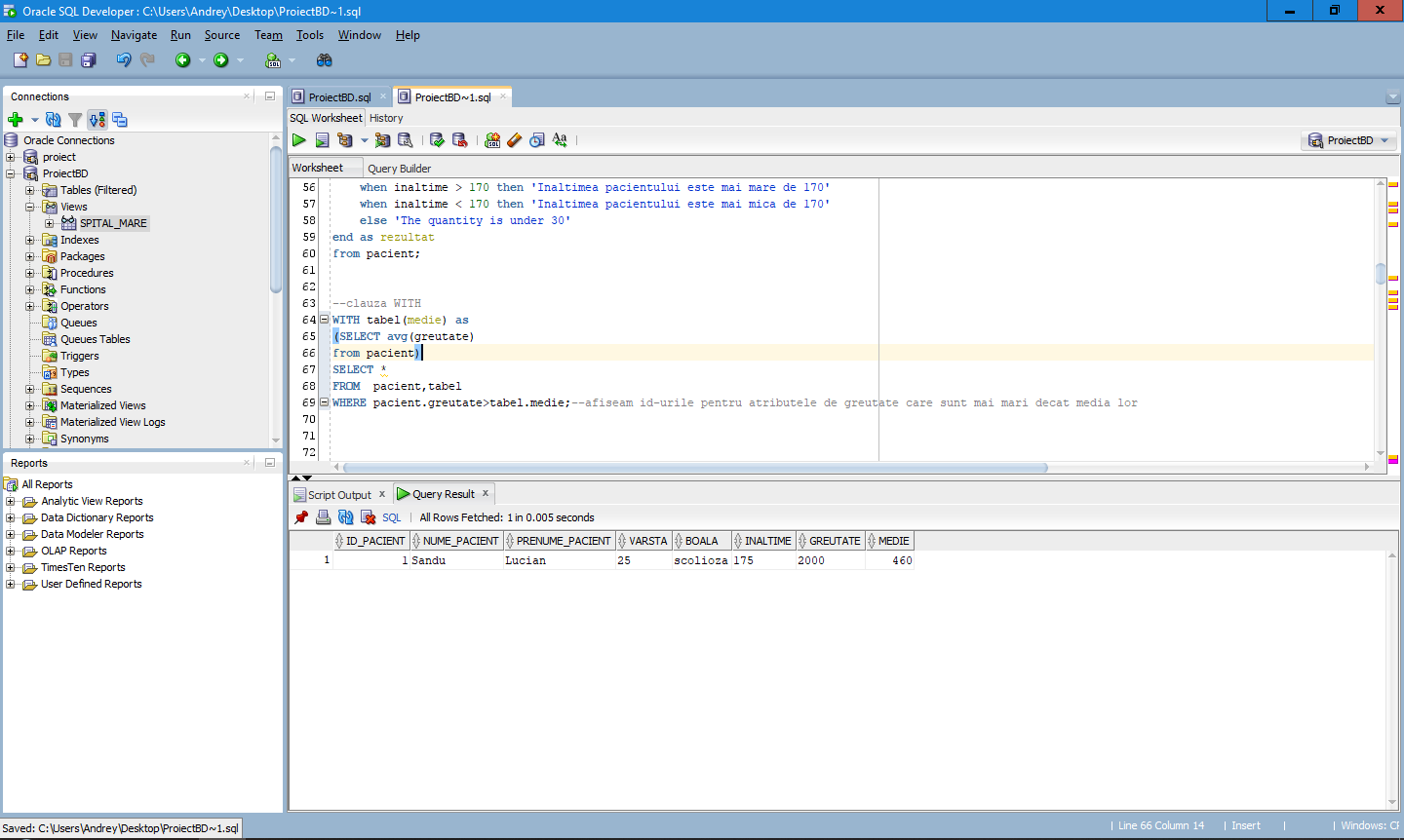
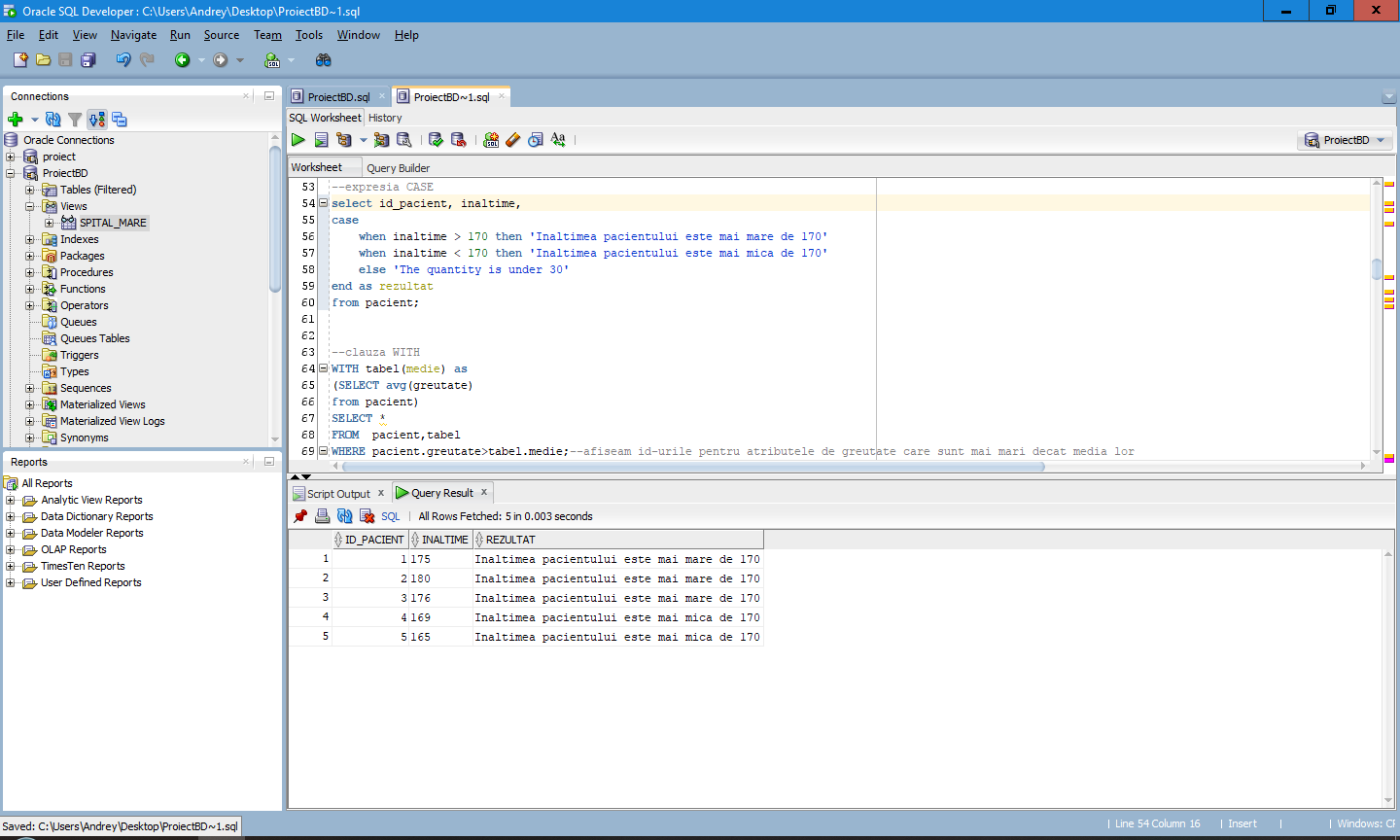
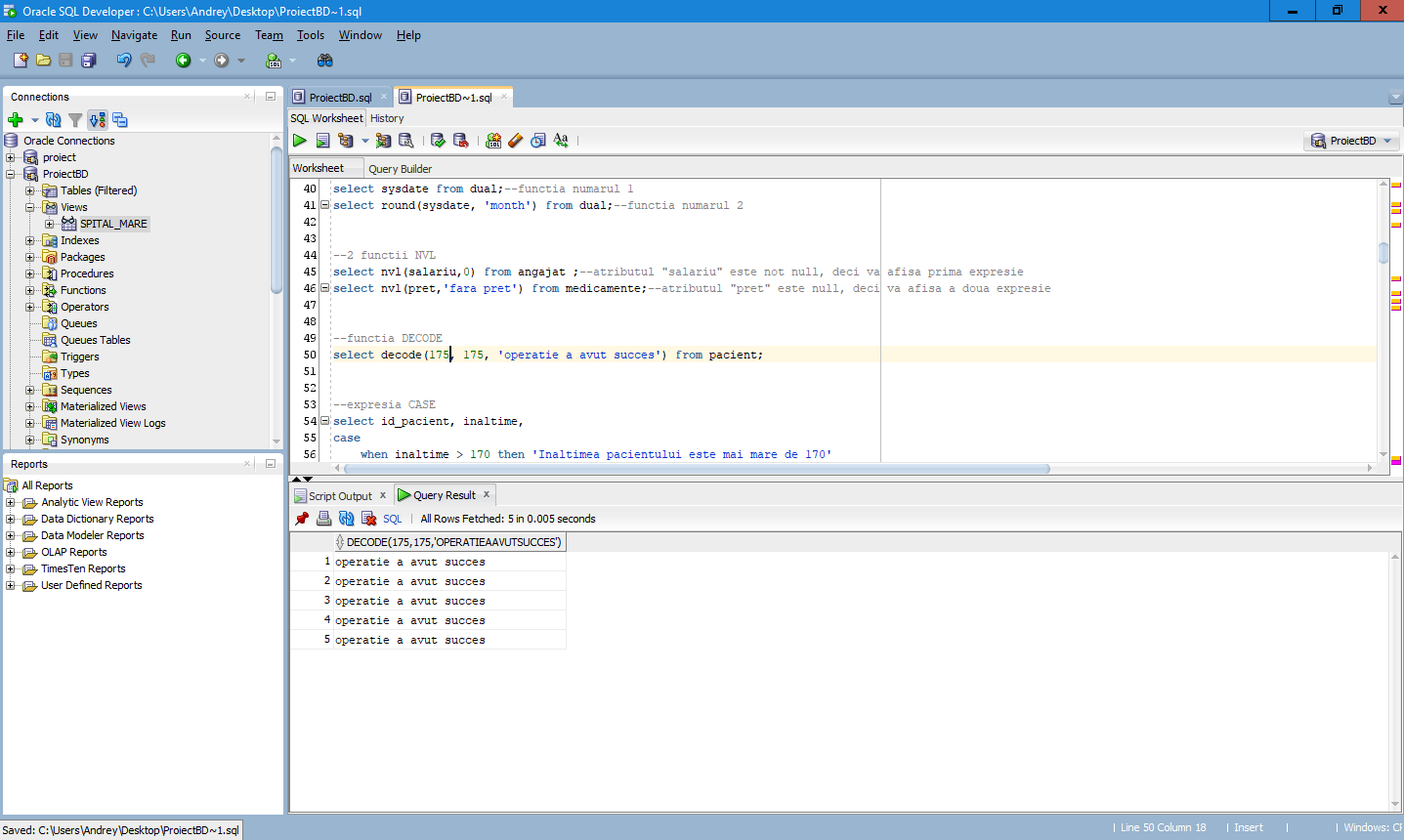
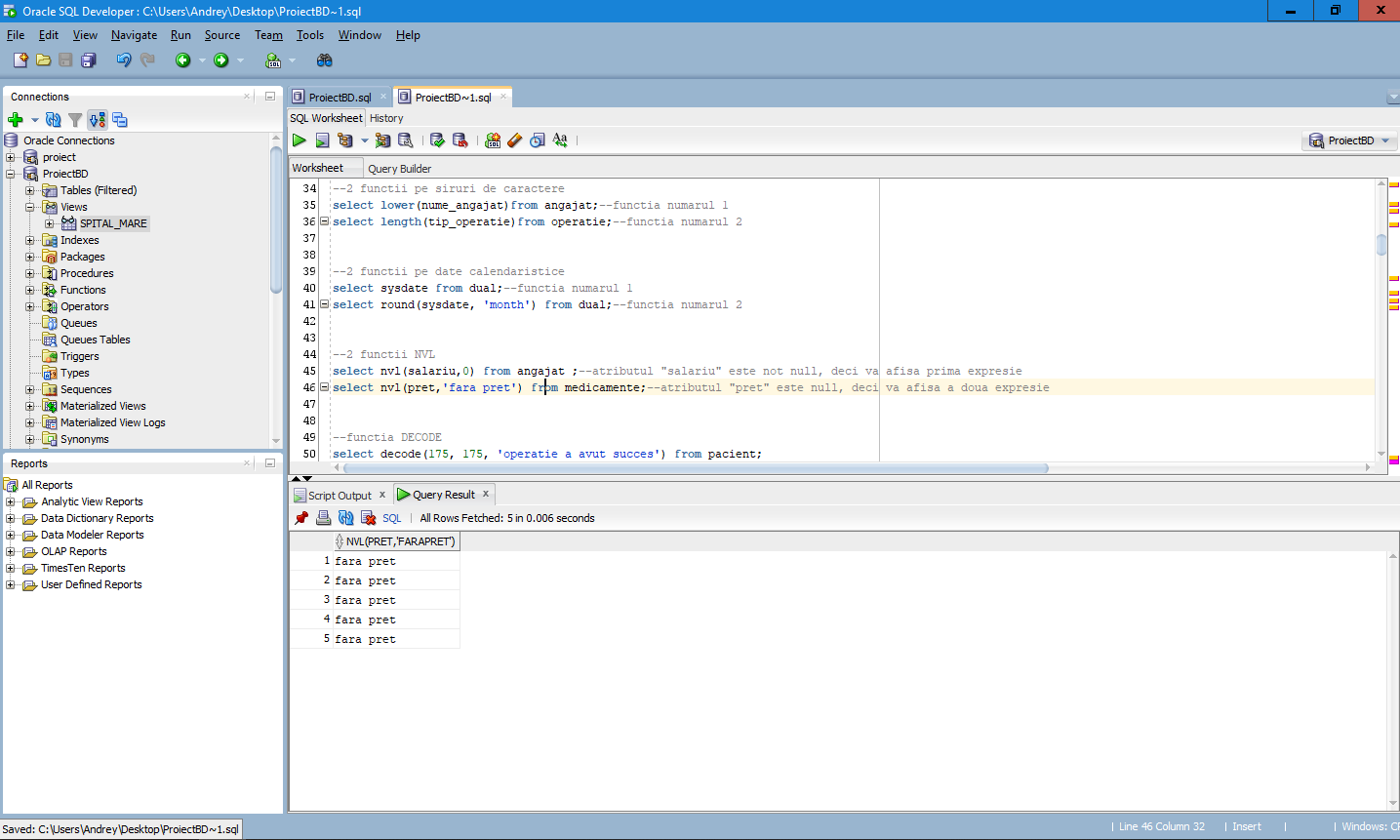
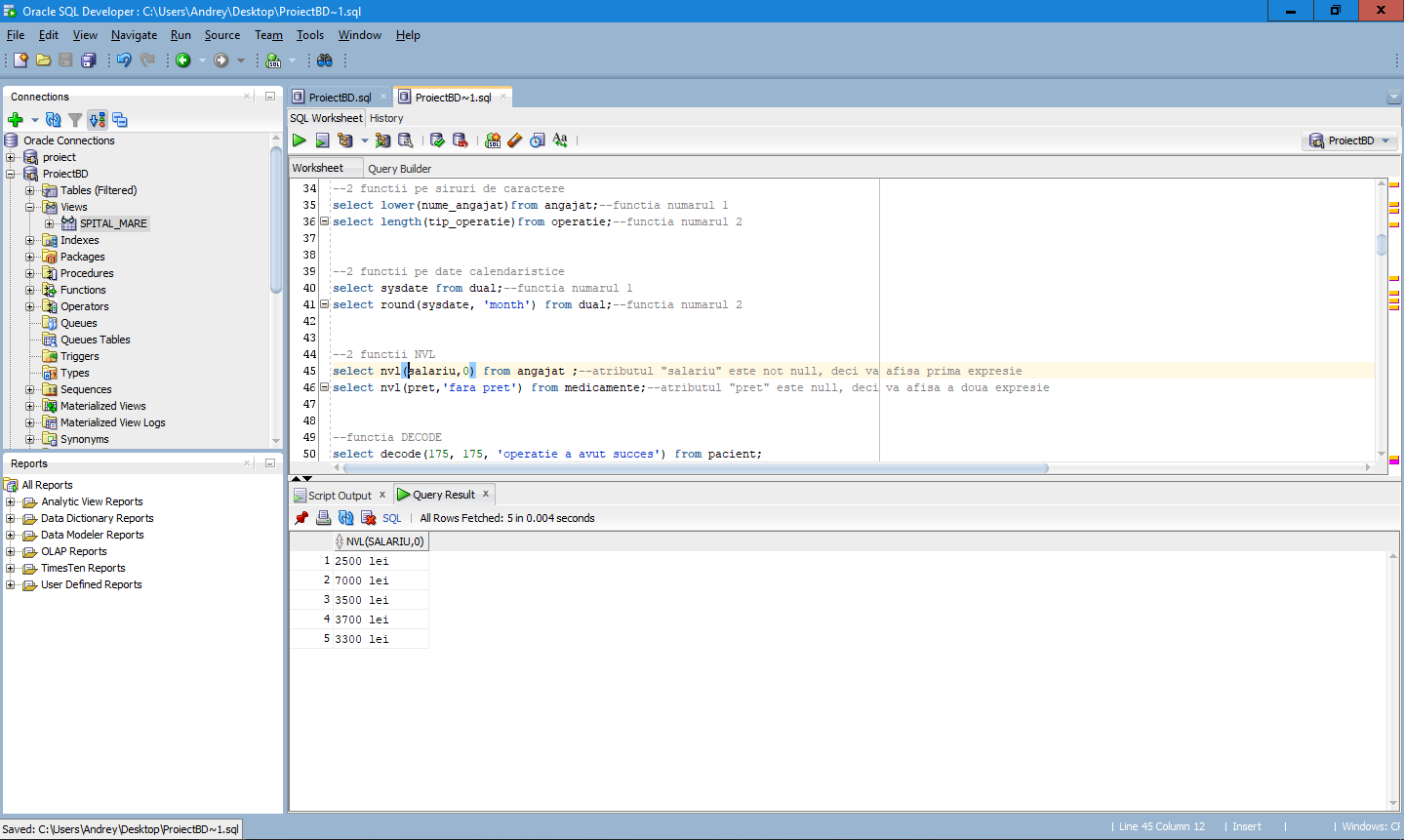
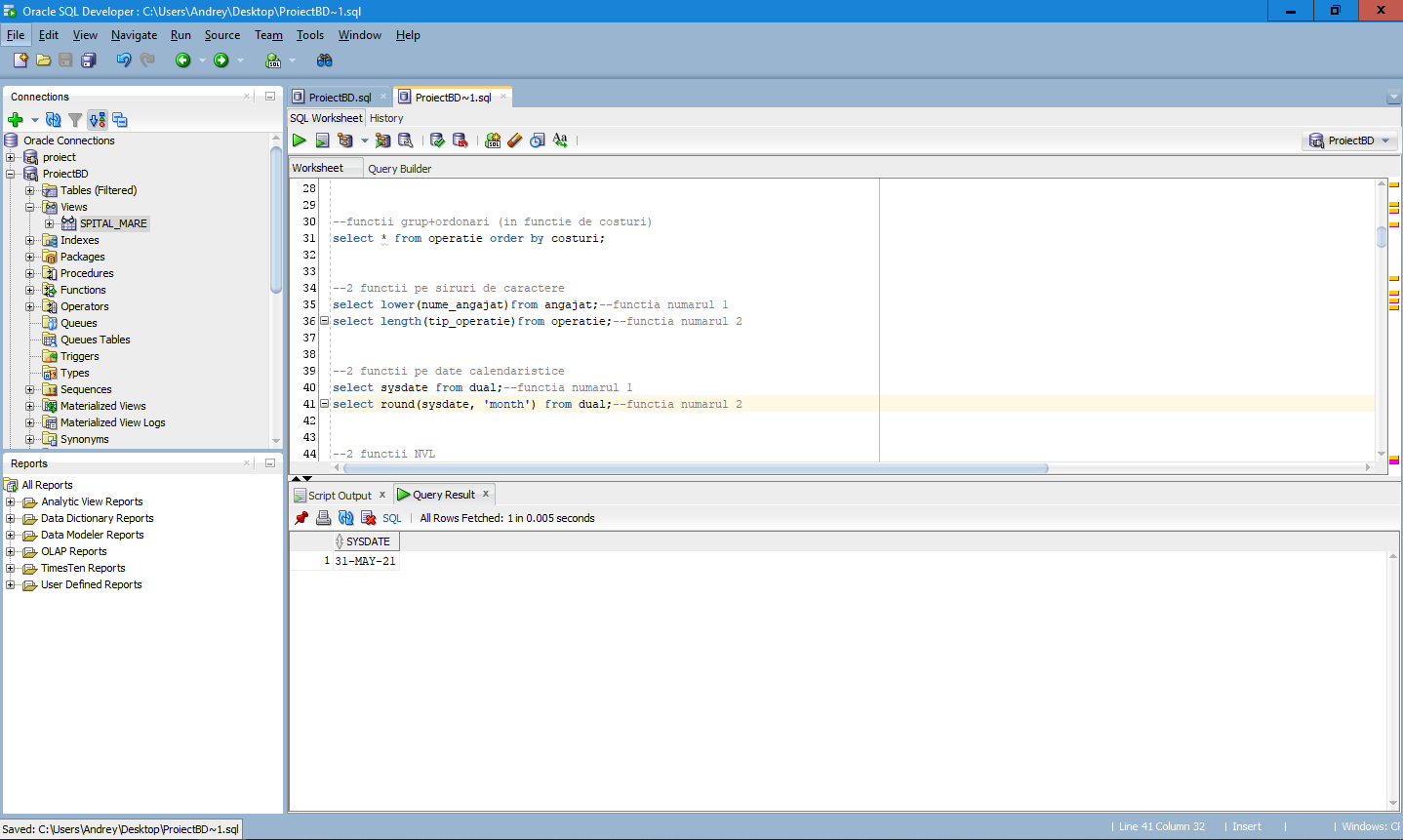
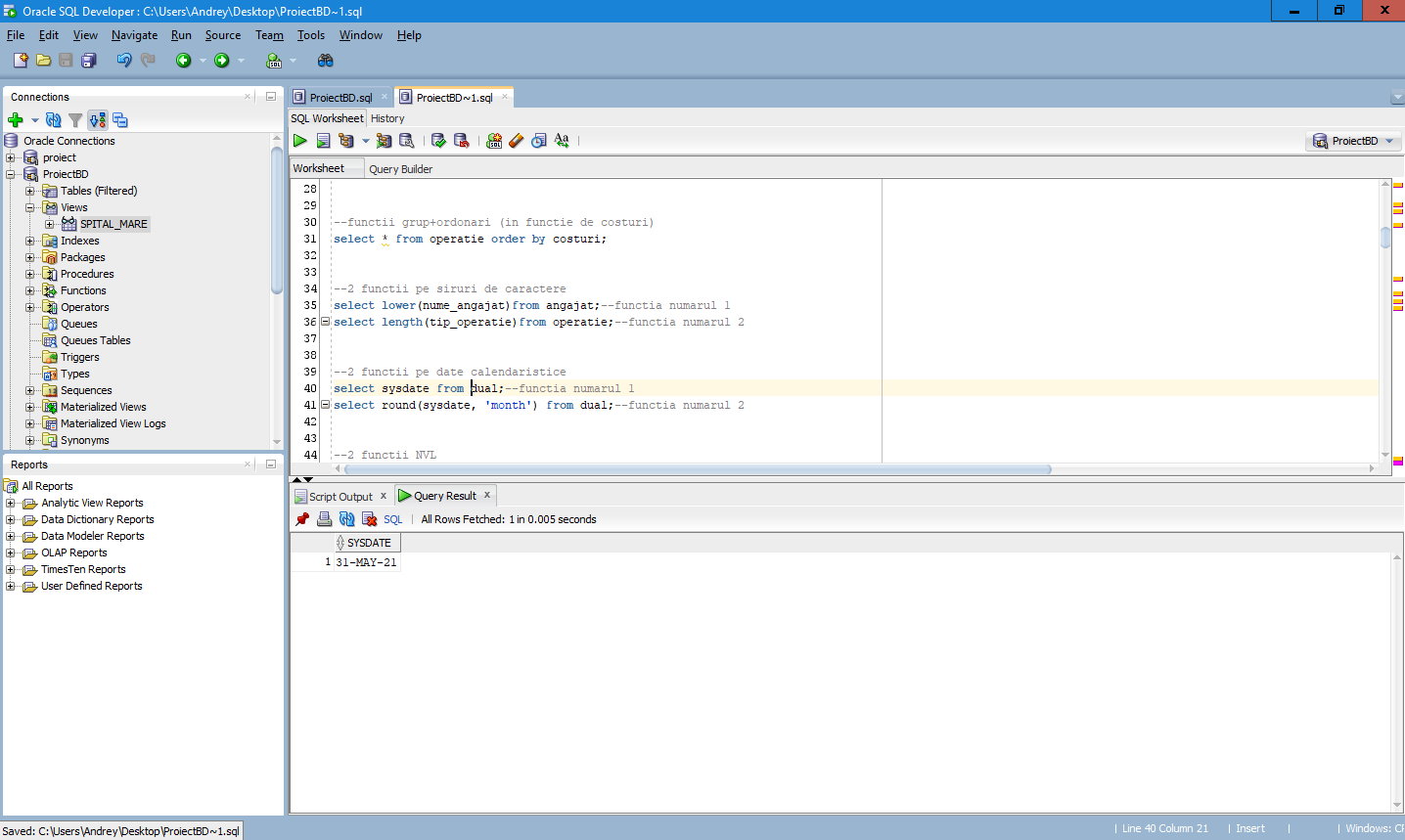
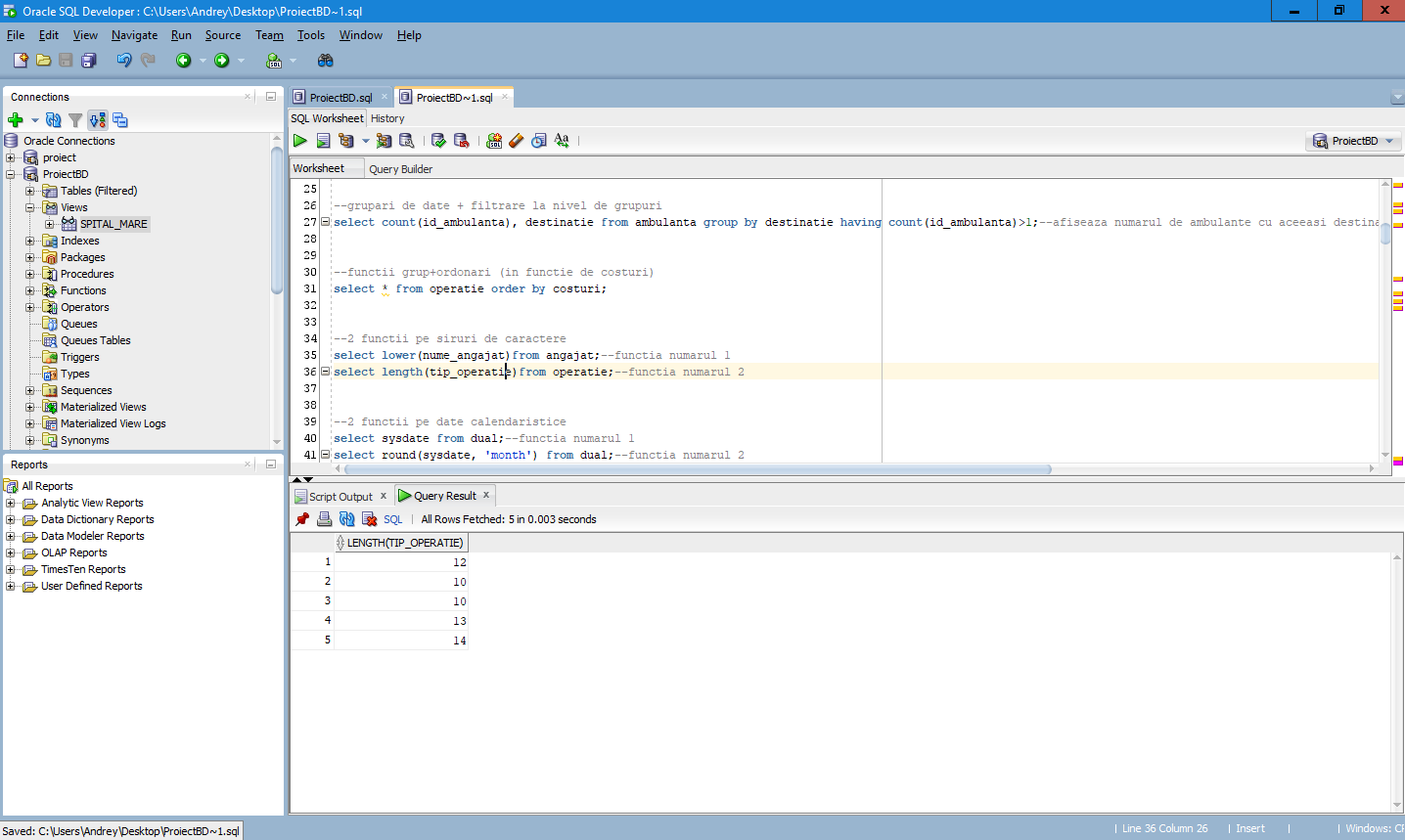


**11) Formulați în limbaj natural și implementați 5 cereri SQL complexe ce vor utiliza, în ansamblul lor, următoarele elemente:**

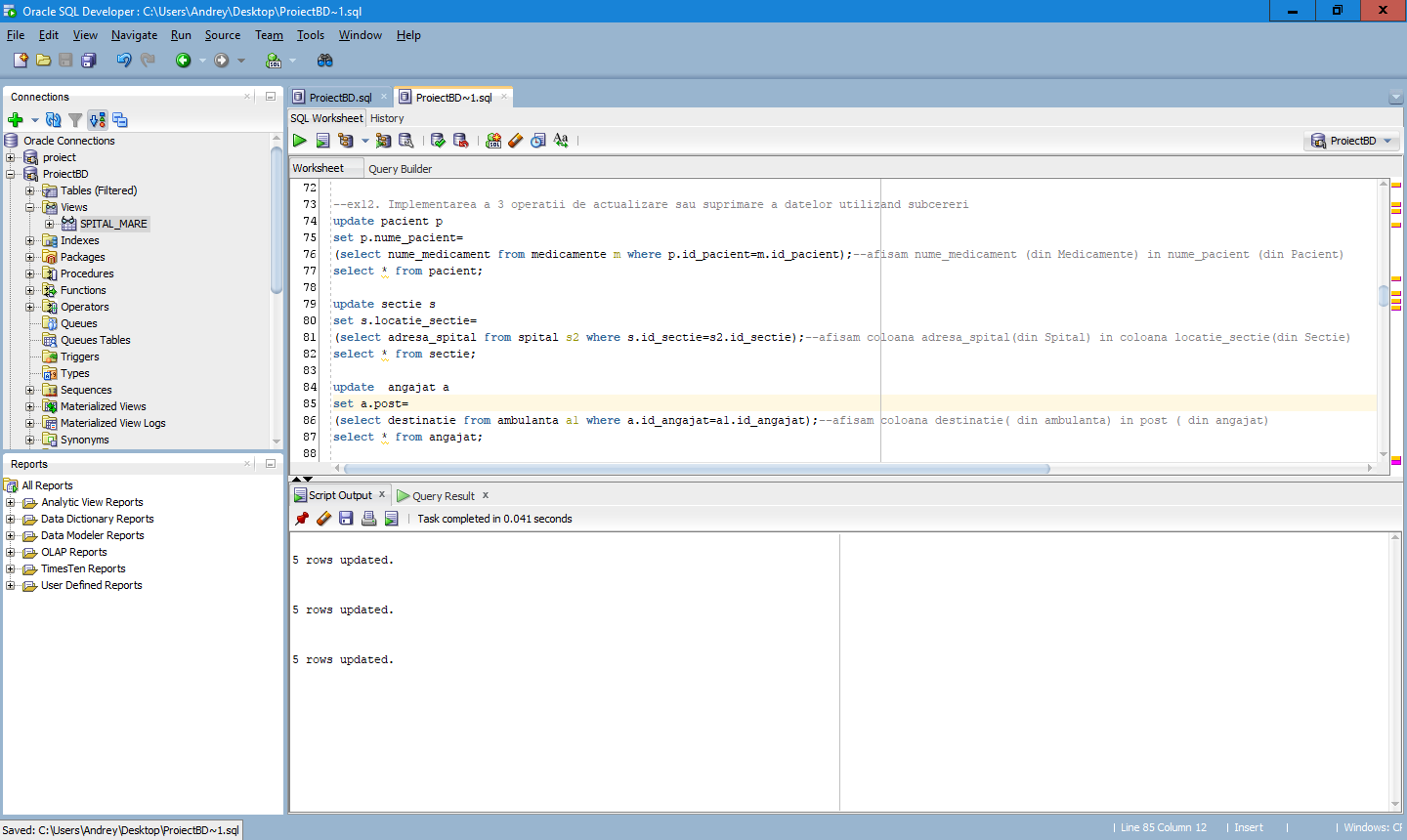
* operație join pe cel puțin 4 t abele
* filtrare la nivel de linii
* subcereri sincronizate în care intervin cel puțin 3 tabele
* subcereri nesincronizate în care intervin cel puțin 3 tabele
* grupări de date, funcții de grup, filtrare la nivel de grupuri
* ordonări
* utilizarea a cel puțin 2 funcții pe șiruri de caractere, 2 funcții pe date calendaristice, a funcțiilor NVL și DECODE, a cel puțin unei expresii CASE
* utilizarea a cel puțin 1 bloc de cerere (clauza WITH)



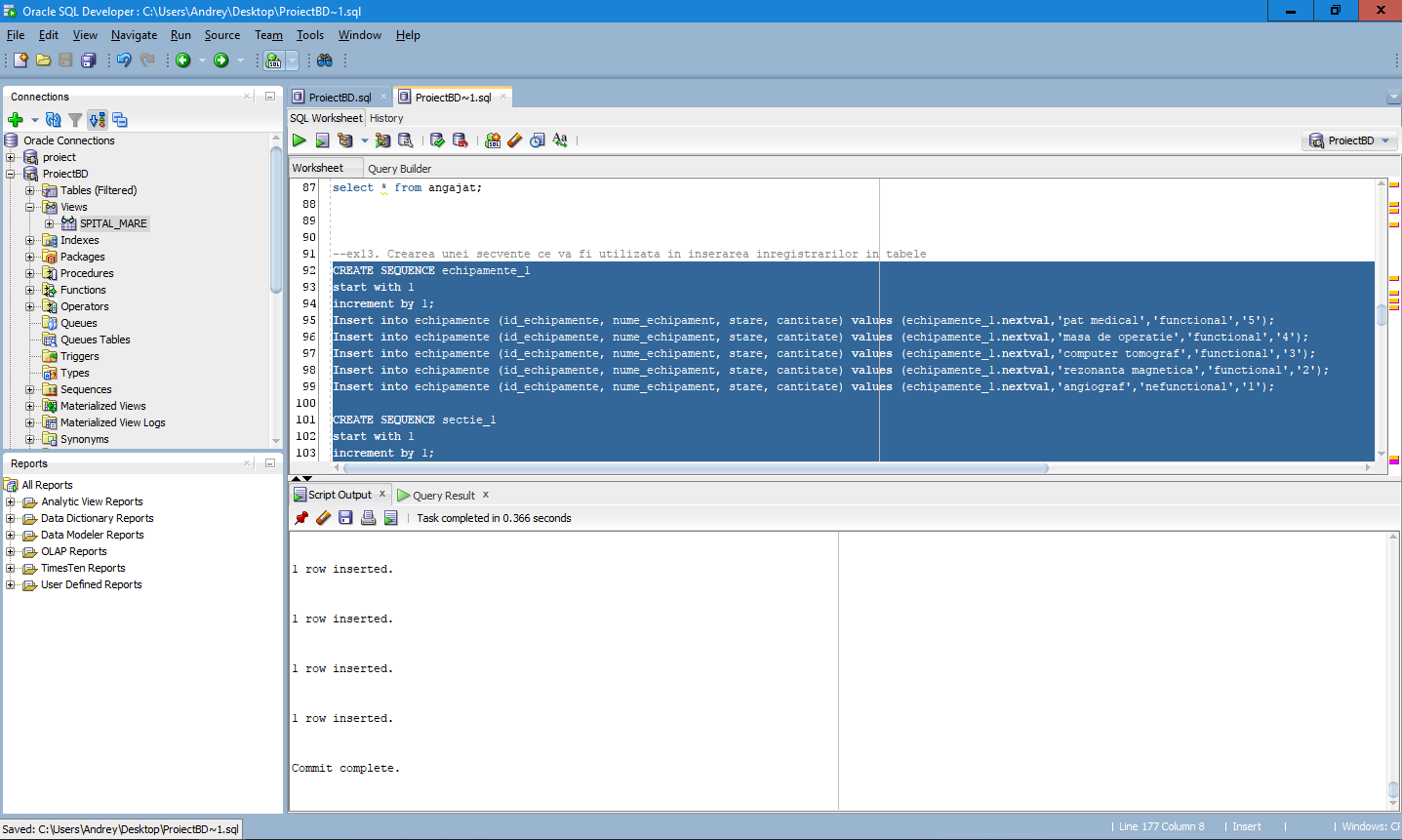
****



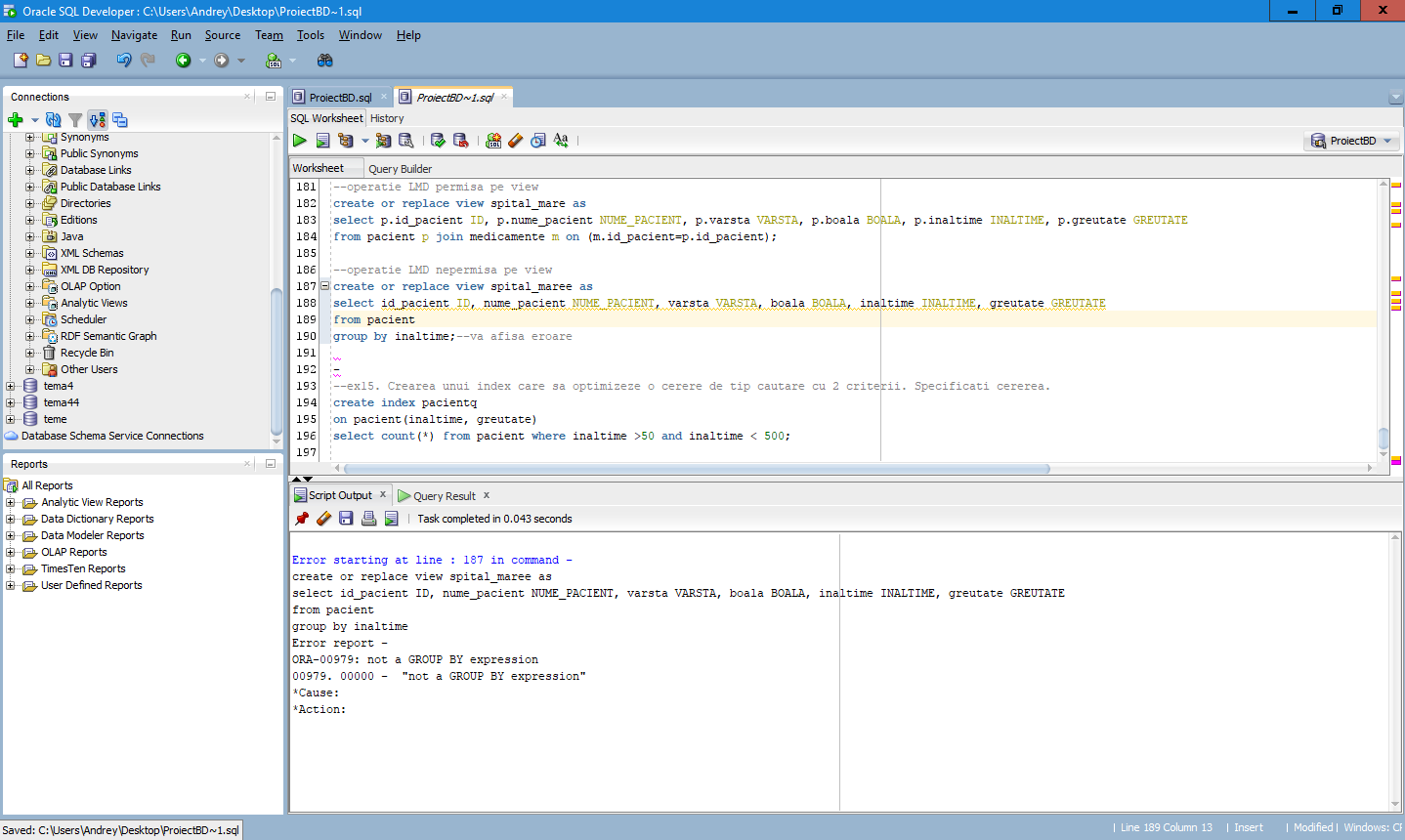
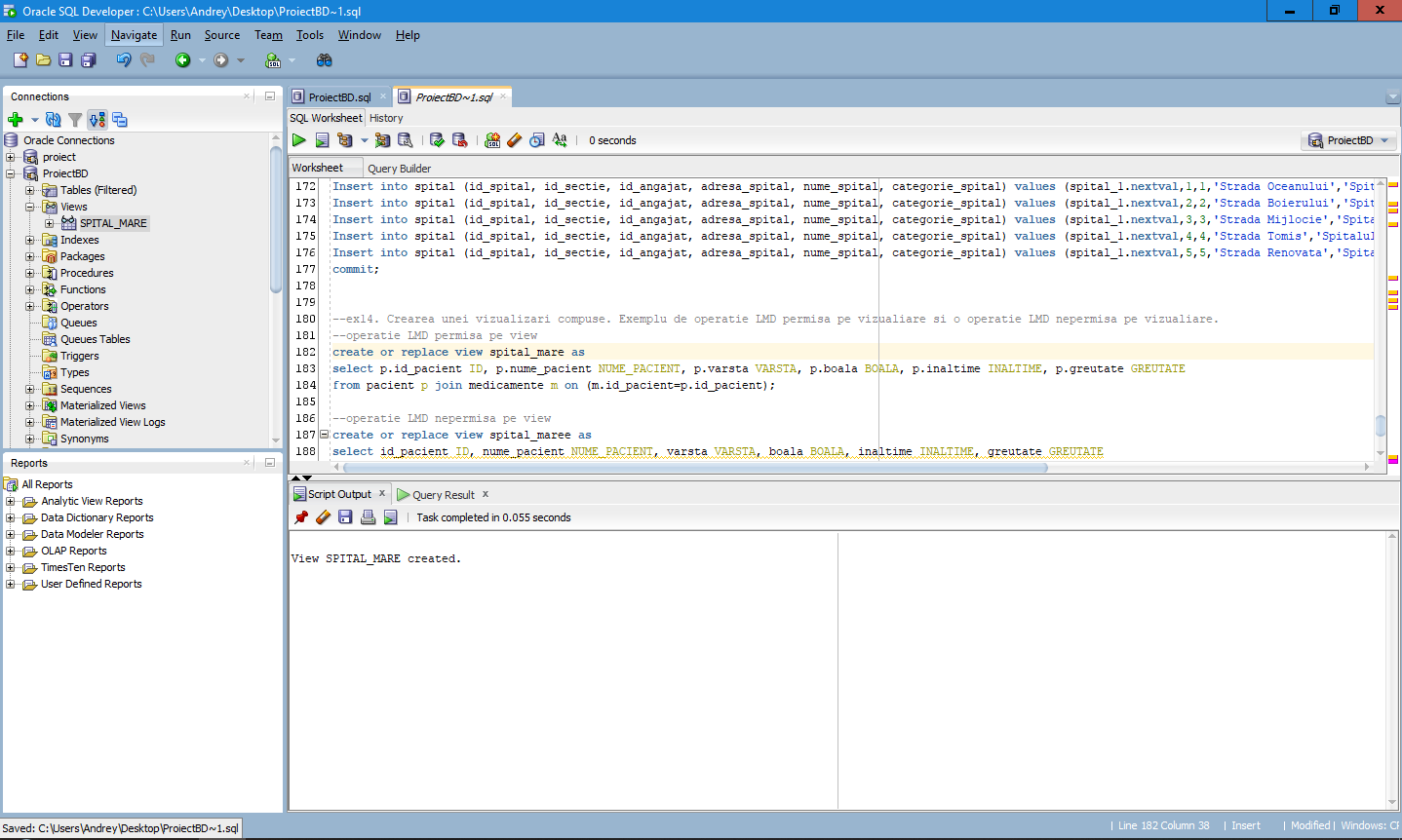
**12) Implementarea a 3 operații de actualizare sau suprimare a datelor utilizând subcereri.**



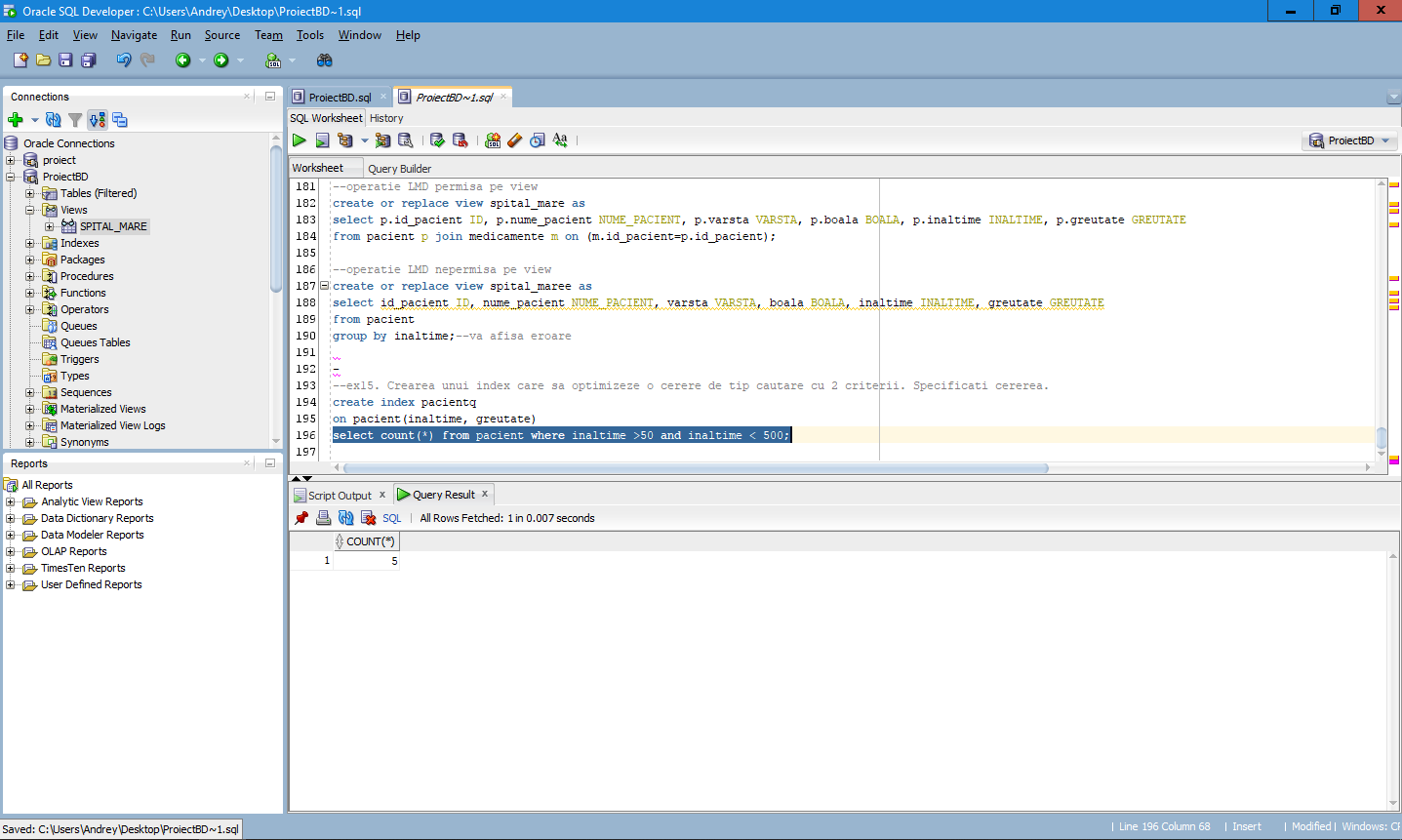
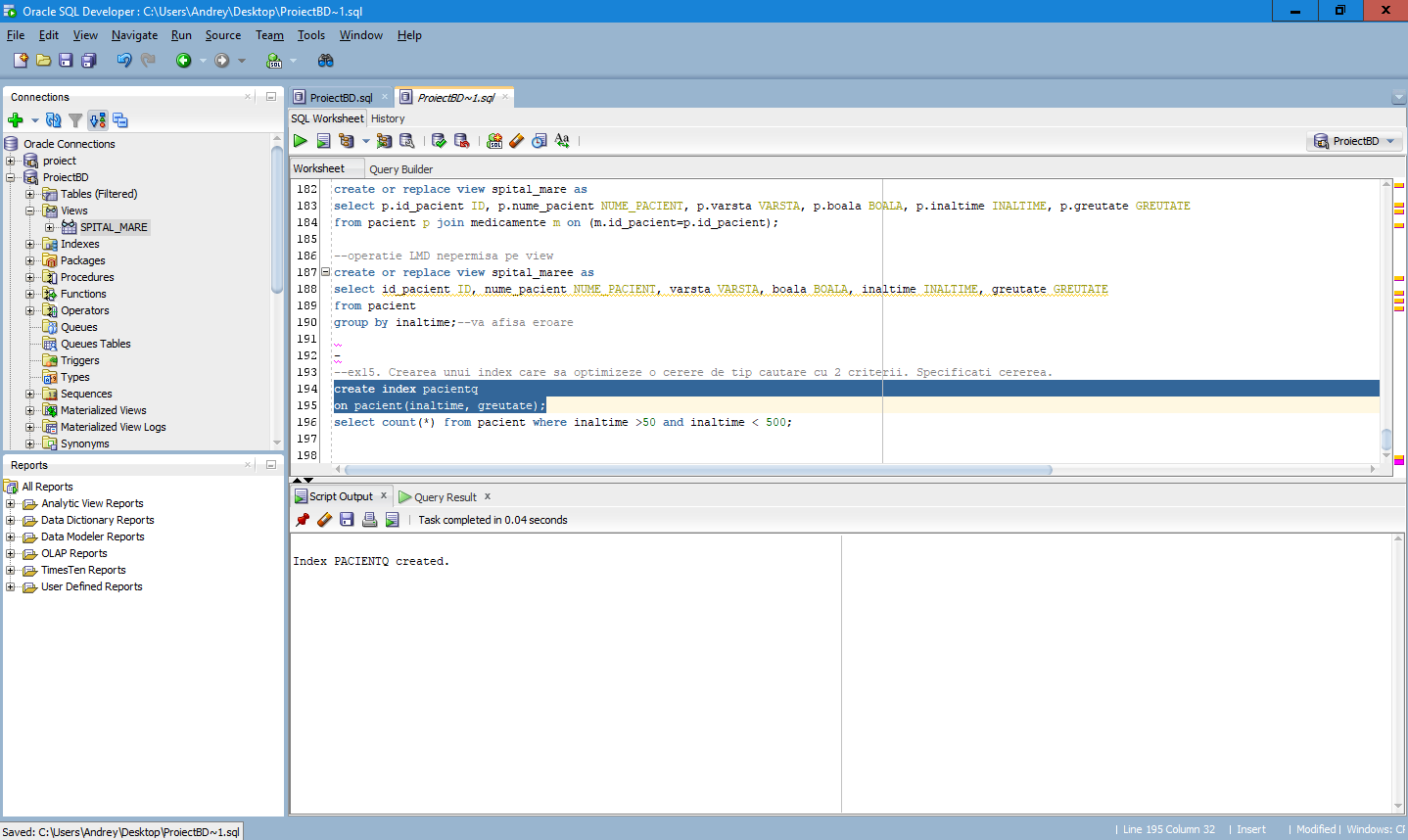
**13) Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele.**



**14) Crearea unei vizualizări compuse. Dați un exemplu de operație LMD permisă pe vizualizarea respectivă și un exemplu de operație LMD permisă.**



**15) Crearea unui index care să optimizeze o cerere de tip căutare cu 2 criterii. Specificați cererea.**



**16) Formulați în limbaj natural și implementați în SQL: o cerere ce utilizează operația outer-join pe minimum 4 tabele și două cereri ce utilizează operația division(minus).**

