Proiect BAZE DE DATE – Neacsu Maria

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

Modelul de date ales gestioneaza informatii si modalitatea de functionare a unui hotel cu servicii de servire masa si organizare de spectacole. Acesta prezinta functionalitati de gestiune a curateniei in camere si buna organizare a angajatilor, modalitate de rezervare a camerelor, modalitate de plata, servire conform standardelor. De asemenea la rezervarea camerelor se stocheaza informatiile personale ale clientului, cat si datele cardului acestuia. Clientii hotelului pot beneficia de anumite spectacole organizate la hotel de catre anumite firme de organizare exterioare.

Clientii fac rezervari, acestea sunt preluate de receptionerii hotelului. Dupa realizarea rezervarii se face plata cu cardul personal de catre client, iar receptionerii valideaza primirea banilor.

Clientii efectueaza diferite comenzi de mancare, care sunt preluate de catre chelneri. Chelnerii predau mai departe comenzile bucatarilor. In momentul in care comanda este gata, aceasta este servita clientului de catre chelner.

La plecarea clientilor, dupa eliberarea camerelor se organizeaza un orar pentru curatenia in camera. Menajerele isi fixeaza o data de inceput si o data de terminare a curateniei in camera.

Hotelul pune la dispozitie clientilor diferite spectacole care sunt organizate de niste firme specializate pe organizarea evenimentelor.

De asemenea, modelul de date ales analizeaza informatiile legate de localizarea hotelului, cat si a angajatilor respectiv clientilor acestuia pentru asigurarea securitatii tuturor.

2*. Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului.*

Modelul de date ales respecta urmatoarele constrangeri:

* Angajatii hotelului, anume receptionerii, menajerele, chelnerii, bucatarii sunt angajati permanent
* Clientii sunt obligati sa plateasca cu cardul
* Receptionerii valideazaa rezervarile facute de catre clienti
* Menajerele sunt obligate sa efectueze curatenia si igienizarea camerelor dupa plecarea clientilor
* Fiecare client este obligat sa isi introduca datele personale si datele cardului la rezervare
* Pentru localizarea angajatilor sau clientilor s-a considerat o singura adresa de baza, anume cea de la domiciliul acestora
* Chelnerii preiau comenzile de la client, iar apoi le transmit bucatarilor
* Un hotel are minim 4 angajati (un bucatar, un receptioner, o menajera si un chelner)
* Toate camerele au minim 12 mp
* Ca rezervarea sa fie valida data de check-in trebuie sa fie mai mica decat data de check-out
* Data de expirare a cardului cu care se efectueaza plata trebuie sa fie mai mare decat anul curent
* Hotelul are maxim 7 stele

*3. Descrierea entităților, incluzând precizarea cheii primare.*

Pentru modelul de date referitor la gestiunea unui hotel avem: CLIENTI, ANGAJAT, RECEPTIONER, MENAJERA, BUCATAR, CHELNER , HOTEL, CAMERA, REZERVARE, PLATA, LOCATIE, SPECTACOL, FIRMA\_ORGANIZARE reprezinta entitati, cat si subentitati.

Vom prezenta entităţile modelului de date, dând o descriere completă a fiecăreia. De asemenea, pentru fiecare entitate se va preciza cheia primară.

CLIENTI = persoana care beneficiaza de serviciile hotelului, anume rezerva si plateste camera pe o perioada determinata, trimite diferite comenzi la bucatarie. Cheia primara a entitatii CLIENT este client\_id.

ANGAJAT = persoana angajata in cadrul hotelului, aceasta are un salariu lunar. Angajatii se impart in receptioneri, menajere, paznici, bucatari si chelneri. Cheia primara a entitatii ANGAJAT este angajat\_id.

RECEPTIONER = subentitate a entitatii ANGAJAT. Receptionerul este angajatul hotelului care se ocupa cu preluarea rezervarilor de catre client, validarea platilor rezervarilor si cu stocarea informatiilor personale ale clientilor.

MENAJERA = subentitate a entitatii ANGAJAT. Menajera este angajata hotelului care se ocupa cu curatenia si igienizarea camerelor dupa plecarea clientilor, cat si pe perioada sederii lor. Menajerele isi fac un orar fix pe care il respecta pentru buna gestionare a curateniei din camera.

BUCATAR = subentitate a entitatii ANGAJAT. Bucatarul este angajatul hotelului care se ocupa cu pregatirea preparatelor comandate de catre client.

CHELNER = subentitate a entitatii ANGAJAT. Chelnerul este angajatul hotelului care se ocupa cu preluarea comenzilor date de catre client. Acesta transmite mai departe comenzile catre bucatari pentru preparare.

HOTEL = entitate ce stocheaza date precum nume, locatie, numar de stele despre hotelul gestionat. Cheia primara a entitatii HOTEL este hotel\_id.

CAMERA = entitate ce reprezinta camera dintr-un hotel anume pe care o rezerva clientii, si care este curatata si igienizata de catre menajere dupa plecarea clientilor. Cheia primara a entitatii CAMERA este camera\_id.

REZERVARE = entitate ce se refera la detaliile rezervarii camerelor de catre client. Rezervarea este primita, iar ulterior acceptata de catre receptionerul hotelului. Cheia primara a entitatii REZERVARE este rezervare\_id.

PLATA = entitate ce face referire la plata rezervarii facute de client. Aceasta poate fi realizata doar prin intermediul cardului personal al clientului. Dupa validarea platii de catre receptioner clientul primeste camera. Cheia primara a entitatii PLATA este plata\_id.

LOCATIE = entitate care stocheaza diferite locatii, precum domiciliul clientilor, domiciliul angajatilor, cat si locatia hotelului. Cheia primara a entitatii LOCATIE este locatie\_id.

FIRMA\_ORGANIZARE = entitate in care se gasesc informatii despre firmele exterioare de organizarea a evenimentelor la hotel. Cheia primara a entitatii FIRMA\_ORGANIZARE este organizator\_id.

SPECTACOL = entitate ce se ocupa cu stocarea informatiilor despre diferitele spectacole care se desfasoara la hoteluri. Acestea sunt organizate de catre firmele exterioare de organizare. Cheia primara a entitatii SPECTACOL este spectacol\_id.

*4. Descrierea relațiilor, incluzând precizarea cardinalității acestora.*

Vom prezenta relaţiile modelului de date, dând o descriere completă a fiecăreia. De fapt, denumirile acestor legături sunt sugestive, reflectând conţinutul acestora şi entităţile pe care le leagă. Pentru fiecare relaţie se va preciza cardinalitatea minimă şi maximă.

ANGAJAT\_are\_LOCATIE = relatie care leaga entitatile ANGAJAT si LOCATIE, reflectand legatura dintre acestea (indica domiciliul angajatului-ului) Cardinalitatea minima, respectiv cardinalitatea maxima este 1:1 (un angajat are o locatie).

HOTEL\_se\_afla\_LOCATIE = relatie care leaga entitatile HOTEL si LOCATIE, reflectand legatura dintre acestea (indica locatia unui hotel) Cardinalitatea minima, respectiv cardinalitatea maxima este 1:1 (un hotel se afla la o locatie).

ANGAJAT\_lucreaza\_la\_HOTEL = relatie care leaga entitatile ANGAJAT si HOTEL, reflectand legatura dintre acestea( angajatul lucreaza la un hotel) Cardinalitatea minima este 1:4 (un hotel are minim patru angajati), iar cardinalitatea maxima este 1:n (un hotel are mai multi angajati).

ANGAJAT\_este\_RECEPTIONER = relatie care leaga entitatea ANGAJAT de subentitatea RECEPTIONER, reflectand legatura dintre acestea (relatia scoate in evidenta care dintre angajatii hotelului ocupa functia de receptioner, anume personalul care preia si valideaza rezervarile si platile clientilor). Cardinalitatea minima, respectiv cardinalitatea maxima este 1:1 (un angajat este un receptioner ).

ANGAJAT\_este\_MENAJERA = relatie care leaga entitatea ANGAJAT de subentitatea MENAJERA, reflectand legatura dintre acestea ( relatia scoate in evidenta care dintre angajatii hotelului ocupa functia de menajera, anume personalul care face curat si igienizeaza camerele dupa plecarea clientilor). Cardinalitatea minima, respectiv cardinalitatea maxima este 1:1 (un angajat este o menajera).

ANGAJAT\_este\_BUCATAR = relatie care leaga entitatea ANGAJAT de subentitatea BUCATAR, reflectand legatura dintre acestea ( relatia scoate in evidenta care dintre angajatii hotelului ocupa functia de bucatar, anume personalul care se ocupa cu prepararea felurilor de mancare comandate de catre clienti). Cardinalitatea minima, respectiv cardinalitatea maxima este 1:1 (un angajat este un bucatar).

ANGAJAT\_este\_CHELNER = relatie care leaga entitatea ANGAJAT de subentitatea CHELNER, reflectand legatura dintre acestea ( relatia scoate in evidenta care dintre angajatii hotelului ocupa functia de chelner, anume personalul care se ocupa cu preluarea comenzilor facute de catre clienti pentru restaurant, iar ulterior transmiterea acestora catre bucatari). Cardinalitatea minima, respectiv cardinalitatea maxima este 1:1 (un angajat este un chelner).

HOTEL\_are\_CAMERA = relatie care leaga entitatile HOTEL si CAMERA, reflectand legatura dintre acestea ( relatia asigneaza hotelului camerale pe care le are). Cardinalitatea minima este 1:1 (un hotel are minim o camera) si cardinalitate maxima 1:n (un hotel are mai multe camera).

CAMERA\_este\_rezervata\_REZERVARE = relatie care leaga entitatile CAMERA si REZERVARE, reflectand legatura dintre acestea ( relatia stabileste rezervarea camerelor de catre client) Cardinalitatea minima este 1:1 (o camera este rezervata o data) si cardinalitate maxima 1:n (o camera este rezervata de mai multe ori).

CLIENTI\_face\_REZERVARE = relatie care leaga entitatile CLIENTI si REZERVARE, reflectand legatura dintre acestea ( relatia gestioneaza rezervarile pe care le fac clientii) Cardinalitatea minima este 1:1 (un client face o rezervare) si cardinalitate maxima 1:n (un client face mai multe rezervari).

ANGAJAT (RECEPTIONER)\_preia\_REZERVARE = relatie care leaga entitatile ANGAJAT de tip receptioner si REZERVARE, reflectand legatura dintre acestea ( relatia indica responsabilitatile receptionerului de a prelua si valida rezervarile facute de catre clienti) Cardinalitatea minima este 1:1 (un receptioner preia o rezervare) si cardinalitate maxima 1:n (un receptioner preia mai multe rezervari).

REZERVARE\_este\_platita\_PLATA = relatie care leaga entitatile REZERVARE si PLATA, reflectand relatia dintre acestea ( relatia stabileste momentul cand rezervarile efectuate de clienti sunt platite). Cardinalitatea minima si cardinalitatea maxima este 1:1 (o rezervare este platita o data).

FIRMA\_ORGANIZARE\_organizeaza\_SPECTACOLE = relatie care leaga entitatile FIRMA\_ORGANIZARE si SPECTACOLE, reflectand relatia dintre acestea ( relatia stabileste care firme organizeaza spectacole). Cardinalitatea minima este 1:1 (o firma organizeaza un spectacol), iar cardinalitatea maxima este 1:n (o firma organizeaza mai multe spectacole).

SPECTACOL\_se\_desfasoara\_HOTEL = relatie care leaga entitatile SPECTACOL si HOTEL, reflectand relatia dintre acestea( ce spectacole se desfasoara la hotel). Cardinalitatea minima este 1:1 (un spectacol se desfasoara la un hotel), iar cea maxima 1:n (mai multe spectacole se desfasoara la un hotel).

ANGAJAT (MENAJERA)\_FACE\_CURAT\_IN\_CAMERA = relatie de tip many-to-many care leaga entitatile ANGAJAT de tip menajera, FACE\_CURAT\_IN si CAMERA, a doua entitate fiind tabela asociativa ( relatia stabileste programul de lucru/curatare al camerelor de catre menajere). Cardinalitatea minima este 1:1 (o menajera face curat intr-o camera) si cardinalitate maxima n:m ( mai multe menajere fac curat in mai multe camera)

CLIENTI\_primeste\_comanda\_de\_la\_ANGAJAT (CHELNER)\_pregatita\_de\_ANGAJAT (BUCATAR) = relatie de tip 3 (many-to-many intre toate cele 3 tabele) ce leaga entitatile ANGAJAT de tip bucatar, ANGAJAT de tip chelner si CLIENTI, reflectand legatura dintre acestea( relatia stabileste de catre ce chelneri au fost serviti clientii, respectiv ce bucatarii au gatit pentru clienti si ce chelneri au preluat mancarea de la bucatari). Cardinalitatea minima este 1:1:1 (un client a fost servit de catre un chelner care a preluat un meni pregatit de un bucatar), iar cardinalitatea maxima este l:n:m (mai multi clienti au fost serviti de catre mai multi chelneri care au preluat meniurile preparate de mai multi bucatari). Denumirea acestei relatii va fi primeste.

*5. Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicite, valori posibile ale atributelor.*

CLIENTI:

client\_id = variabila de tip NUMBER, care reprezinta codul clientului din baza de date;

nume = variabila de tip VARCHAR2 de lungime maxima 15, care reprezinta numele de familie al clientulului;

prenume = variabila de tip VARCHAR2 de lungime maxima 15, care reprezinta prenumele clientului;

nr\_de\_tel = variabila de tip VARCHAR2 de lungime maxima 20, care reprezinta numarul de telefon al clientului;

locatie\_id = variabila de tip NUMBER, care reprezinta numarul locatiei din tabela LOCATIE, unde gasim domiciliul clientului;

ANGAJAT:

angajat\_id = variabila de tip NUMBER, care reprezinta codul angajatului hotelului din baza de date;

nume = variabila de tip VARCHAR2 de lungime maxima 15, care reprezinta numele de familie al angajatului;

prenume = variabila de tip VARCHAR2 de lungime maxima 15, care reprezinta prenumele angajatului;

salariu = variabila de tip NUMBER care este mai mare decat 0 si reprezinta salariul lunar al angajatului;

data\_angajare = variabila de tip DATE ce reprezinta data la care s-a angajat;

locatie\_id = variabila de tip NUMBER, care reprezinta numarul locatiei din tabela LOCATIE, unde gasim domiciliul angajatului;

hotel\_id = variabila de tip NUMBER, care reprezinta numarul hotelului din tabela HOTEL, unde gasim informatii despre hotelul unde lucreaza angajatul;

tip\_angajat = variabila de tip VARCHAR2 de lungime maxima 15, ce indica tipul de angajat al hotelului;

nr\_chelner = variabila de tip NUMBER, care reprezinta numarul chelnerului, in cazul in care angajatul curent nu este chelner, nr\_chelner va avea valoarea NULL;

nr\_menajera = variabila de tip NUMBER, care reprezinta numarul menajerei, in cazul in care angajatul curent nu este menajera, nr\_menajera va avea valoarea NULL;

nr\_receptioner = variabila de tip NUMBER, care reprezinta numarul receptionerului, in cazul in care angajatul curent nu este receptioner, nr\_receptioner va avea valoarea NULL;

nr\_bucatar = variabila de tip NUMBER, care reprezinta numarul bucatarului, in cazul in care angajatul curent nu este bucatar, nr\_bucatar va avea valoarea NULL;

LOCATIE:

locatie\_id = variabila de tip NUMBER, care reprezinta codul unei anumite locatii din baza de date;

strada = variabila de tip VARCHAR2, care reprezinta denumirea strazii respectivei locatii;

numar = variabila de tip NUMBER, mai mare decat 0 care reprezinta numarul strazii respectivei locatii;

orasul = variabila de tip VARCHAR2, care reprezinta orasul in care se afla locatia respective;

tara = variabila de tip VARCHAR2, care reprezinta tara in care se afla locatia respective;

HOTEL:

hotel\_id = variabila de tip NUMBER, care reprezinta codul unui hotel din baza de date;

nume\_hotel = variabila de tip VARCHAR2, care reprezinta numele hotelului respective;

stele = variabila de tip NUMBER mai mare decat 0 si mai mica sau egala cu 7, care reprezinta numarul de stele aferente hotelului;

locatie\_id = variabila de tip NUMBER, care reprezinta numarul locatiei din tabela LOCATIE, unde gasim informatii despre adresa hotelului;

CAMERA:

camera\_id = variabila de tip NUMBER, care reprezinta codul unei anumite camera ce apartine unui hotel din baza de date;

etaj = variabila de tip NUMBER, care reprezinta etajul la care se gaseste camera;

suprafata = variabila de tip NUMBER, mai mare decat 12, care reprezinta suprafata camerei;

hotel\_id = variabila de tip NUMBER, care reprezinta numarul hotelului din tabela HOTEL, unde gasim informatii despre hotelul din care face parte camera;

REZERVARE:

rezervare\_id = variabila de tip NUMBER, care reprezinta numarul unei rezervari facute de catre un client la un hotel;

checkindata = variabila de tip DATE, mai mica sau egala cu checkoutdata care reprezinta data de sosire a unui client la hotel;

checkoutdata = variabila de tip DATE, mai mare sau egala cu checkindata care reprezinta data de plecare a unui client de la hotel;

client\_id = variabila de tip NUMBER, care reprezinta numarul clientului din tabela CLIENTI, unde gasim informatii despre clientul care a efectuat rezervarea;

camera\_id = variabila de tip NUMBER, care reprezinta numarul camerei rezervate din tabela CAMERA;

angajat\_id = variabila de tip NUMBER, care reprezinta numarul angajatului de tip receptioner din tabela ANGAJAT, unde gasim informatii despre receptionerul care a preluat rezervarea;

PLATA:

plata\_id = variabila de tip NUMBER, ce reprezinta numarul unei plati efectuate de client in urma rezervarii camerei;

nrcard = variabila de tip VARCHAR2 de lungime maxima 20, ce reprezinta numarul cardului personal cu care clientul efectueaza plata;

dataexp = variabila de tip DATE, ce reprezinta data de expirare a cardului cu care se face rezervarea;

cvv = variabila de tip NUMBER, ce reprezinta codul de securitate al cardului;

rezervare\_id = variabila de tip NUMBER, ce reprezinta numarul rezervarii din tabela REZERVARE;

SPECTACOL:

spectacol\_id = variabila de tip NUMBER, ce reprezinta numarul unui spectacol difuzat la un hotel ;

nume = variabila de tip VARCHAR2, ce reprezinta numele spectacolului;

hotel\_id = variabila de NUMBER, ce reprezinta numarul hotelului din tabela HOTEL, unde gasim informatii despre hotelul la care se tine respectivul spectacol;

organizator\_id = variabila de tip NUMBER, ce reprezinta numarul firmei de organizare din tabela FIRMA\_ORGANIZARE, ce face referire la firma care organizeaza un anumit spectacol;

FIRMA\_ORGANIZARE:

organizator\_id = variabila de tip NUMBER, ce reprezinta numarul de ordine al unei firme de organizare de spectacole;

nume\_organizator = variabila de tip VARCHAR2, ce reprezinta numele firmei de organizare spectacole;

rating = variabila de tip NUMBER, ce reprezinta cat de apreciata este o firma de organizare spectacole;

FACE\_CURAT\_IN:

camera\_id = variabila de tip NUMBER, ce reprezinta numarul camerei din tabela CAMERA, unde gasim informatii despre camera in care se efectueaza curatenia;

nr\_menajera = variabila de tip NUMBER, ce reprezinta numarul angajatului de tip menajera din tabela ANGAJAT, unde gasim menajera care efectueaza curatenia in camera;

dataincepere = variabila de tip DATA ce reprezinta data de inceput a curateniei efectuate de menajere in camera;

dataterminare = variabila de tip DATA, mai mare sau egala decat dataincepere ce reprezinta momentul terminarii curateniei in camera;

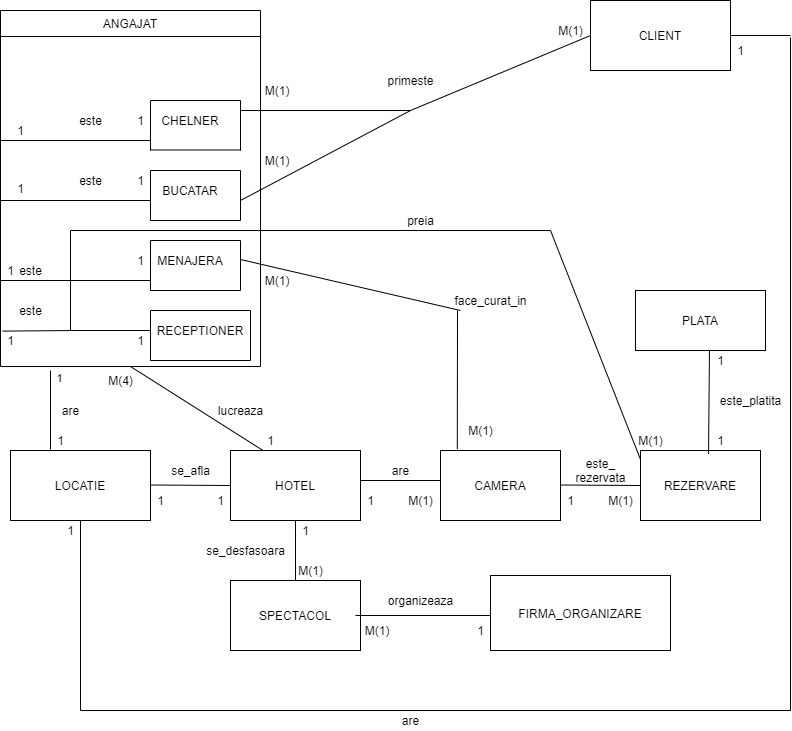
PRIMESTE:

nr\_chelner = variabila de tip NUMBER, ce reprezinta numarul angajatului de tip chelner din tabela ANGAJAT ;

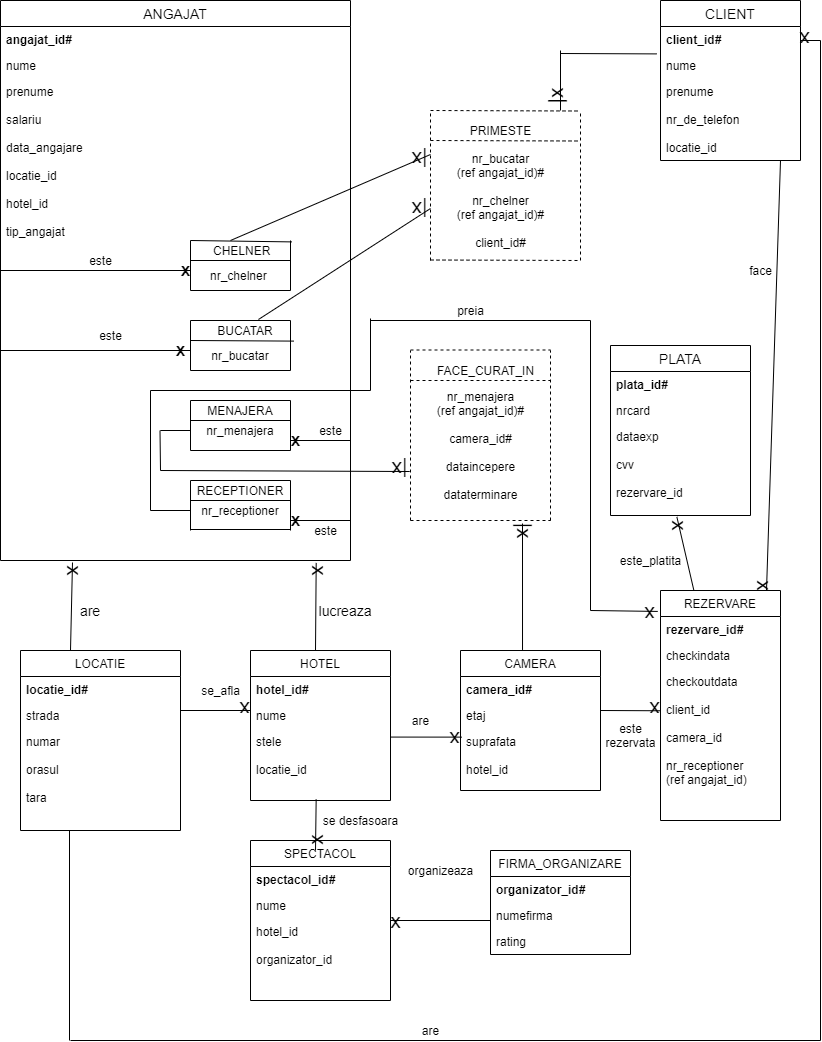
nr\_bucatar = variabila de tip NUMBER, ce reprezinta numarul angajatului de tip bucatar din tabela ANGAJAT;

client\_id = variabila de tip NUMBER, ce face referire la numarul clientului din tabela CLIENTI;

*6. Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5.*



*7. Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectata la punctul 6. Diagrama conceptuală obținută trebuie să conțină minimum 6 tabele (fără considerarea subentităților), dintre care cel puțin un tabel asociativ.*



*8. Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale proiectata la punctul 7.*

ANGAJAT(angajat\_id#, nume, prenume, salariu, data\_angajare, locatie\_id, hotel\_id, tip\_angajat, nr\_chelner, nr\_menajera, nr\_receptioner, nr\_bucatar)

CLIENTI(client\_id#, nume, prenume, nr\_de\_tel, locatie\_id)

LOCATIE(locatie\_id#, strada, numar, orasul, tara)

HOTEL(hotel\_id#, nume, stele, locatie\_id)

CAMERA(camera\_id#, etaj, suprafata, hotel\_id)

SPECTACOL(spectacol\_id#, nume, hotel\_id, organizator\_id)

FIRMA\_ORGANIZARE(organizator\_id#, nume, rating)

REZERVARE(rezervare\_id#, checkindata, checkoutdata, client\_id, camera\_id, angajat\_id)

PLATA(plata\_id#, nrcard, dataexp, cvv, rezervare\_id)

FACE\_CURAT\_IN(camera\_id#, angajat\_id#, dataincepere, dataterminare)

PRIMESTE(client\_id#, nr\_chelner(referire la angajat\_id)#, nr\_bucatar(referire la angajat\_id)#)

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

Non-FN1:

* HOTEL(hotel\_id#, nume\_hotel, stele, locatie\_id, coduri\_camere)
* CAMERA(camera\_id#, etaj, suprafata, hotel\_id)

Nu este in FN1, deoarece coloana coduri\_camera nu este indivizibila/atomica

FN1:

* HOTEL(hotel\_id#, nume\_hotel, stele, locatie\_id)
* CAMERA(camera\_id#, etaj, suprafata, hotel\_id)

Non-FN2:

* ANGAJAT(angajat\_id#, nume, prenume, salariu, data\_angajare, locatie\_id, hotel\_id, tip\_angajat, nr\_chelner, nr\_menajera, nr\_receptioner, nr\_bucatar)
* FACE\_CURAT\_IN(camera\_id#, nr\_menajera#, dataincepere, dataterminare, etaj)
* CAMERA(camera\_id#, suprafata, hotel\_id)

Nu este in FN2, deoarece coloana etaj depinde doar de camera\_id#, nu si de nr\_menajera#

FN2:

* ANGAJAT(angajat\_id#, nume, prenume, salariu, data\_angajare, locatie\_id, hotel\_id, tip\_angajat, nr\_chelner, nr\_menajera, nr\_receptioner, nr\_bucatar)
* FACE\_CURAT\_IN(camera\_id#, nr\_menajera#, dataincepere, dataterminare)
* CAMERA(camera\_id#, etaj, suprafata, hotel\_id)

Non-FN3:

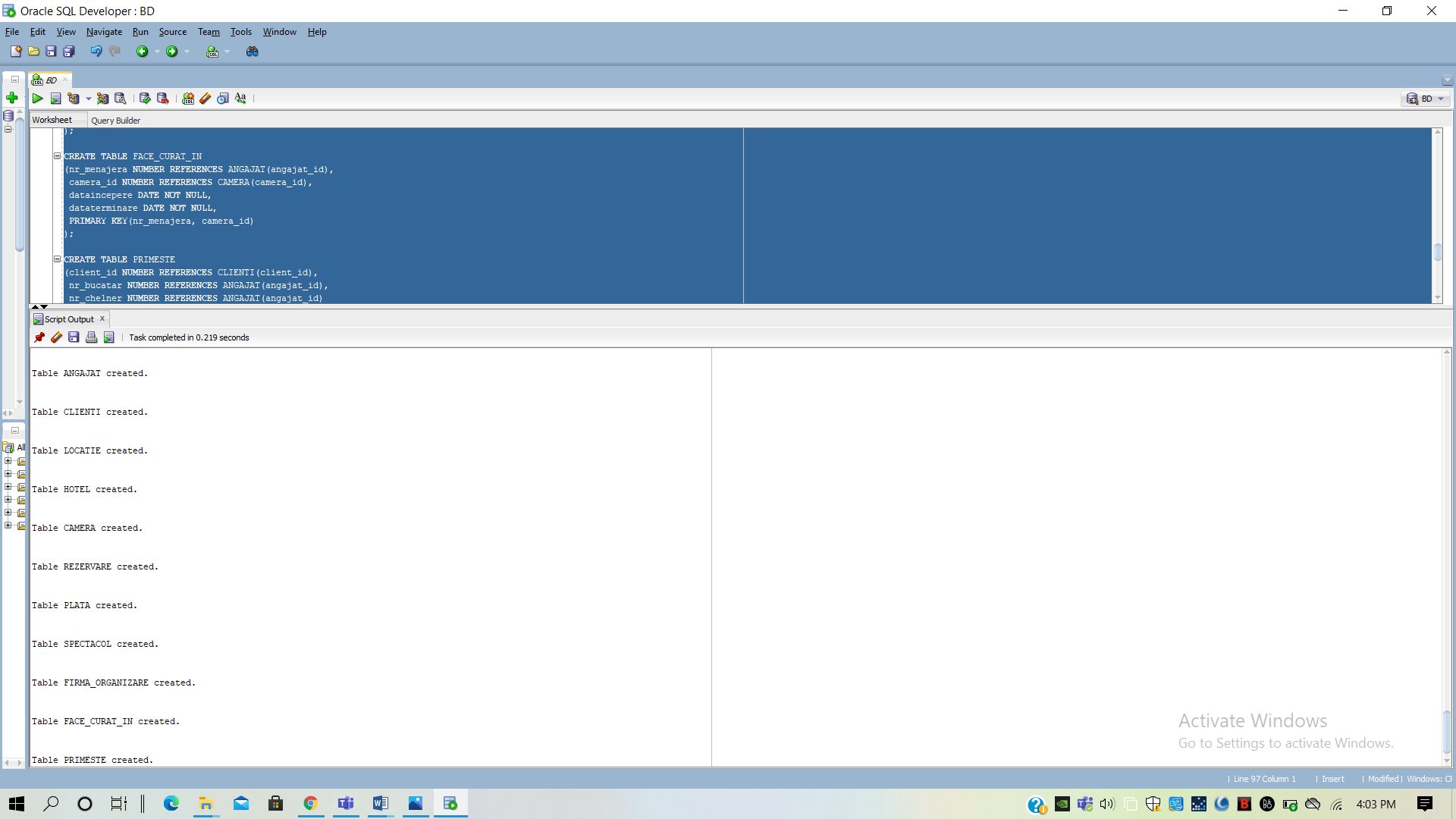
* CLIENTI(client\_id#, nume, prenume, nr\_de\_telefon, locatie\_id)
* REZERVARE(rezervare\_id#, checkindata, checkoutdata, client\_id, nume, camera\_id, nr\_receptioner)

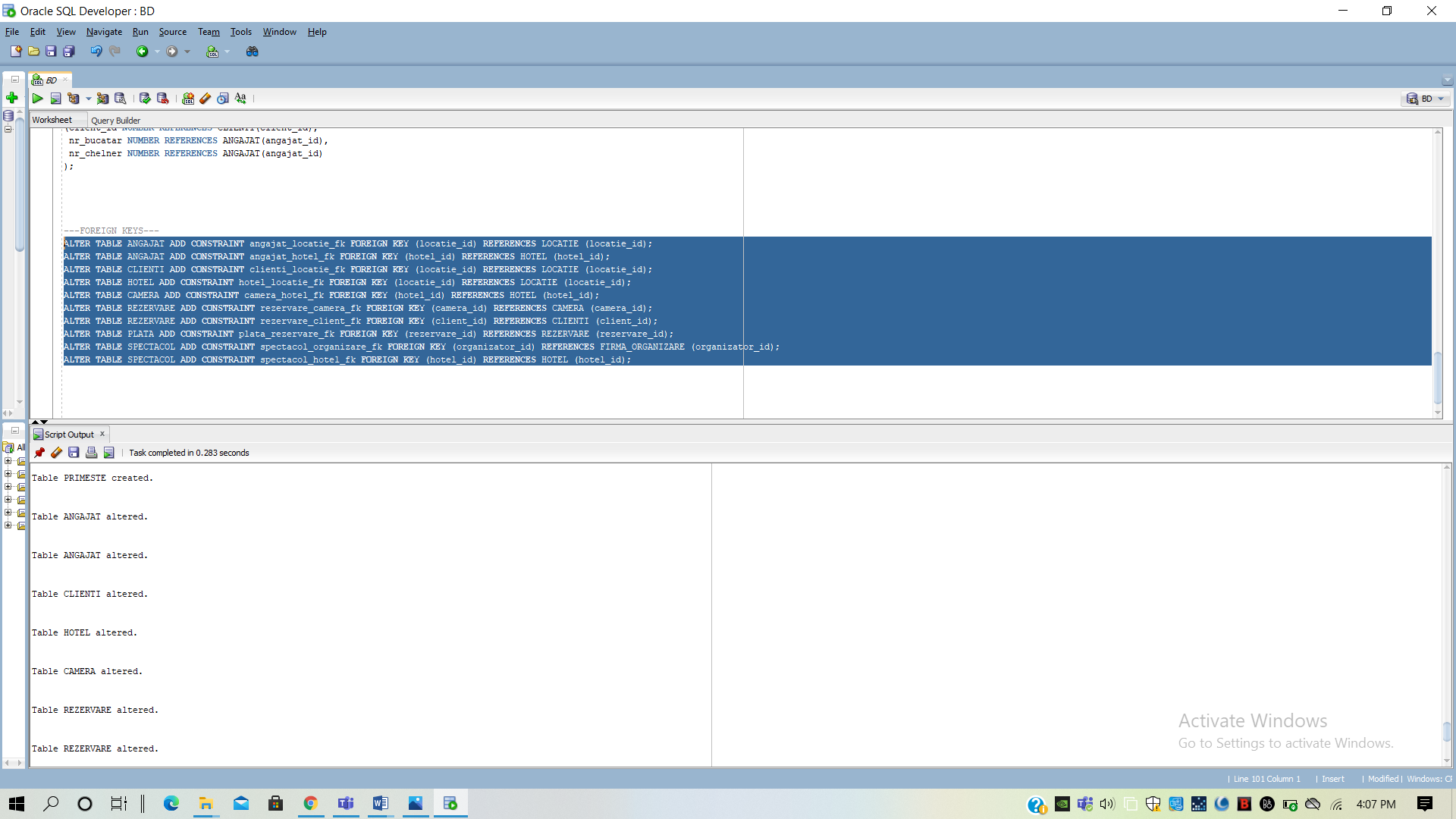
Nu este in FN3, deoarece in tabela REZERVARE coloana nume depinde de client\_id si nu de cheia primara a tabelei, anume rezervare\_id#.

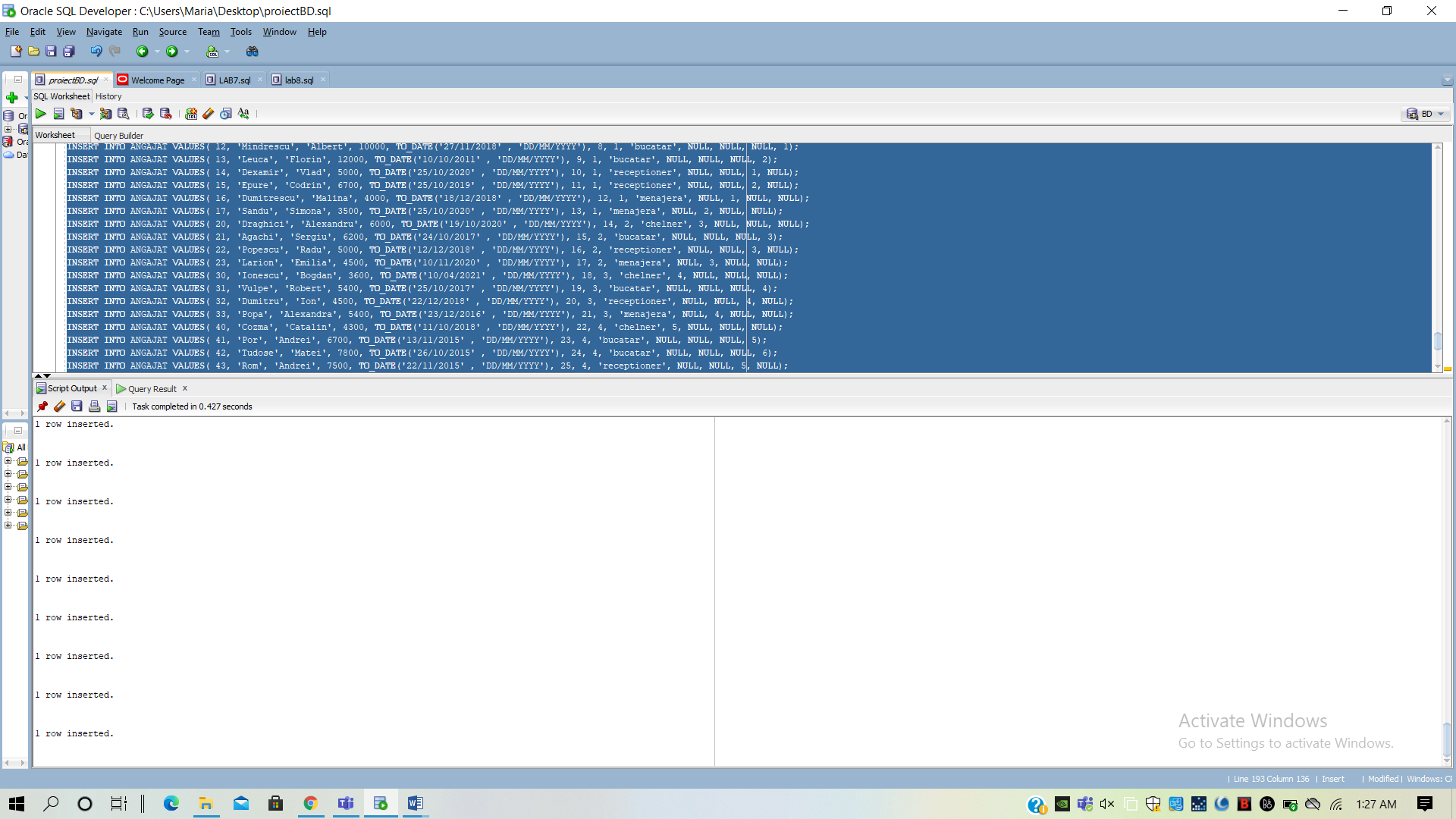
FN3:

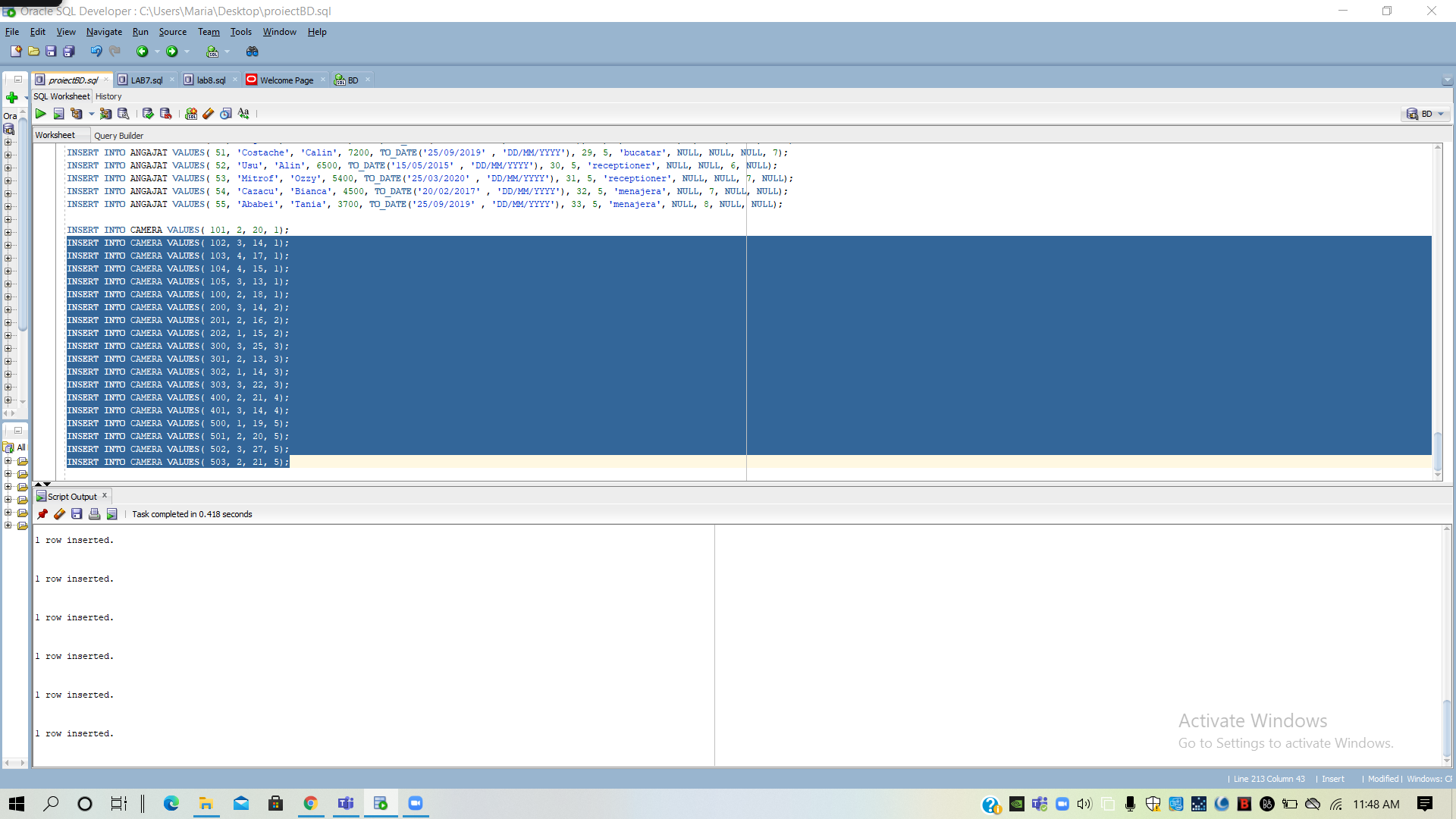
* CLIENTI(client\_id#, nume, prenume, nr\_de\_telefon, locatie\_id)
* REZERVARE(rezervare\_id#, checkindata, checkoutdata, client\_id, camera\_id, nr\_receptioner)

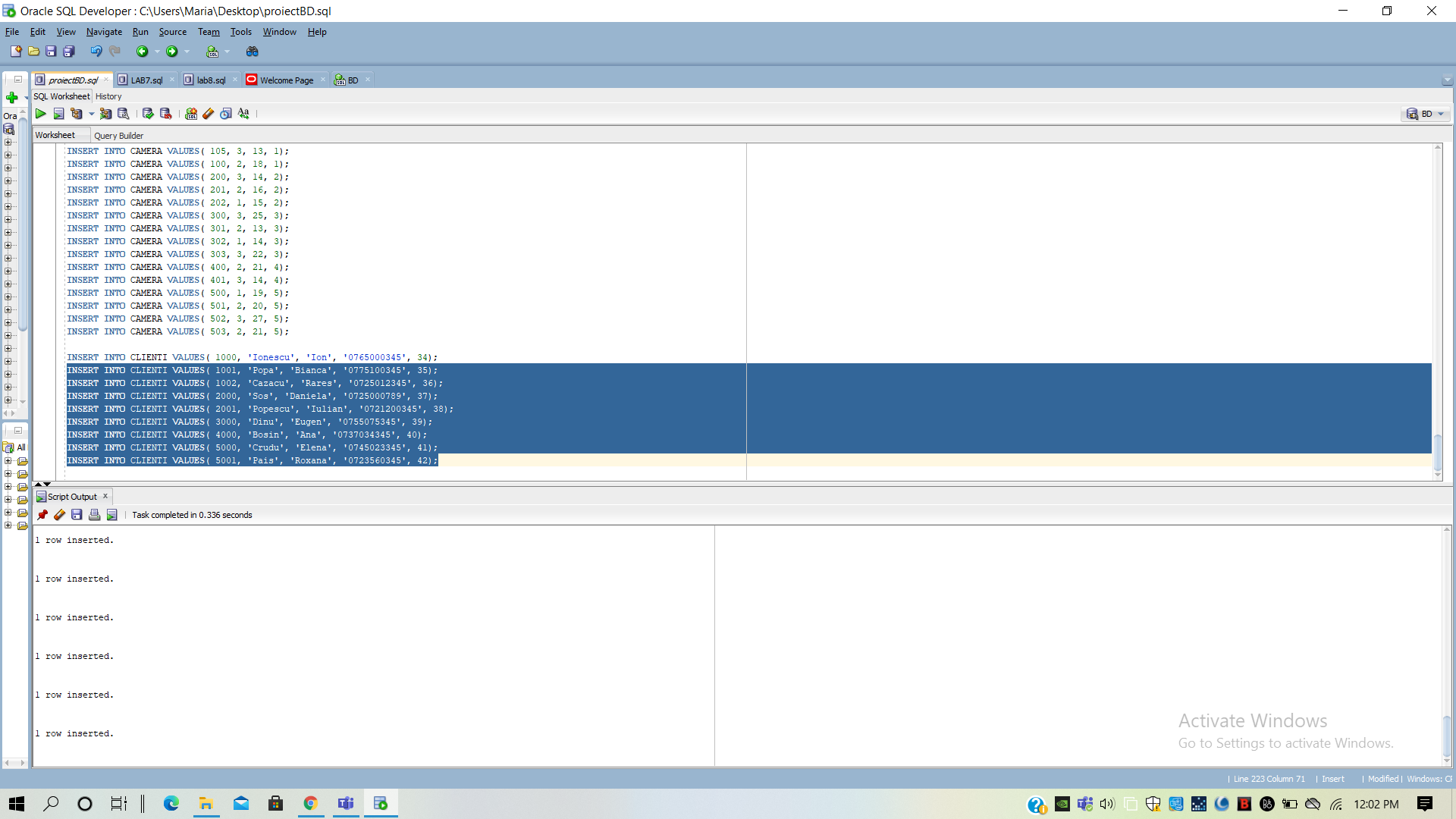
*10. Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea (minimum 5 înregistrări în fiecare tabel neasociativ; minimum 10 înregistrări în tabelele asociative).*

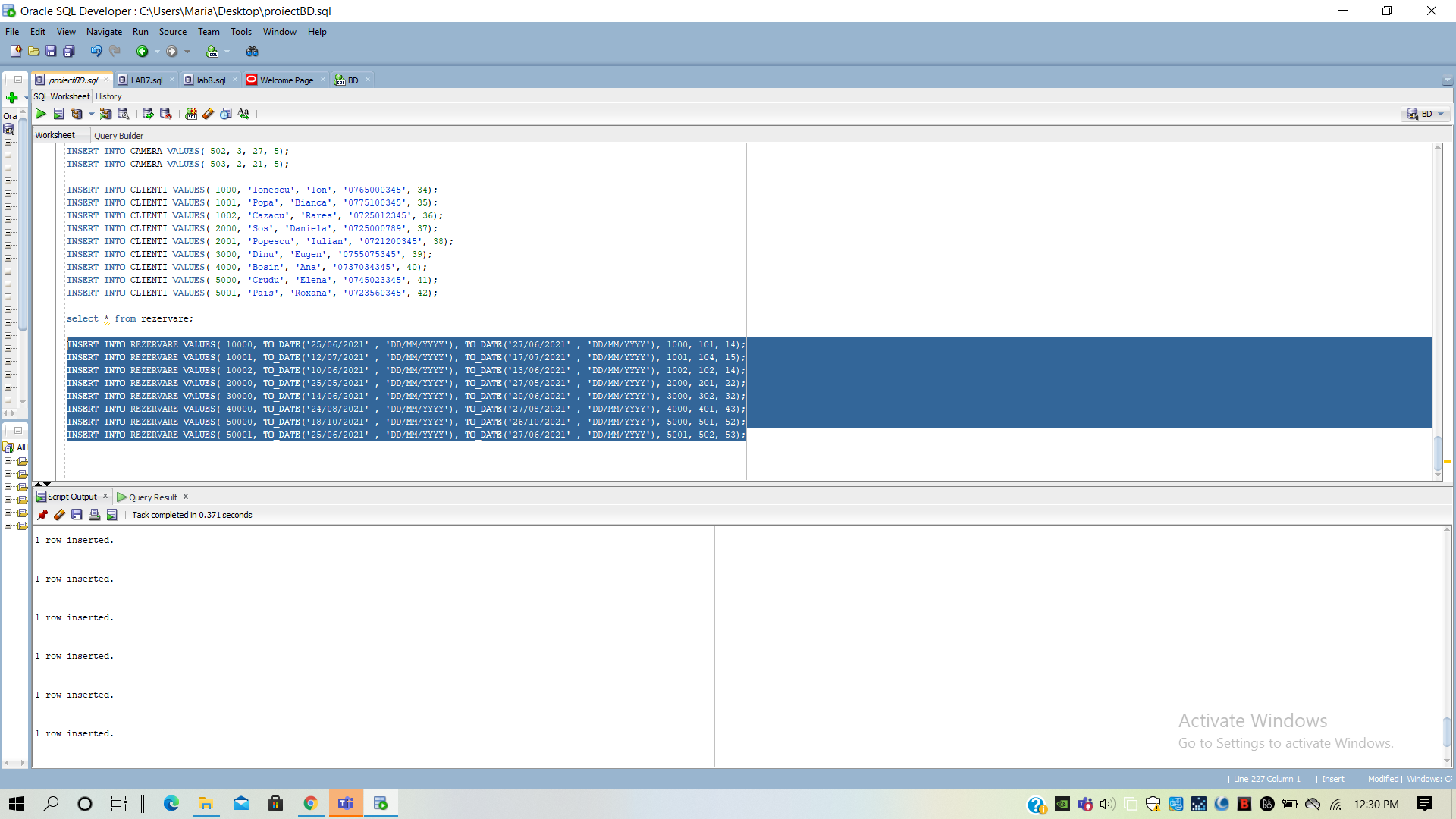


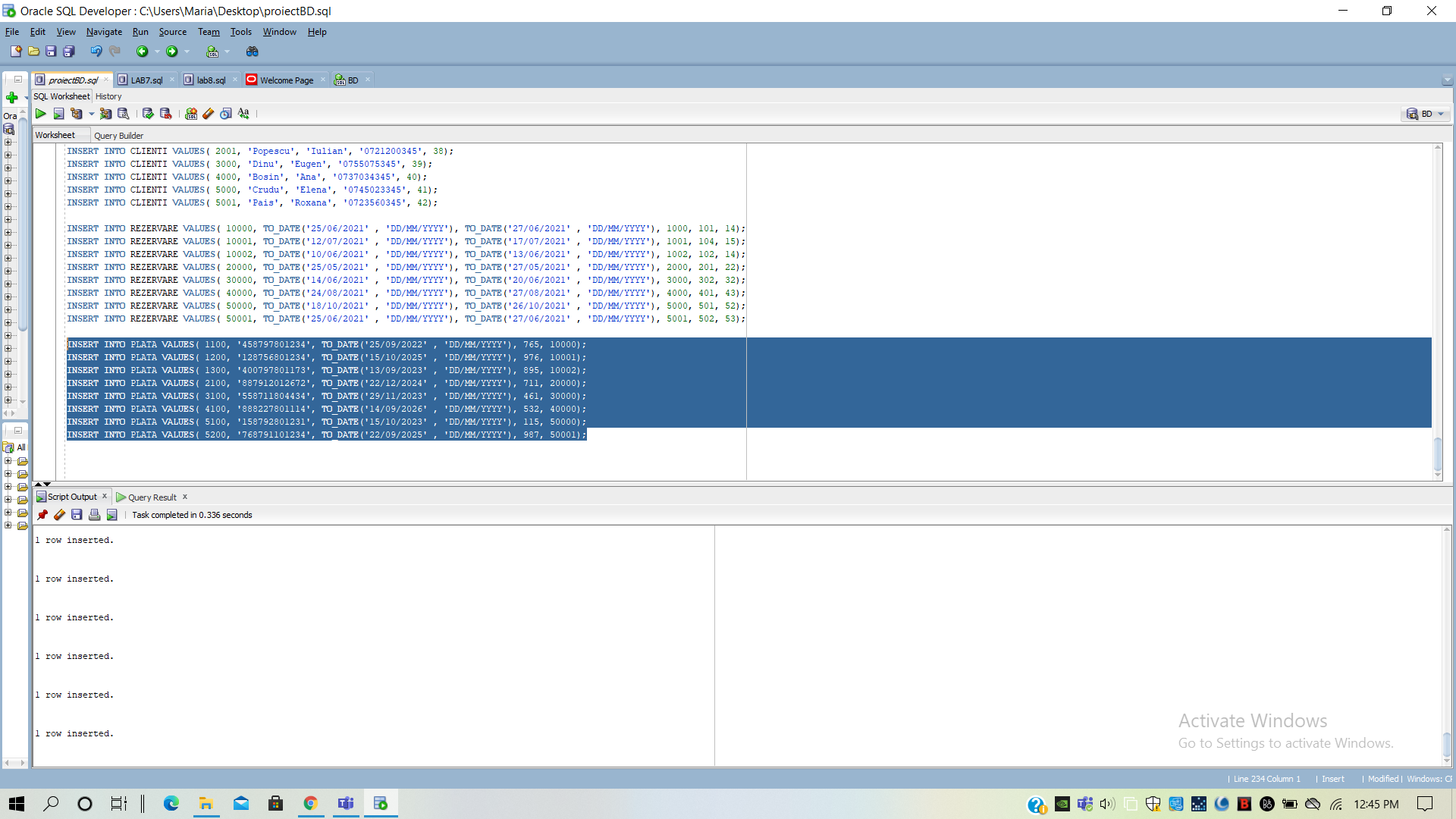


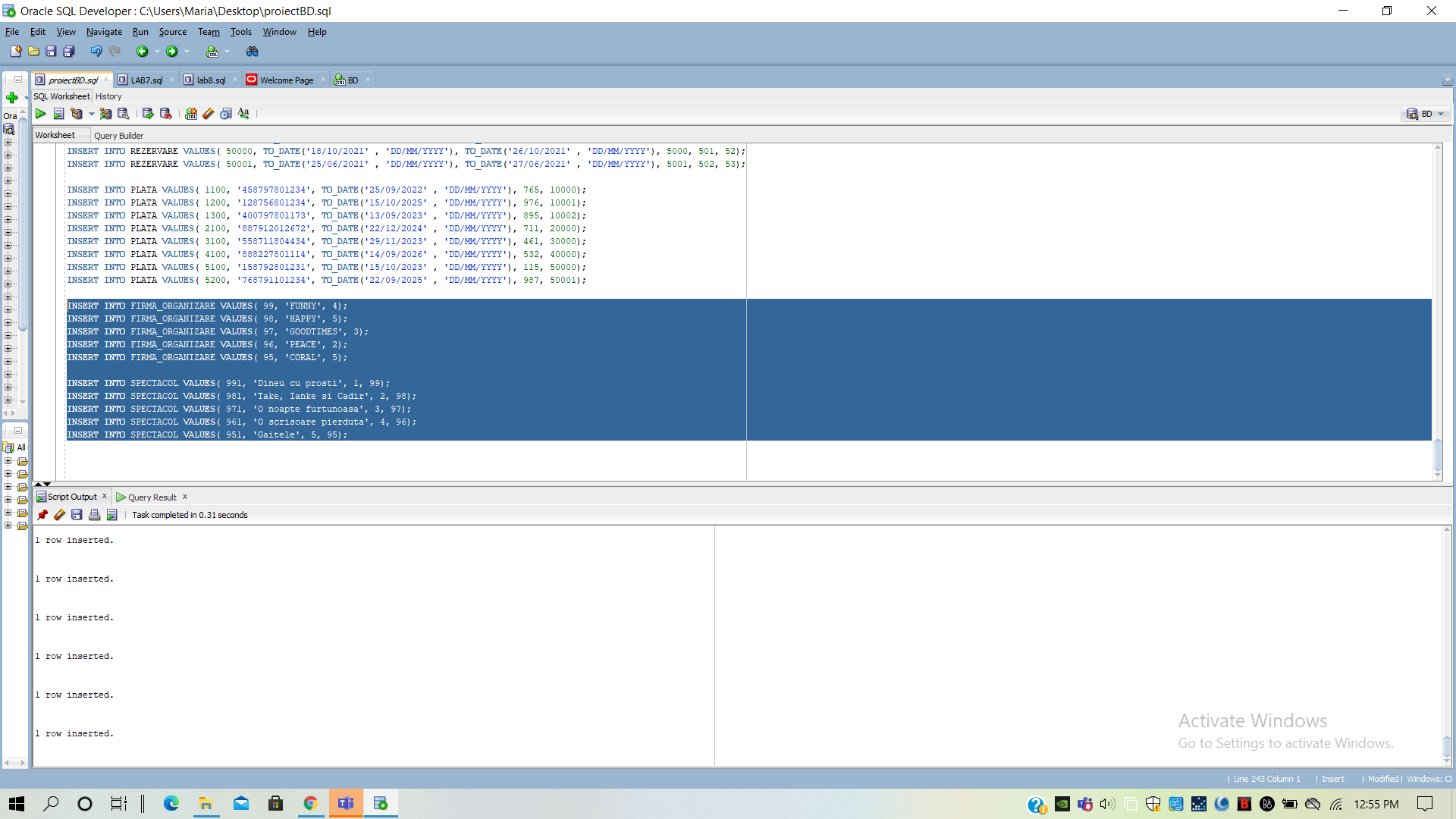


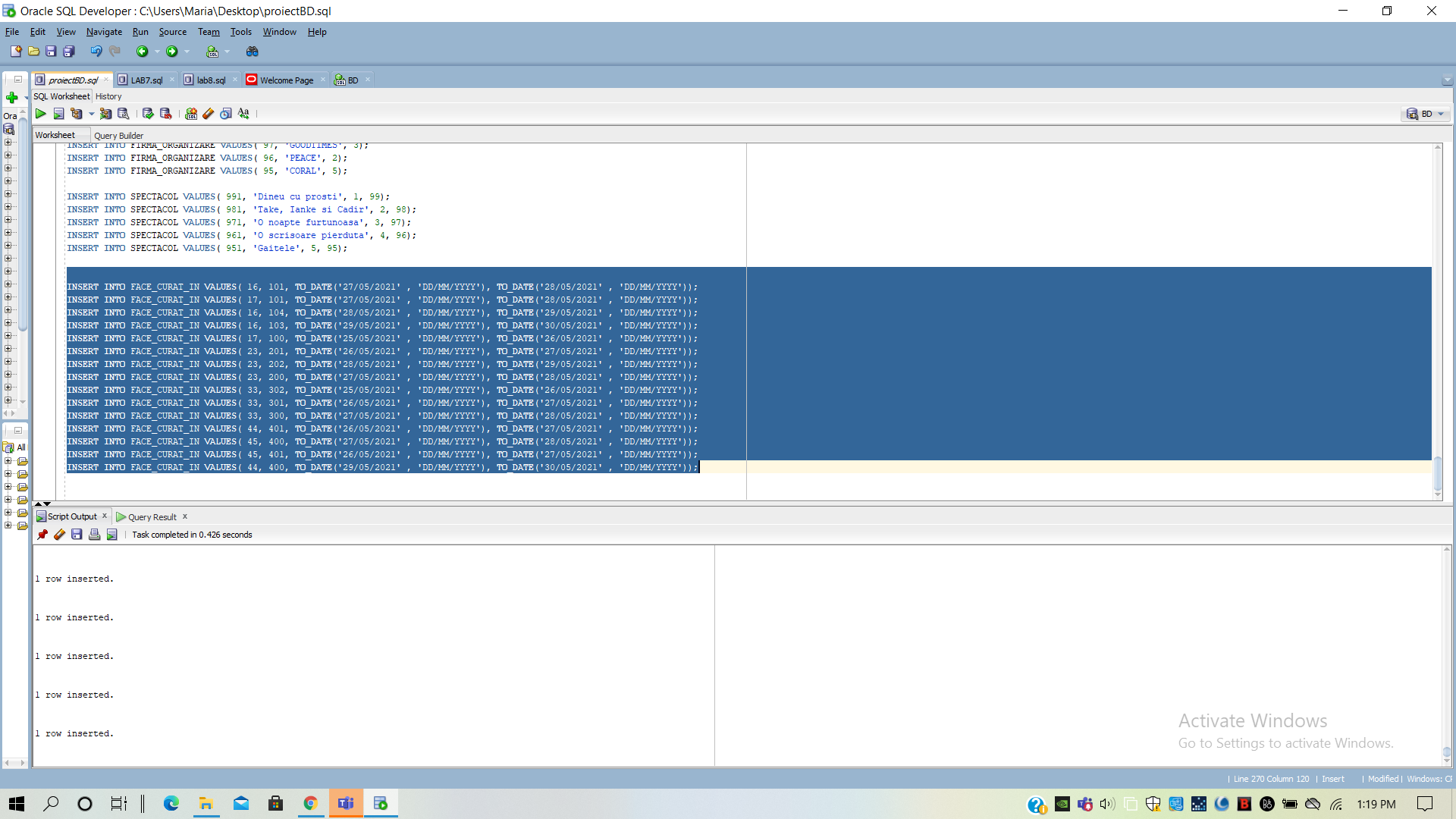


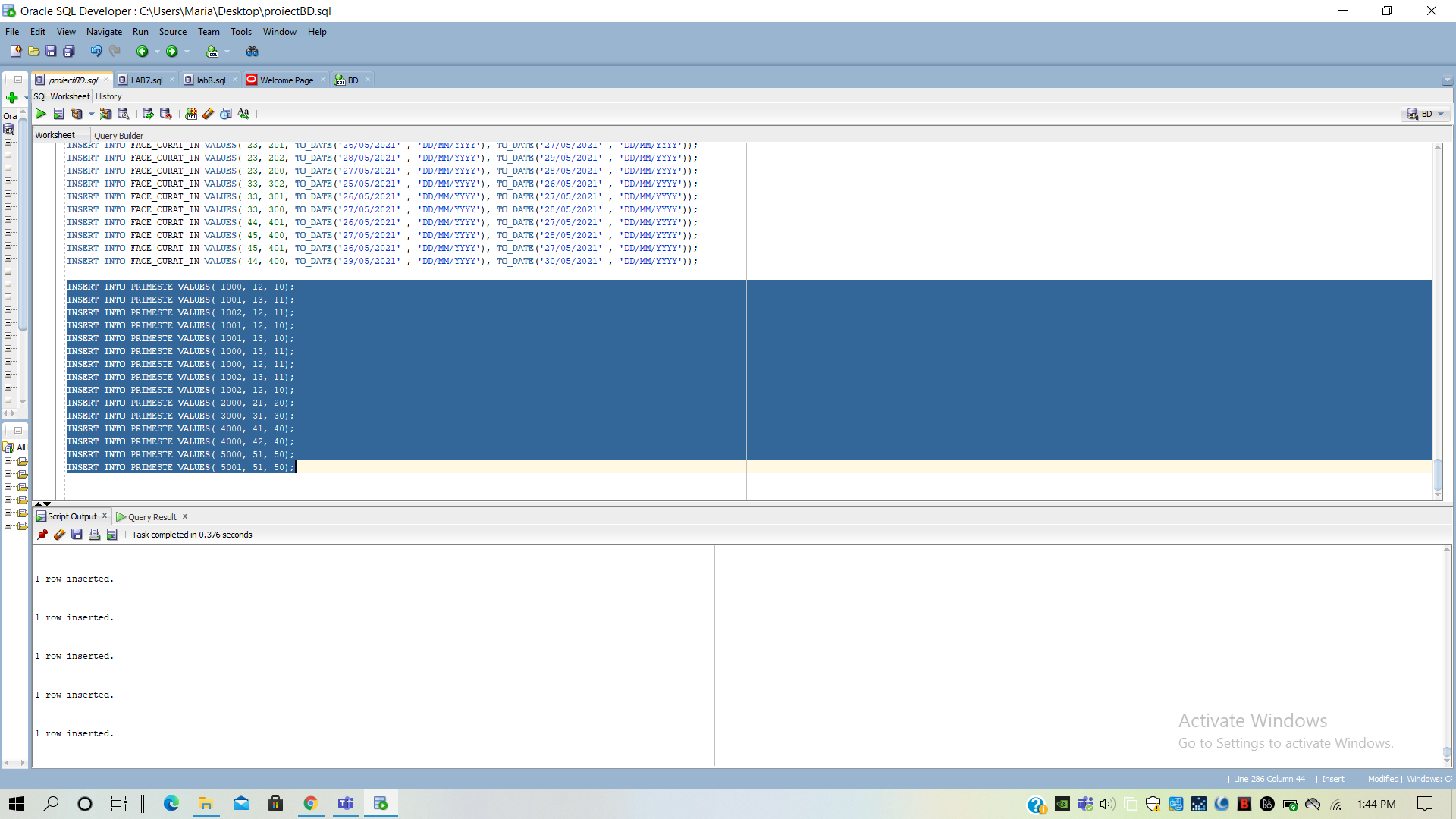


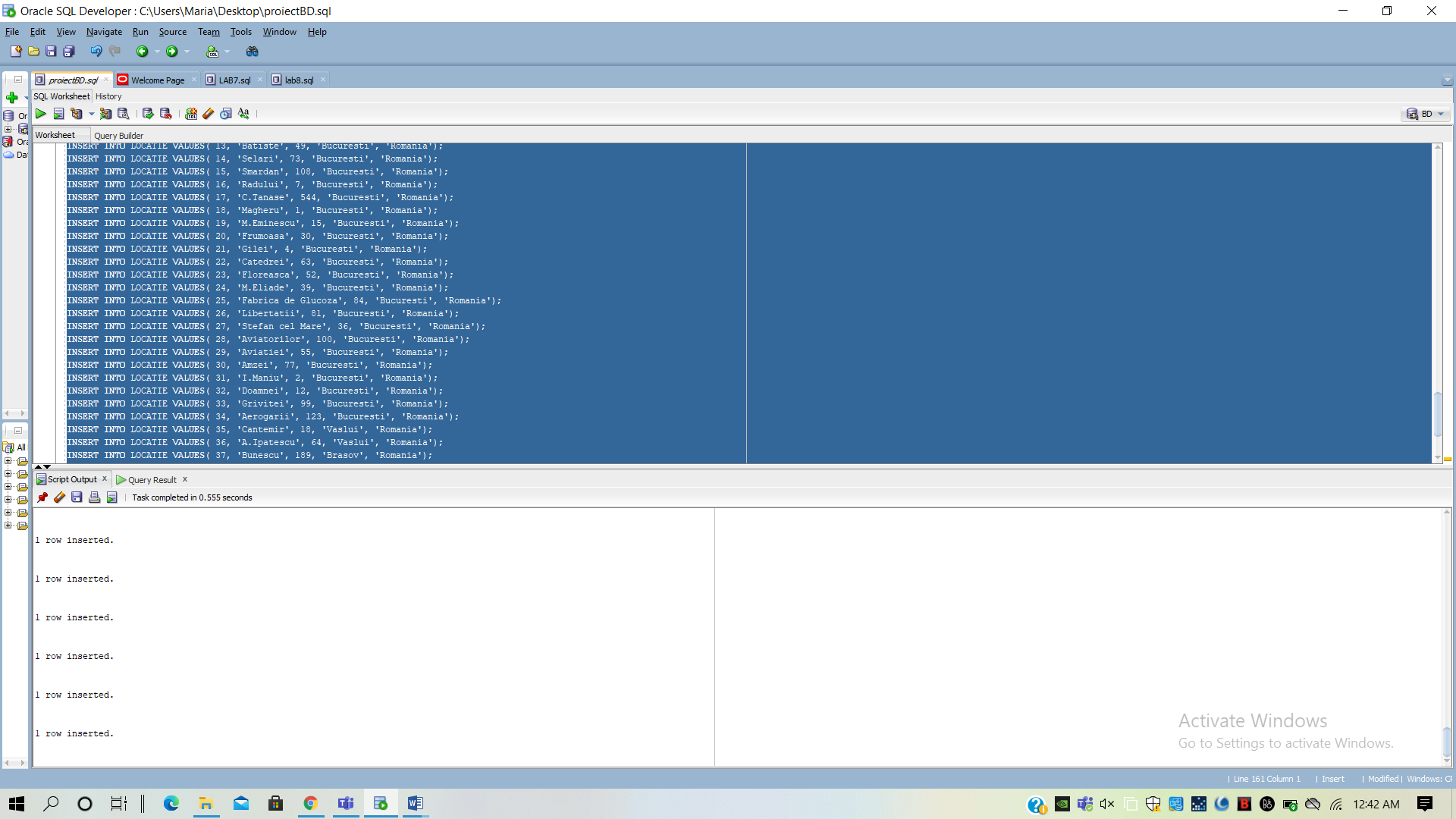


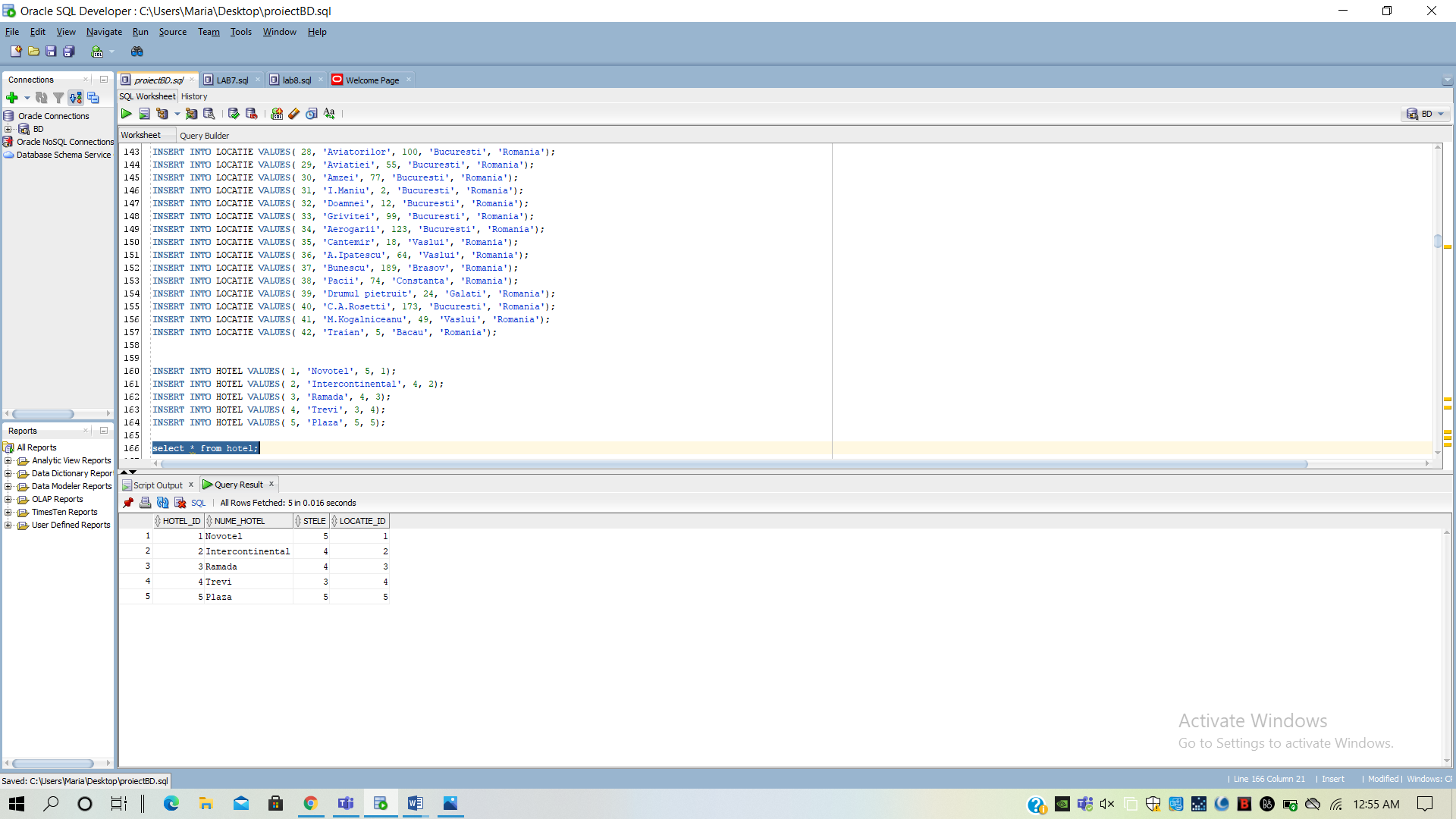












*11. Formulați în limbaj natural și implementați 5 cereri SQL complexe ce vor utiliza, în ansamblul lor, următoarele elemente:*

*• operație join pe cel puțin 4 tabele*

*• filtrare la nivel de linii*

*• subcereri sincronizate în care intervin cel puțin 3 tabele*

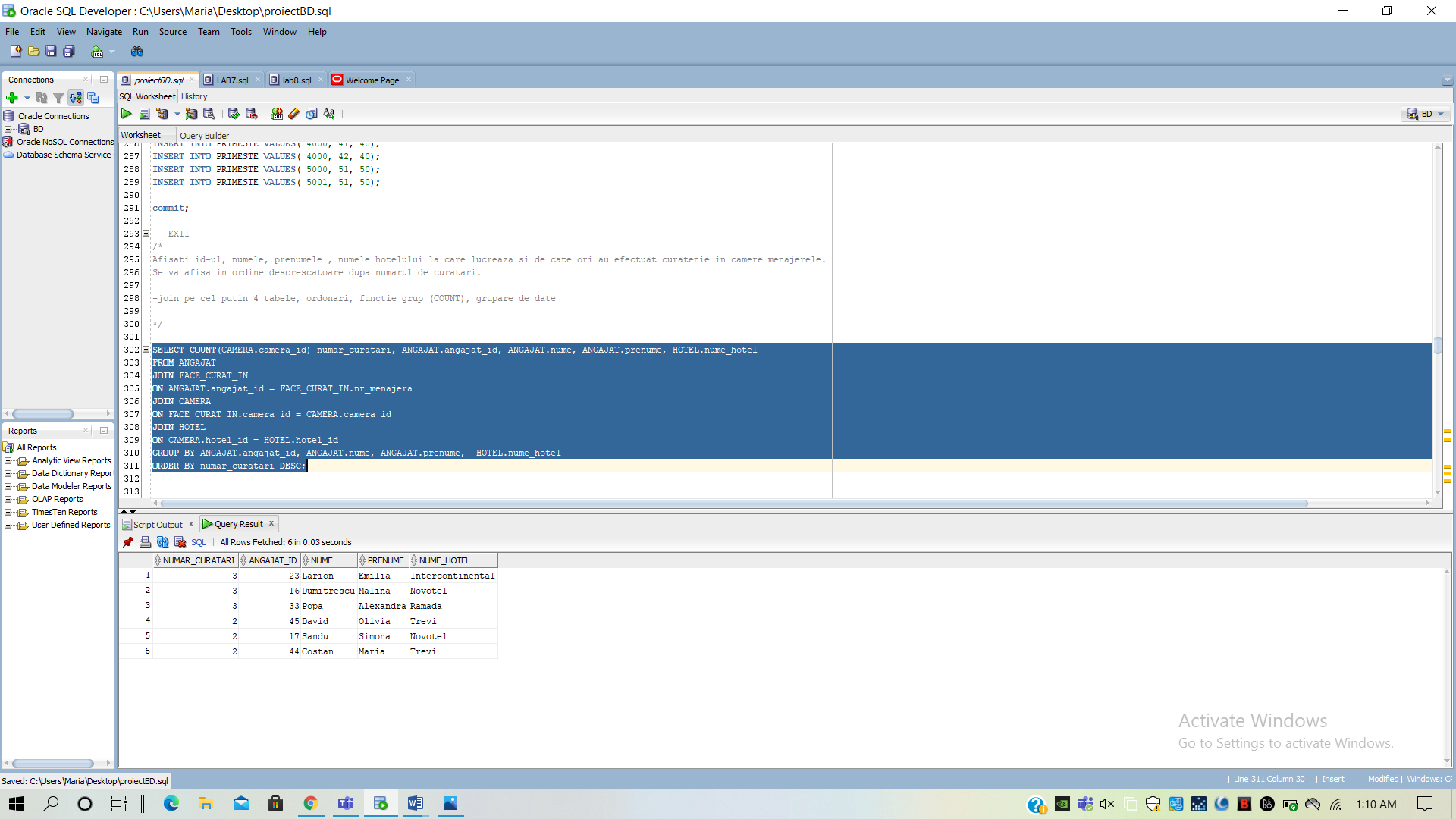
*• subcereri nesincronizate în care intervin cel puțin 3 tabele*

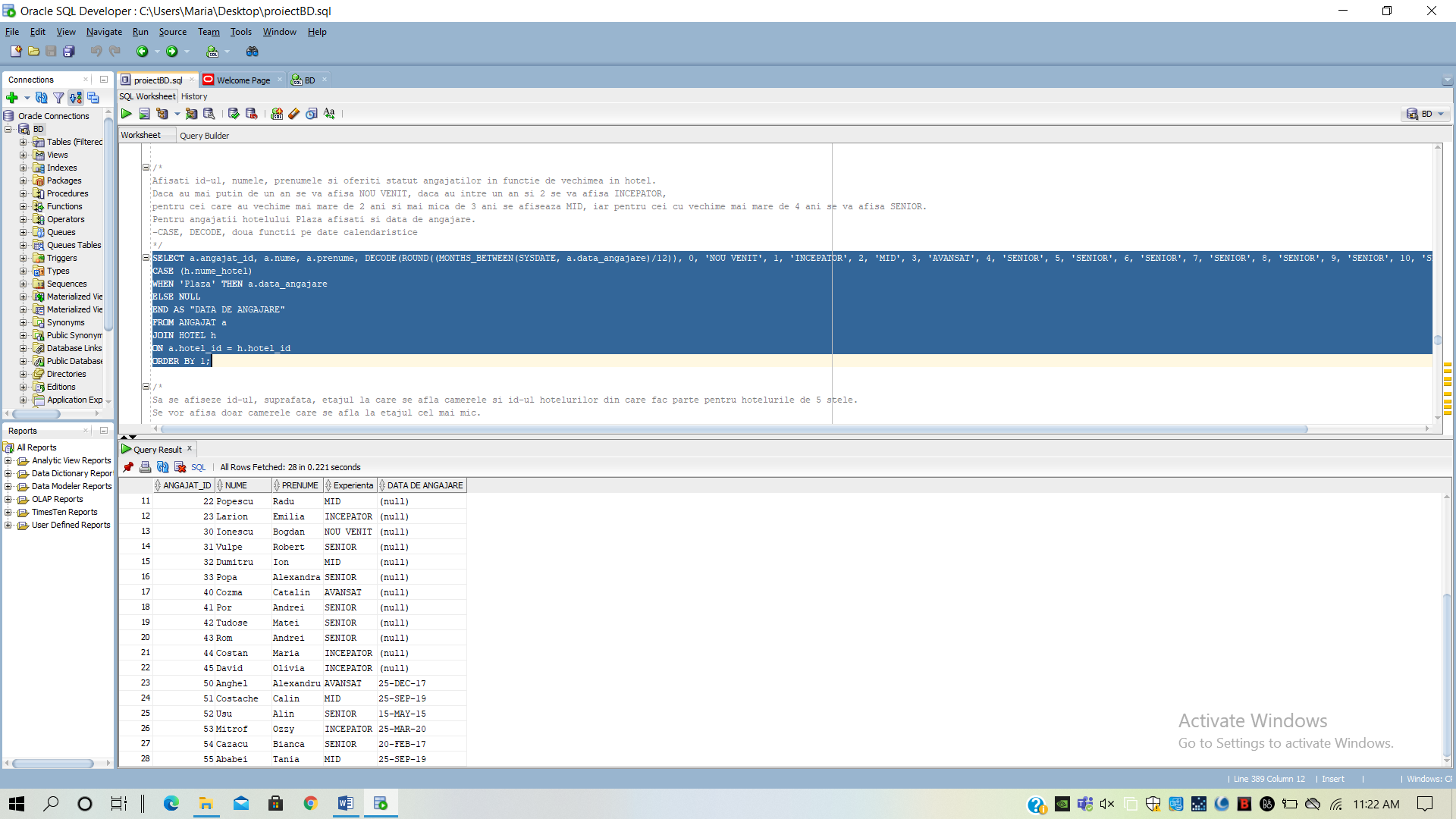
*• grupări de date, funcții grup, filtrare la nivel de grupuri*

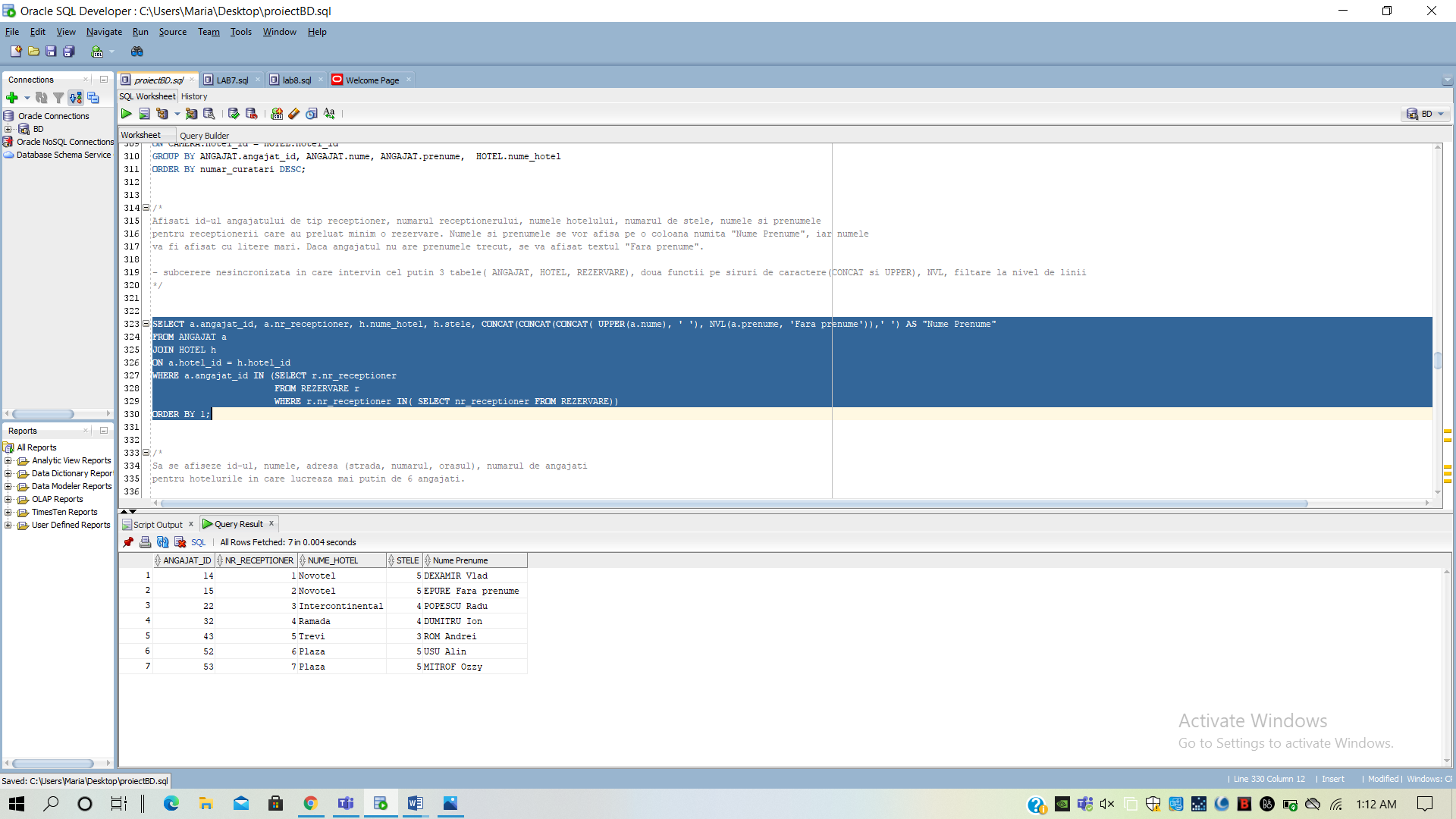
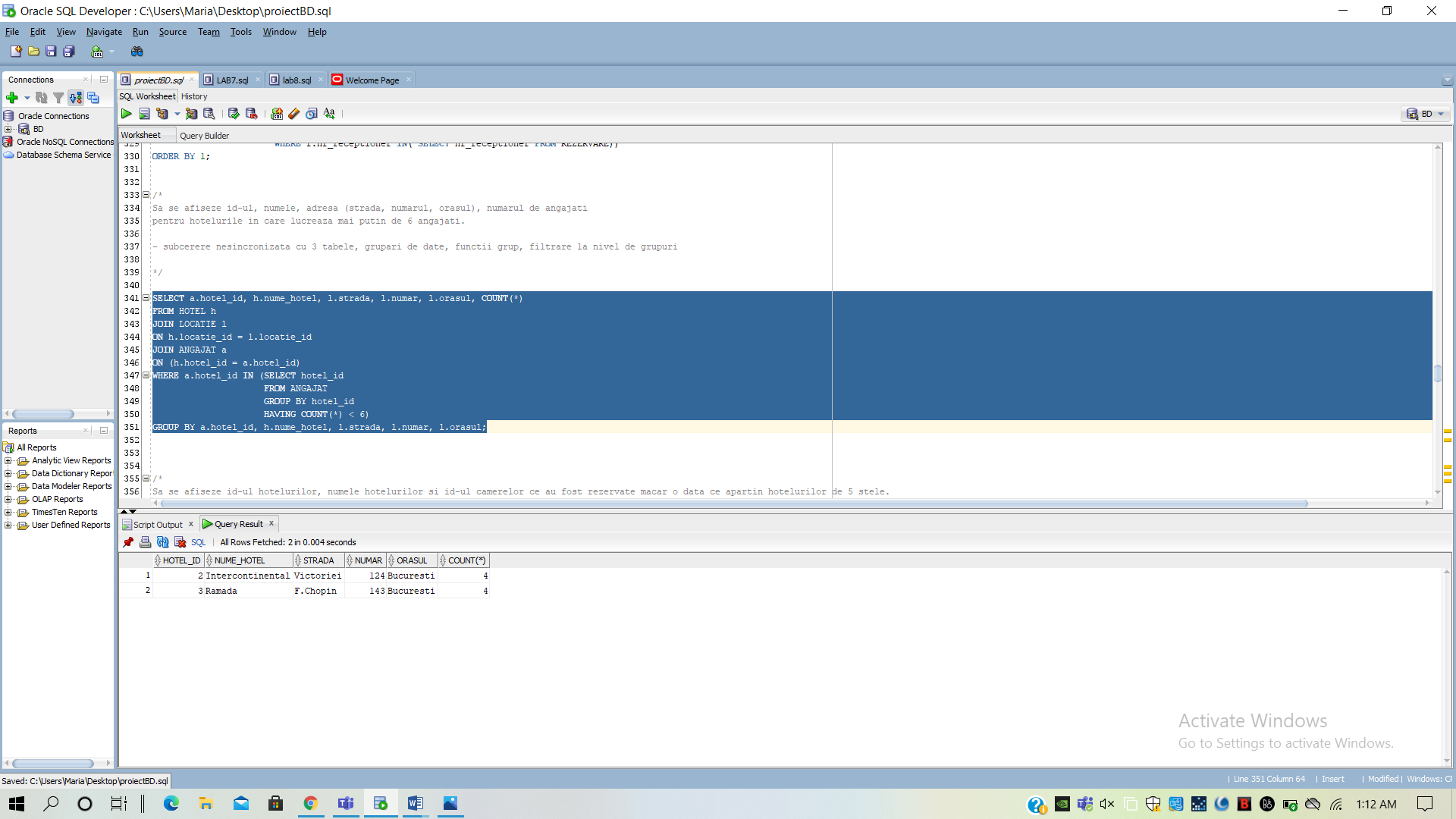
*• ordonări*

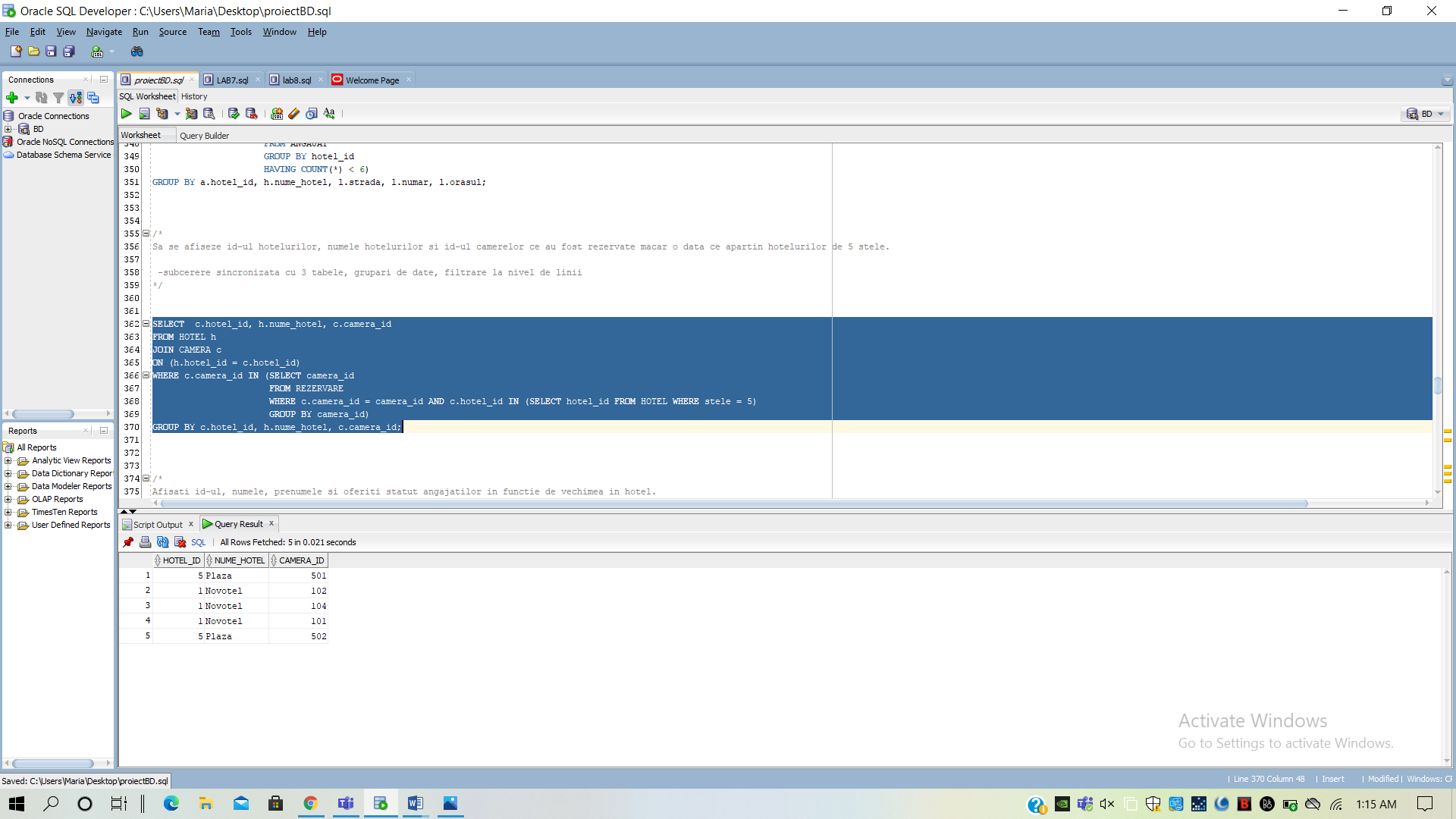
*• utilizarea a cel puțin 2 funcții pe șiruri de caractere, 2 funcții pe date calendaristice, a funcțiilor NVL și DECODE, a cel puțin unei expresii CASE*

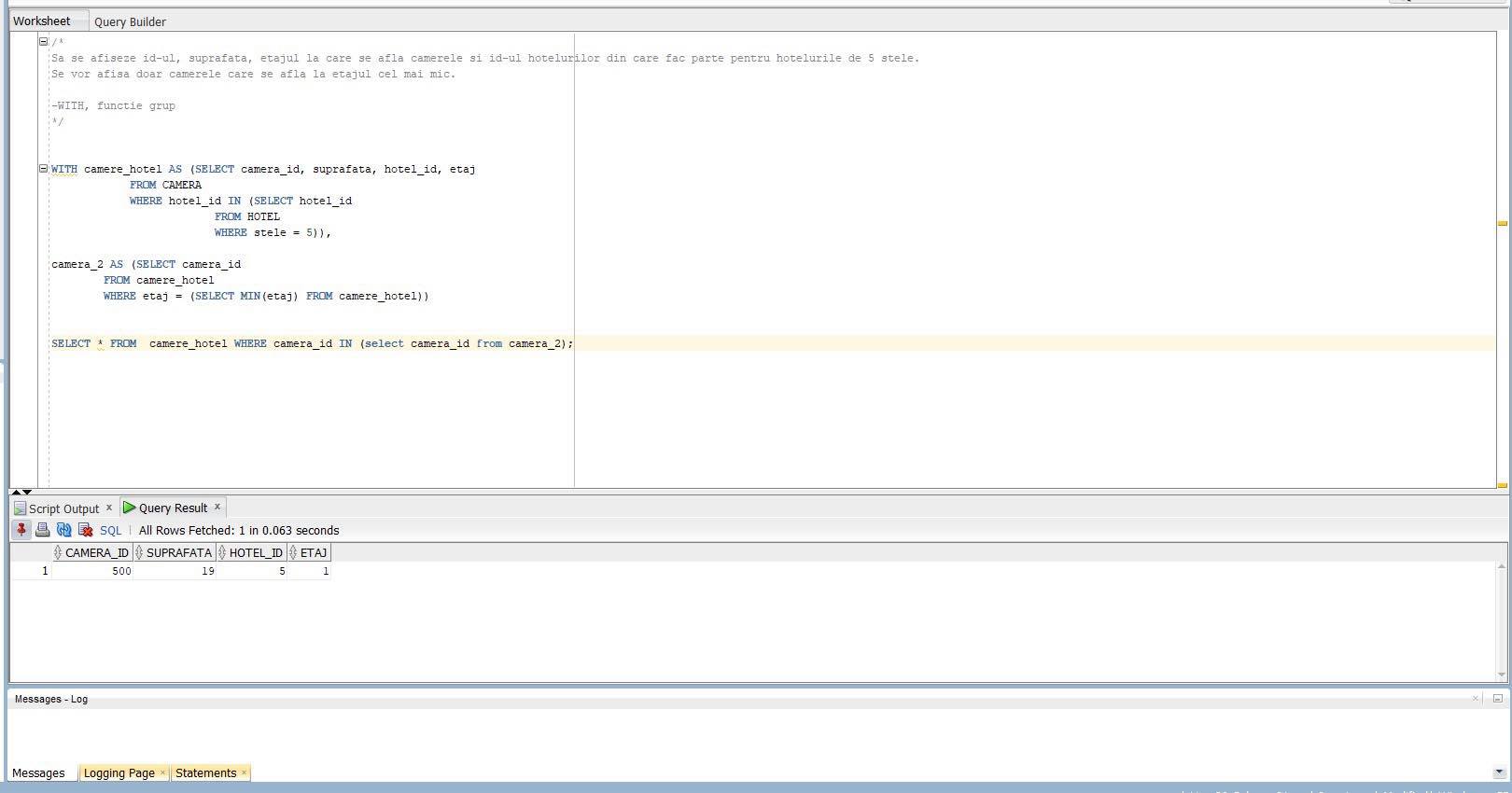
*• utilizarea a cel puțin 1 bloc de cerere (clauza WITH)*



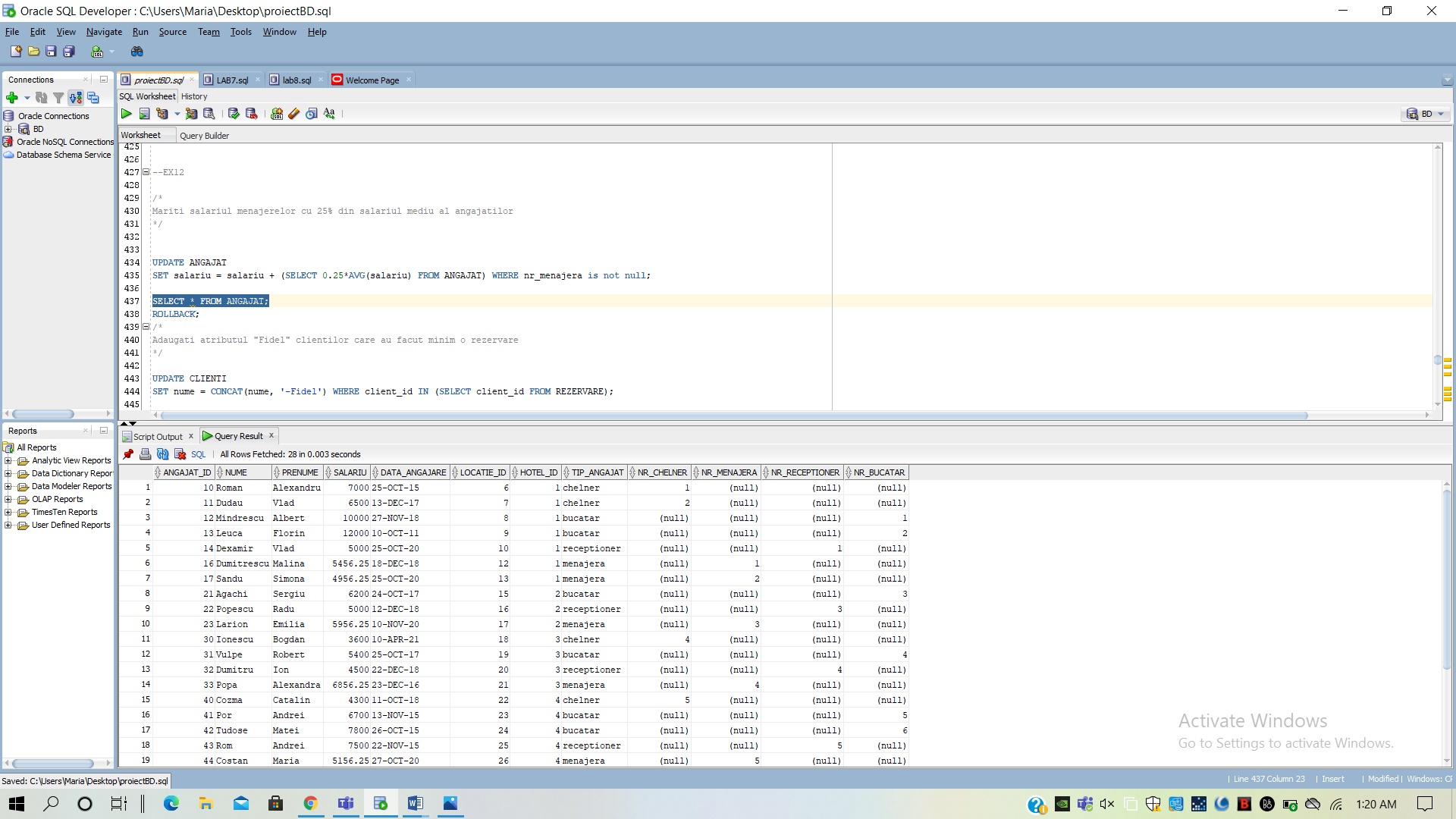


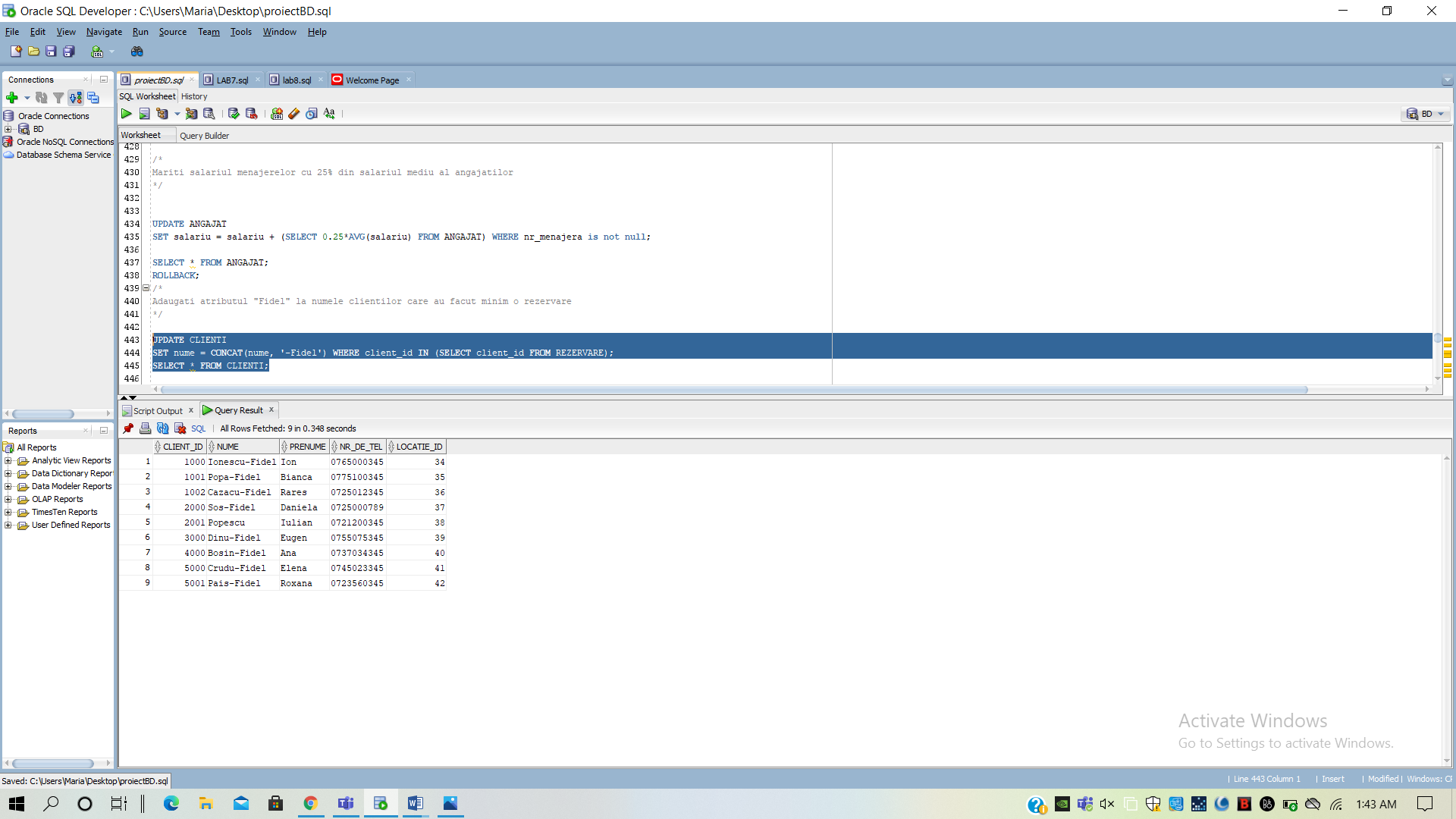
  


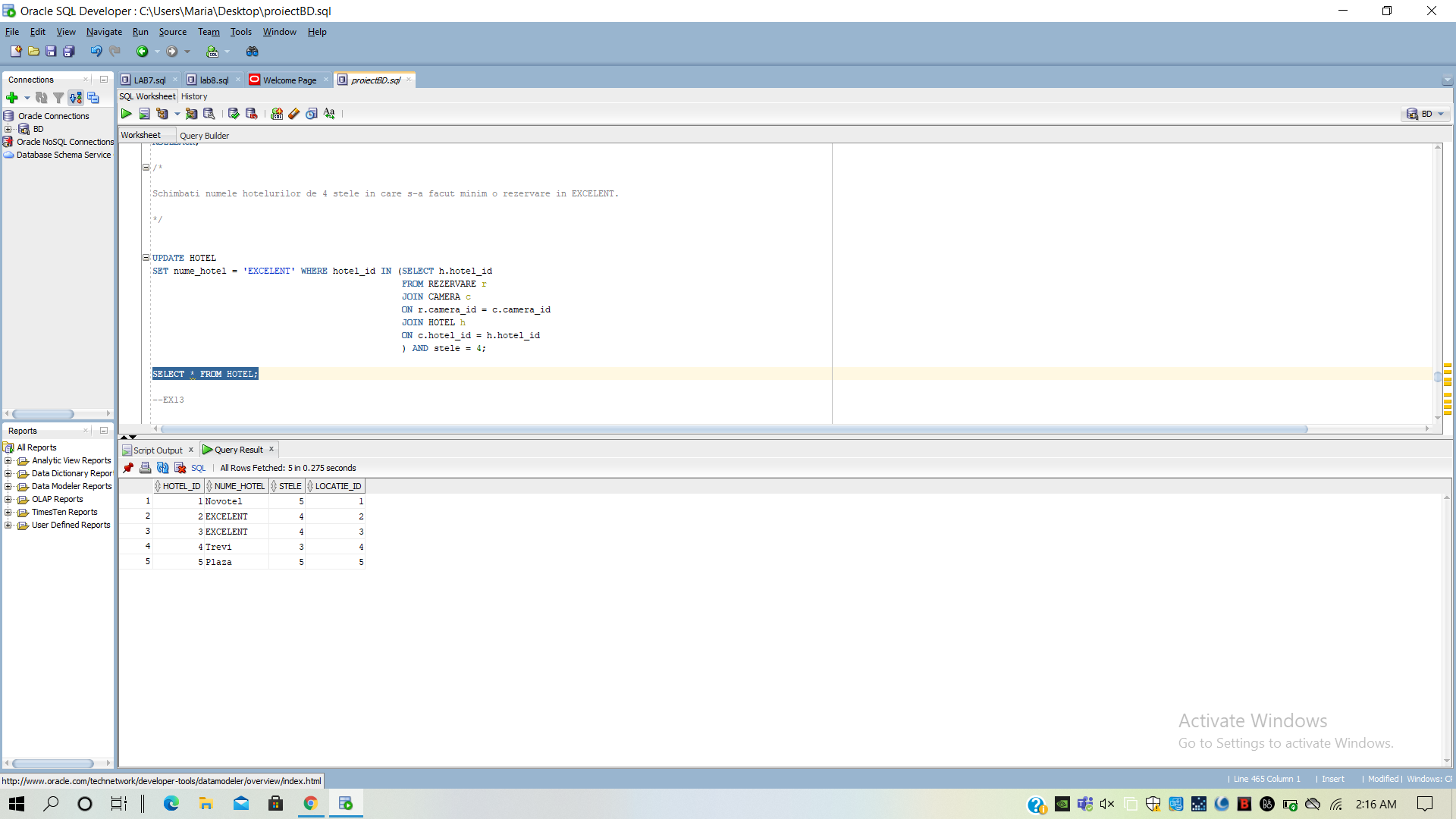




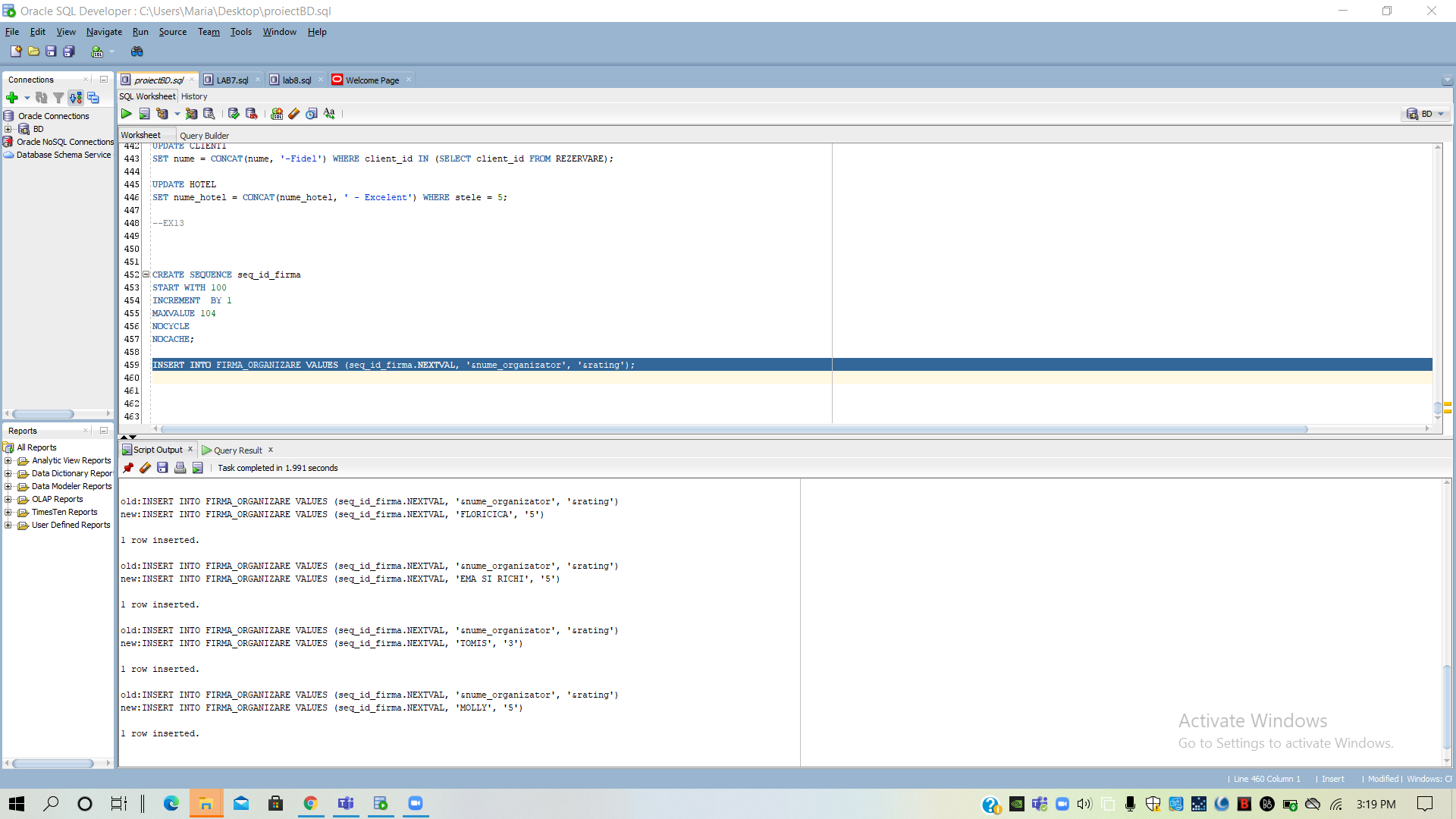
*12. Implementarea a 3 operații de actualizare sau suprimare a datelor utilizând subcereri.*

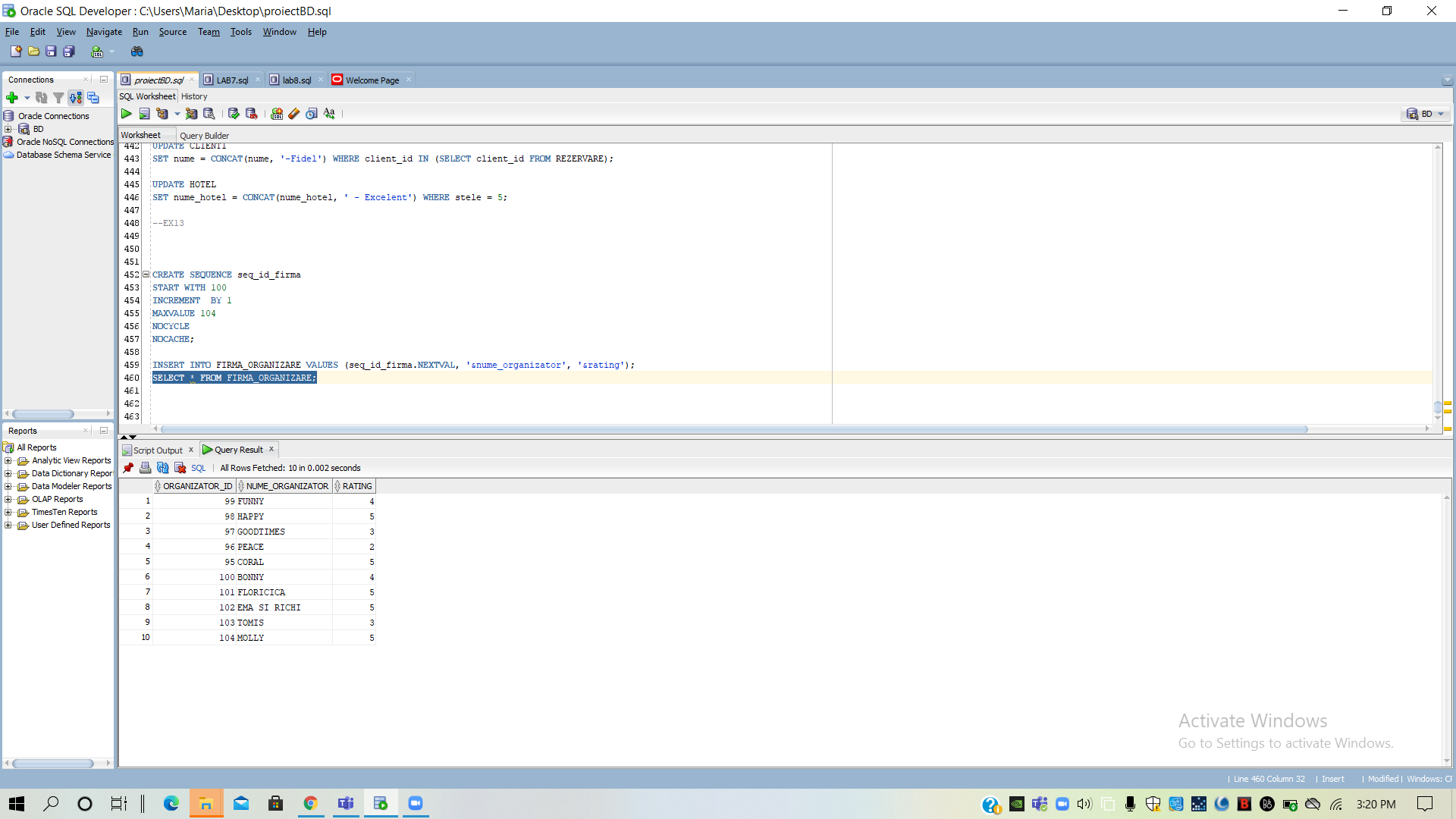


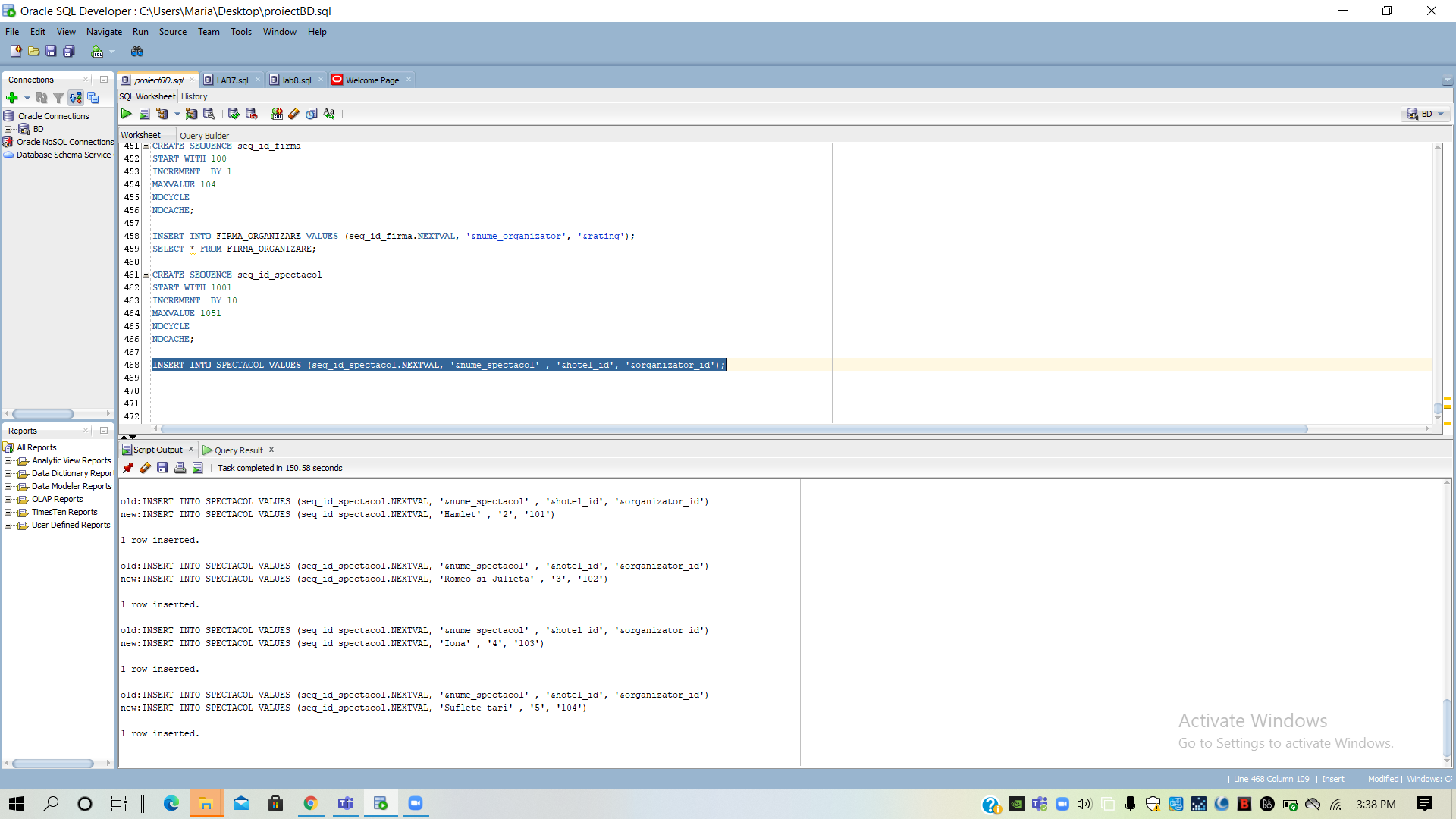


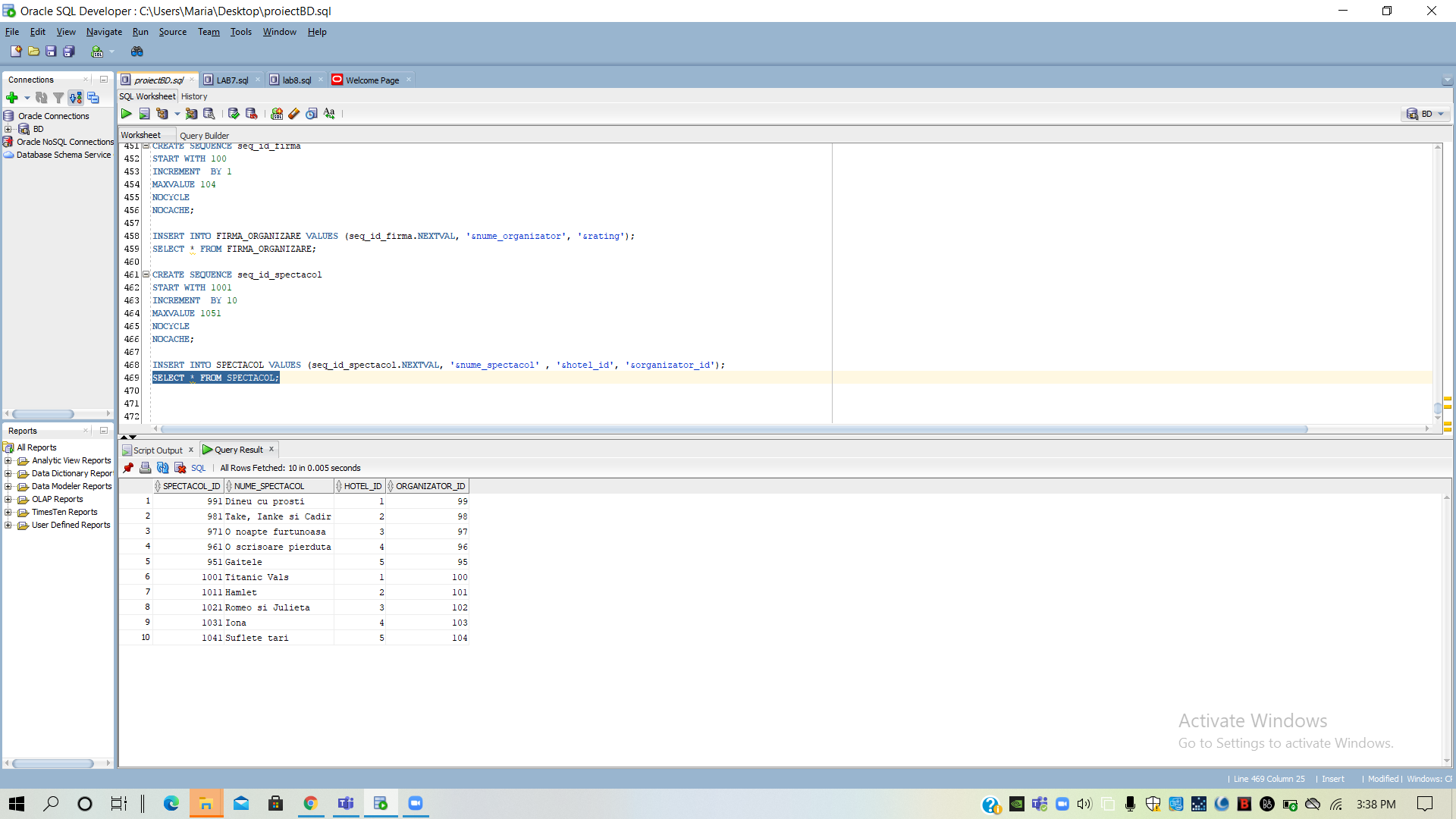


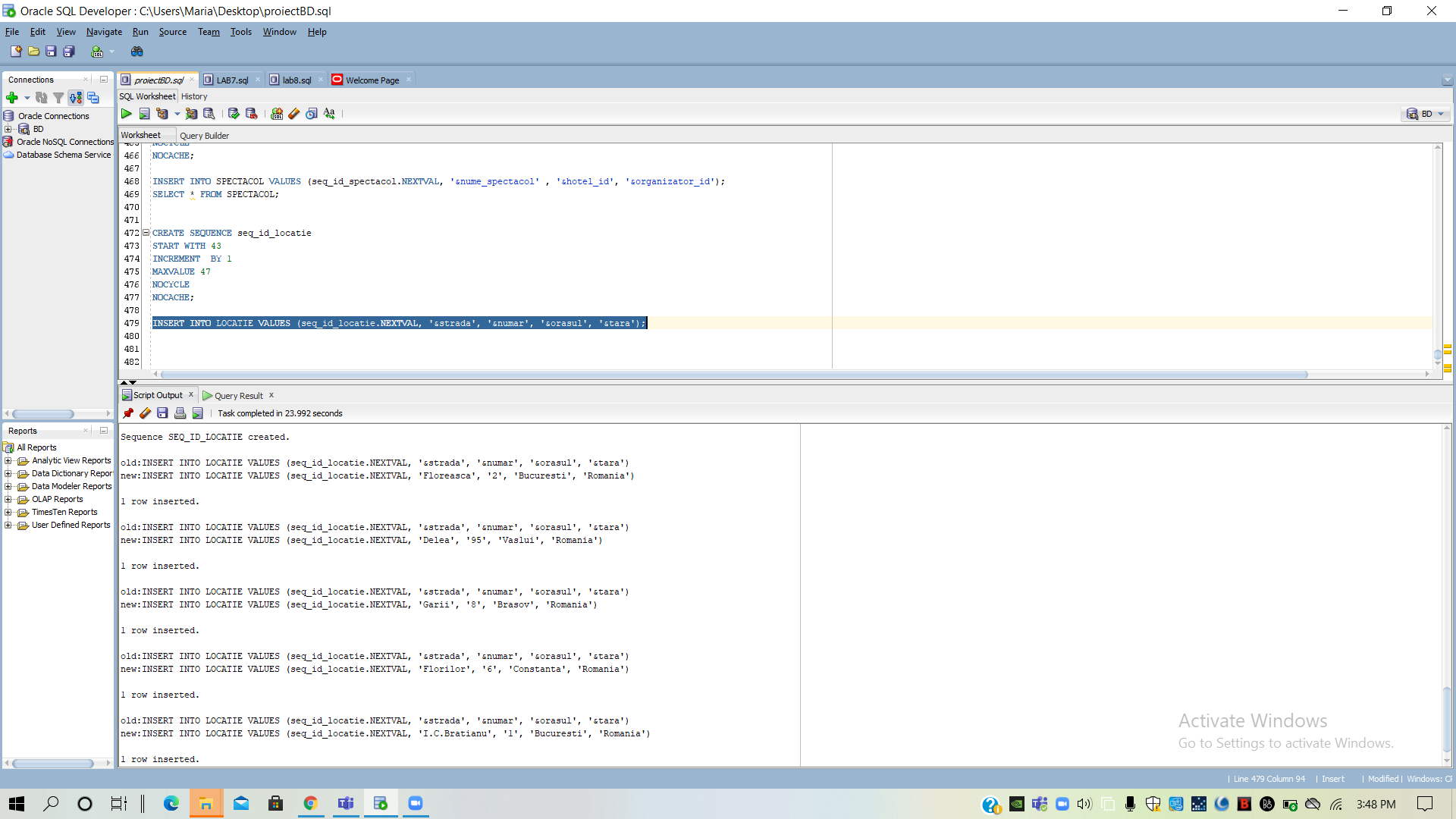
*13. Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 10).*

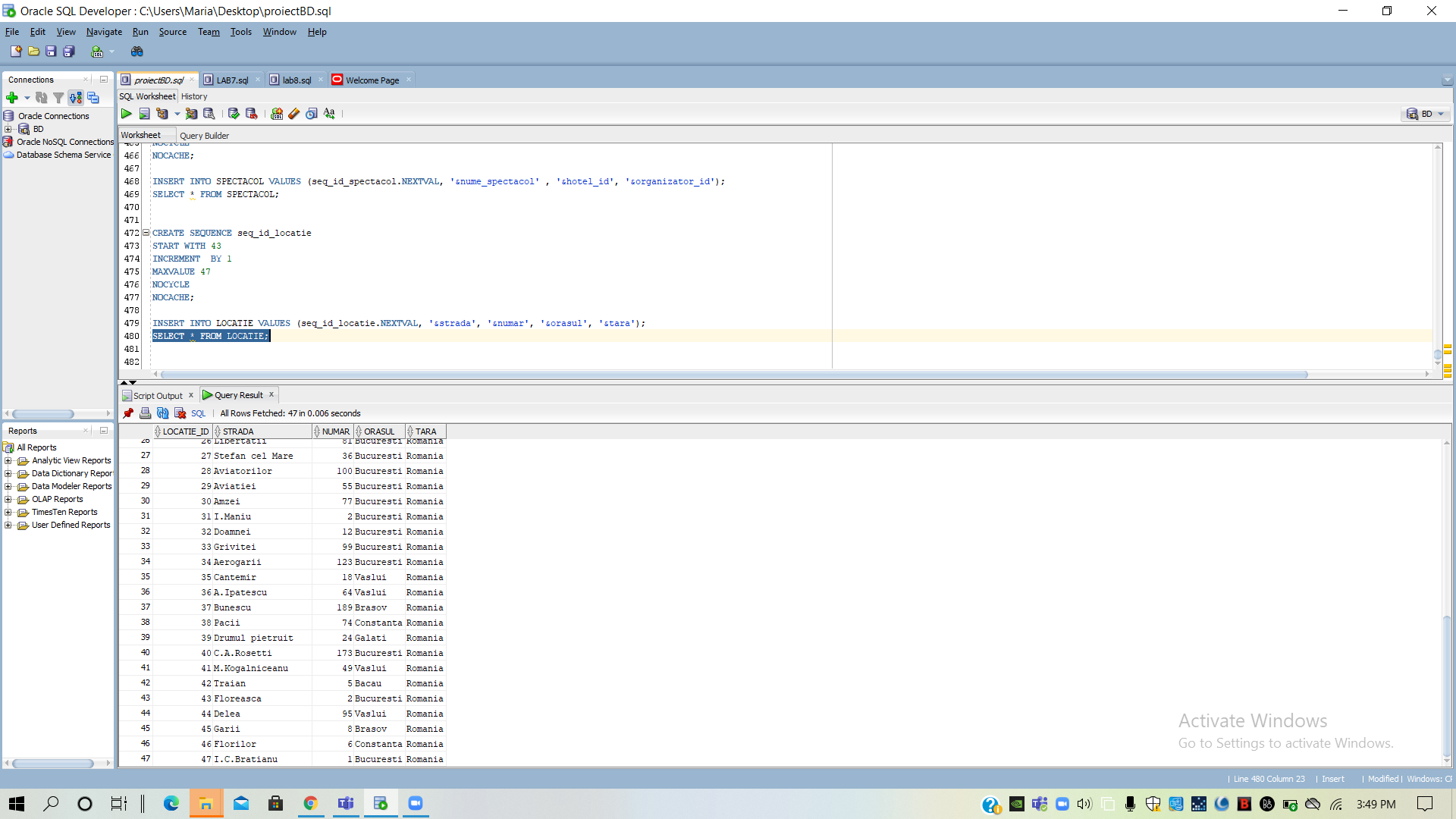












*16. Formulați în limbaj natural și implementați în SQL: o cerere ce utilizează operația outerjoin pe minimum 4 tabele și două cereri ce utilizează operația division.*

