PROIECT BAZE DE DATE

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

Compania ”InterRails” oferă servicii de transport feroviar cetățenilor Uniunii Europene. Această companie deține un număr de trenuri care fac legătura între diferite orașe europene pe liniile de cale ferată deja existente între acestea.

Pentru fiecare tren al companiei se cunoaște numărul său, anul fabricației, numele firmei producătoare, data achiziționării și data ultimei revizii .

Călătorii se înregistrează cu datele personale pe platforma online a companiei, de unde achiziționează bilete. Călătorii pot dispune de diferite reduceri pentru biletele de tren. Un bilet include numele călătorului, numărul trenului, data și ora plecării, orașul de plecare, data și ora sosiri, orașul sosirii și prețul biletului respectiv (eventual și reducerea împreună cu prețul final). In plus, biletele pot include și opțiunea de meniu la vagonul de restaurant al trenului (va fi ales la cumpărarea biletului).

Aceasta companie are personal în tren (controlori, chelneri, bucătari), care asigură toate condițiile necesare unei călătorii cât mai plăcute pentru pasageri. Controlorii verifică biletele, iar chelnerii au rolul de a servi călătorilor meniurile pe care aceștia le aleg în timpul călătoriei și care sunt preparate cu iscusință de către bucătarii aleși ai companiei.

1. Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului.

* Trenurile circulă pe drumuri directe între orașele pe care le leagă, fără stații intermediare.
* Un conductor nu poate opera mai multe trenuri în aceeași zi.
* Un călător poate achiziționa bilete doar pentru el însuși.
* Pentru un anumit bilet nu se poate aplica mai mult de o reducere.
* Un controlor poate valida o singură dată un anumit bilet de la un călător.
* Un singur chelner poate servi la un moment dat un anumit meniu unui călător.
* Un bilet poate să cuprindă maximum un meniu la vagonul restaurant.
* Nu pot pleca două trenuri în același timp, din aceeași stație.

1. Descrierea entităților, incluzând precizarea cheii primare.

|  |  |  |
| --- | --- | --- |
| Entitate | Cheie primară | Observații |
| tren | nr\_tren | Informații specifice despre trenurile deținute de companie |
| oraș plecare | id\_oras | Informații generale despre orașele de plecare |
| oraș sosire | id\_oras | Informații generale despre orașele de sosire |
| călător | id\_calator | Informații despre călătorii înregistrați pe platforma online |
| bilet | id\_calator, data\_plecare, ora\_plecare | Detalii despre bilete |
| meniu | id\_meniu | Meniurile pe care călătorii le pot alege la vagonul restaurant |
| staff | id\_member | Informații despre membrii staff-ului din trenuri |
| controlor | id\_member | Sub-entitate a entității staff; informații despre controlori |
| chelner | id\_member | Sub-entitate a entității staff;  informații despre chelneri |
| bucătar | id\_member | Sub-entitate a entității staff;  informații despre bucătari |

1. Descrierea relațiilor, incluzând precizarea cardinalității acestora.

|  |  |  |
| --- | --- | --- |
| Relație | Cardinalitate | Observații |
| apare\_plecare | oraș – bilet one-to-many | Pe bilet apare un singur oraș de plecare |
| apare\_sosire | oraș – bilet one-to-many | Pe bilet apare un singur oraș de sosire |
| eliberat\_pentru | tren – bilet one-to-many | Biletele sunt eliberate pentru anumite trenuri |
| lucreaza\_in | staff – tren many-to-many | Persoanele din staff pot să lucreze în mai multe trenuri, iar în fiecare tren lucrează mai multe persoane |
| cumpara | călător – bilet one-to-many | Fiecare călător poate cumpăra mai multe bilete, însă un bilet aparține unui singur călător |
| contine | meniu – bilet one-to-many | Un bilet poate să includă maxim un meniu (sau niciunul). Meniurile pot apărea pe mai multe bilete |
| controlează | controlor – bilet many-to-many | Un controlor poate să controleze mai multe bilete, iar un bilet se poate să fie controlat de mai multe ori, de controlori diferiți. |
| gateste | bucătar – meniu many-to-many | Un bucătar poate să gătească mai multe meniuri, iar un meniu poate fi gătit de mai mulți bucătari |
| serveste | chelner – (călător, meniu) one-to-one  călător – (chelner, meniu)  one-to-one  meniu – (călător, chelner)  one-to-many | Un chelner poate servi mai multe meniuri aceluiași client. |
| ISA | chelner – staff one-to-one | Relație de tip ISA între super-entitatea staff și sub-entitatea chelner |
| ISA | controlor – staff one-to-one | Relație de tip ISA între super-entitatea staff și sub-entitatea controlor |
| ISA | bucătar – staff one-to-one | Relație de tip ISA între super-entitatea staff și sub-entitatea bucătar |
| legate | oraș plecare – oraș sosire many-to-many | Simbolizează liniile de cale ferată existente între orașe |

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

ENTITATE: ORAȘ PLECARE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atribut | Tip | Dimensiune | Valori posibile și valori default | Observații  Obligatoriu/Opțional |
| id\_oras | string | 8 (maxim) |  | PK - obligatoriu |
| nume\_tara | string | 40 (maxim) |  | obligatoriu |
| nume | string | 40 (maxim) |  | obligatoriu |

ENTITATE: ORAȘ SOSIRE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atribut | Tip | Dimensiune | Valori posibile și valori default | Observații  Obligatoriu/Opțional |
| id\_oras | string | 8 (maxim) |  | PK - obligatoriu |
| nume\_tara | string | 40 (maxim) |  | obligatoriu |
| nume | string | 40 (maxim) |  | obligatoriu |

ENTITATE: TREN

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atribut | Tip | Dimensiune | Valori posibile și valori default | Observații  Obligatoriu/Opțional |
| nr\_tren | integer | 6 (maxim) |  | PK - obligatoriu |
| an\_fabricatie | integer | 4 | [2010, 2022], default 2010 |  |
| data\_achizitie | date |  | data curenta |  |
| firma | string | 30 (maxim) |  | obligatoriu |
| ultima\_revizie | date |  |  | Poate să fie NULL în cazul în care nu a fost făcută încă o revizie a trenului |
| nr\_vagoane | integer | 2 (maxim) | [1, 10], default 1 |  |

ENTITATE: CĂLĂTOR

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atribut | Tip | Dimensiune | Valori posibile și valori default | Observații  Obligatoriu/Opțional |
| id\_calator | integer | 8 (maxim) |  | PK - obligatoriu |
| nume\_complet | string | 100 (maxim) |  | Atribut multiplu; obligatoriu |
| varsta | integer | 3 (maxim) |  | obligatoriu |
| adrese email | string | 100 (maxim) |  | Atribut multiplu; un client poate avea maxim 2 adrese de email pentru un cont (separate prin ,) – cel puțin una obligatoriu |
| telefon | string | 20 (maxim) |  | Un călător poate avea cel mult un număr de telefon asociat - obligatoriu |
| gen | char | 1 | M / F | Poate fi NULL (în acest caz se presupune că persoana nu dorește sa menționeze acest detaliu) |

ENTITATE: BILET

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atribut | Tip | Dimensiune | Valori posibile și valori default | Observații  Obligatoriu/Opțional |
| id\_calator | integer | 8 (maxim) |  | Corespunde cu id\_calator din CĂLĂTOR – face parte din PK (obligatoriu) |
| ora\_plecare | string | 5 |  | Ora o să fie dată în formatul „HH:MM” (ceas digital cu 24 de ore) – face parte din PK (obligatoriu) |
| ora\_sosire | string | 5 |  | Ora o să fie dată în formatul „HH:MM” (ceas digital cu 24 de ore) |
| data\_plecare | date |  |  | face parte din PK (obligatoriu) |
| data\_sosire | date |  |  | obligatoriu |
| pret\_initial | float | 3 – parte întreagă  2 – parte fracționară |  | obligatoriu |
| tip\_reducere | string | 20 (maxim) |  | tipul reducerii care este aplicată biletului (poate fi NULL în cazul în care biletul nu este redus) |
| valoare\_redusa | float | 1 – parte intregă  2 – parte fracționară |  | Valoare reală de forma 0.ab, reprezentând cat o să se reducă din valoarea biletului |
| pret\_final | float | 3 – parte întreagă  2 – parte fracționară |  | prețul final (după o eventuală reducere aplicată) - obligatoriu |
| nr\_tren | integer | 6 (maxim) |  | corespunde cu nr\_tren din TREN - obligatoriu |
| nume\_calator | string | 100 (maxim) |  | obligatoriu |
| denumire\_meniu | string | 50 (maxim) |  | denumirea meniului ales de către călător în momentul cumpărării biletului (poate fi NULL daca nu a ales nimic); corespunde cu denumire din MENIU |
| oras\_plecare | string | 8 (maxim) |  | obligatoriu – corespunde cu id\_oras din ORAS PLECARE |
| oras\_sosire | string | 8 (maxim) |  | obligatoriu – corespunde cu id\_oras din ORAS SOSIRE |

ENTITATE: MENIU

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atribut | Tip | Dimensiune | Valori posibile și valori default | Observații  Obligatoriu/Opțional |
| denumire | string | 50 (maxim) |  | PK - obligatoriu |
| rating | integer | 1 | [1, 5], default 1 | cât de popular este acest meniu |

ENTITATE: STAFF

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atribut | Tip | Dimensiune | Valori posibile și valori default | Observații  Obligatoriu/Opțional |
| id\_membru | integer | 3 (maxim) |  | PK - obligatoriu |
| nume\_complet | string | 100 (maxim) |  | obligatoriu |
| data\_angajare | date |  | data curenta |  |
| salariu | integer | 5 (maxim) |  | obligatoriu |

ENTITATE: CONTROLOR

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atribut | Tip | Dimensiune | Valori posibile și valori default | Observații  Obligatoriu/Opțional |
| id\_membru | integer | 3 (maxim) |  | PK - obligatoriu |
| politete | integer | 1 | [1, 5], default 1 | calificativ acordat în funcție de politețea fiecărui controlor |

ENTITATE: BUCĂTAR

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atribut | Tip | Dimensiune | Valori posibile și valori default | Observații  Obligatoriu/Opțional |
| id\_membru | integer | 3 (maxim) |  | PK - obligatoriu |
| stele | integer | 1 | [1, 5], default 1 | numărul de stele acordat de comisia internațională de bucătari |

ENTITATE: CHELNER

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atribut | Tip | Dimensiune | Valori posibile și valori default | Observații  Obligatoriu/Opțional |
| id\_membru | integer | 3 (maxim) |  | PK - obligatoriu |
| rating | integer | 1 | [1, 5], default 1 | calificativ general pentru chelner |

RELAȚIE: LUCREAZĂ ÎN

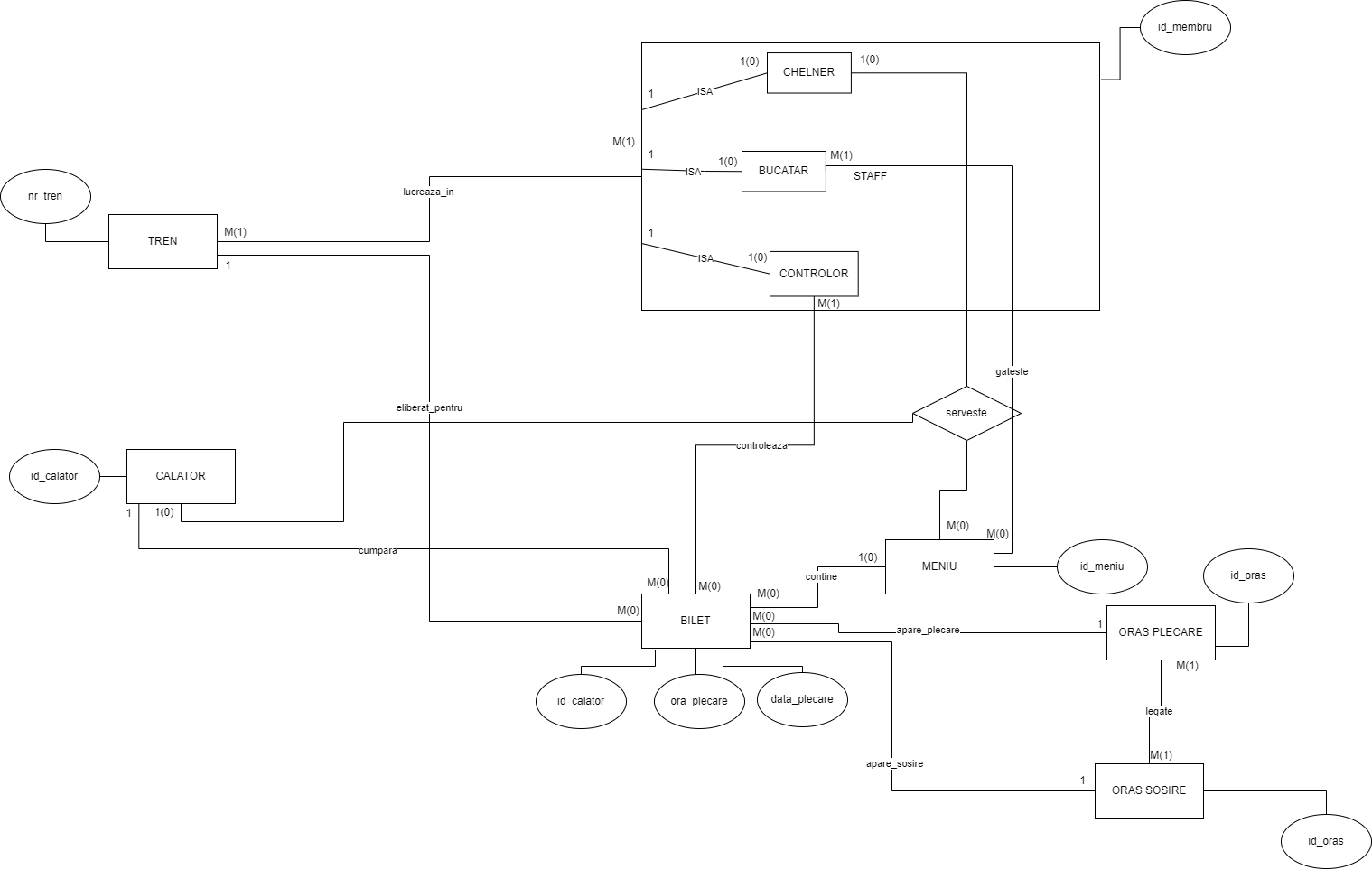
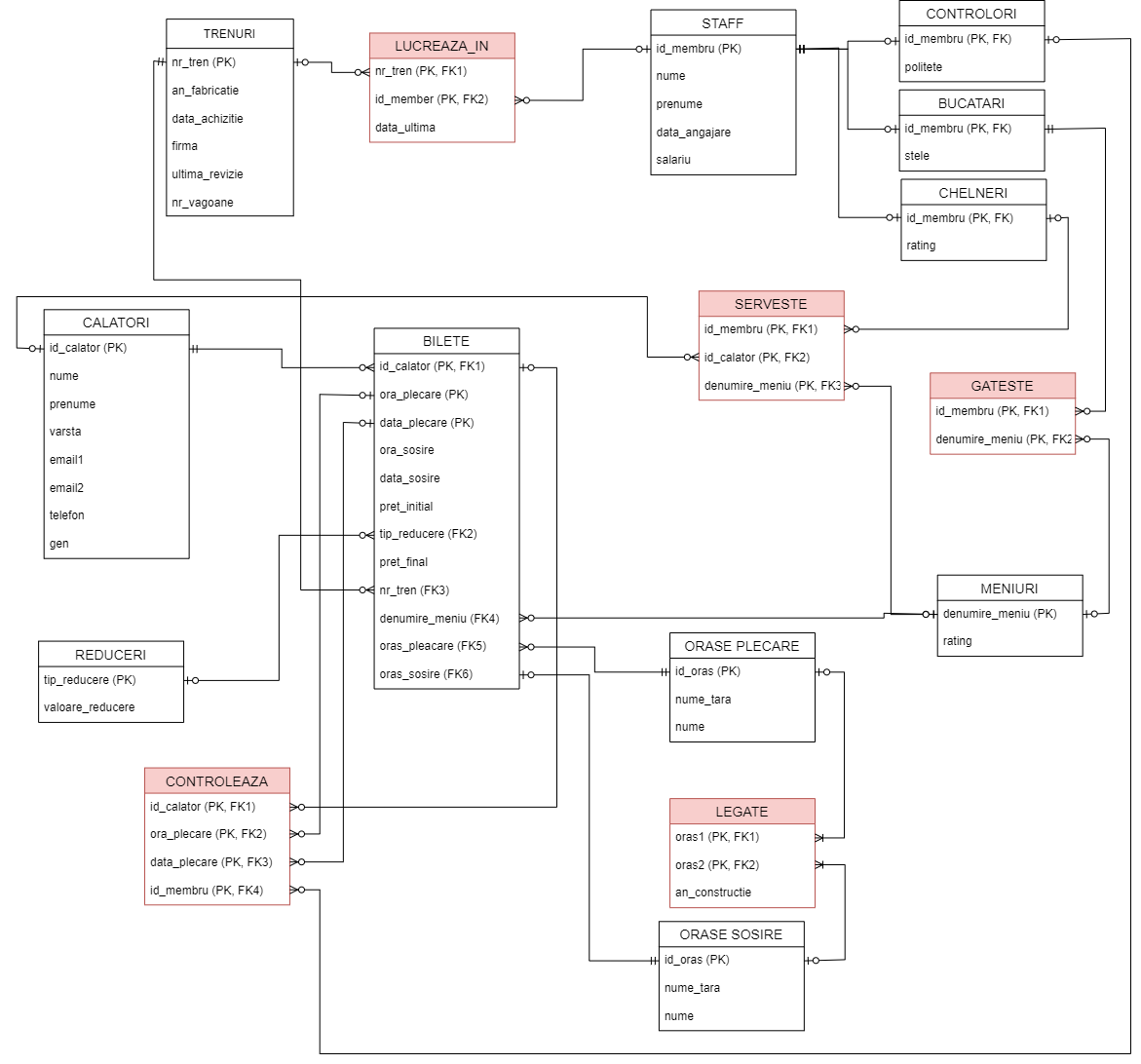
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atribut | Tip | Dimensiune | Valori posibile și valori default | Observații  Obligatoriu/Opțional |
| data\_ultima | date |  | data curenta | ultima dată când persoana respectivă a lucrat în acel tren |

RELAȚIE: SERVESTE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atribut | Tip | Dimensiune | Valori posibile și valori default | Observații  Obligatoriu/Opțional |
| nota\_client | integer | 1 | [1, 10], default 1 | nota acordată de către călător per total (atât pentru servire, cât și pentru meniu) |

RELAȚIE: LEGATE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Atribut | Tip | Dimensiune | Valori posibile și valori default | Observații  Obligatoriu/Opțional |
| an\_constructie | integer | 4 | [1970, 2022], default 1970 | anul înființării liniei de cale ferată |

1. Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5.
2. Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6. Diagrama conceptuală obținută trebuie să conțină minimum 6 tabele (fără considerarea subentităților), dintre care cel puțin un tabel asociativ.
3. Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale proiectate la punctul 7.

* CALATORI(id\_calator#, nume, prenume, varsta, email1, email2, telefon, gen)
* TRENURI(nr\_tren#, an\_fabricatie, data\_achizitie, firma, ultima\_revizie, nr\_vagoane)
* STAFF(id\_member#, nume, prenume, data\_angajare, salariu)
* CONTROLORI(id\_member#, politete)
* CHELNERI(id\_member#, rating)
* BUCATARI(id\_member#, stele)
* MENIURI(denumire\_meniu#, rating)
* ORASE PLECARE(id\_oras#, nume, nume\_tara)
* ORASE SOSIRE(id\_oras#, nume, nume\_tara)
* BILETE(id\_calator#, ora\_plecare#, data\_plecare#, ora\_sosire, data\_sosire, pret\_initial, tip\_reducere, pret\_final, nr\_tren, denumire\_meniu, oras\_plecare, oras\_sosire)
* REDUCERI(tip\_reducere#, valoare\_reducere)
* LUCREAZA\_IN(nr\_tren#, id\_member#, data\_ultima)
* SERVESTE(id\_member#, id\_calator#, denumire\_meniu, nota\_client)
* GATESTE(id\_member#, denumire\_meniu#)
* LEGATE(oras\_plecare#, oras\_sosire#, an\_constructie)
* CONTROLEAZA(id\_calator#, ora\_plecare#, data\_plecare#, id\_membru#)

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

**Aducerea la FN1:**

Analizez toate tabelele si observ ca am atribute multiple în tabelele: călători (nume\_complet și adrese\_email), bilete (nume\_călător), staff (nume\_complet).

În acest sens, o sa despart în fiecare tabel atributele multiple cu atribute atomice, pentru a respecta regulile pentru FN1.

Procedez în felul următor:

* Atributul **nume\_complet** din tabelul călători o să se înlocuiască cu atributele **nume** si **prenume**
* Atributul **adrese\_email** din tabelul călători o să se înlocuiască cu atributele **email1** și **email2** (din moment ce este permis călătorilor sa asocieze cel mult doua adrese de email cu contul lor de pe site)
* Atributul **nume\_călător** (ce semnifică numele complet al călătorului) din tabelul bilete o să se înlocuiască cu atributele **nume\_călător** și **prenume\_călător**
* Atributul **nume\_complet** din tabelul staff o să se înlocuiască cu atributele **nume** și **prenume**

În acest moment toate tabelele conțin atribute atomice, deci baza de date se află în FN1.

**Aducerea la FN2:**

Analizez din nou toate tabelele, de data aceasta pentru a găsi atribute care depind doar de o parte din cheia primară.

Observ că în tabelul bilete apar atributele **nume\_călător** și **prenume\_călător**, care depind doar de coloana **id\_călător** (coloană care face parte din cheia primară a tabelului bilete).

Pentru a rezolva problema ar trebui sa creez un nou tabel cu cheia primară **id\_călător** și să pun atributele **nume\_călător** și **prenume\_călător** în acest tabel, eliminându-le apoi din tabelul bilete.

Din moment ce eu am deja tabelul călători care include coloanele **nume** și **prenume** nu mai este necesar să creez un nou tabel. Mai rămâne doar să elimin atributele **nume\_călător** și **prenume\_călător** din tabelul bilete.

După această prelucrare, în baza mea de date nu mai există dependențe ale unor atribute care nu fac parte din cheia primară de o parte din aceasta, fapt ce înseamnă că baza de date se află în FN2.

**Aducerea la FN3:**

Analizez încă o dată tabelele bazei mele de date. De această dată, caut dependențe tranzitive de forma pk -> non-pk -> non-pk.

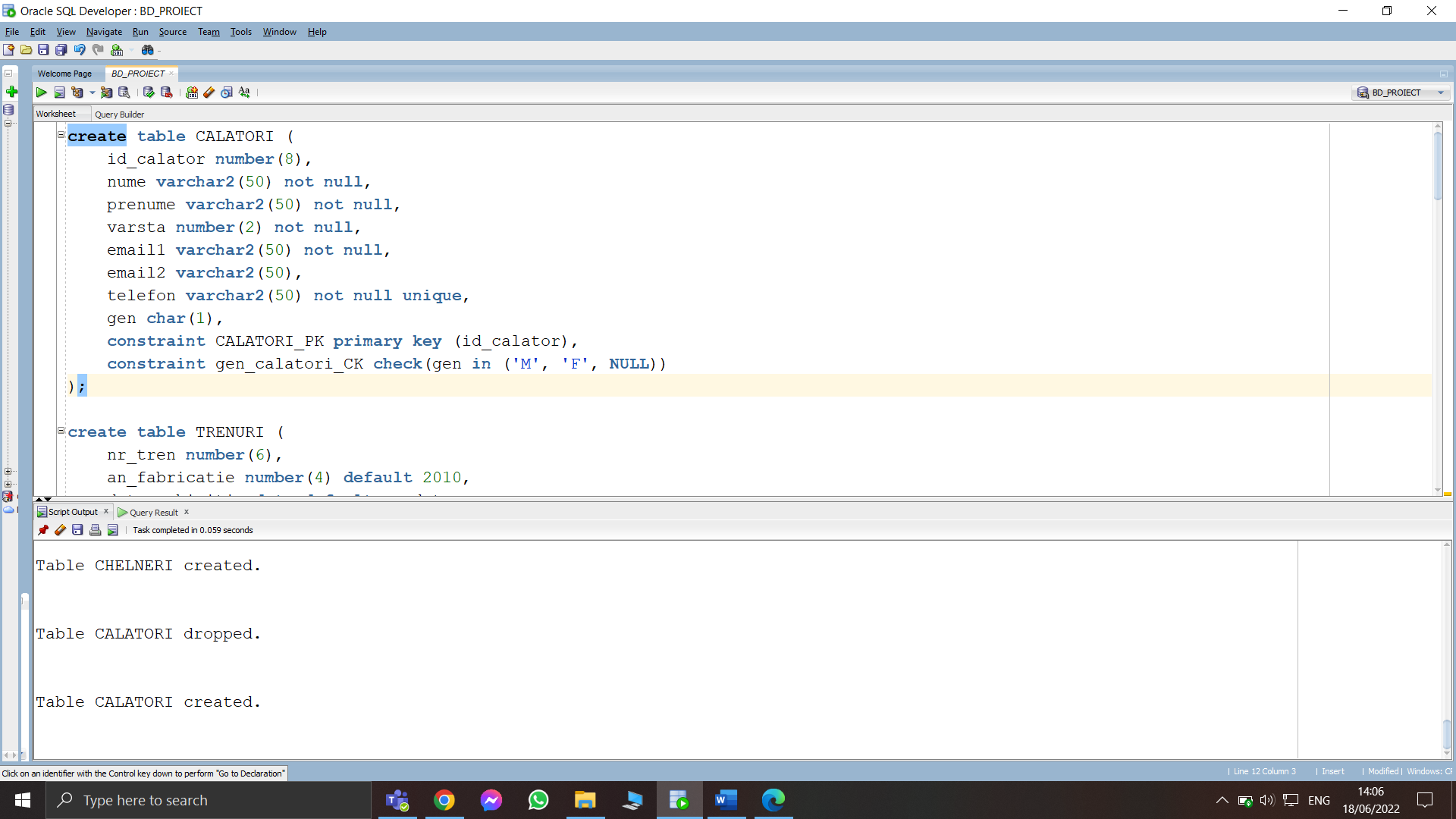
Observ că în tabelul bilete am coloana valoare\_reducere, care depinde de coloana tip\_reducere. Aceasta este o dependență tranzitivă ce trebuie eliminată.

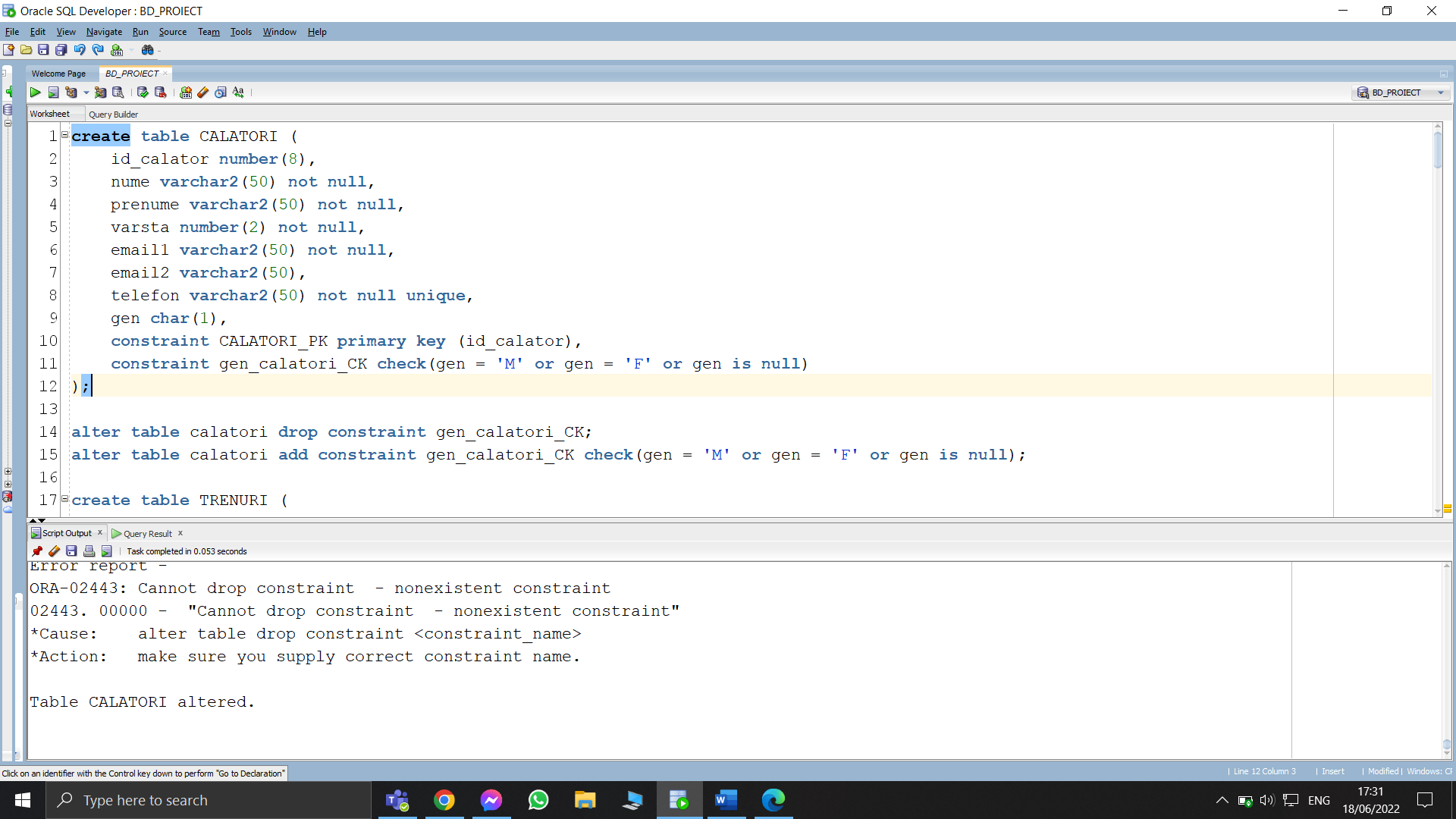
Pentru a elimina dependența tranzitivă este nevoie sa creez un nou tabel pe care o sa îl numesc REDUCERI, în care o să am drept cheie primară **tip\_reducere** (cu aceleași detalii ca în tabelul bilete). Pe lângă această coloană voi mai adăuga una, **valoare\_redusă**, ce reprezintă procentul redus din prețul inițial al biletului (va conține numere de tip float sub forma 0.ab). În urma acestei modificări o sa pot șterge coloana **valoare\_redusă** din tabelul bilete fără pierdere de informație.

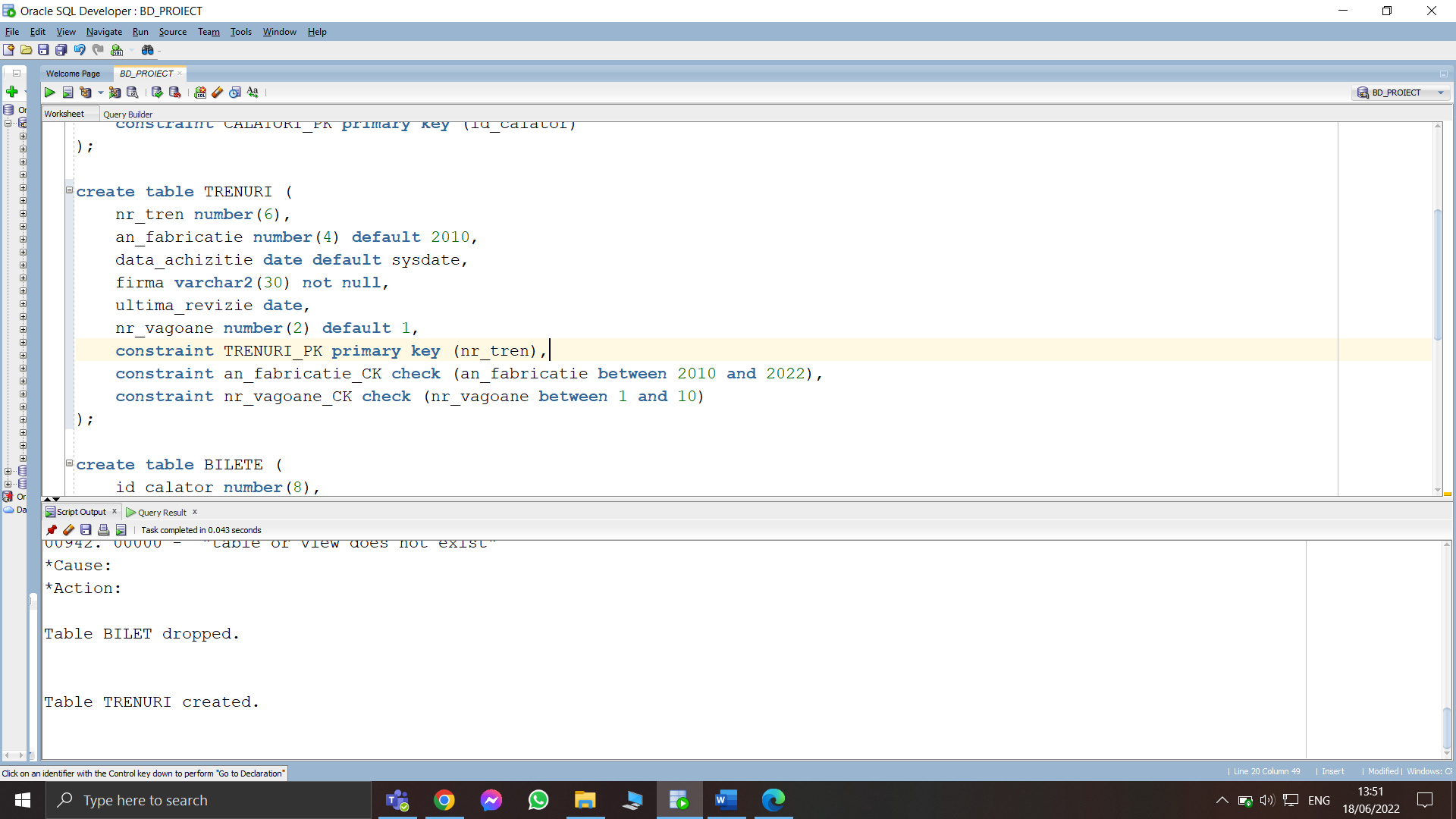
Cum alte dependențe tranzitive nu mai sunt, în momentul de față baza mea de date se află in FN3.

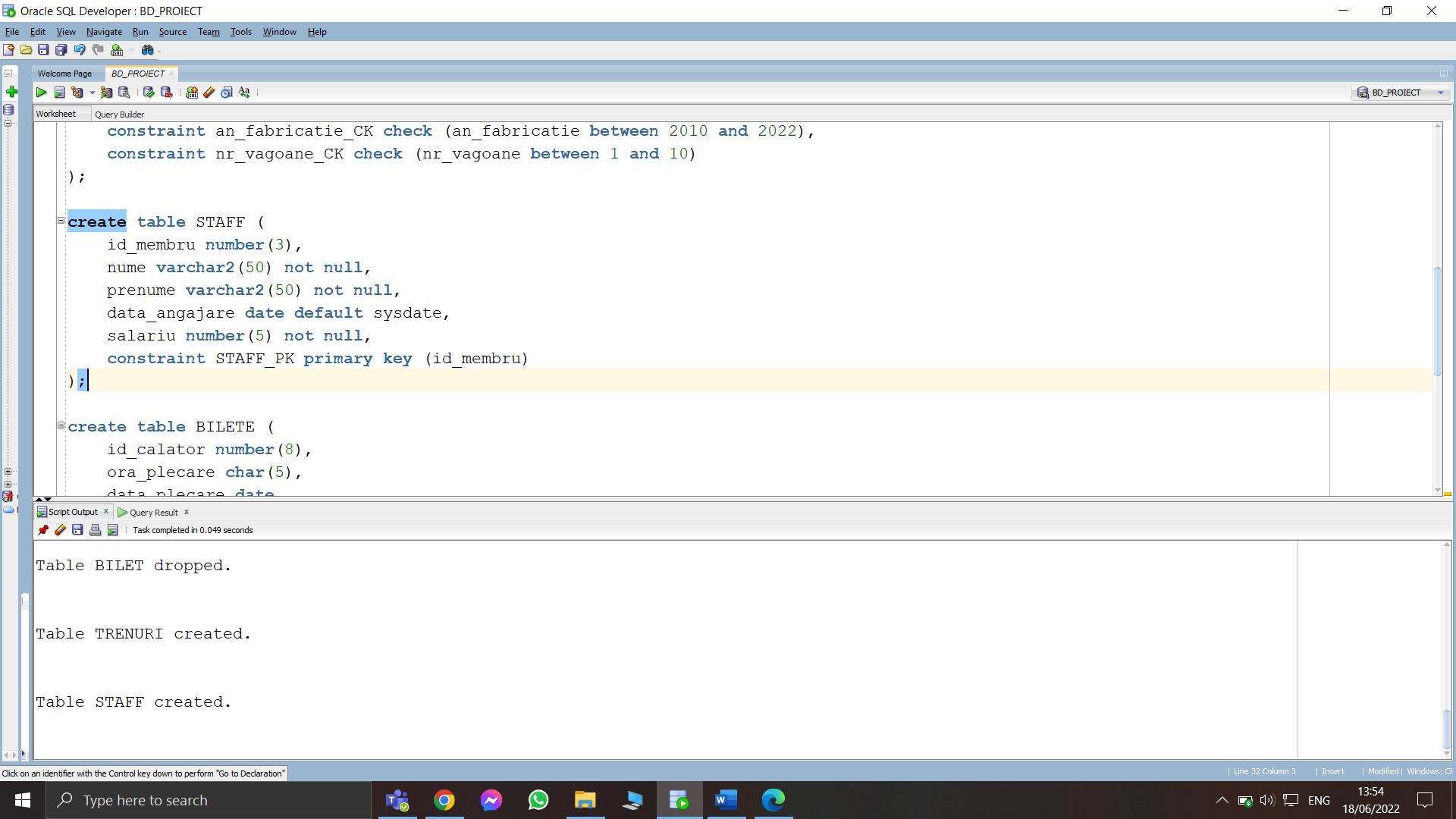
1. Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea (minimum 5 înregistrări în fiecare tabel neasociativ; minimum 10 înregistrări în tabelele asociative).

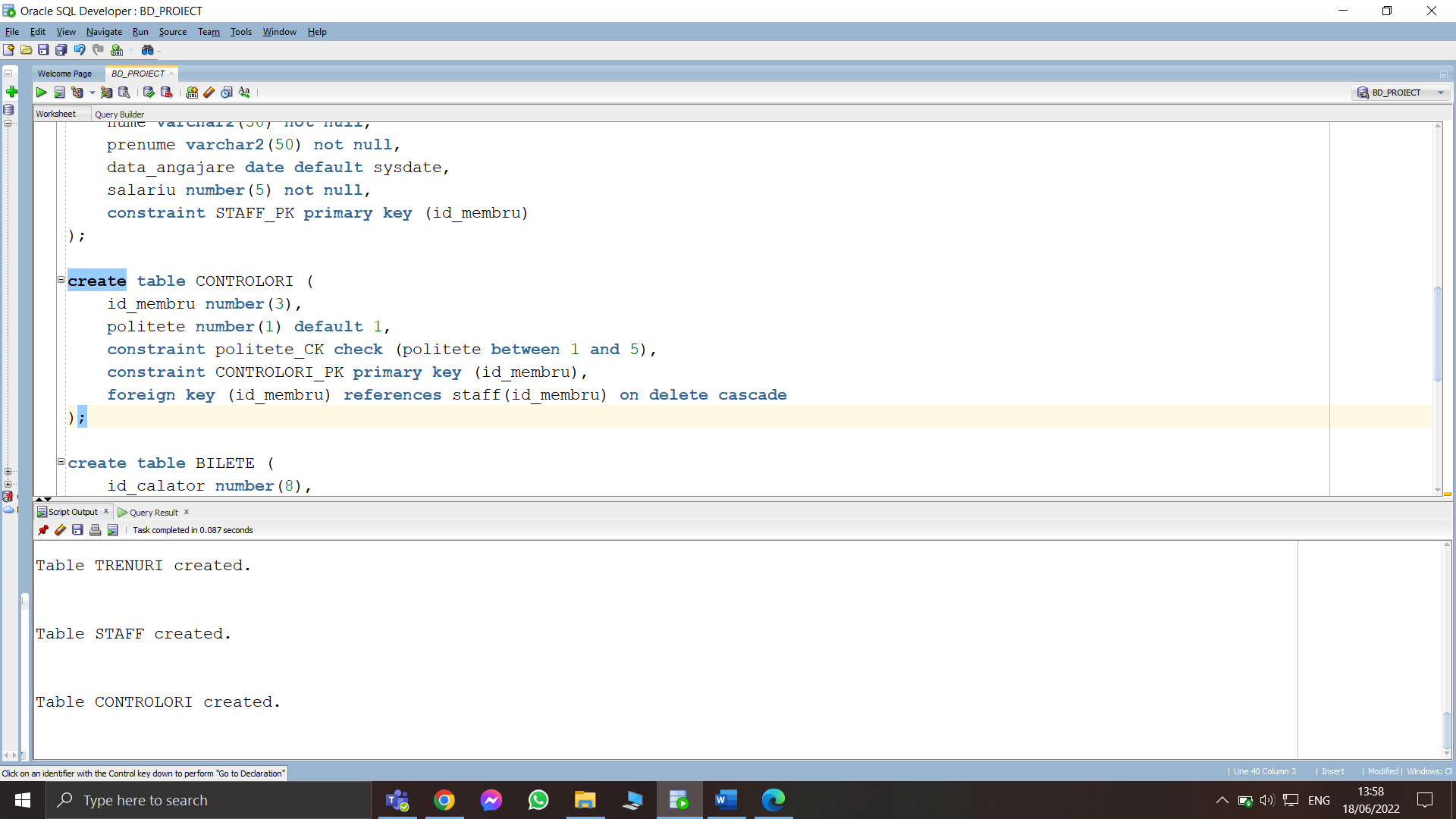
// greseala la constrangerea de check –> modificarea mai jos

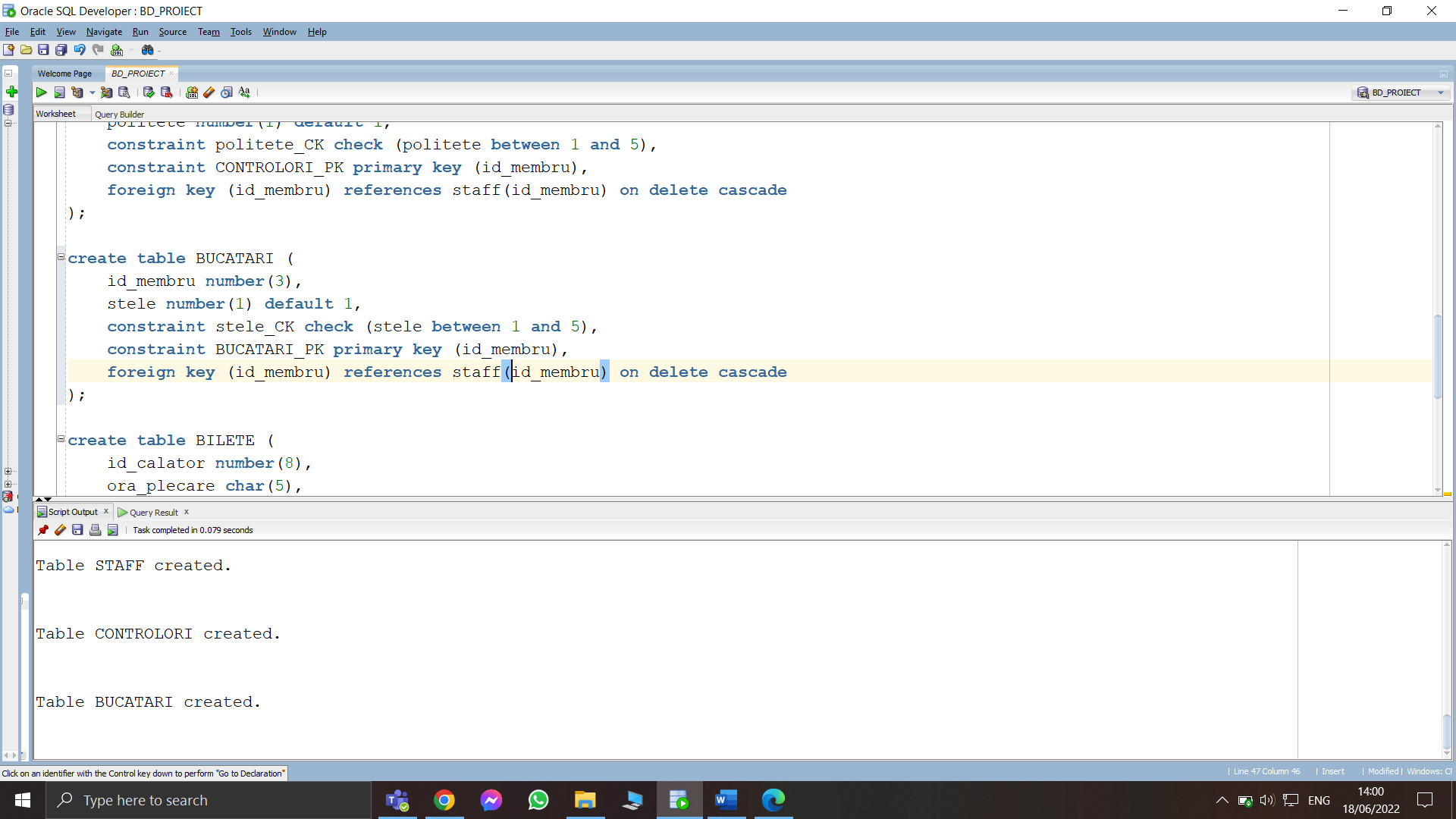


// am modificat constragerea pentru ca initial nu a fost corecta

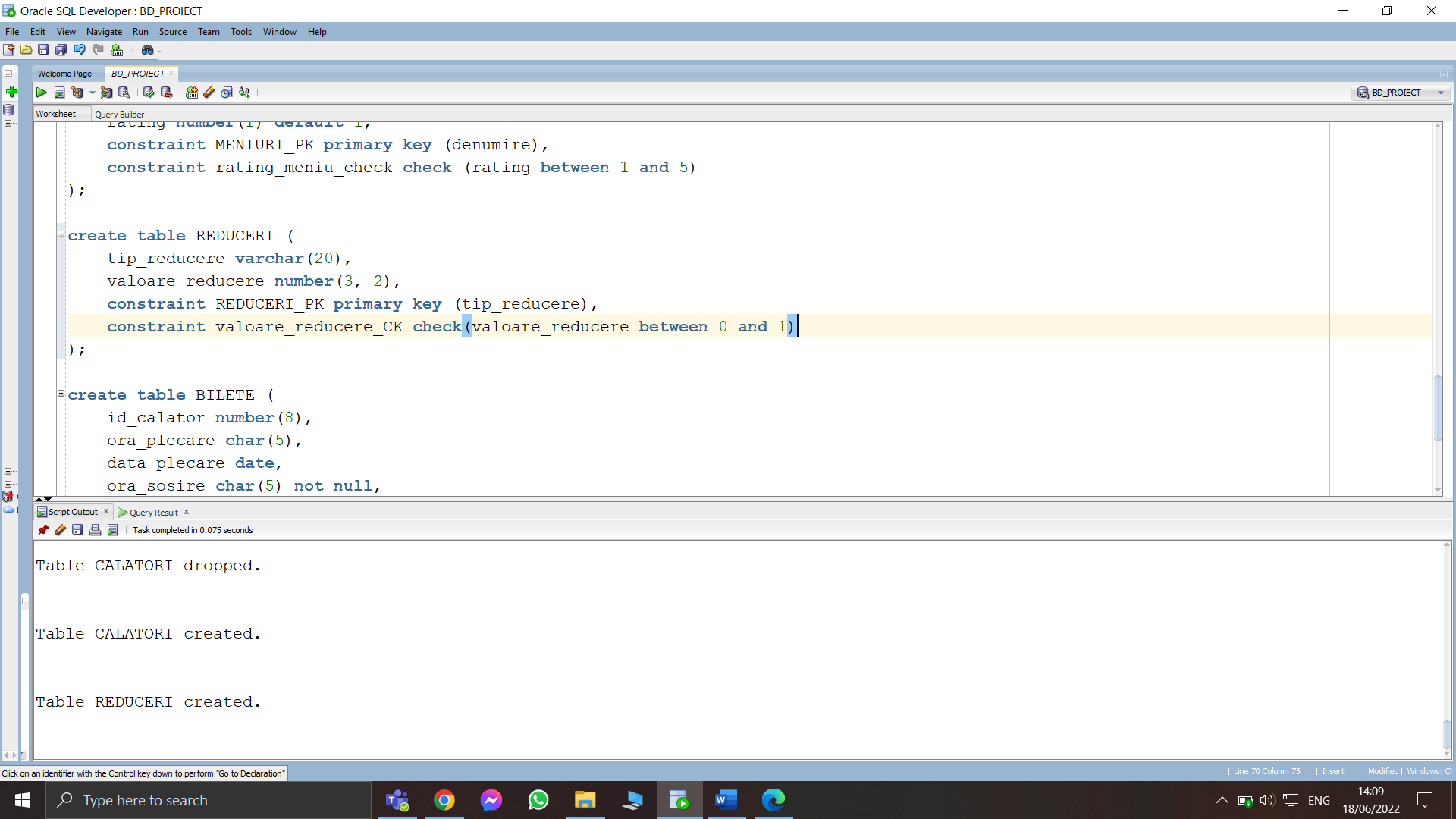




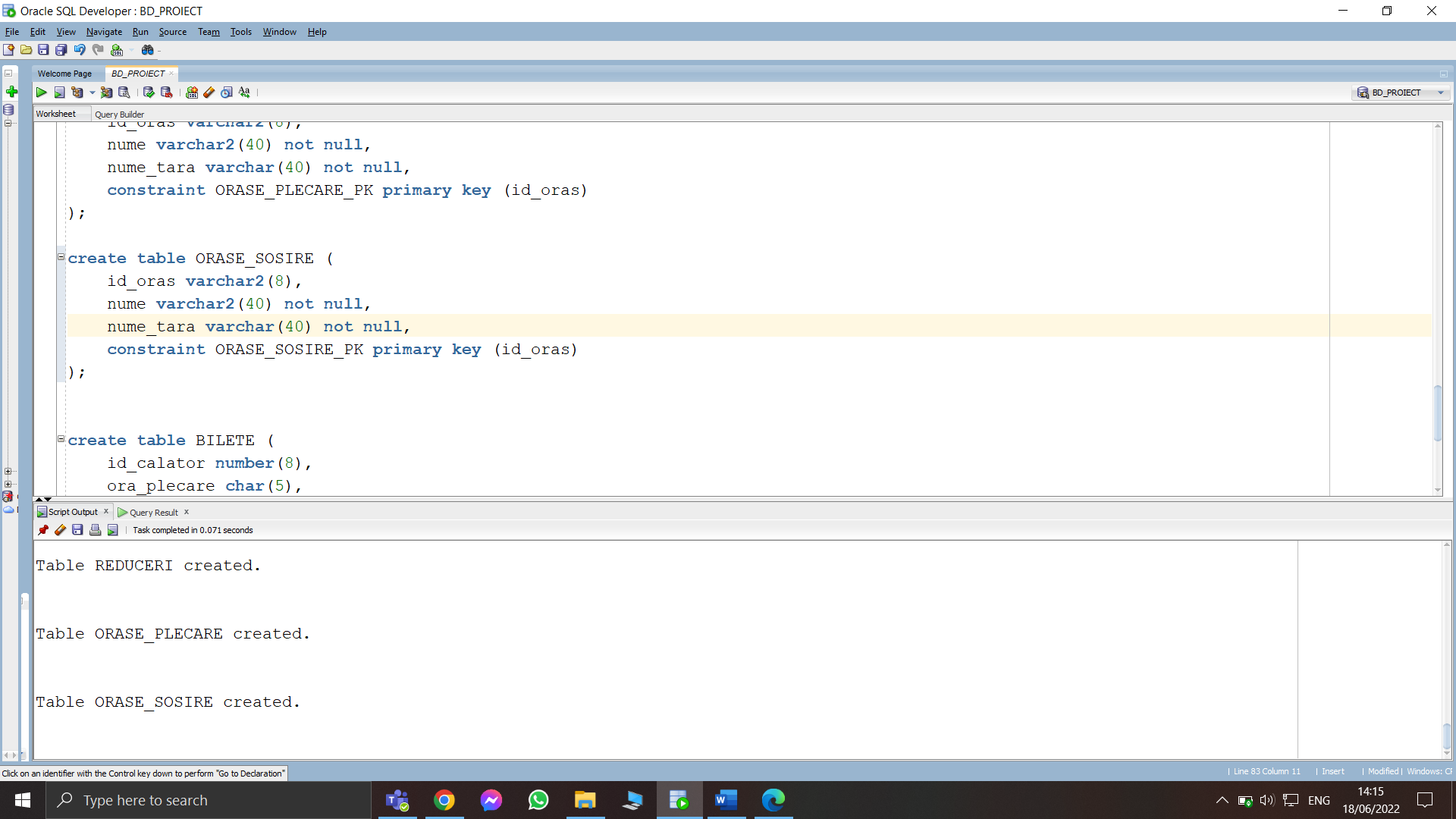


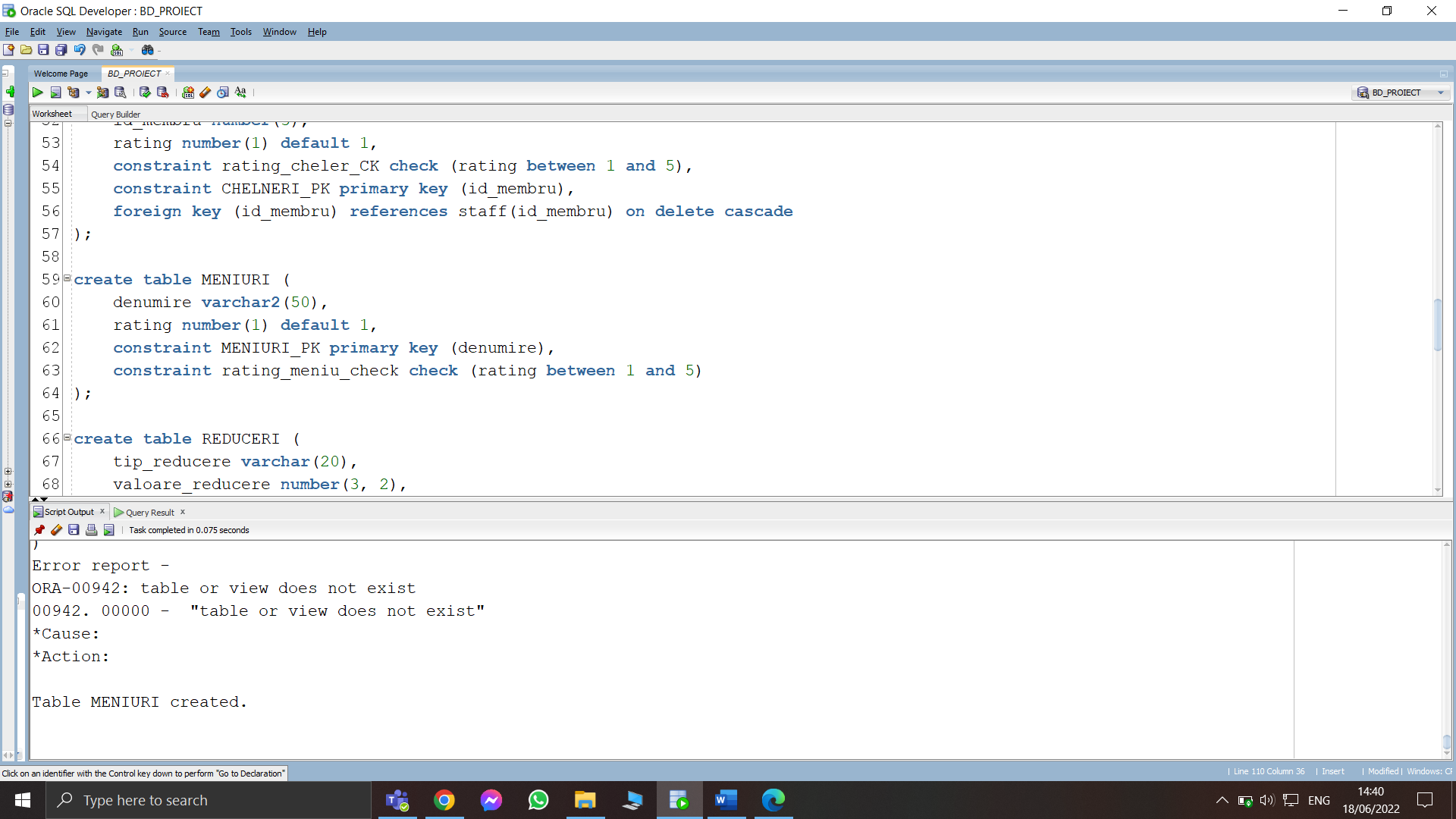


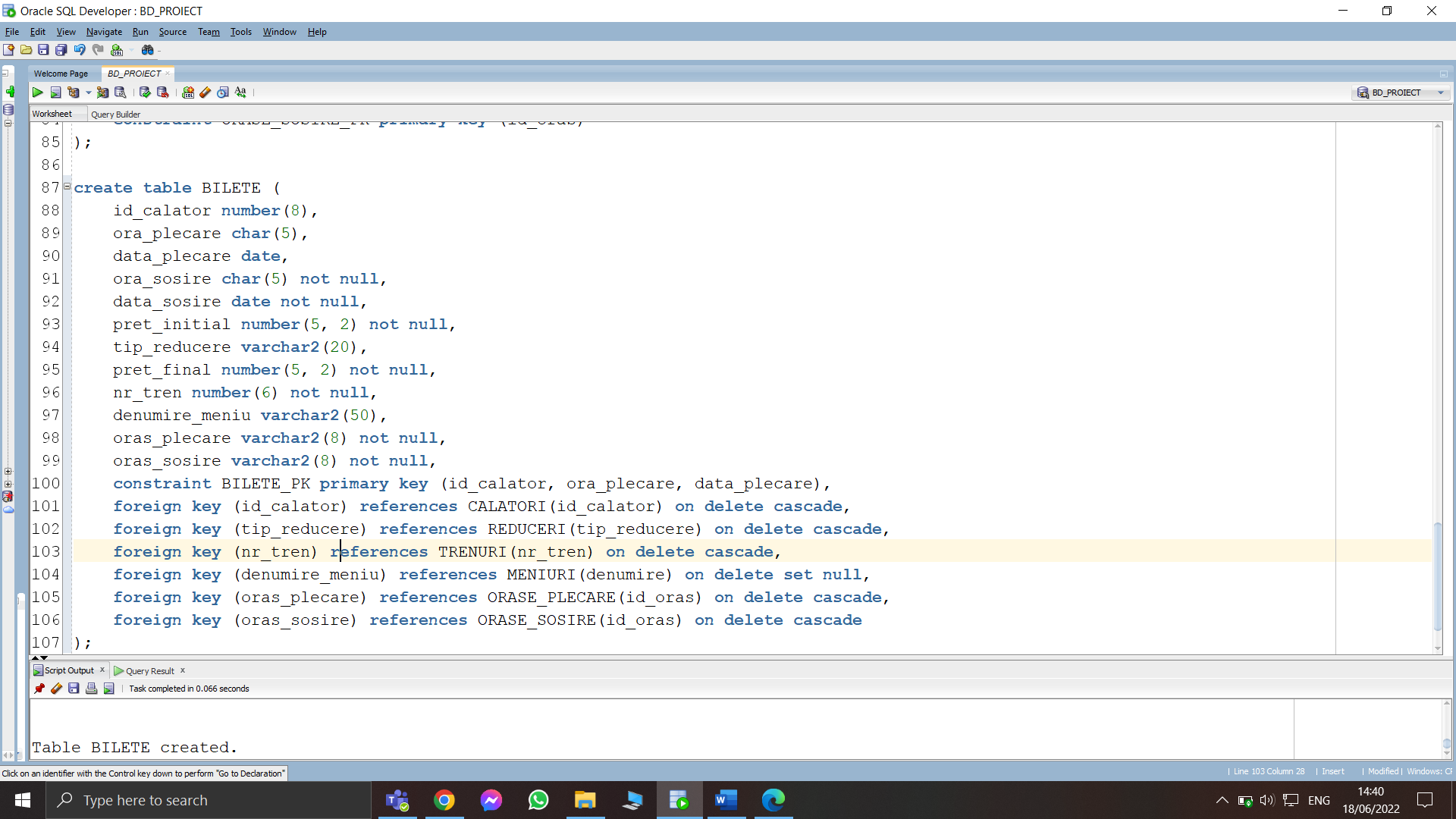


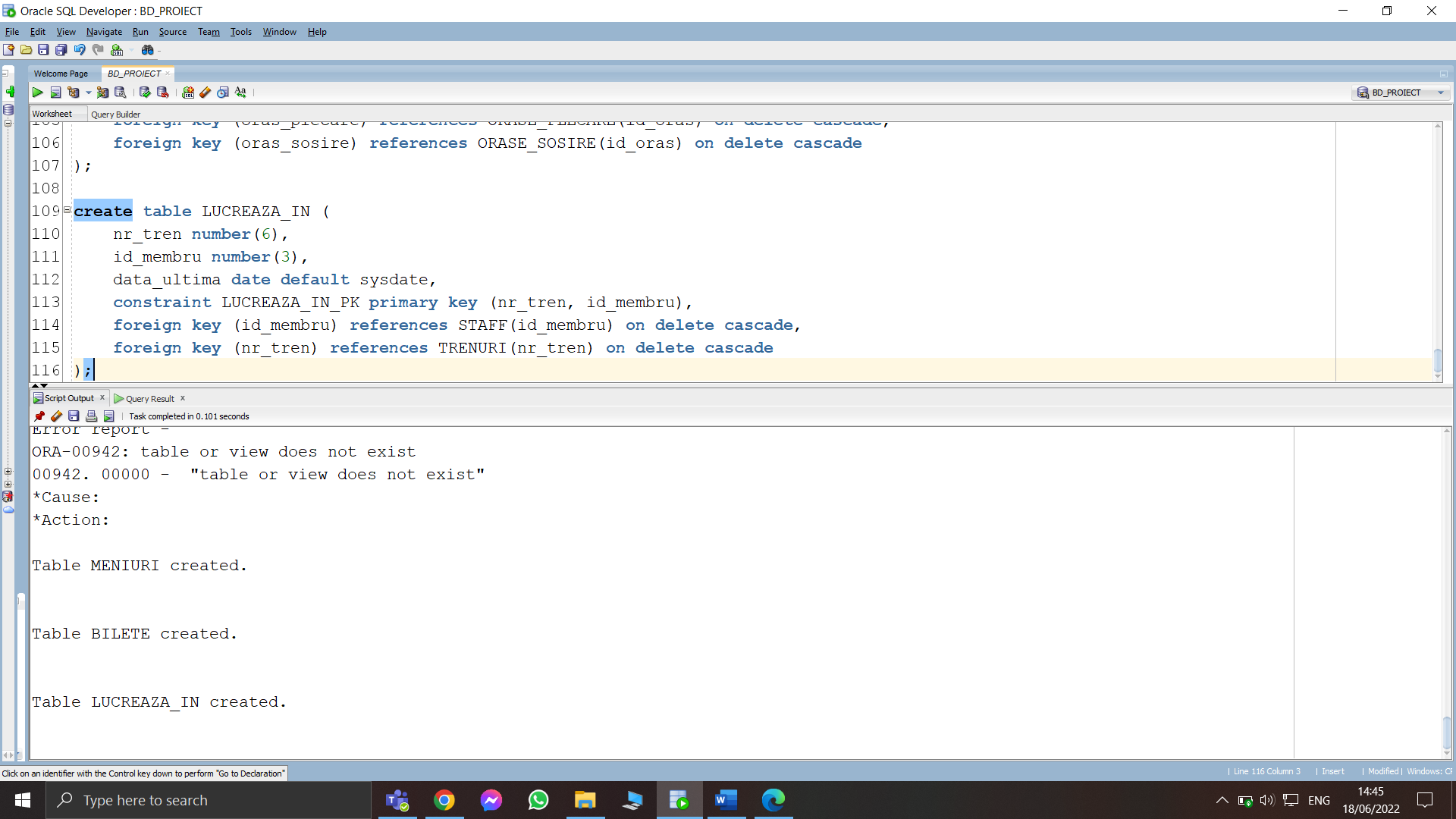


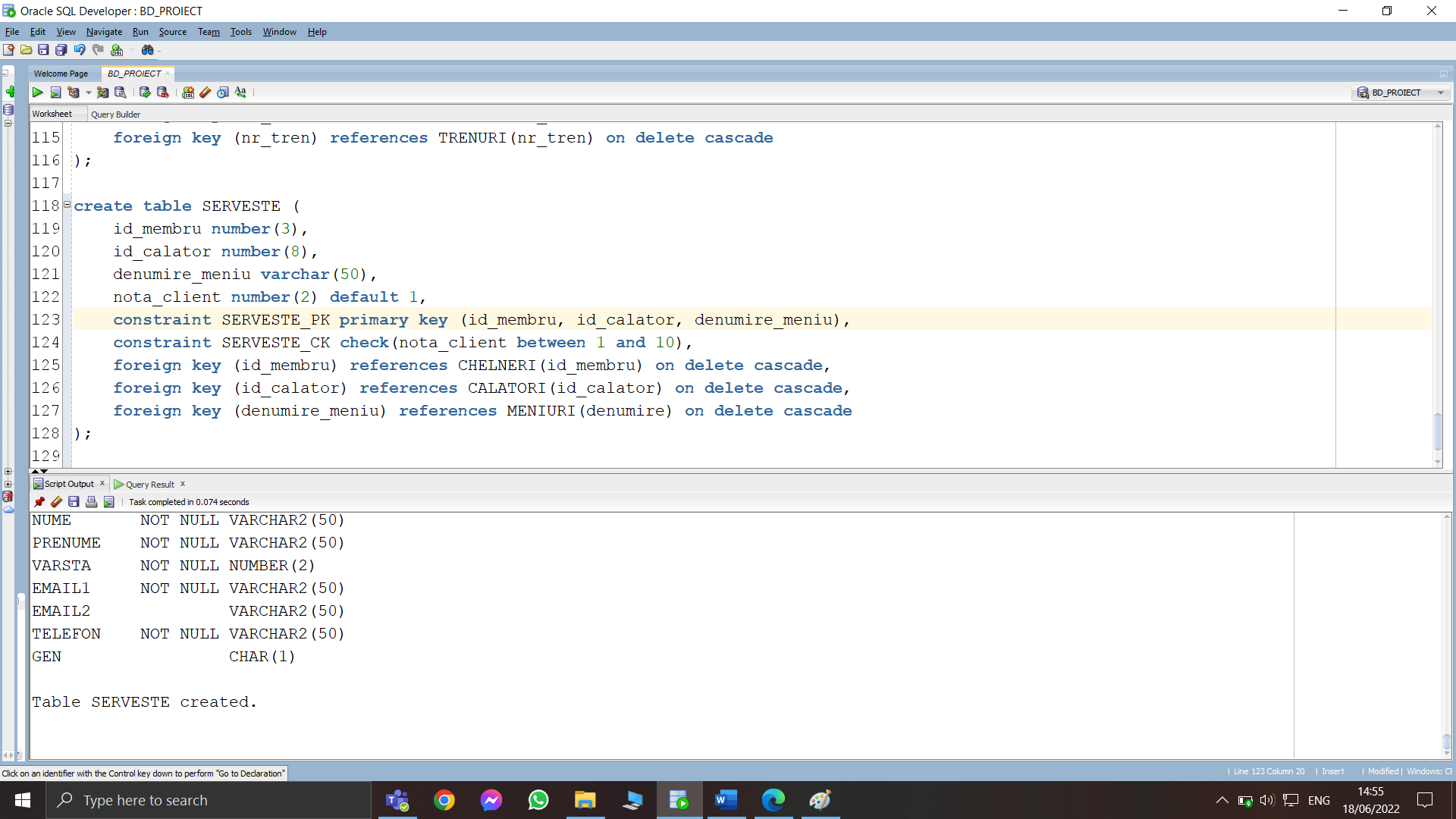


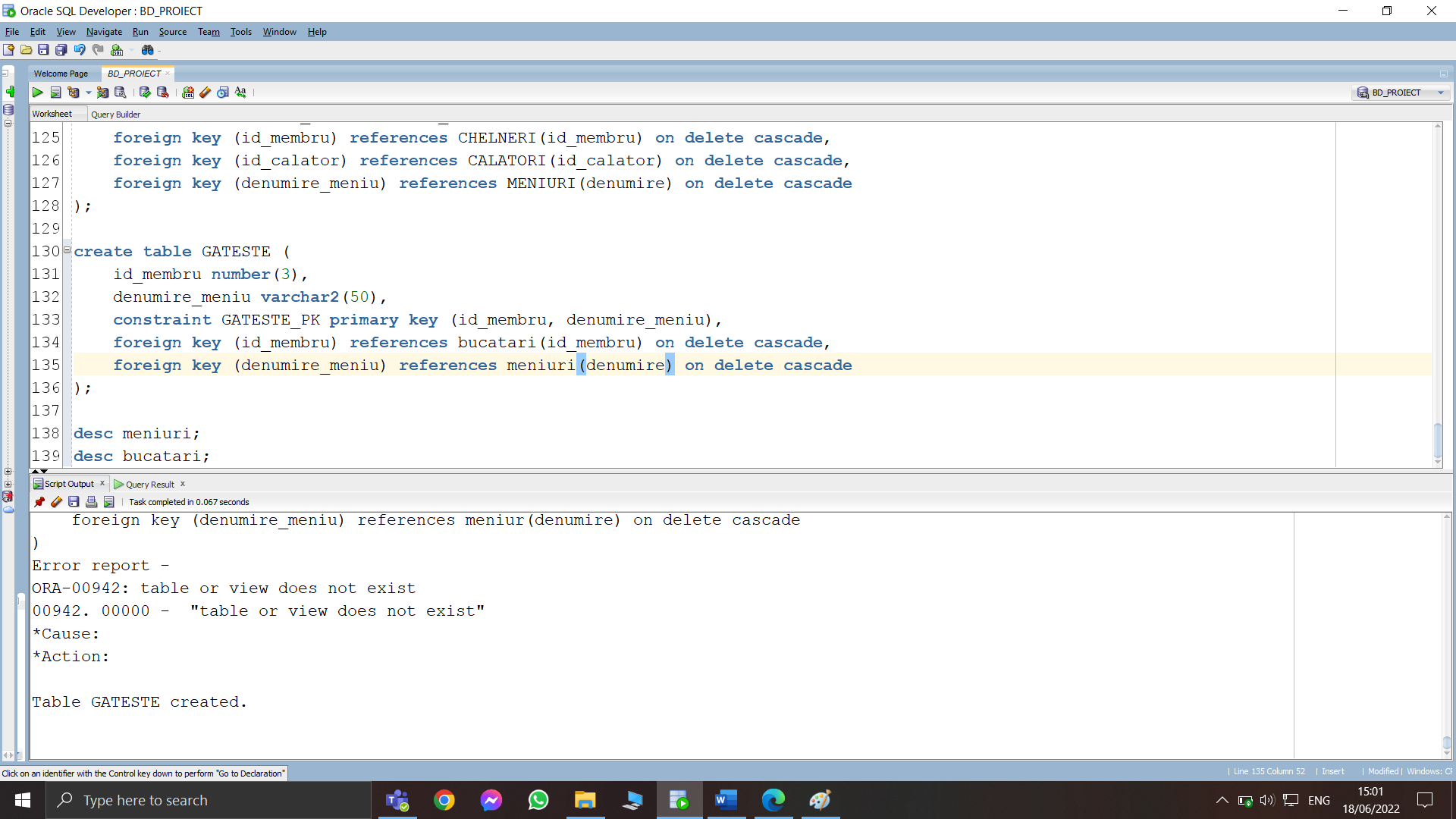


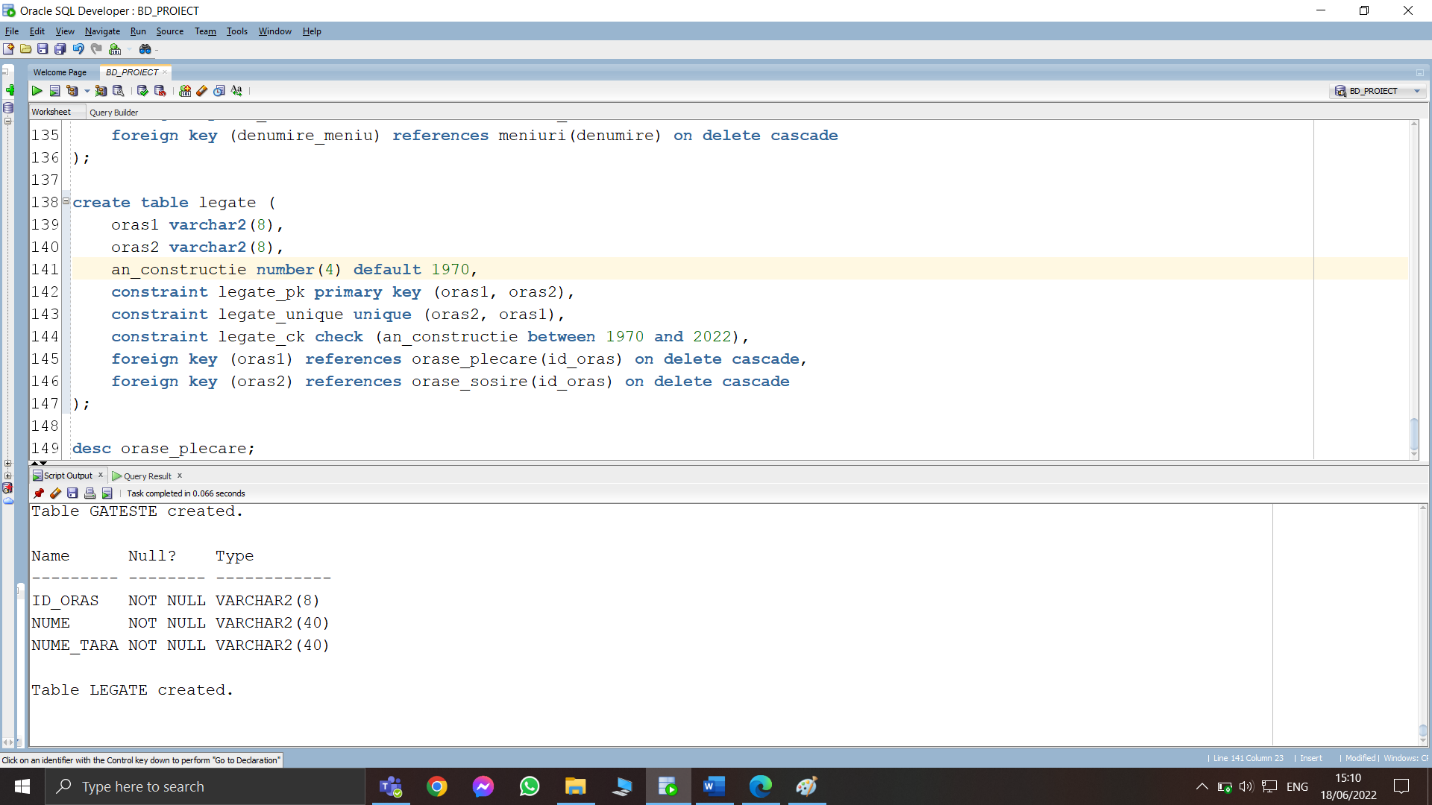


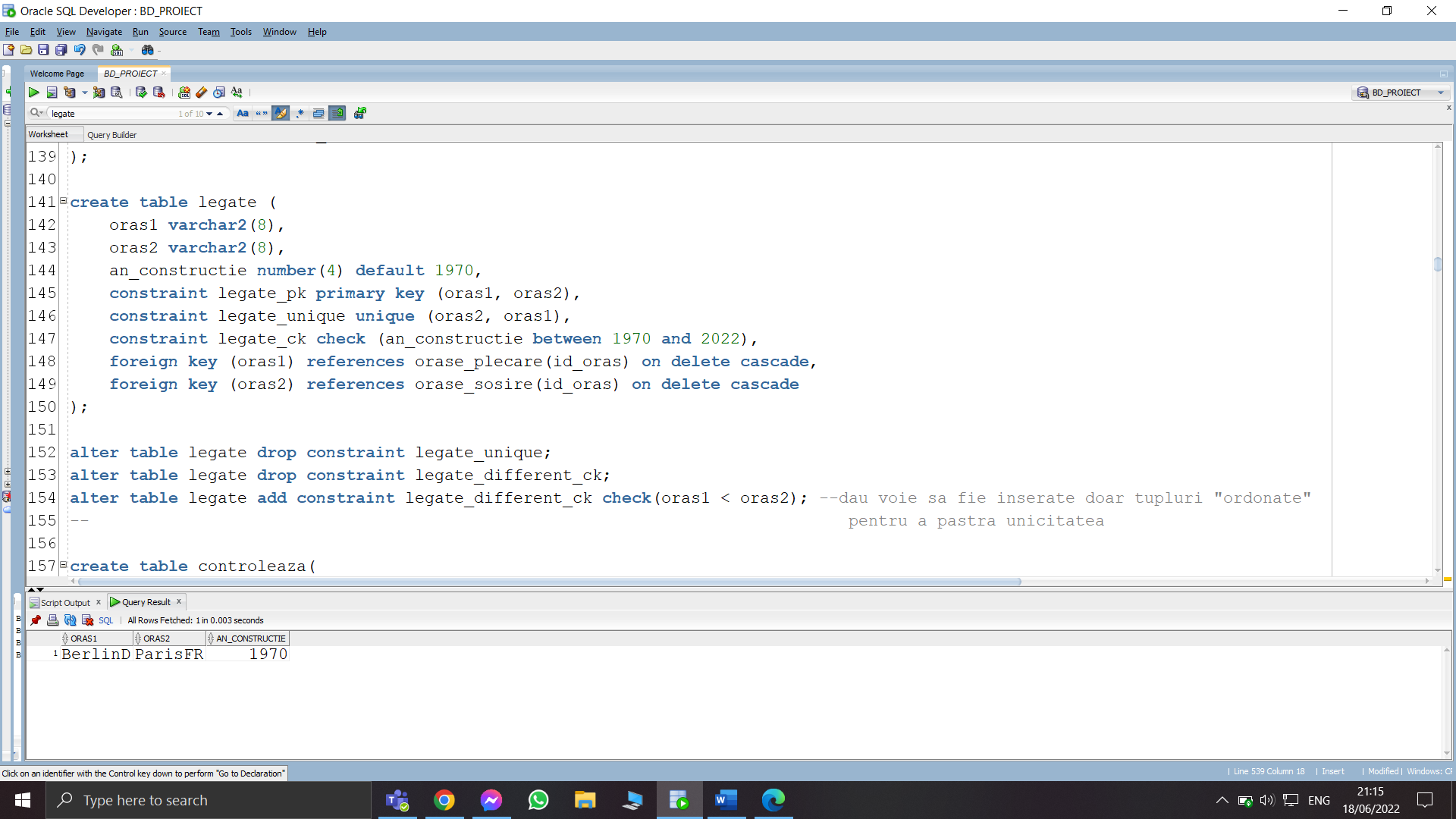


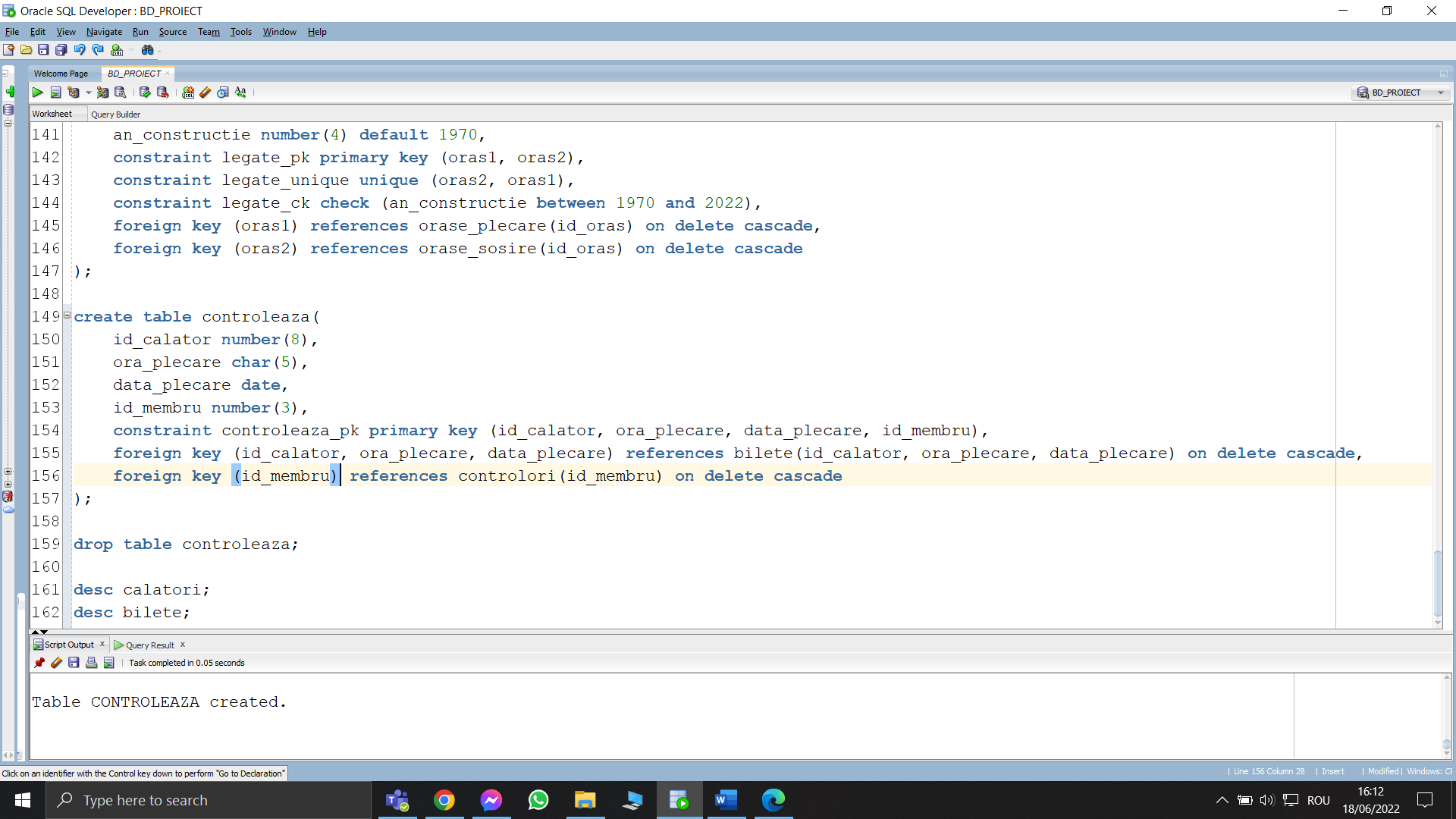




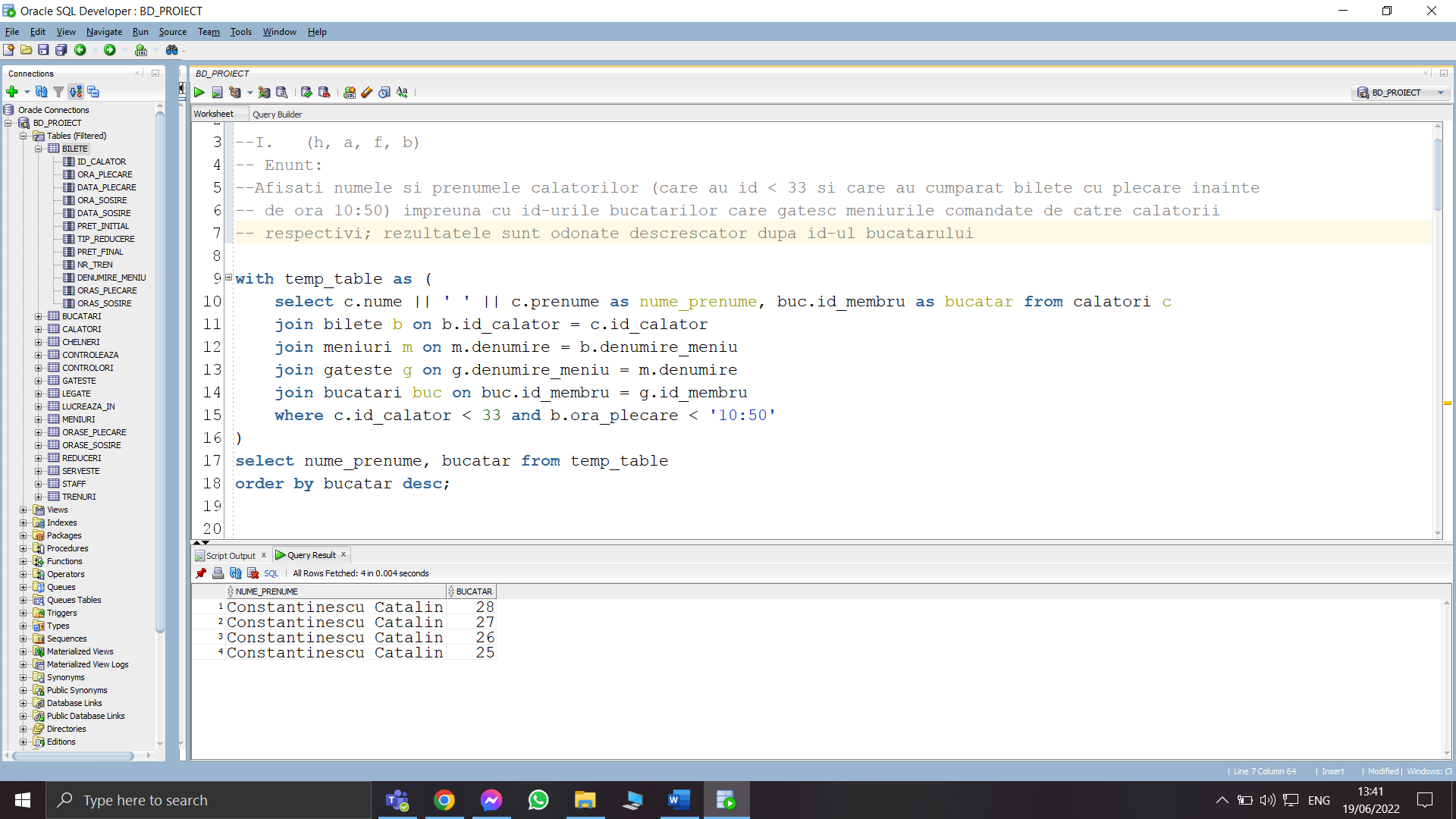
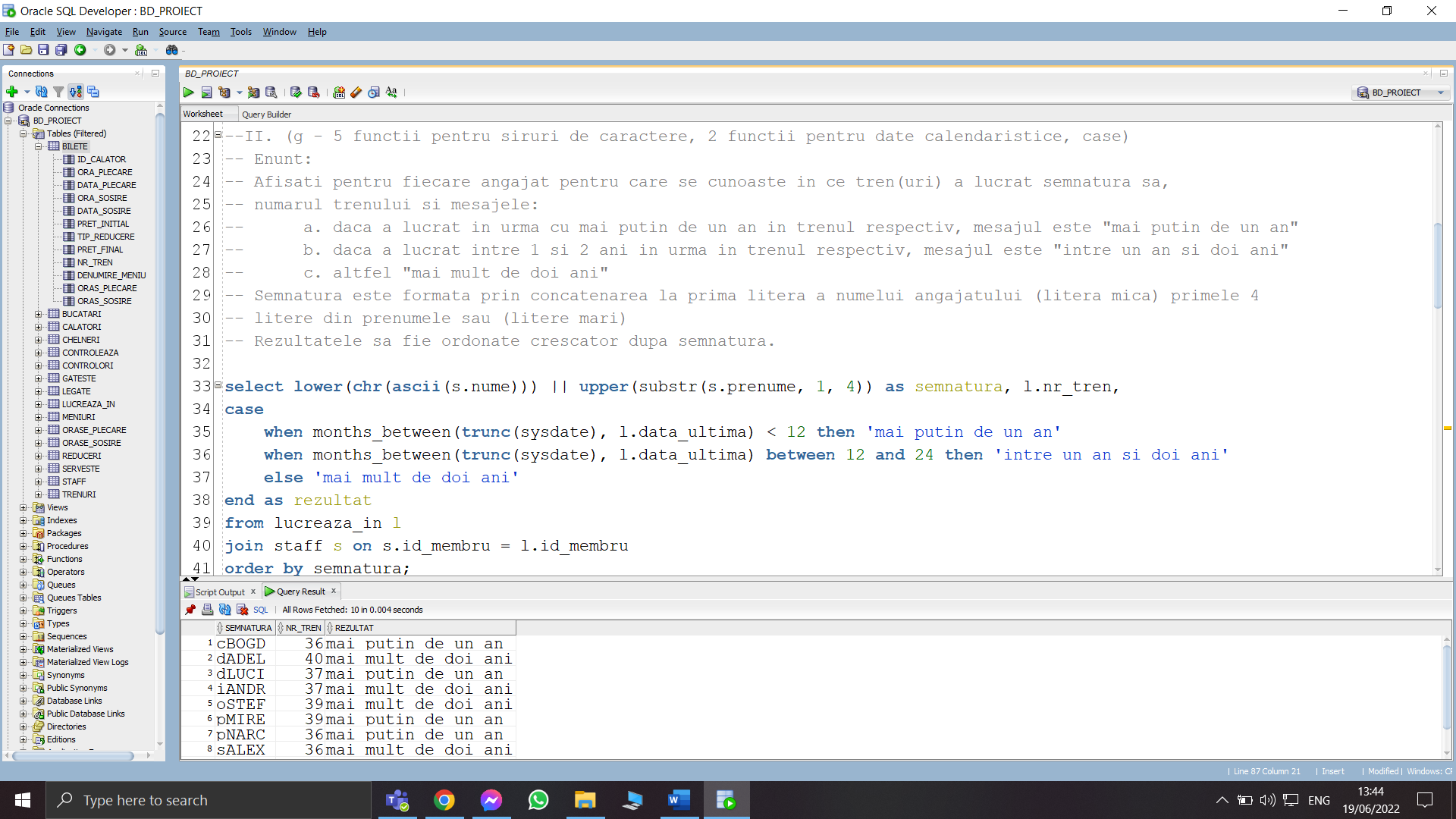


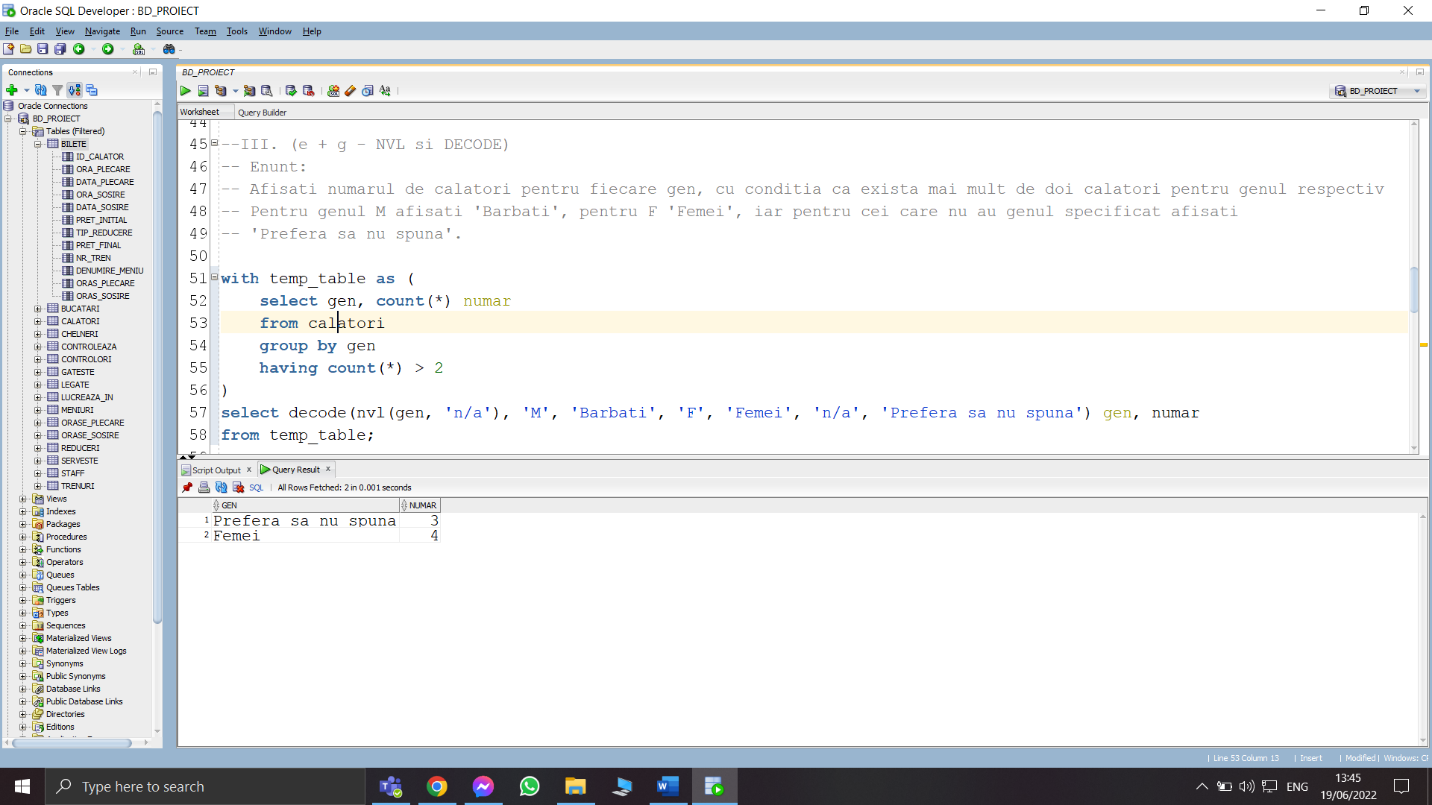
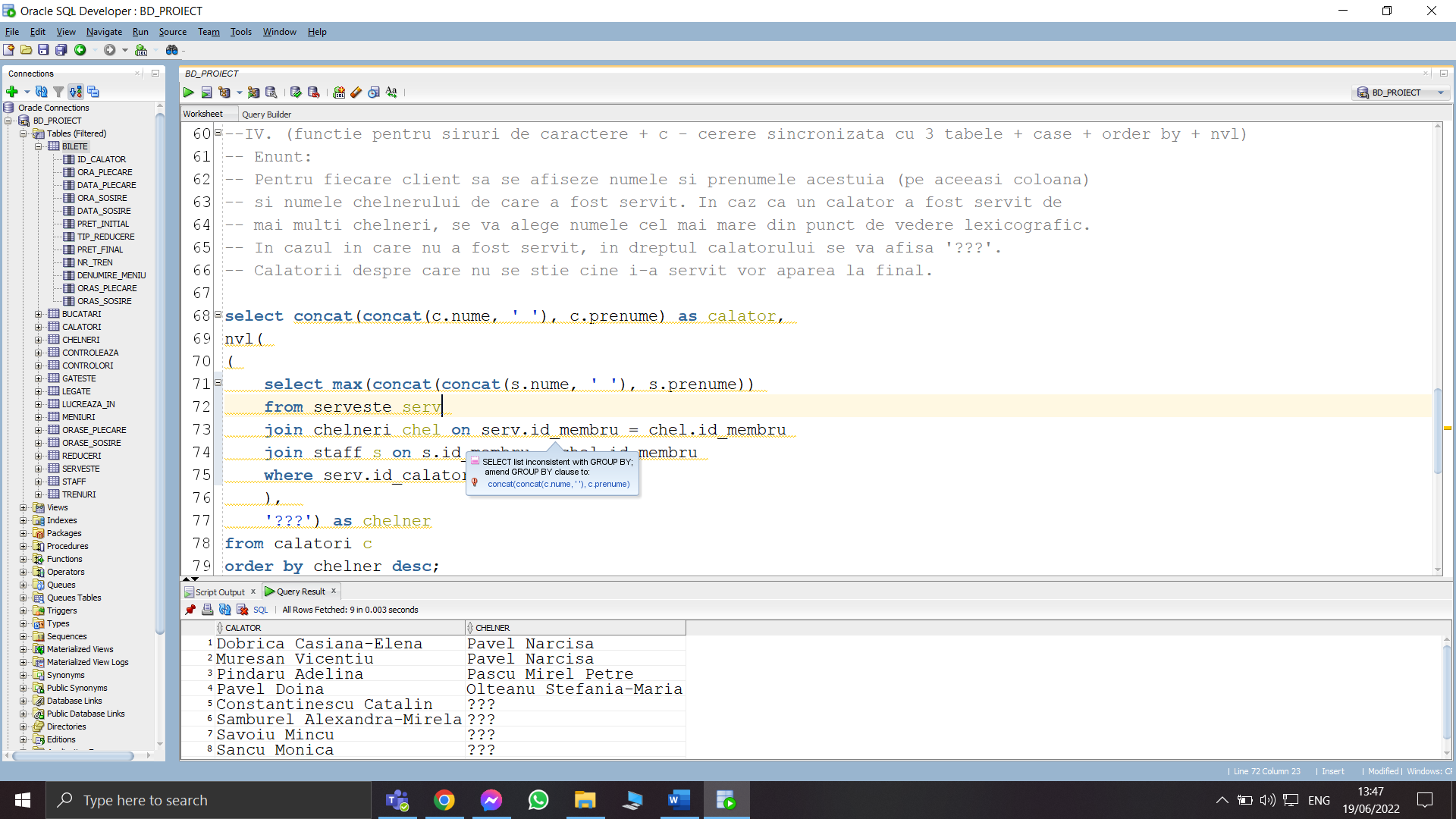
 // am modificat constrangeri in acest tabel pentru a nu permite sa fie inserate (oras1, oras2) si (oras2, oras1); am sters constrangerea legate\_unique si am adaugat constrangerea din urmatorul screen-shot

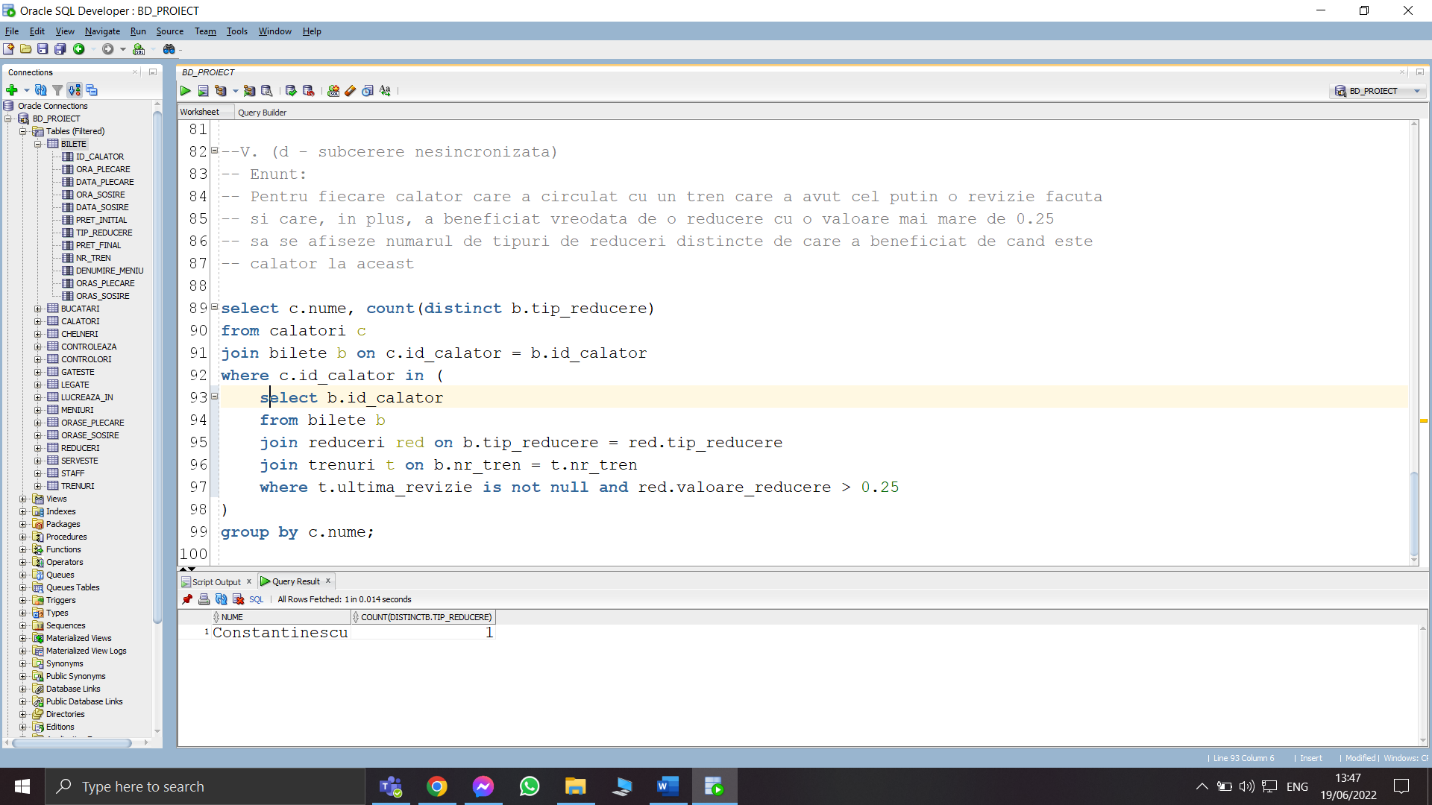
// cu aceasta constrangere permit sa fie inserate doar tupluri ordonate (care au oras1 < oras2), iar de unicitate se va ocupa cheia primara

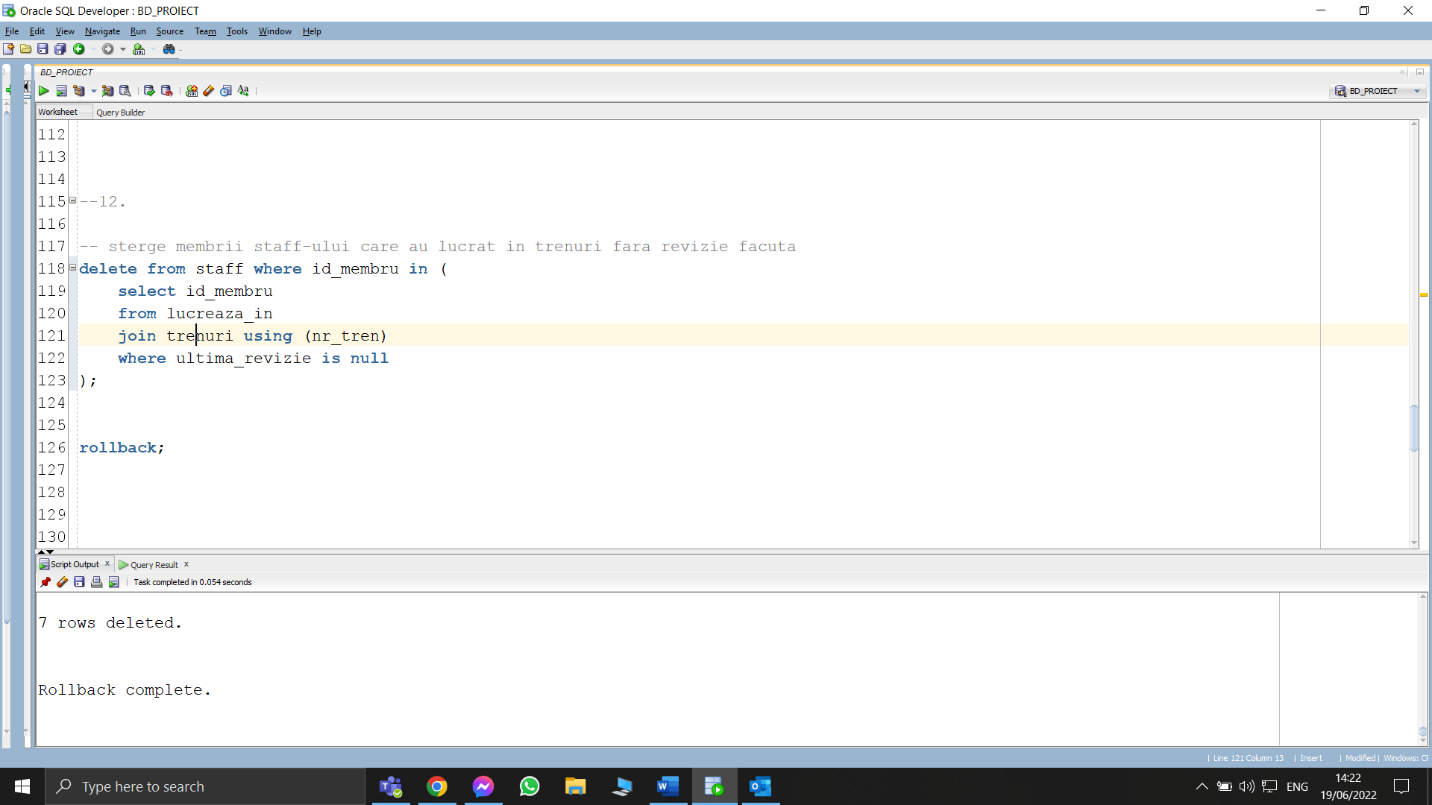


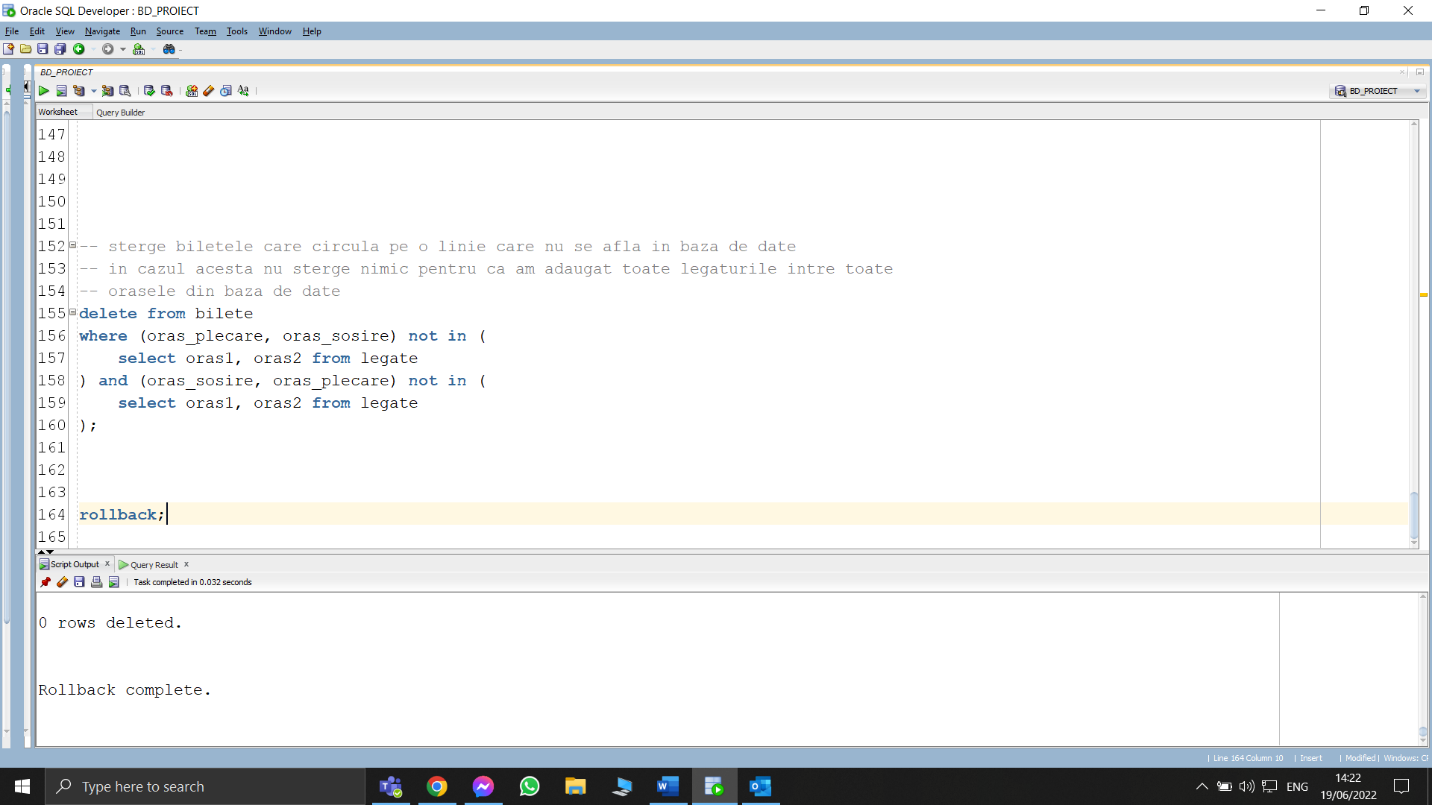
1. Formulați în limbaj natural și implementați 5 cereri SQL complexe ce vor utiliza, în ansamblul lor, următoarele elemente:
   1. operație join pe cel puțin 4 tabele
   2. filtrare la nivel de linii
   3. subcereri sincronizate în care intervin cel puțin 3 tabele
   4. subcereri nesincronizate în care intervin cel puțin 3 tabele
   5. grupări de date, funcții grup, filtrare la nivel de grupuri
   6. ordonări
   7. 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
   8. utilizarea a cel puțin 1 bloc de cerere (clauza WITH)

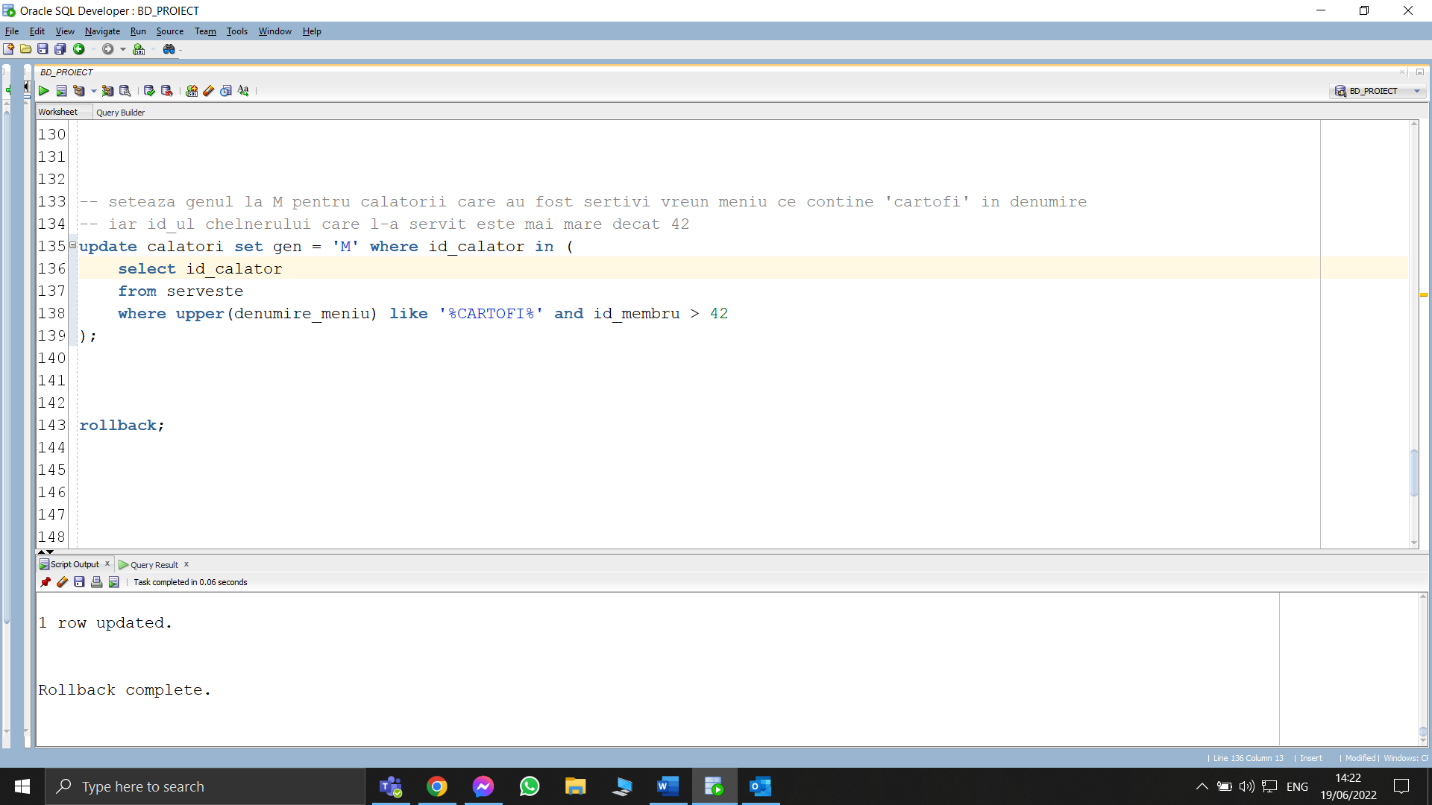


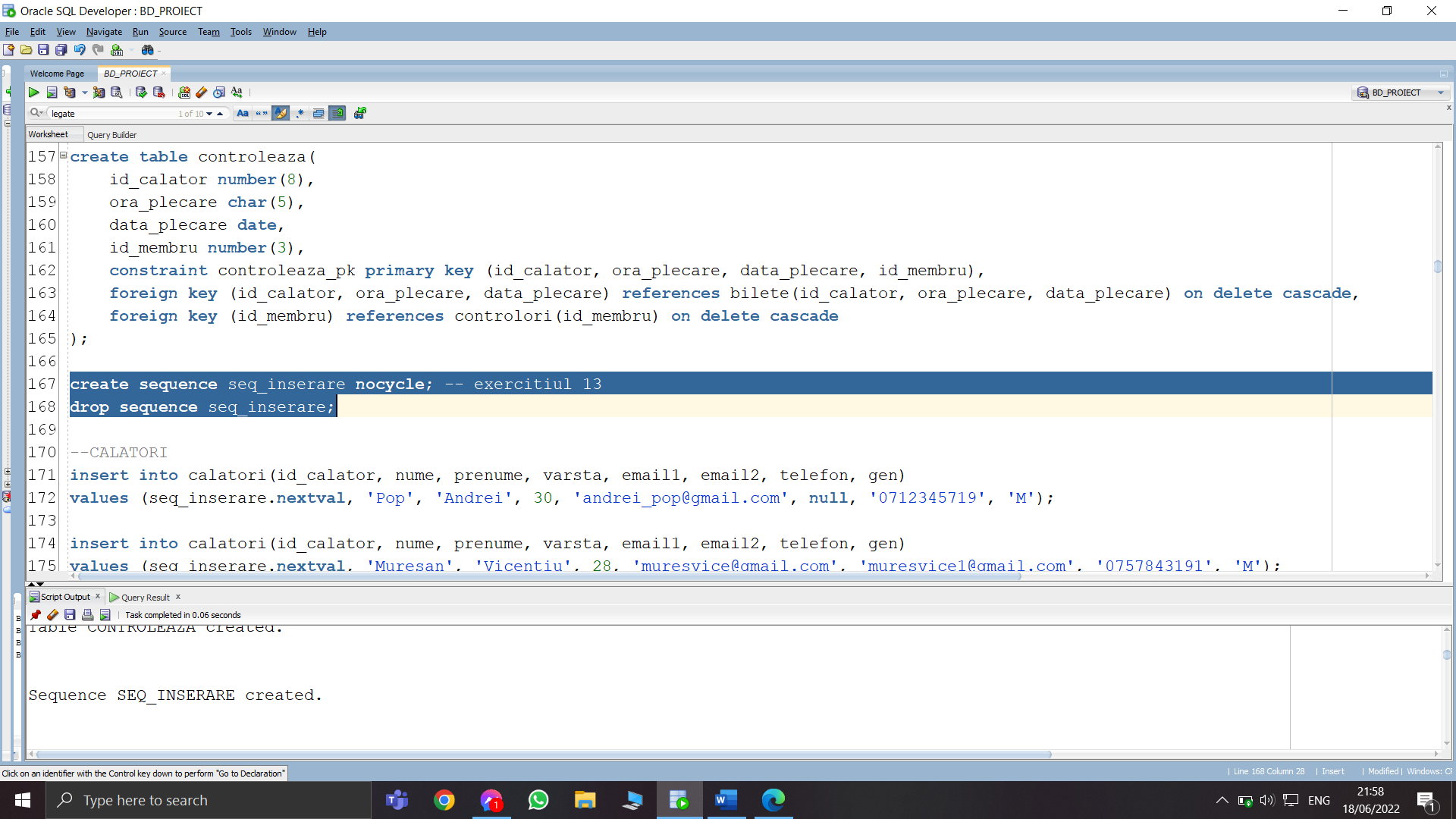




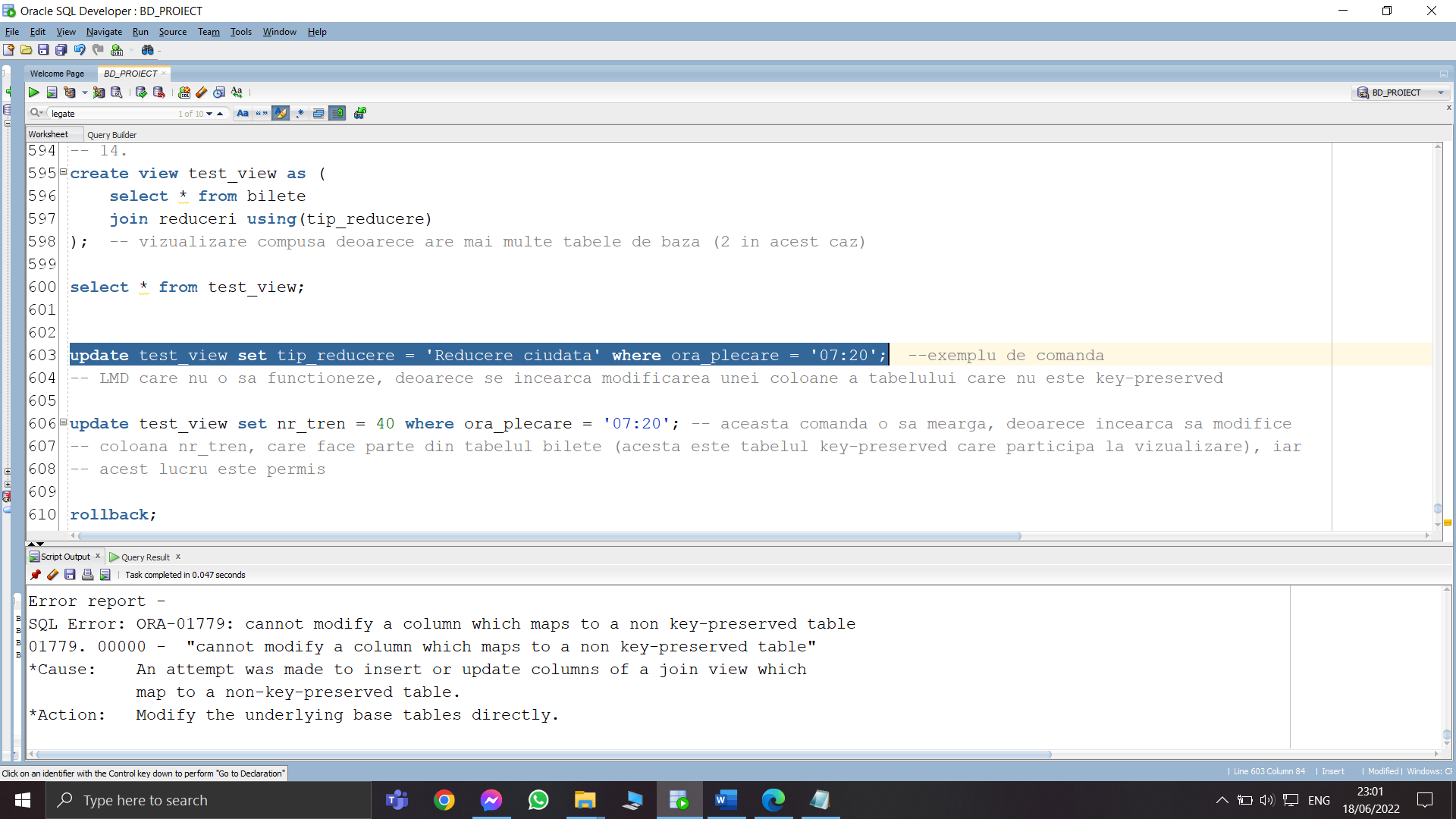
1. Implementarea a 3 operații de actualizare sau suprimare a datelor utilizând subcereri.



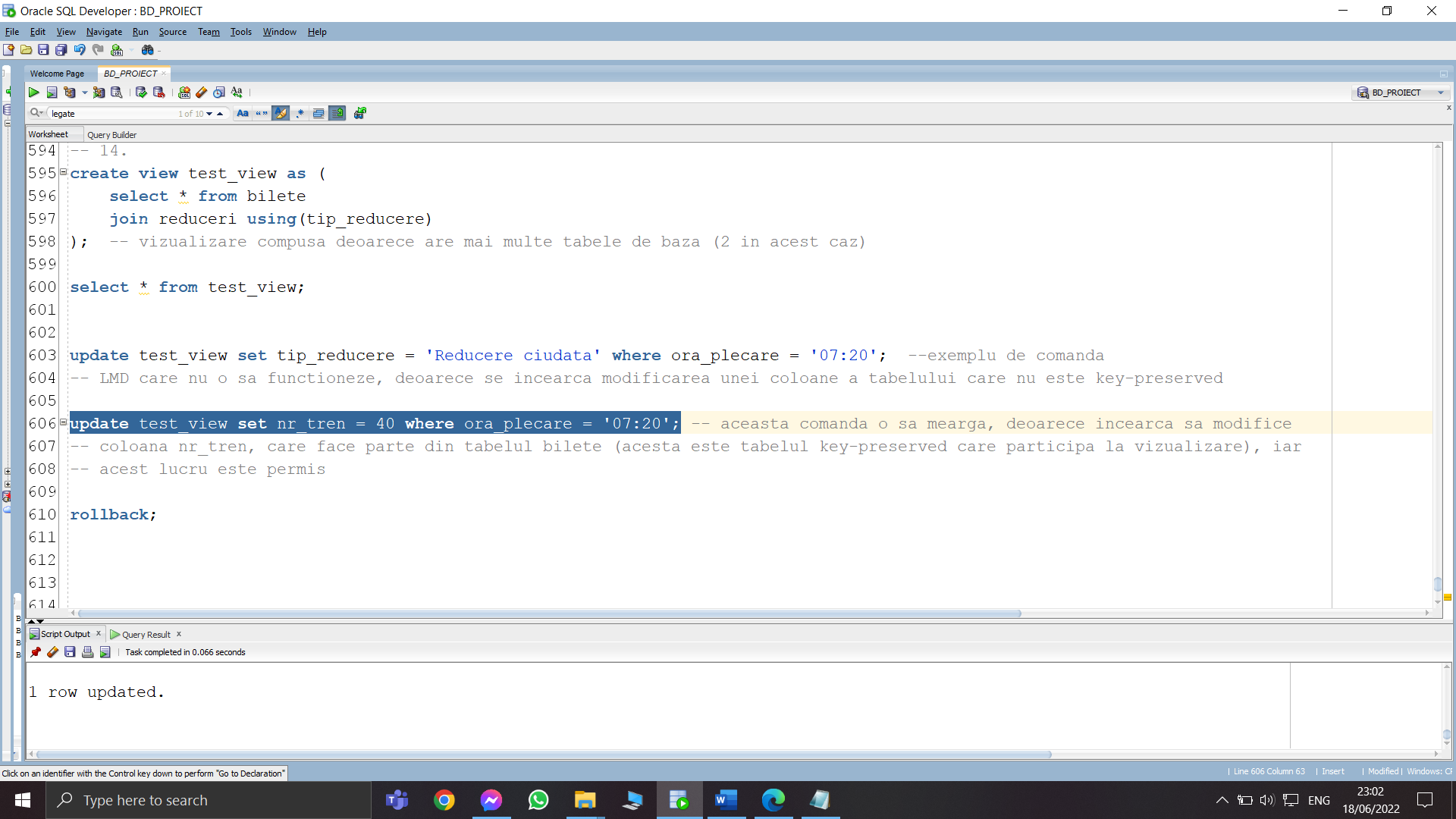


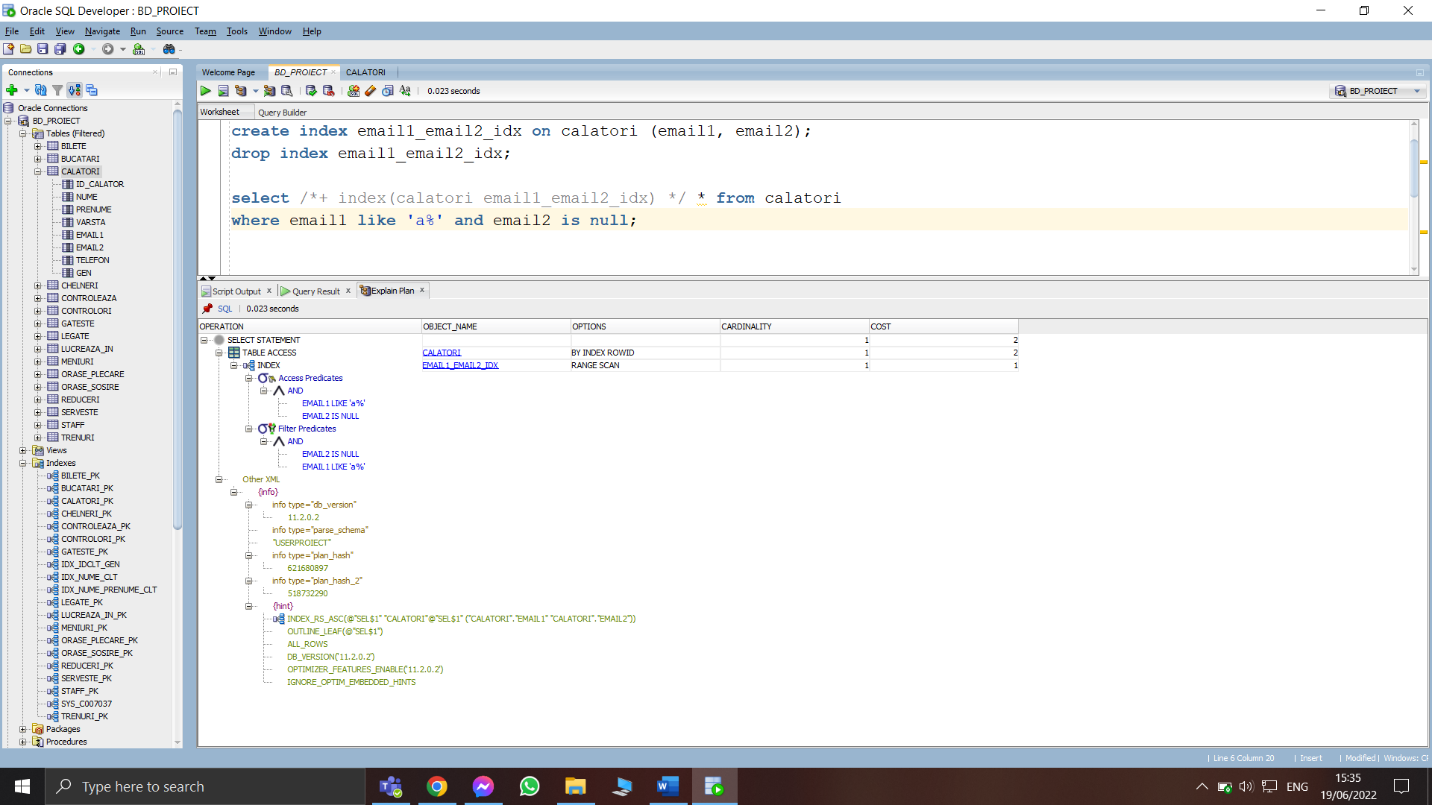
1. Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 10).
2. Crearea unei vizualizări compuse. Dați un exemplu de operație LMD permisă pe vizualizarea respectivă și un exemplu de operație LMD nepermisă.

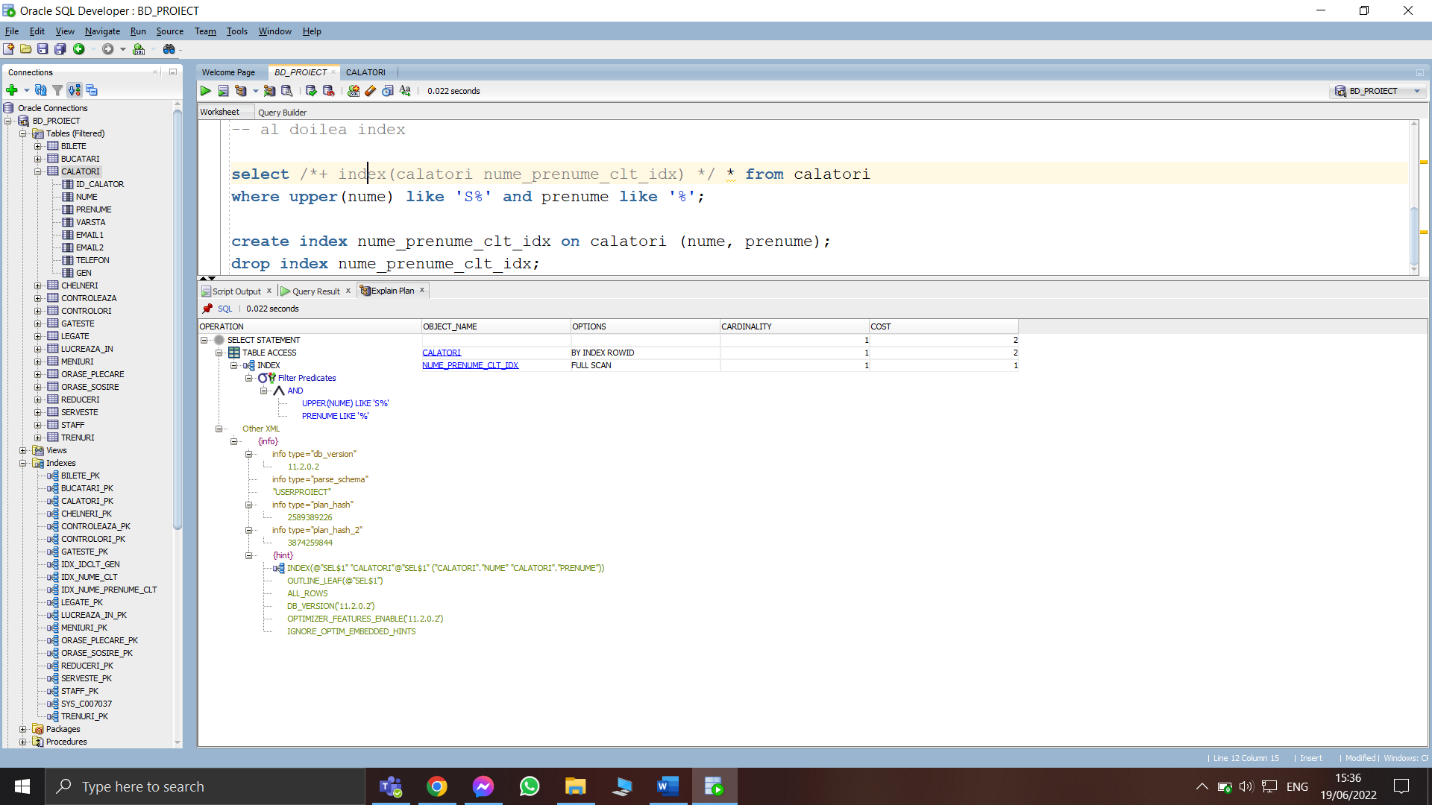
operația LMD care **nu** este permisă



operația LMD care este permisă



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



1. 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.

