**Gradina Zoologica**

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

Aceasta baza de date reprezinta un model de management al unei gradini zoologice, avand o structura relationala, bazata pe tabele interconectate intre ele prin chei primare si chei externe. Astfel putem stoca si gestiona date despre: gradina zoo, angajati, animale, specii, custi, locatii, spectacole, bilete, leguturi, vizitatori, program, ingrijire.

Tabela **GRADINA\_ZOO** stocheaza informatii despre gradina zoologica precum marimea acesteia si orasul din care face parte. Aceasta este in relatie one to many cu tabela ANGAJAT, in relatie one to many cu tabela ANIMAL, in relatie one to many cu tabela SPECTACOL, in relatie one to one cu tabela PROGRAM si in relatie one to many cu tabela VIZITATOR.

Tabela **ANGAJAT** stocheaza informatii despre angajati precum nume, prenume, varsta, gen si salariu. Aceasta este in relatie many to one cu tabela GRADINA\_ZOO si in relatie many to one cu tabela INGRIJIRE.

Tabela **ANIMAL** stocheaza informatii despre animalele din gradina zoologica precum: categoria acestora, numarul acestora, precum si numele acestora. Aceasta este in relatie many to one cu tabela GRADINA\_ZOO, in relatie many to one cu tabela LEGATURA, in relatie one to many cu tabela SPECIE si in relatie many to one cu tabela INGRIJIRE.

Tabela **BILET** stocheaza informatii despre biletele de la spectacolele din gradina zoologica precum: pretul acestuia si categoria acestuia. Aceasta este in relatie many to one cu tabela SPECTACOL.

Tabela **SPECTACOL** stocheaza informatii despre spectacolele care se desfasoara in gradina zoologica precum: ora si numarul maxim de locuri. Aceasta este in relatie one to many cu tabela BILET, in relatie many to one cu tabela LEGATURI si in relatie many to one cu tabela GRADINA\_ZOO si in relatie one to one cu tabela ORA.

Tabela **LEGATURA** face legatura intre tabela ANIMAL si tabela SPECTACOL ca sa transforme o relatie many to many in doua, una one to many intre tabela LEGATURA si tabela ANIMAL si una many to one intre tabela SPECTACOL si tabela LEGATURA.

Tabela **SPECIE** stocheaza informatii despre speciile de animale precum: rasa acestora si numarul de familii. Aceasta este intr-o relatie one to many cu tabela ANIMAL.

Tabela **CUSCA** stocheaza informatii despre custile de animale precum: materialul acestora si dimensiunea acestora. Aceasta este in relatie many to one cu tabela LOCATIE si in relatie many to one cu tabela INGRIJIRE.

Tabela **LOCATIE** stocheaza informatii despre locurile din gradina zoologica in care se afla fiecare cusca precum: pozitiile acestora si distanta de la intrare in gradina zoologica. Acesta este in relatie one to many cu tabela CUSCA.

Tabela **VIZITATOR** stocheaza informatii despre vizitatorii din gradina zoologica precum: categoria de varsta a acestora. Aceasta este in relatie many to one cu tabela GRADINA\_ZOO.

Tabela **PROGRAM** stocheaza informatii despre programul gradinii zoologice precum: zua saptamanii, ora de deschidere si ora de inchidere. Aceasta este in relatie one to one cu tabela GRADINA\_ZOO.

Tabela **INGRIJIRE** face legatura dintre tabela ANIMAL, tabela ANGAJAT si tabela CUSCA ca sa transforme 3 relatii many to many in 3,una one to many intre tabela ANGAJAT si tabela INGRIJIRE, una one to many intre tabela CUSCA si tabela INGRIJIRE si alta one to many intre tabela ANIMAL si tabela INGRIJIRE.

Tabela **ORA** stocheaza informatii despre ora la care este programat spectacol precum: ora. Aceasta este in relatie one to one cu tabela SPECTACOL.

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

In tabela GRADINA\_ZOO, id\_animal, id\_angajat, id\_vizitator, id\_program sunt foreign key.

In tabela ANGAJAT, nume si prenume sunt not null.

In tabela ANIMAL, categorie si nume\_animal sunt not null si id\_specie este foreign key.

In tabela BILET, categorie\_bilet este not null.

In tabela SPECTACOL, nr\_max\_locuri este not null si id\_bilet si id\_ora sunt foreign key.

In tabela LEGATURA, id\_animal si id\_spectacol sunt foreign key.

In tabela SPECIE, rasa este not null.

In tabela CUSCA, material este not null si id\_locatie este foreign key.

In tabela INGRIJIRE, id\_animal si id\_angajat si id\_cusca sunt foreign key.

In tabela ORA, ora este not null.

3. Descrierea entităților, incluzând precizarea cheii primare.

Tabela GRADINA\_ZOO este tabela principala, de la ea pornind si restul tabelelor. Primary key-ul este id\_gradina\_zoo.

Tabela ANGAJAT retine informatiile despre angajat. Primary key-ul este id\_angajat.

Tabela ANIMAL retine informatii despre animale. Primary key-ul este id\_animal.

Tabela BILET retine informatii despre bilete. Primary key-ul este id\_bilet.

Tabela SPECTACOL retine informatii despre spectacole. Primary key-ul este id\_spectacol.

Tabela LEGATURA retine informatii despre legatura dintre tabela ANIMAL si tabela SPECTACOL. Primary key-ul este (id\_animal, id\_spectacol).

Tabela SPECIE retine informatii despre speciile de animale. Primary key-ul este id\_specie.

Tabela CUSCA retine informatii despre custiile in care se afla animalele. Primary key-ul este id\_cusca.

Tabela LOCATIE retine informatii despre locatiile in care se afla custiile cu animale. Primary key-ul este id\_locatie.

Tabela VIZITATOR retine informatii despre vizitatorii care vin la gradina zoologica. Primary key-ul este id\_vizitator.

Tabela PROGRAM retine informatii despre programul gradinii zoologice. Primary key-ul este id\_program.

Tabela INGRIJIRE retine informatii despre legatura dintre tabela ANIMAL si tabela ANGAJAT si tabela CUSCA. Primary key-ul este (id\_animal, id\_angajat, id\_cusc).

Tabela ORA retine informatii despre ora la care incepe spectacolul. Primary key-ul este id\_ora.

4. Descrierea relațiilor, incluzând precizarea cardinalității acestora.

Tabela GRADINA\_ZOO este in relatie one to many cu tabela ANGAJAT, in relatie one to many cu tabela ANIMAL, in relatie one to many cu tabela SPECTACOL, in relatie one to one cu tabela PROGRAM si in relatie one to many cu tabela VIZITATOR.

Tabela ANGAJAT este in relatie many to one cu tabela GRADINA\_ZOO si in relatie many to one cu tabela INGRIJIRE.

Tabela ANIMAL este in relatie many to one cu tabela GRADINA\_ZOO, in relatie many to one cu Tabela LEGATURA, in relatie one to many cu tabela SPECIE.

Tabela BILET este in relatie many to one cu tabela SPECTACOL.

Tabela SPECTACOL este in relatie one to many cu tabela BILET, in relatie many to one cu tabela LEGATURI si in relatie many to one cu tabela GRADINA\_ZOO si in relatie one to one cu tabela ORA.

Tabela LEGATURA transforma o relatie many to many in doua, una one to many intre tabela LEGATURA si tabela ANIMAL si una many to one intre tabela SPECTACOL si tabela LEGATURA.

Tabela SPECIE este intr-o relatie one to many cu tabela ANIMAL.

Tabela CUSCA este in relatie many to one cu tabela LOCATIE si in relatie many to one cu tabela INGRIJIRE.

Tabela LOCATIE este in relatie one to many cu tabela CUSCA.

Tabela VIZITATOR este in relatie many to one cu tabela GRADINA\_ZOO.

Tabela PROGRAM este in relatie one to one cu tabela GRADINA\_ZOO.

Tabela INGRIJIRE transforma 3 relatii many to many in 3,una one to many intre tabela ANGAJAT si tabela INGRIJIRE, una one to many intre tabela CUSCA si tabela INGRIJIRE si alta one to many intre tabela ANIMAL si tabela INGRIJIRE.

Tabela ORA este in relatie one to one cu tabela SPECTACOL.

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

Tabela GRADINA\_ZOO:

ID\_GRADINA\_ZOO: Număr întreg de maximum 9 cifre, cheie primară.

MARIME: Număr întreg de maximum 7 cifre.

ORAS: Șir de caractere cu lungimea maximă de 30 de caractere.

ID\_ANIMAL: Număr întreg de maximum 9 cifre, cheie externă referențiază coloana ID\_ANIMAL din tabela ANIMAL.

ID\_ANGAJAT: Număr întreg de maximum 9 cifre, cheie externă referențiază coloana ID\_ANGAJAT din tabela ANGAJAT.

ID\_VIZITATOR: Număr întreg de maximum 9 cifre, cheie externă referențiază coloana ID\_VIZITATOR din tabela VIZITATOR.

ID\_PROGRAM: Număr întreg de maximum 9 cifre, cheie externă referențiază coloana ID\_PROGRAM din tabela PROGRAM.

Tabela ANGAJAT:

ID\_ANGAJAT: Număr întreg de maximum 9 cifre, cheie primară.

NUME: Șir de caractere cu lungimea maximă de 20 de caractere, obligatoriu (NOT NULL).

PRENUME: Șir de caractere cu lungimea maximă de 20 de caractere, obligatoriu (NOT NULL).

VARSTA: Număr întreg de maximum 3 cifre.

GEN: Șir de caractere cu lungimea maximă de 20 de caractere.

SALARIU: Număr întreg de maximum 9 cifre.

Tabela ANIMAL:

ID\_ANIMAL: Număr întreg de maximum 9 cifre, cheie primară.

CATEGORIE: Șir de caractere cu lungimea maximă de 20 de caractere, obligatoriu (NOT NULL).

NR\_ANIMALE: Număr întreg de maximum 5 cifre.

NUME\_ANIMAL: Șir de caractere cu lungimea maximă de 50 de caractere, obligatoriu (NOT NULL).

ID\_SPECIE: Număr întreg de maximum 9 cifre, cheie externă referențiază coloana ID\_SPECIE din tabela SPECIE.

Tabela BILET:

ID\_BILET: Număr întreg de maximum 9 cifre, cheie primară.

PRET: Număr întreg de maximum 4 cifre, obligatoriu (NOT NULL).

CATEGORIE\_BILET: Șir de caractere cu lungimea maximă de 20 de caractere, obligatoriu (NOT NULL).

Tabela SPECTACOL:

ID\_SPECTACOL: Număr întreg de maximum 9 cifre, cheie primară.

NR\_MAX\_LOCURI: Număr întreg de maximum 6 cifre, obligatoriu (NOT NULL).

ID\_ORA: Număr întreg de maximum 9 cifre, cheie externă referențiază coloana ID\_ORA din tabela ORA.

ID\_BILET: Număr întreg de maximum 9 cifre, cheie externă referențiază coloana ID\_BILET din tabela BILET.

Tabela LEGATURA:

ID\_ANIMAL: Număr întreg de maximum 9 cifre, cheie externă referențiază coloana ID\_ANIMAL din tabela ANIMAL.

ID\_SPECTACOL: Număr întreg de maximum 9 cifre, cheie externă referențiază coloana ID\_SPECTACOL din tabela SPECTACOL.

Cheie primară formată din combinația coloanelor ID\_ANIMAL și ID\_SPECTACOL.

Tabela SPECIE:

ID\_SPECIE: Număr întreg de maximum 9 cifre, cheie primară.

RASA: Șir de caractere cu lungimea maximă de 50 de caractere, obligatoriu (NOT NULL).

NR\_FAMILII: Număr întreg de maximum 5 cifre.

Tabela CUSCA:

ID\_CUSCA: Număr întreg de maximum 9 cifre, cheie primară.

MATERIAL: Șir de caractere cu lungimea maximă de 20 de caractere, obligatoriu (NOT NULL).

DIMENSIUNE: Număr întreg de maximum 5 cifre.

ID\_LOCATIE: Număr întreg de maximum 9 cifre, cheie externă referențiază coloana ID\_LOCATIE din tabela LOCATIE.

Tabela LOCATIE:

ID\_LOCATIE: Număr întreg de maximum 9 cifre, cheie primară.

POZITIE: Șir de caractere cu lungimea maximă de 20 de caractere.

DISTANTA: Număr întreg de maximum 4 cifre.

Tabela VIZITATOR:

ID\_VIZITATOR: Număr întreg de maximum 9 cifre, cheie primară.

CATEGORIE\_VARSTA: Șir de caractere cu lungimea maximă de 20 de caractere.

Tabela PROGRAM:

ID\_PROGRAM: Număr întreg de maximum 9 cifre, cheie primară.

ZIUA\_SAPTAMANII: Șir de caractere cu lungimea maximă de 20 de caractere.

ORA\_DESCHIDERII: Număr întreg de maximum 3 cifre.

ORA\_INCHIDERII: Număr întreg de maximum 3 cifre.

Tabela INGRIJIRE:

ID\_ANIMAL: Număr întreg de maximum 9 cifre, cheie externă referențiază coloana ID\_ANIMAL din tabela ANIMAL.

ID\_ANGAJAT: Număr întreg de maximum 9 cifre, cheie externă referențiază coloana ID\_ANGAJAT din tabela ANGAJAT.

ID\_CUSCA: Număr întreg de maximum 9 cifre, cheie externă referențiază coloana ID\_CUSCA din tabela CUSCA.

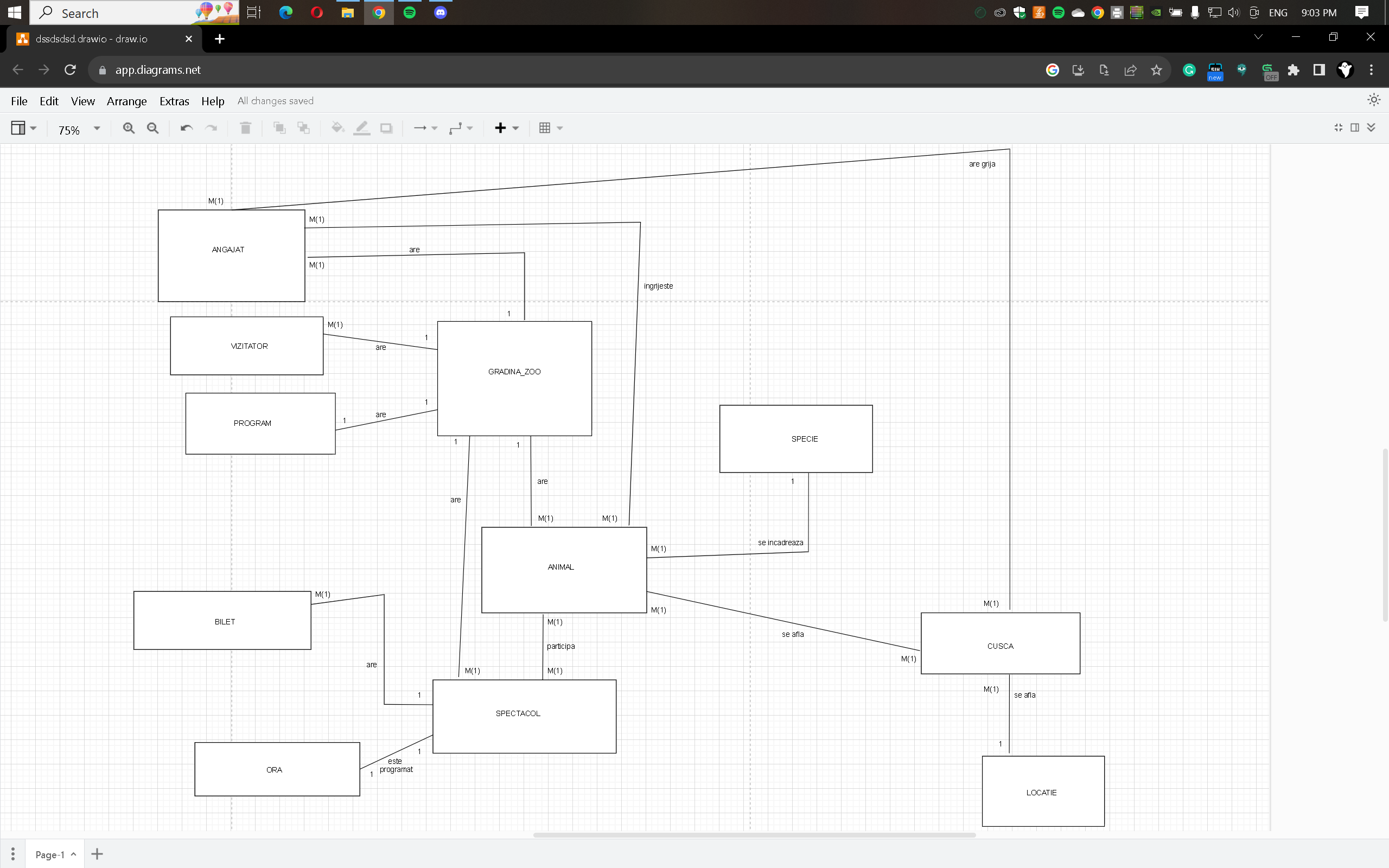
Cheie primară formată din combinația coloanelor ID\_ANIMAL, ID\_ANGAJAT și ID\_CUSCA.

Tabela ORA:

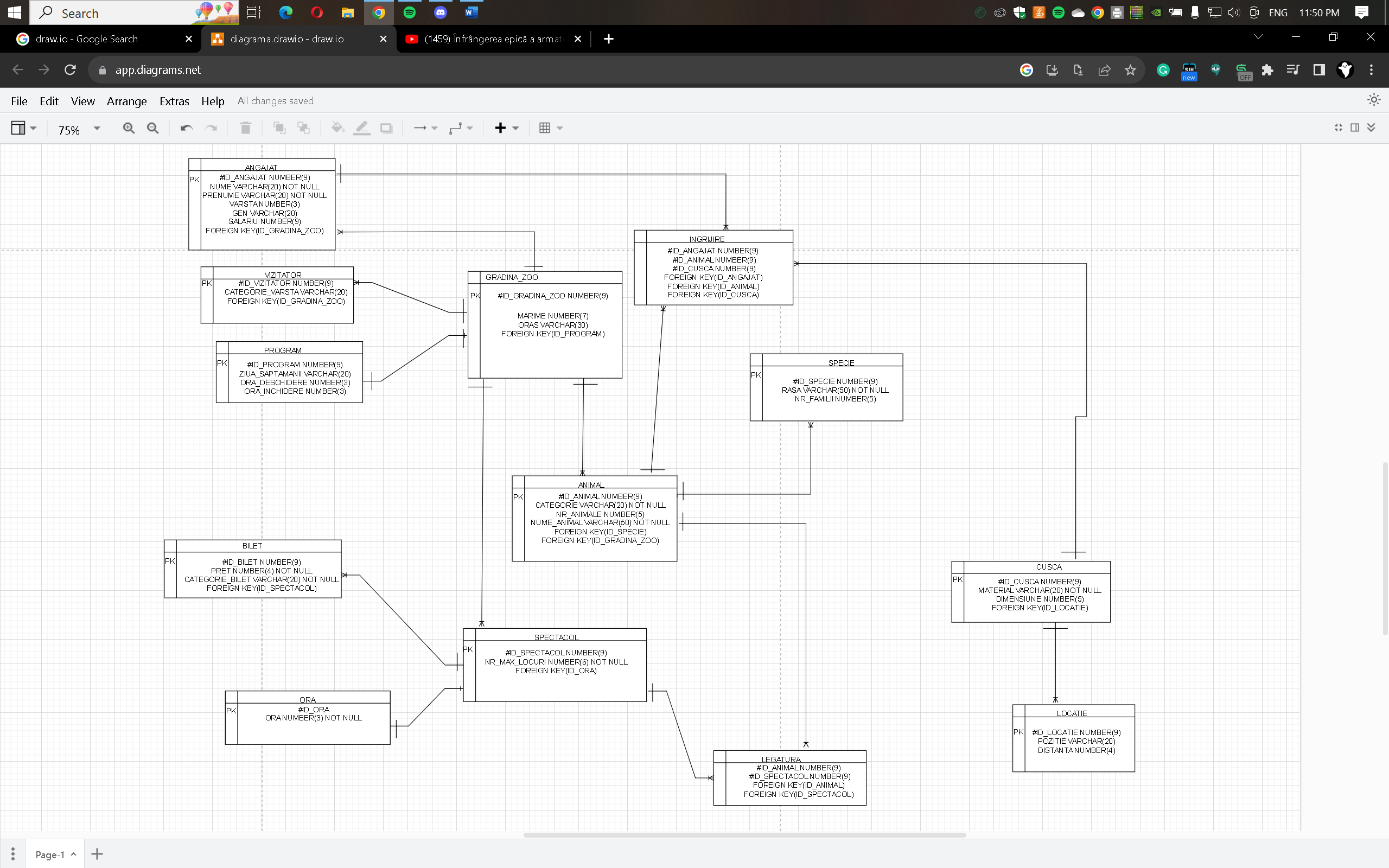
ID\_ORA: Număr întreg de maximum 3 cifre, cheie primară.

ORA: Număr întreg de maximum 3 cifre, obligatoriu (NOT NULL).

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



7. 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.



8. Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale proiectate la punctul 7.

GRADINA\_ZOO(id\_gradina\_zoo#, marime, oras, id\_animal, id\_angajat, id\_vizitator, id\_program)

ANGAJAT(id\_angajat#, nume, prenume, varsta, gen, salariu)

ANIMAL(id\_animal#, categorie, nr\_animale, nume\_animal, id\_specie)

BILET(id\_bilet#, pret, categorie\_bilet)

SPECTACOL(id\_spectacol#, nr\_max\_locuri, id\_bilet, id\_ora)

LEGATURA(id\_animal#, id\_spectacol#, id\_animal, id\_spectacol)

SPECIE(id\_specie#, rasa, nr\_familii)

CUSCA(id\_cusca#, material, dimensiune, id\_locatie)

LOCATIE(id\_locatie#, pozitie, distanta)

VIZITATOR(id\_vizitator#, categorie\_varsta)

PROGRAM(id\_program#, ziua\_saptamanii, ora\_deschidere, ora\_inchidere)

INGRIJIRE(id\_angajat#, id\_animal#, id\_cusca#)

ORA(id\_ora#, ora)

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

Tabela ANIMAL- NON FN1:

|  |  |
| --- | --- |
| #id\_animal | id\_specie |
| 1 | 1, 2 |
| 2 | 3, 4 |

Transformare in FN1:

|  |  |
| --- | --- |
| #id\_animal | id\_specie |
| 1 | 1 |
| 1 | 2 |
| 2 | 3 |
| 2 | 4 |

Tabela SPECTACOL- NON FN2:

|  |  |  |
| --- | --- | --- |
| #id\_spectacol | nr\_max\_locuri | id\_bilet |
| 1 | 100 | 7 |
| 2 | 140 | 4 |
| 1 | 130 | 3 |
| 4 | 120 | 7 |
| 4 | 100 | 7 |
| 2 | 150 | 3 |

Transformare in FN2( vom crea 2 tabele ):

|  |  |
| --- | --- |
| #id\_spectacol | nr\_max\_locuri |
| 1 | 100 |
| 2 | 140 |
| 1 | 130 |
| 4 | 120 |
| 4 | 100 |

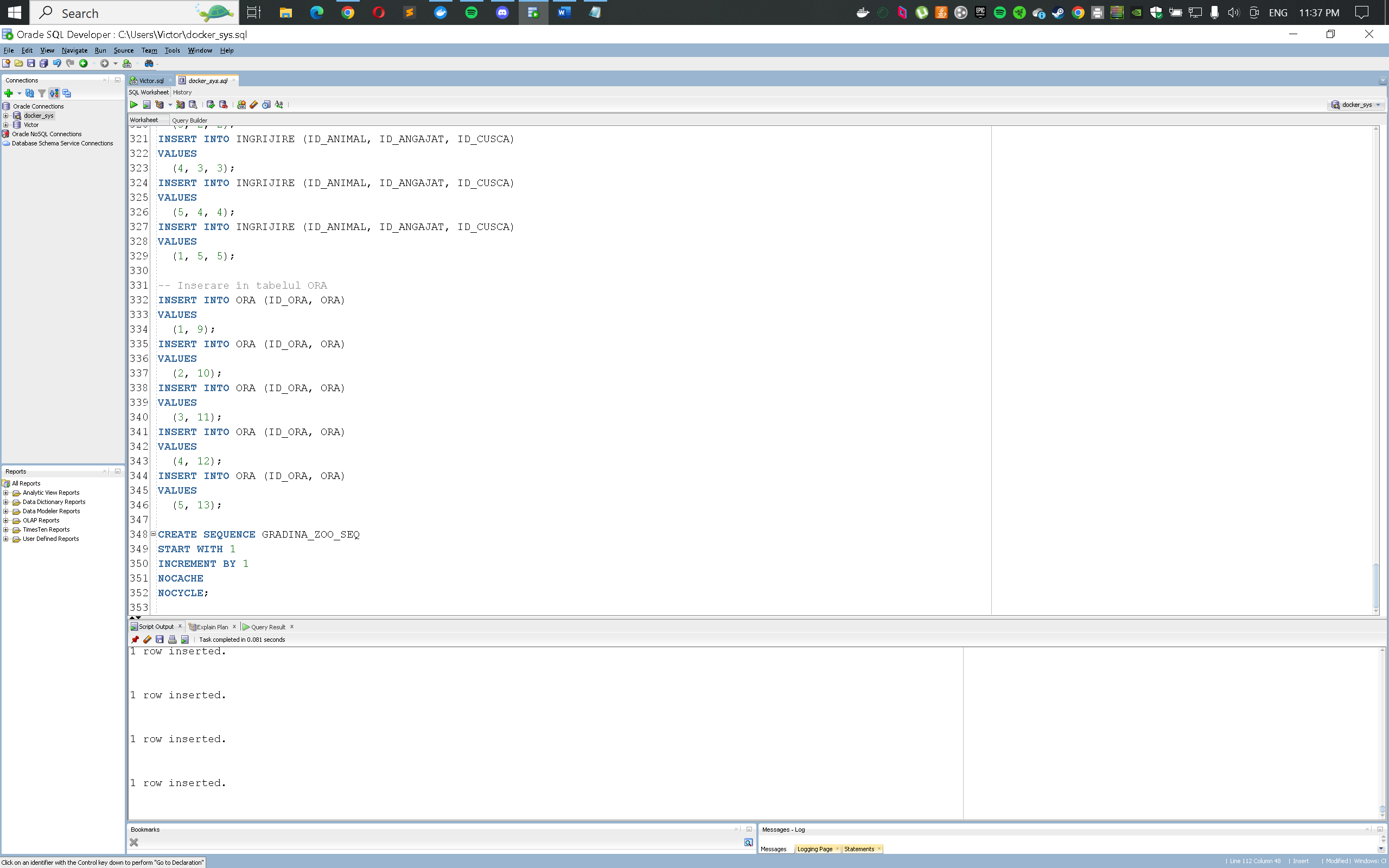
|  |  |
| --- | --- |
| nr\_max\_locuri | Id\_bilet |
| 100 | 7 |
| 140 | 4 |
| 120 | 7 |
| 130 | 3 |
| 150 | 3 |

Transformarea in FN3 se poate realiza astfel:

Cream tabelul LEGATURA. Tabelul LEGATURA are o cheie primara compozita care consta in doua chei straine(ID\_ANIMAL, ID\_SPECTACOL). Cheile straine fac trimitere catre tabelelle ANIMAL si SPECTACOL. Aceste este un NON-FN3, deoarece cheia primara a LEGATURA este compusa din doua chei straine, cee ace incalca prima forma normala.

Pentru a transforma LEGATURA intr-un table FN3, trebuie adaugata o coloanal separata ca cheie primara, iar cheile strained trebuie utilizate pentru a stabilii relatiile intre tabele. De exemplu o noua coloana ID\_LEGATURA ar putea fi adaugata ca cheie primara a tabelului LEGATURA. Apoi cheile straine ID\_ANIMAL, ID\_SPECTACOL ar putea fi utilizate pentru a stabili relatii intre tabelele ANIMAL si SPECTACOL.

10. Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 11).



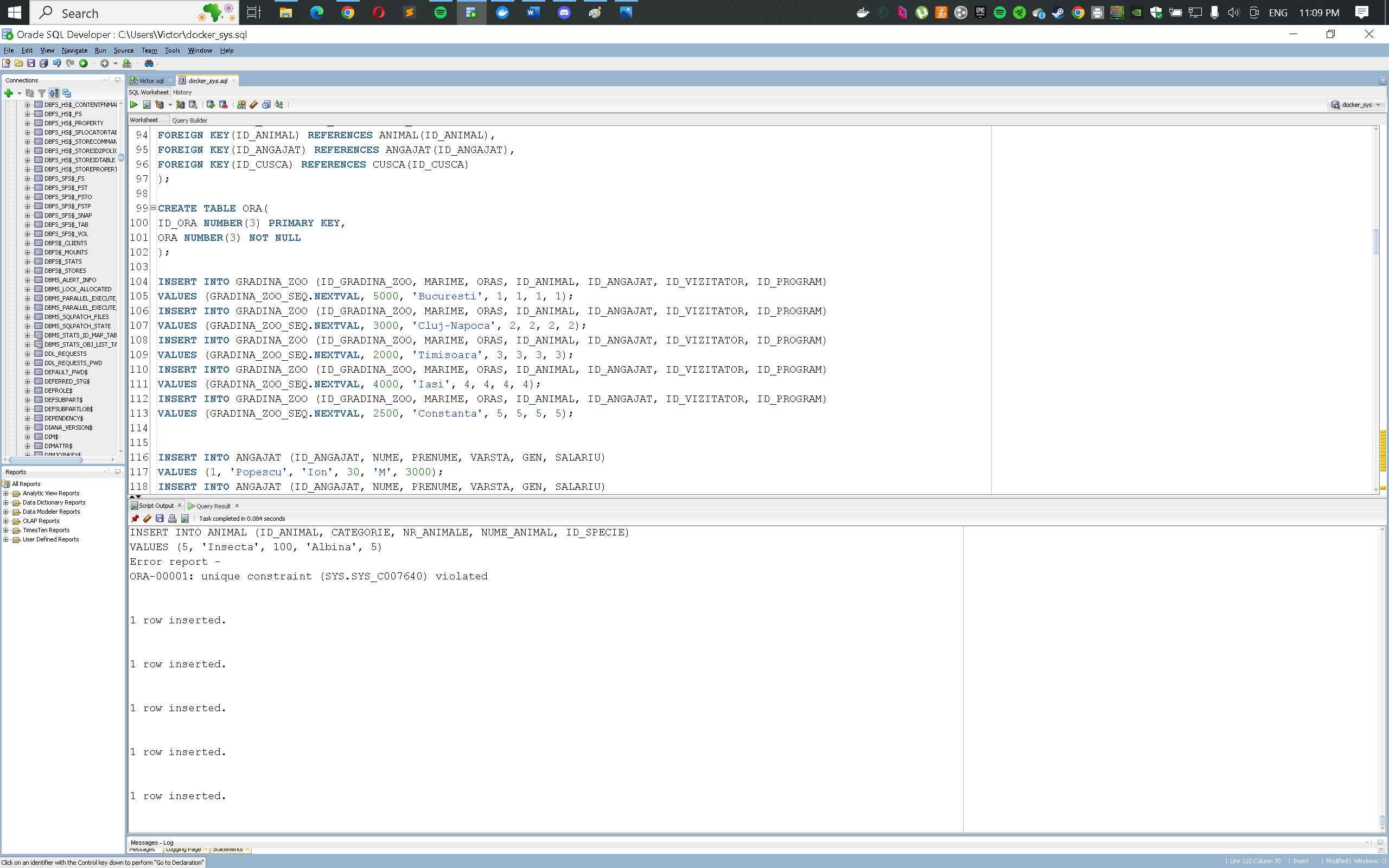
CREATE SEQUENCE GRADINA\_ZOO\_SEQ

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;



INSERT INTO GRADINA\_ZOO (ID\_GRADINA\_ZOO, MARIME, ORAS, ID\_ANIMAL, ID\_ANGAJAT, ID\_VIZITATOR, ID\_PROGRAM)

VALUES (GRADINA\_ZOO\_SEQ.NEXTVAL, 5000, 'Bucuresti', 1, 1, 1, 1);

INSERT INTO GRADINA\_ZOO (ID\_GRADINA\_ZOO, MARIME, ORAS, ID\_ANIMAL, ID\_ANGAJAT, ID\_VIZITATOR, ID\_PROGRAM)

VALUES (GRADINA\_ZOO\_SEQ.NEXTVAL, 3000, 'Cluj-Napoca', 2, 2, 2, 2);

INSERT INTO GRADINA\_ZOO (ID\_GRADINA\_ZOO, MARIME, ORAS, ID\_ANIMAL, ID\_ANGAJAT, ID\_VIZITATOR, ID\_PROGRAM)

VALUES (GRADINA\_ZOO\_SEQ.NEXTVAL, 2000, 'Timisoara', 3, 3, 3, 3);

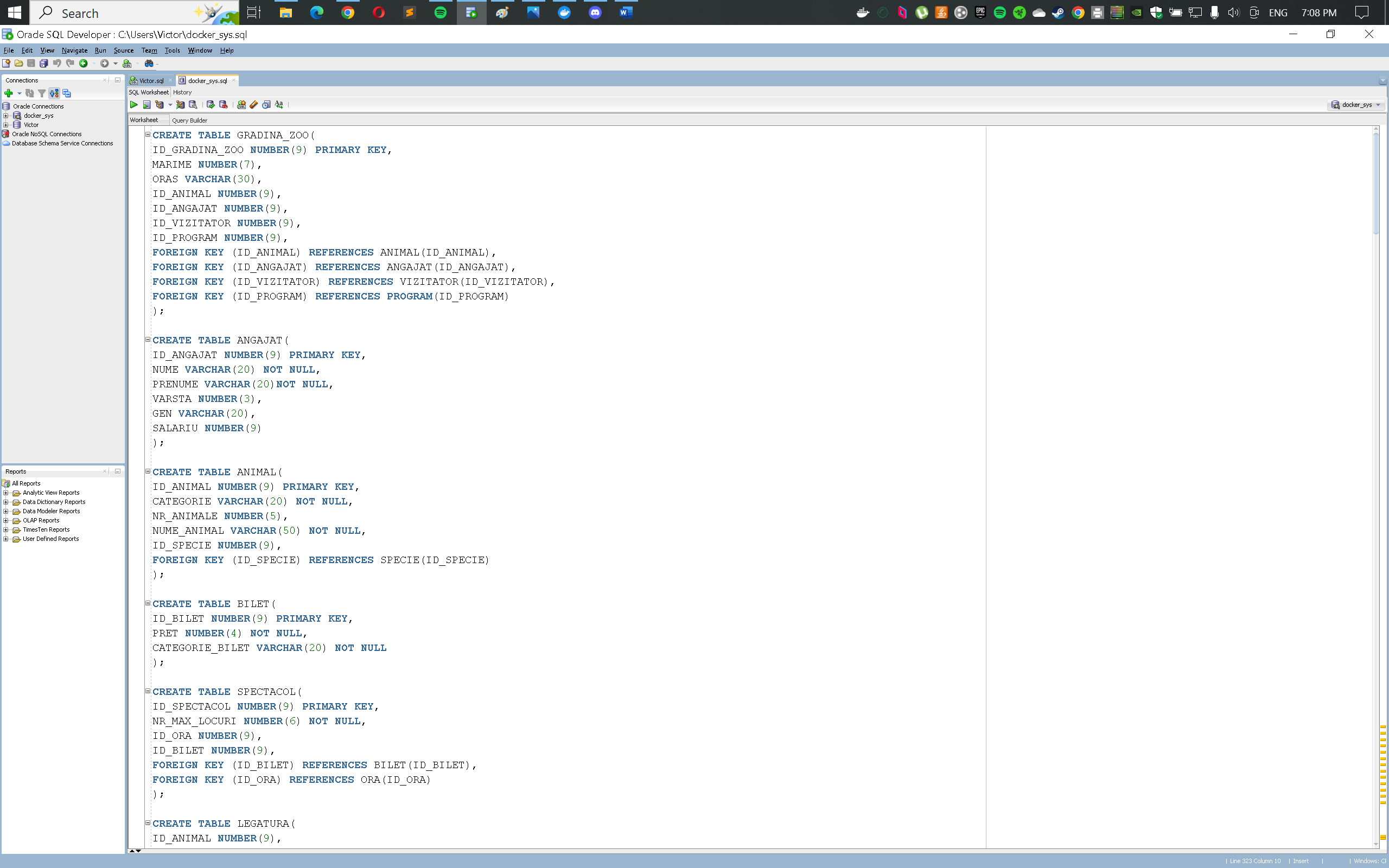
INSERT INTO GRADINA\_ZOO (ID\_GRADINA\_ZOO, MARIME, ORAS, ID\_ANIMAL, ID\_ANGAJAT, ID\_VIZITATOR, ID\_PROGRAM)

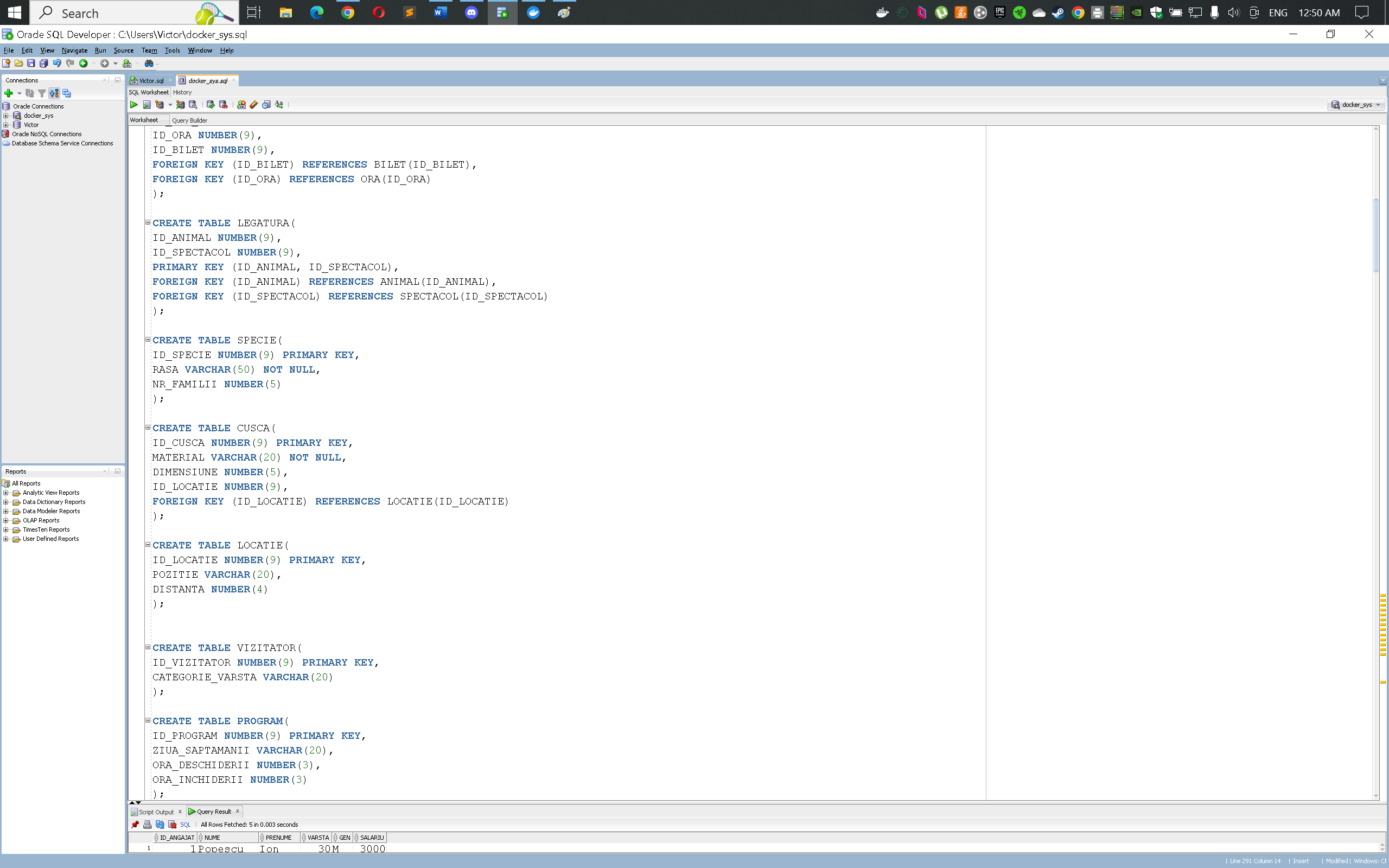
VALUES (GRADINA\_ZOO\_SEQ.NEXTVAL, 4000, 'Iasi', 4, 4, 4, 4);

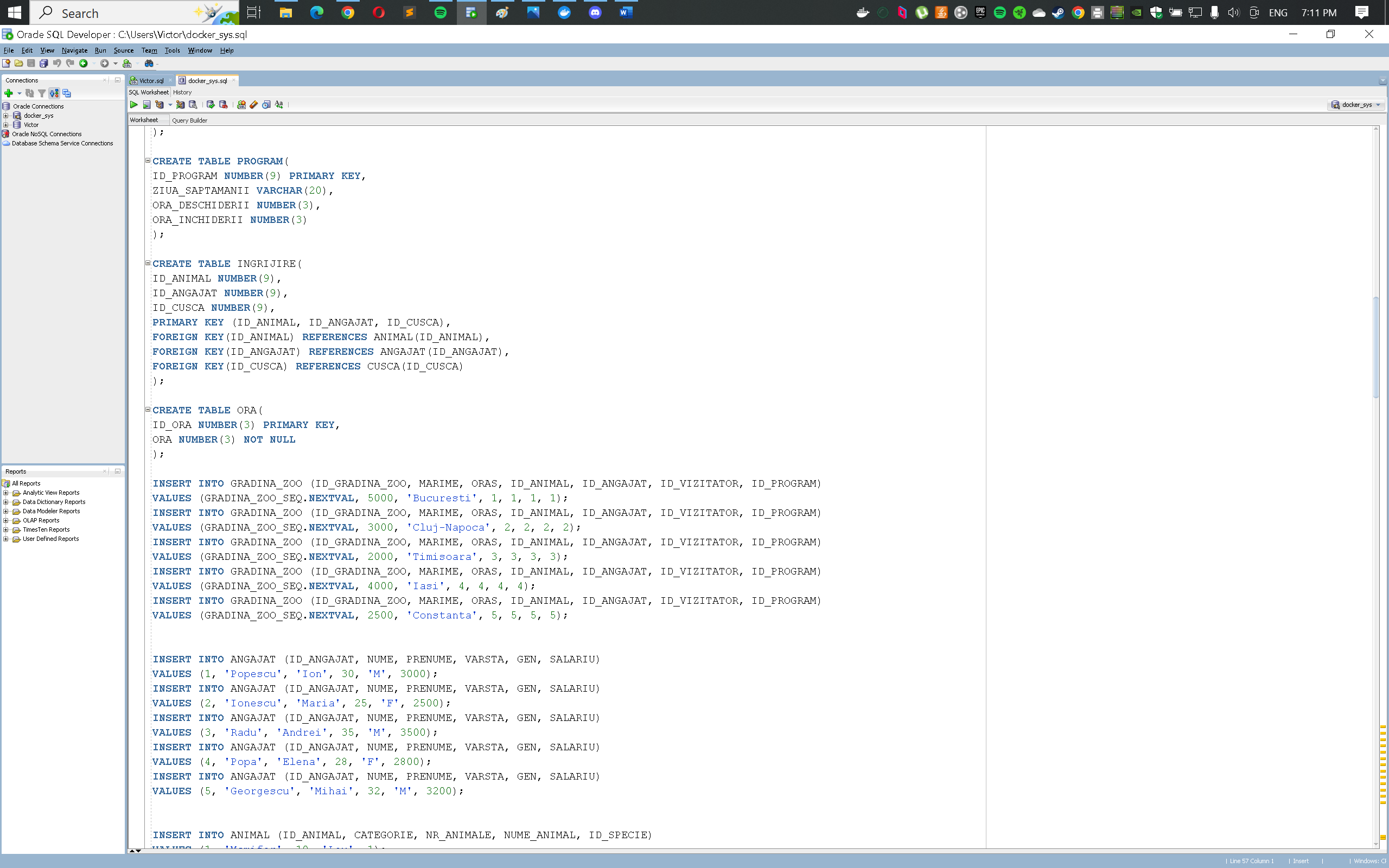
INSERT INTO GRADINA\_ZOO (ID\_GRADINA\_ZOO, MARIME, ORAS, ID\_ANIMAL, ID\_ANGAJAT, ID\_VIZITATOR, ID\_PROGRAM)

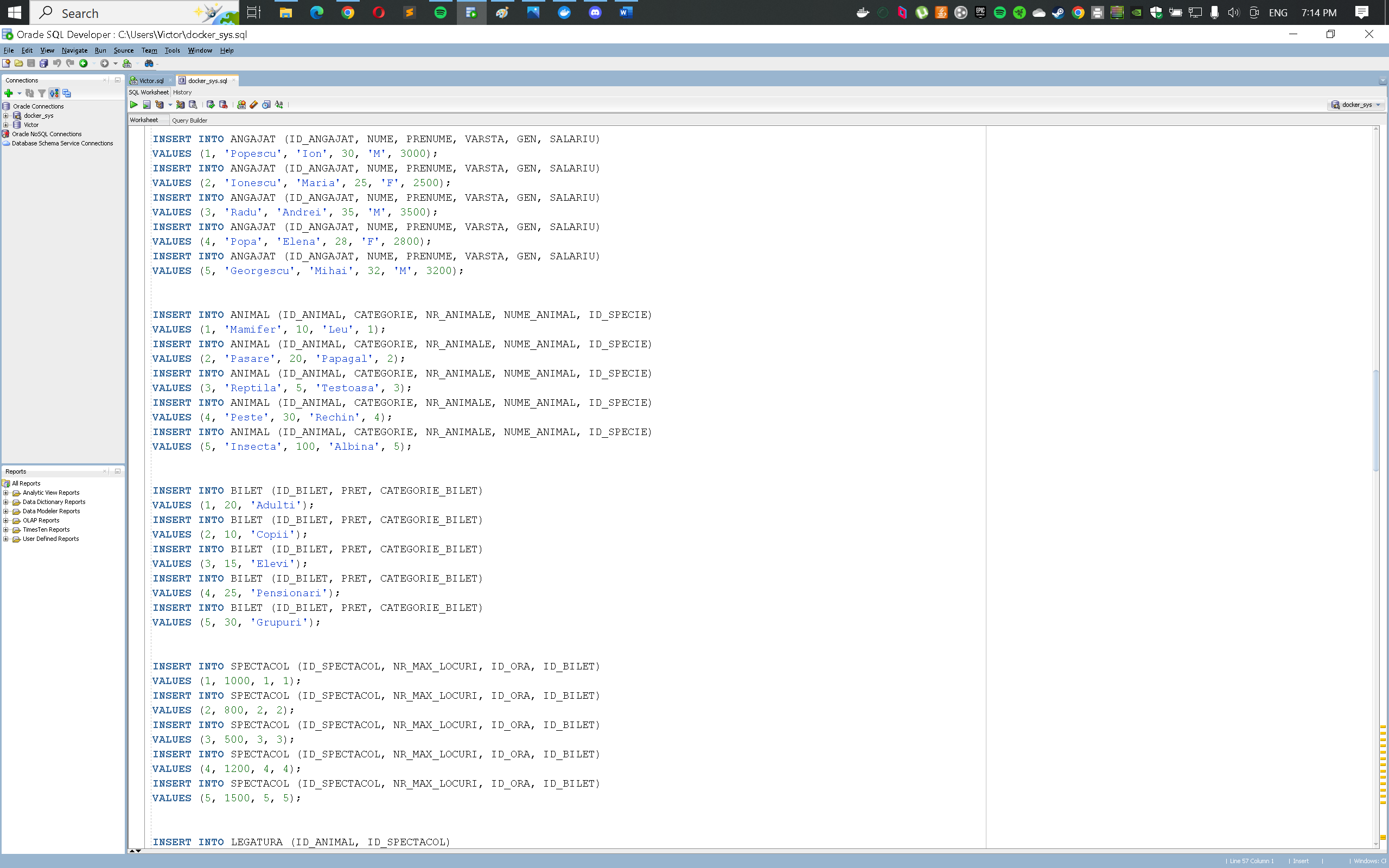
VALUES (GRADINA\_ZOO\_SEQ.NEXTVAL, 2500, 'Constanta', 5, 5, 5, 5);

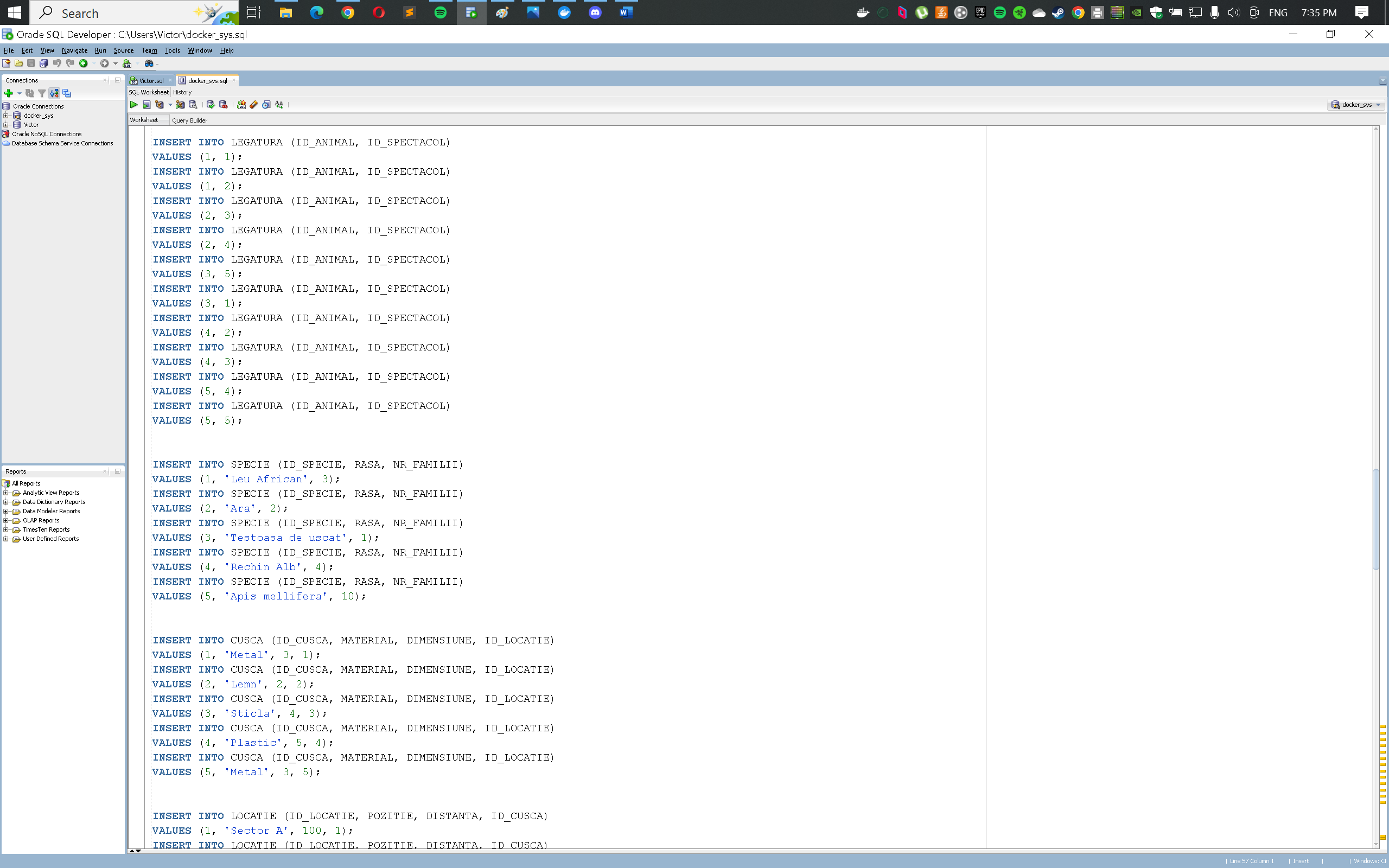
11. Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea (minimum 5 înregistrări în fiecare tabel neasociativ; minimum 10 înregistrări în tabelele asociative).

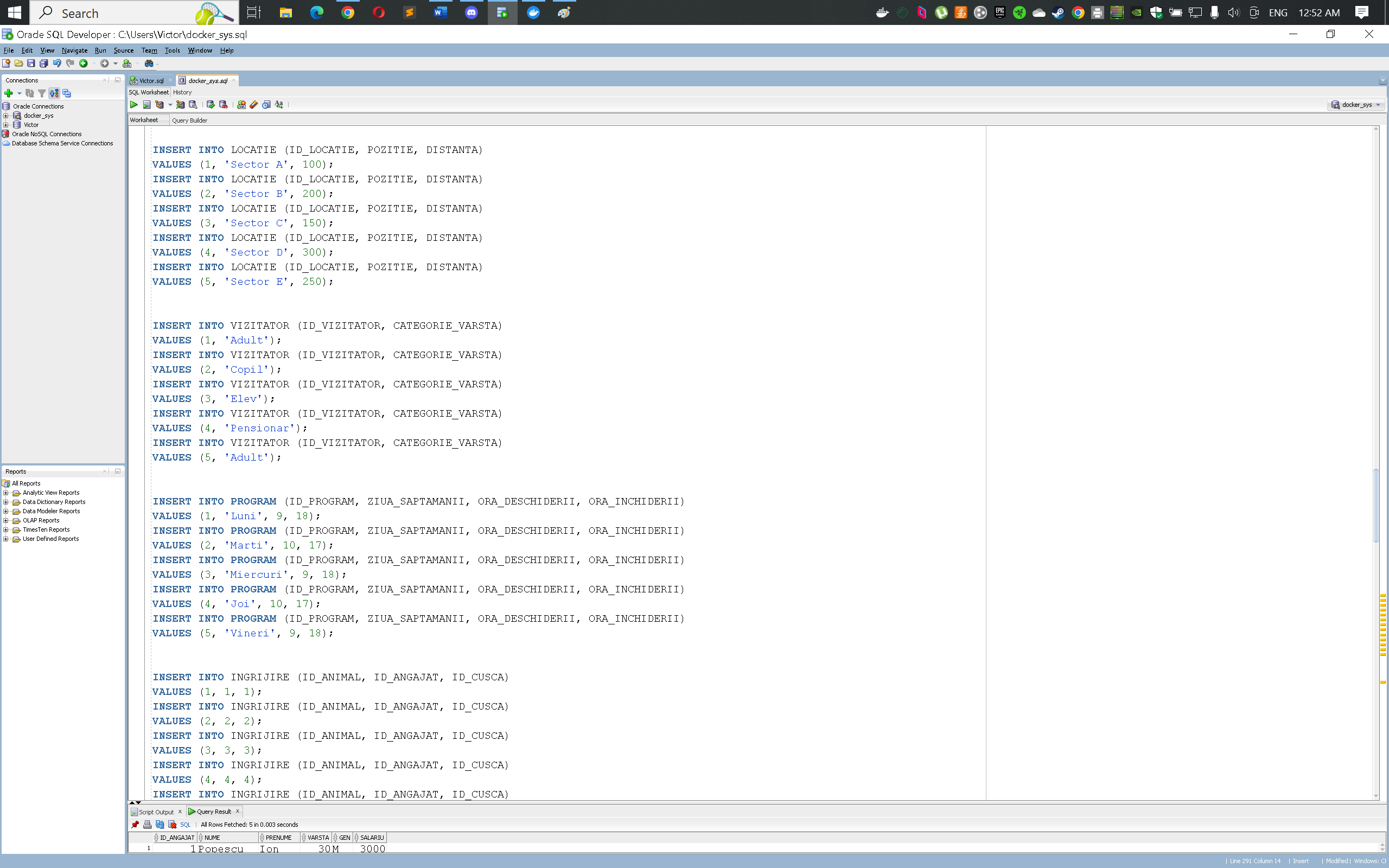


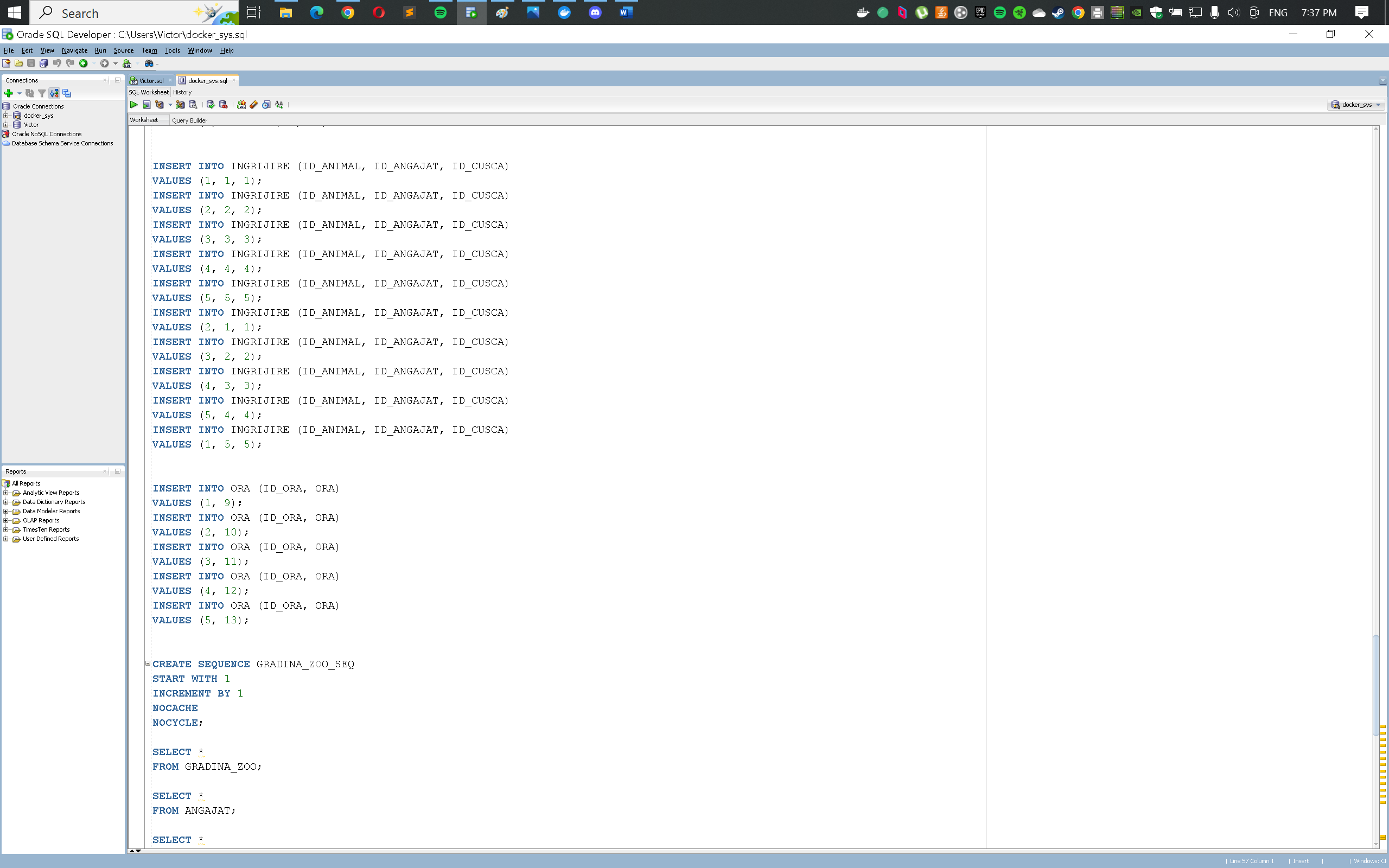


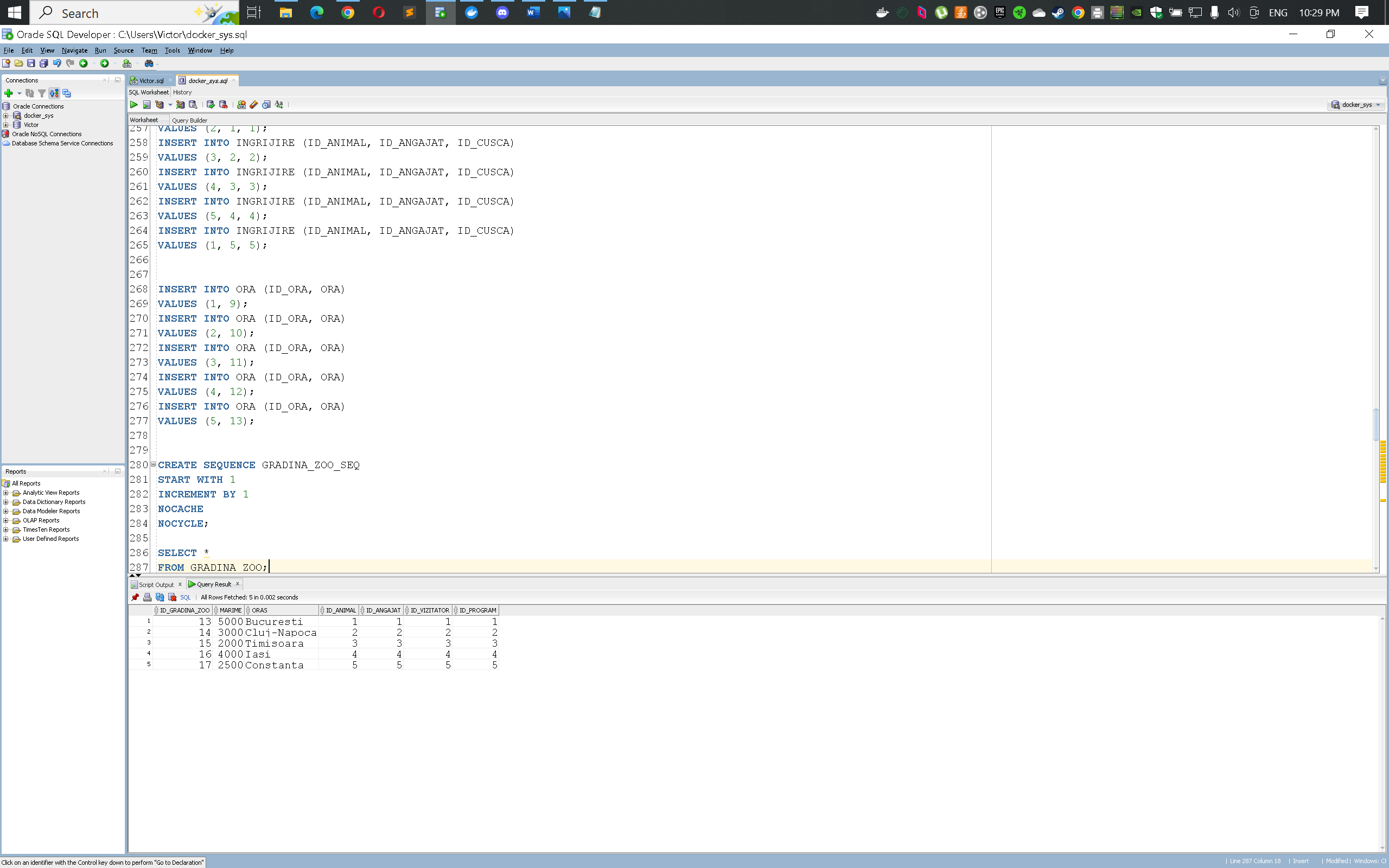


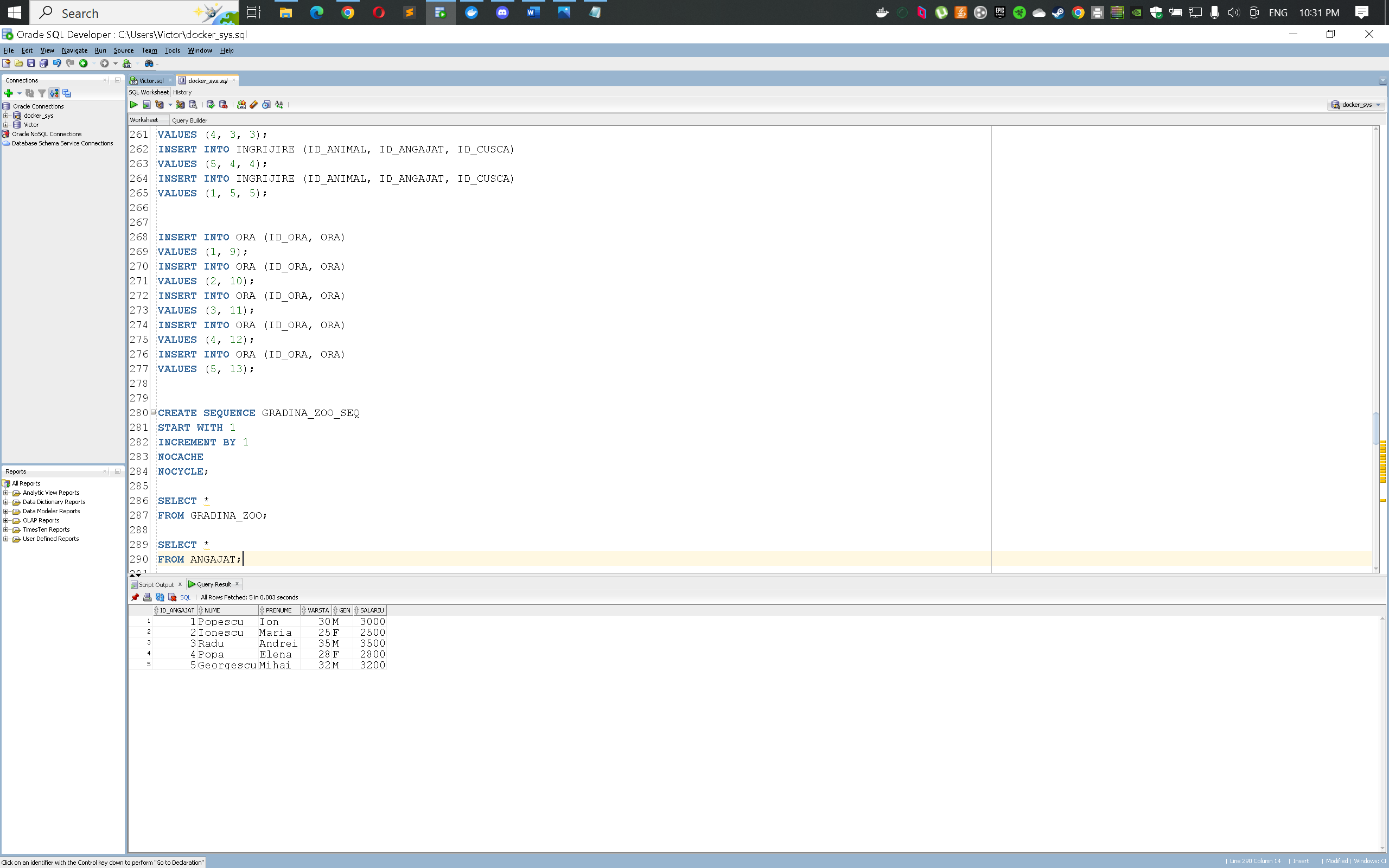


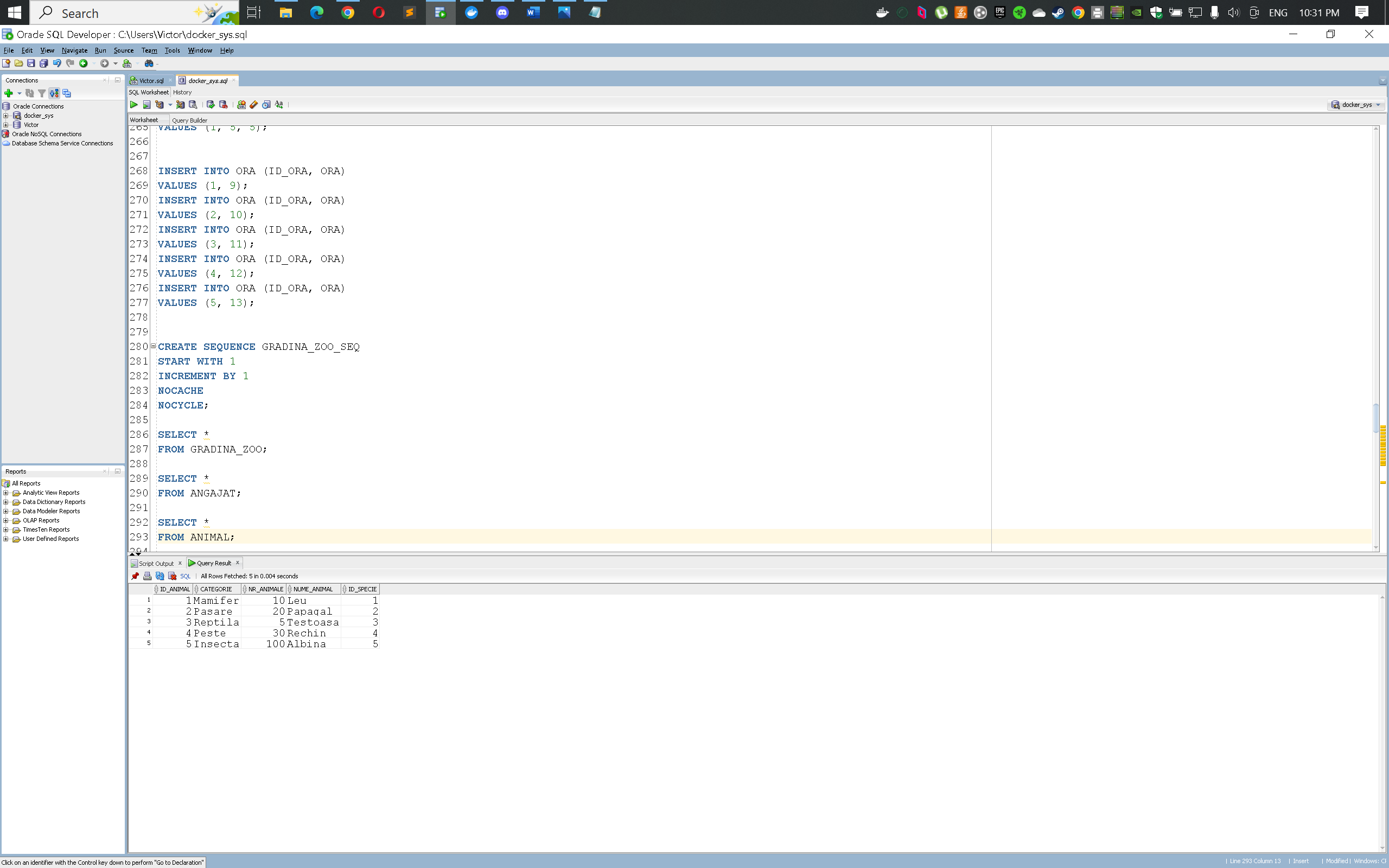


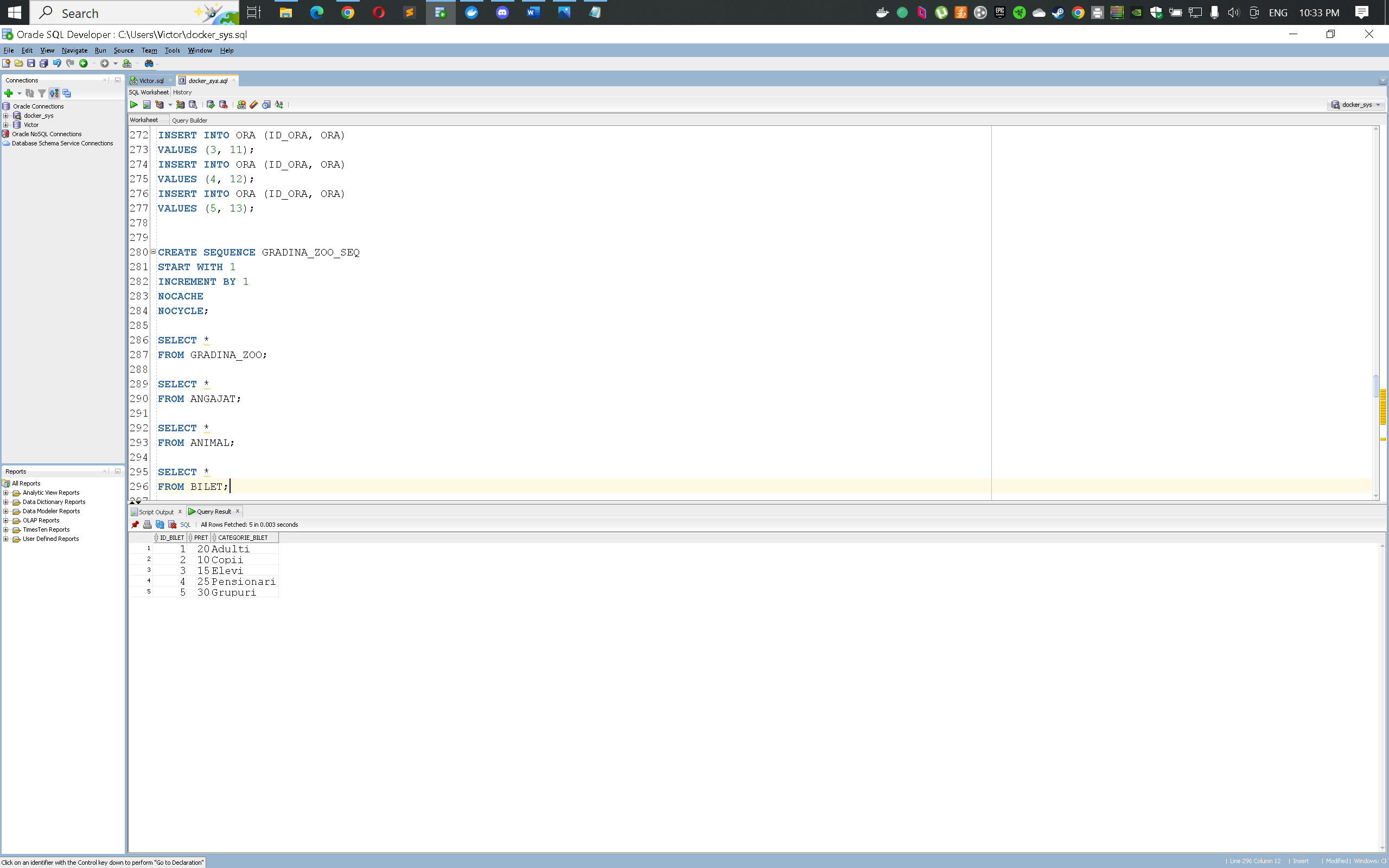


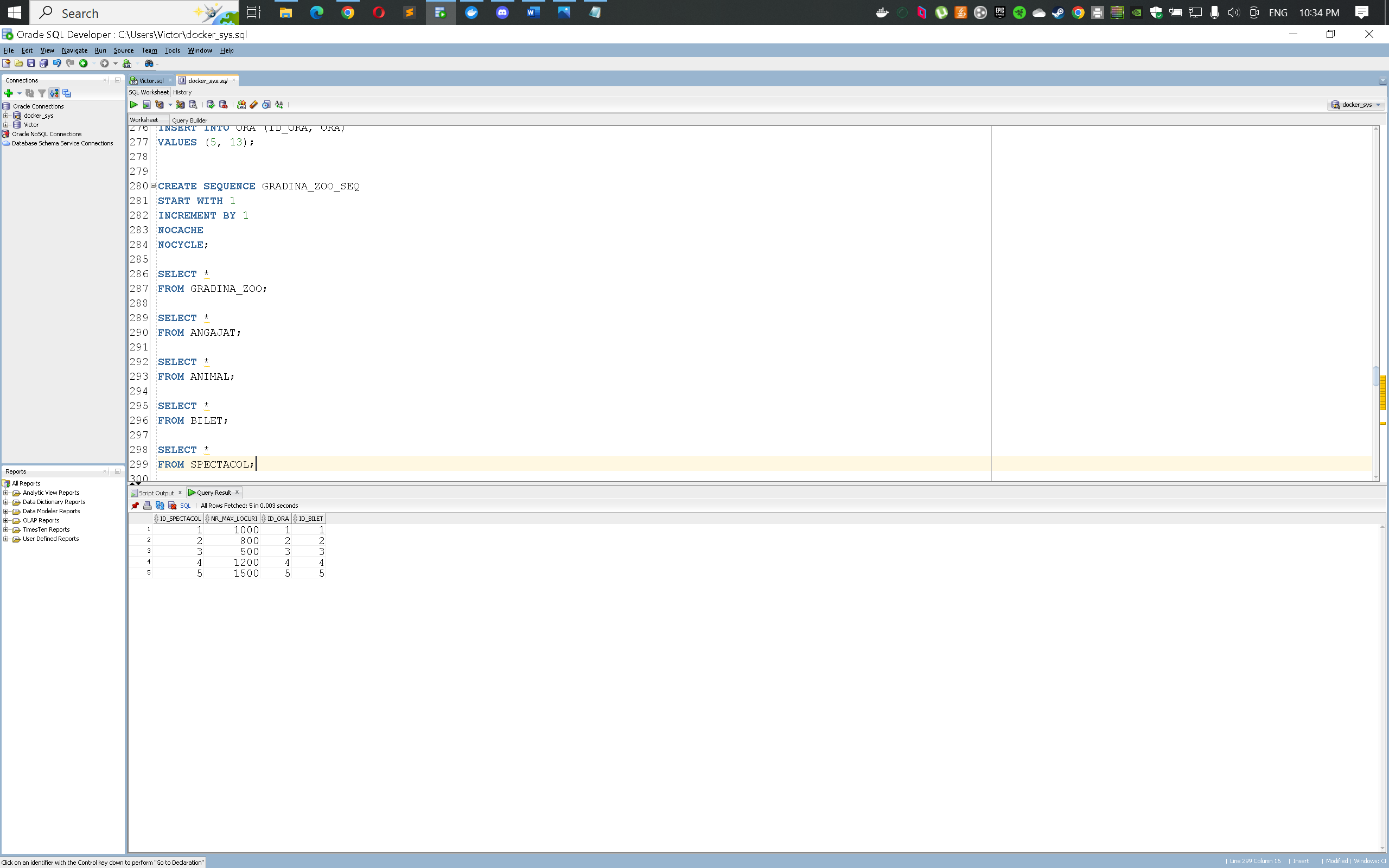


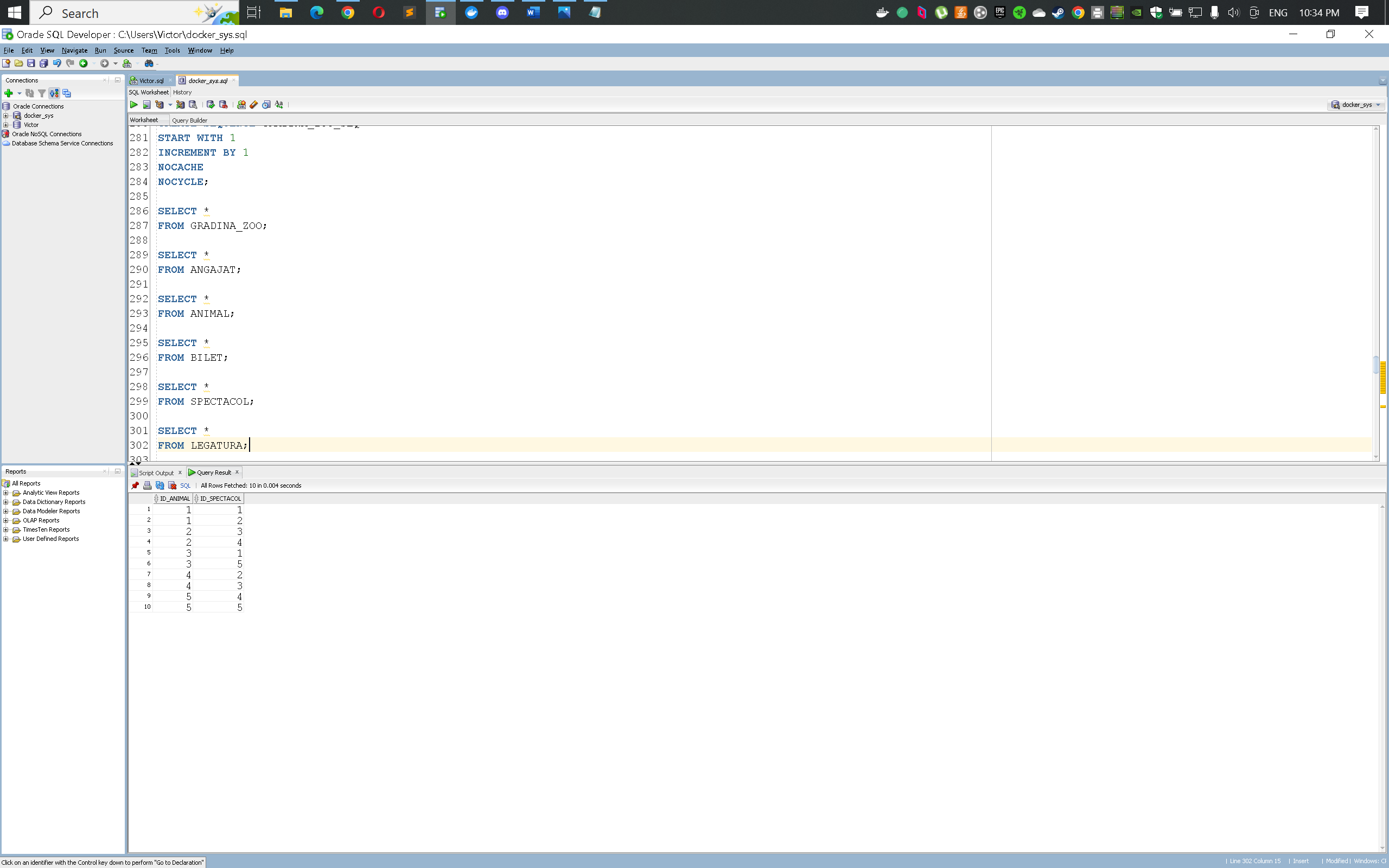


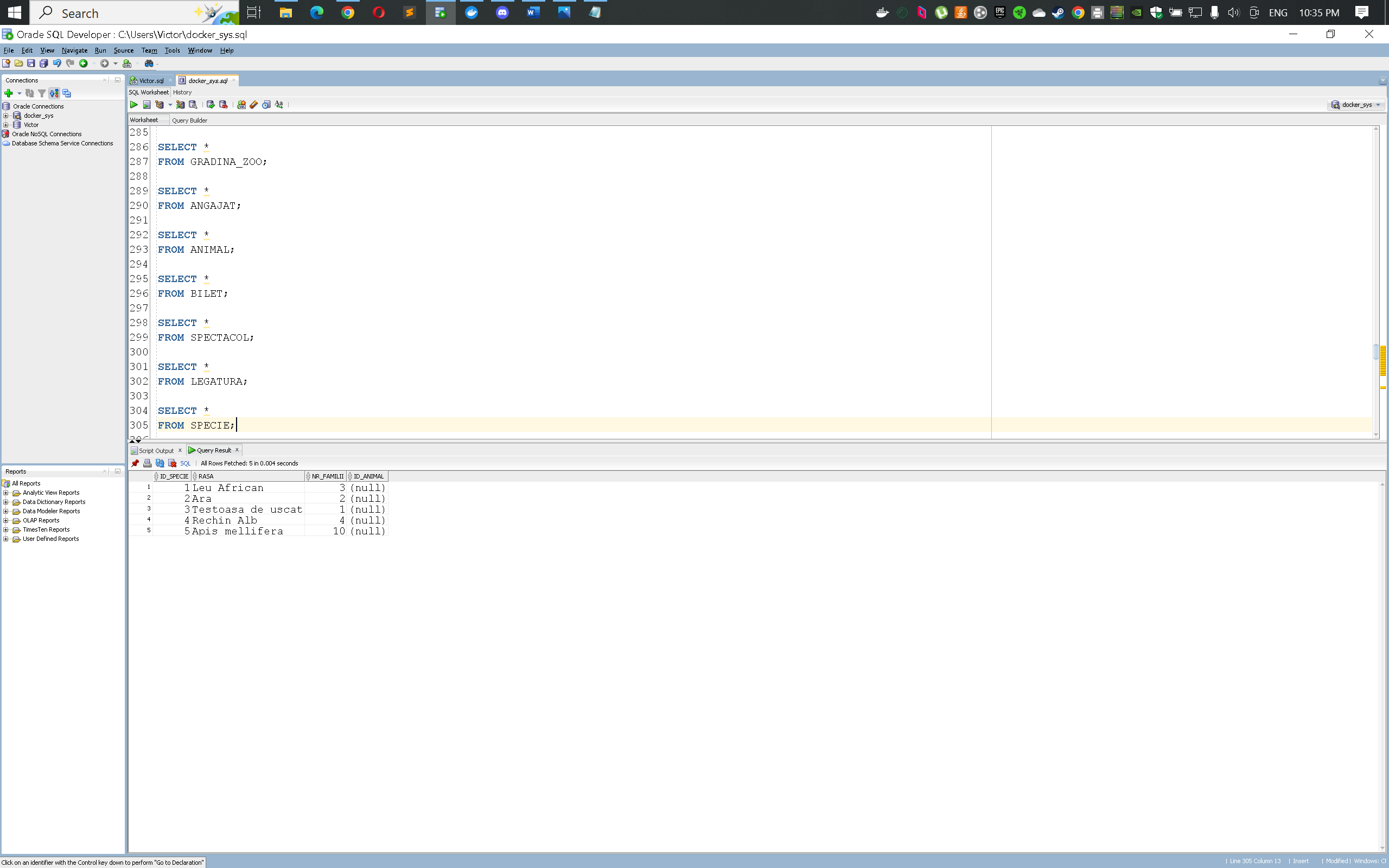


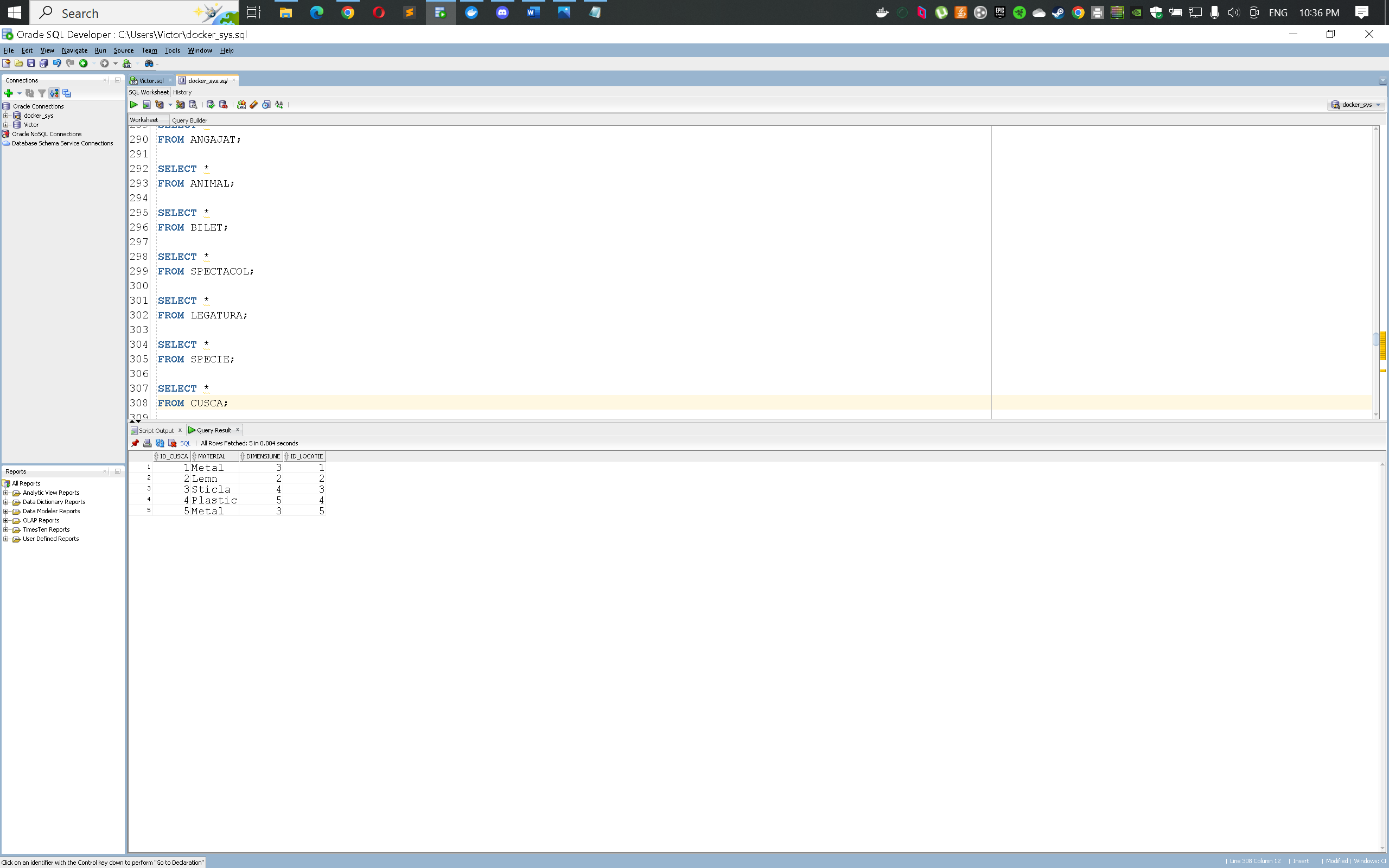


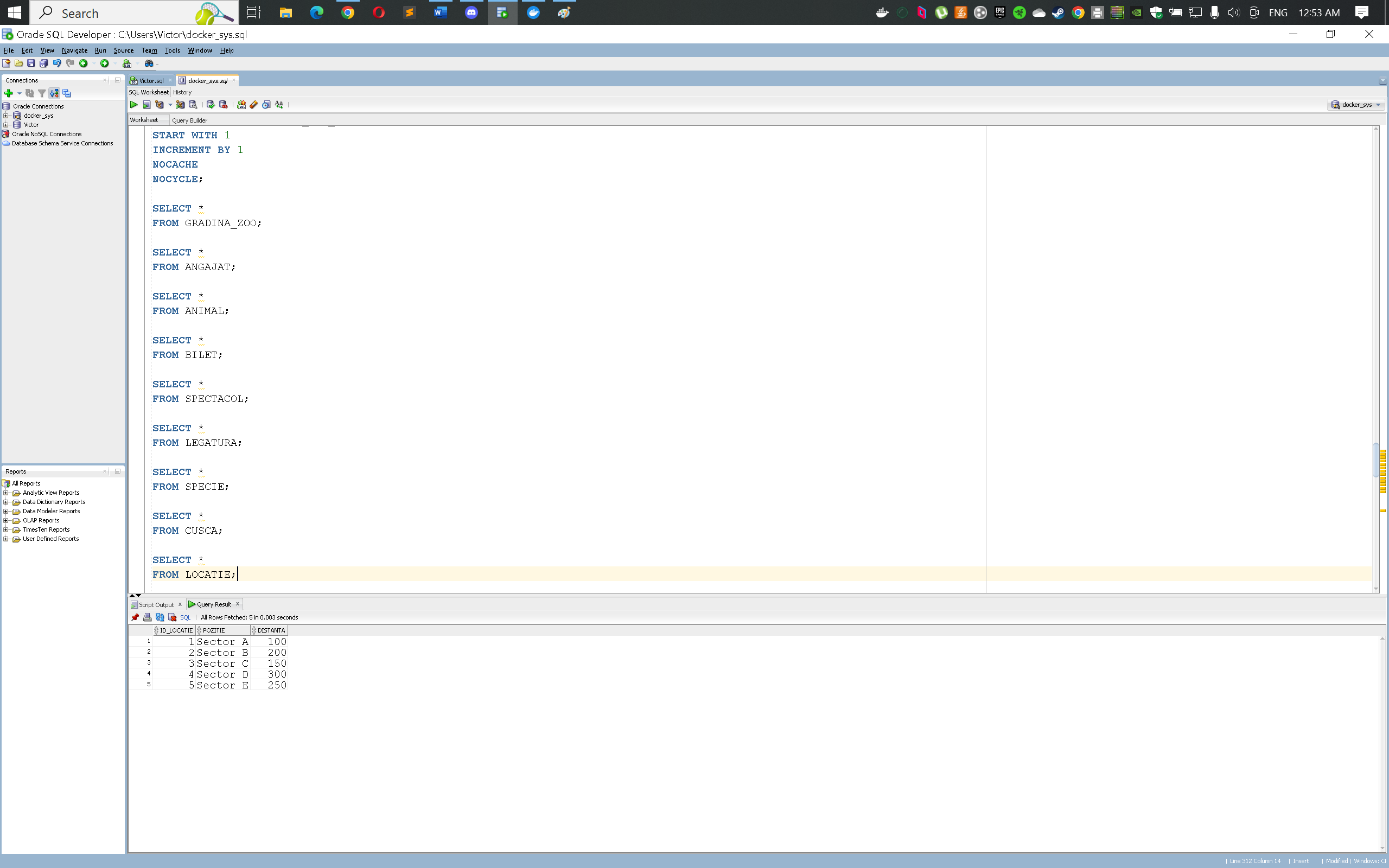


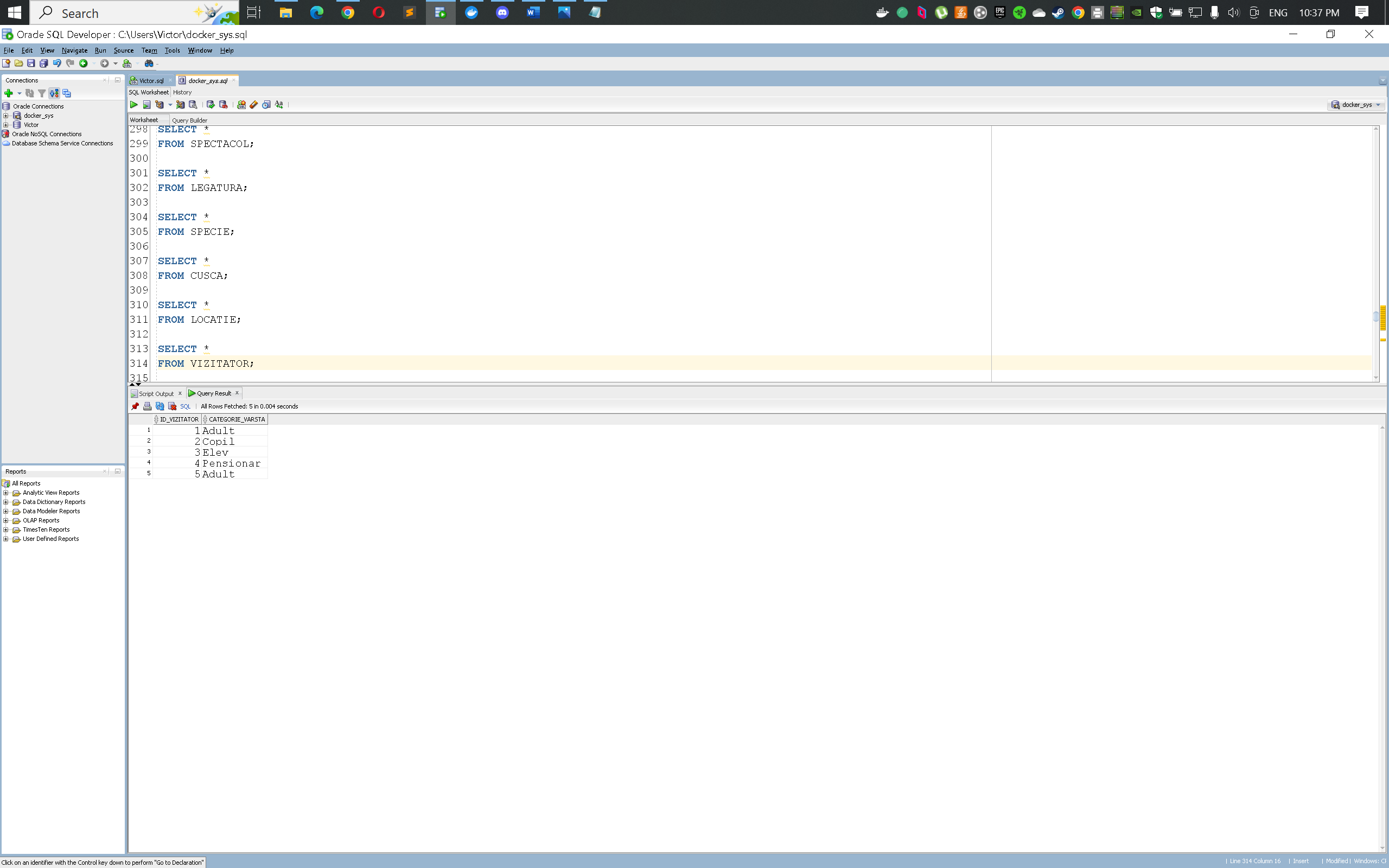


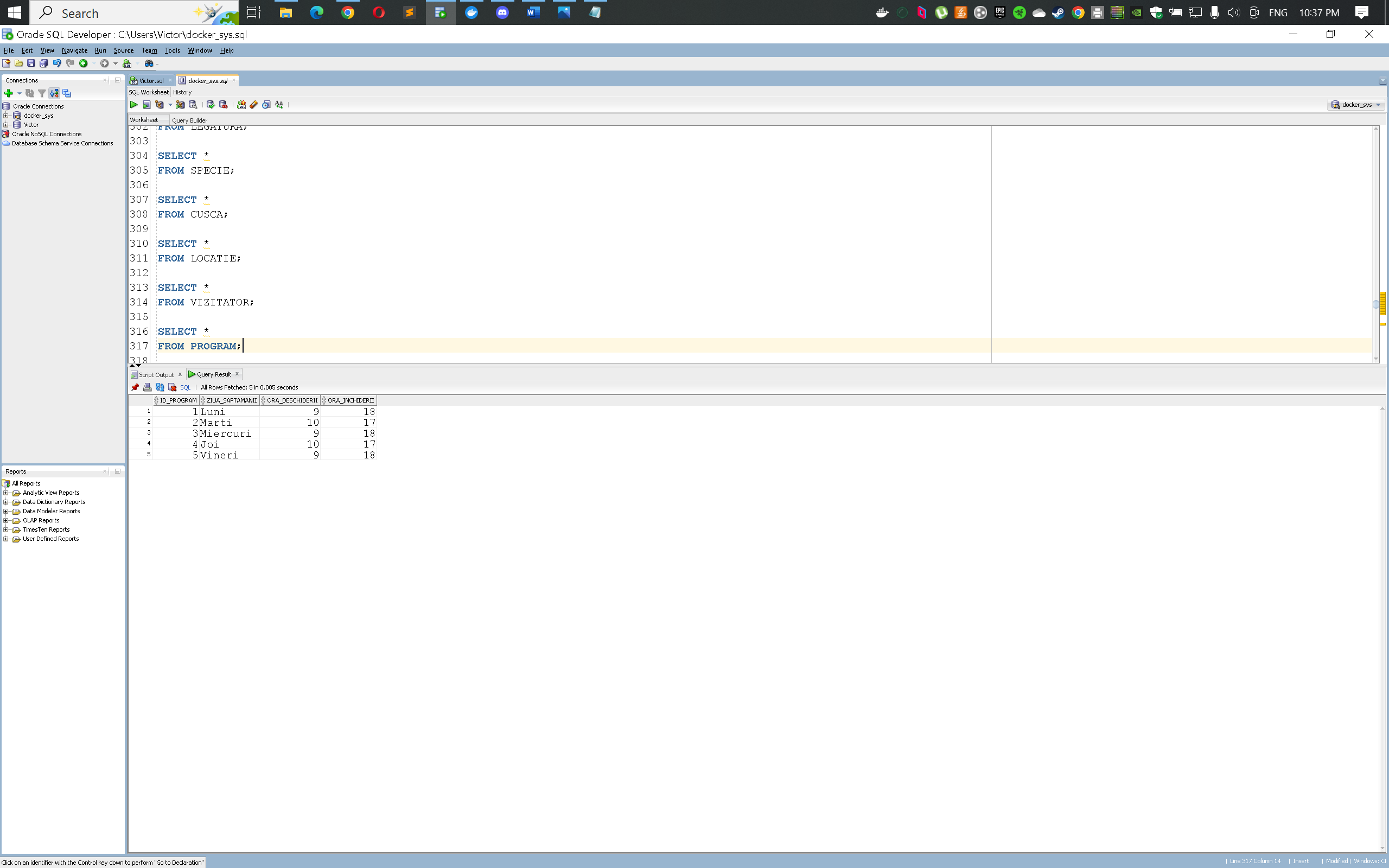


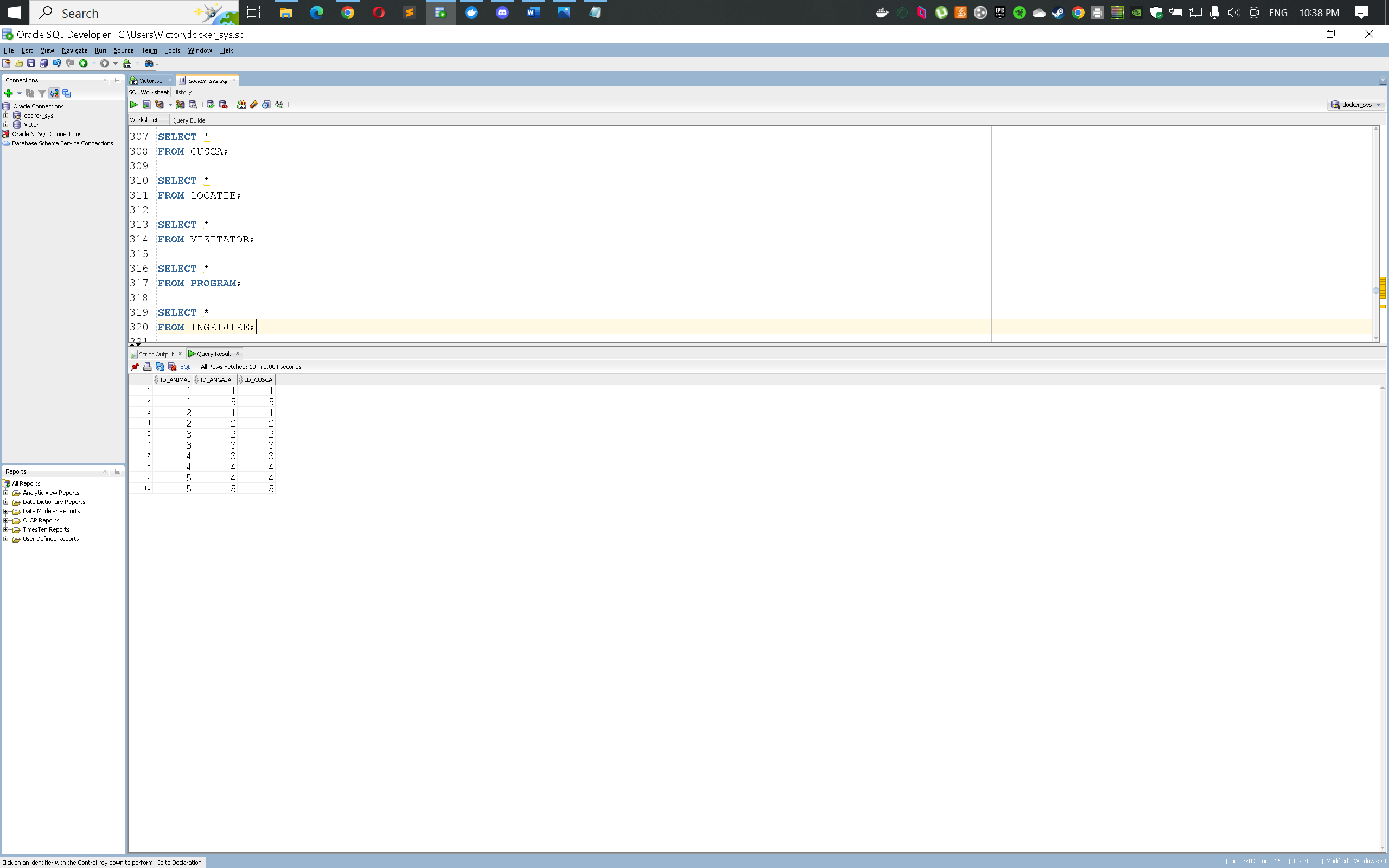


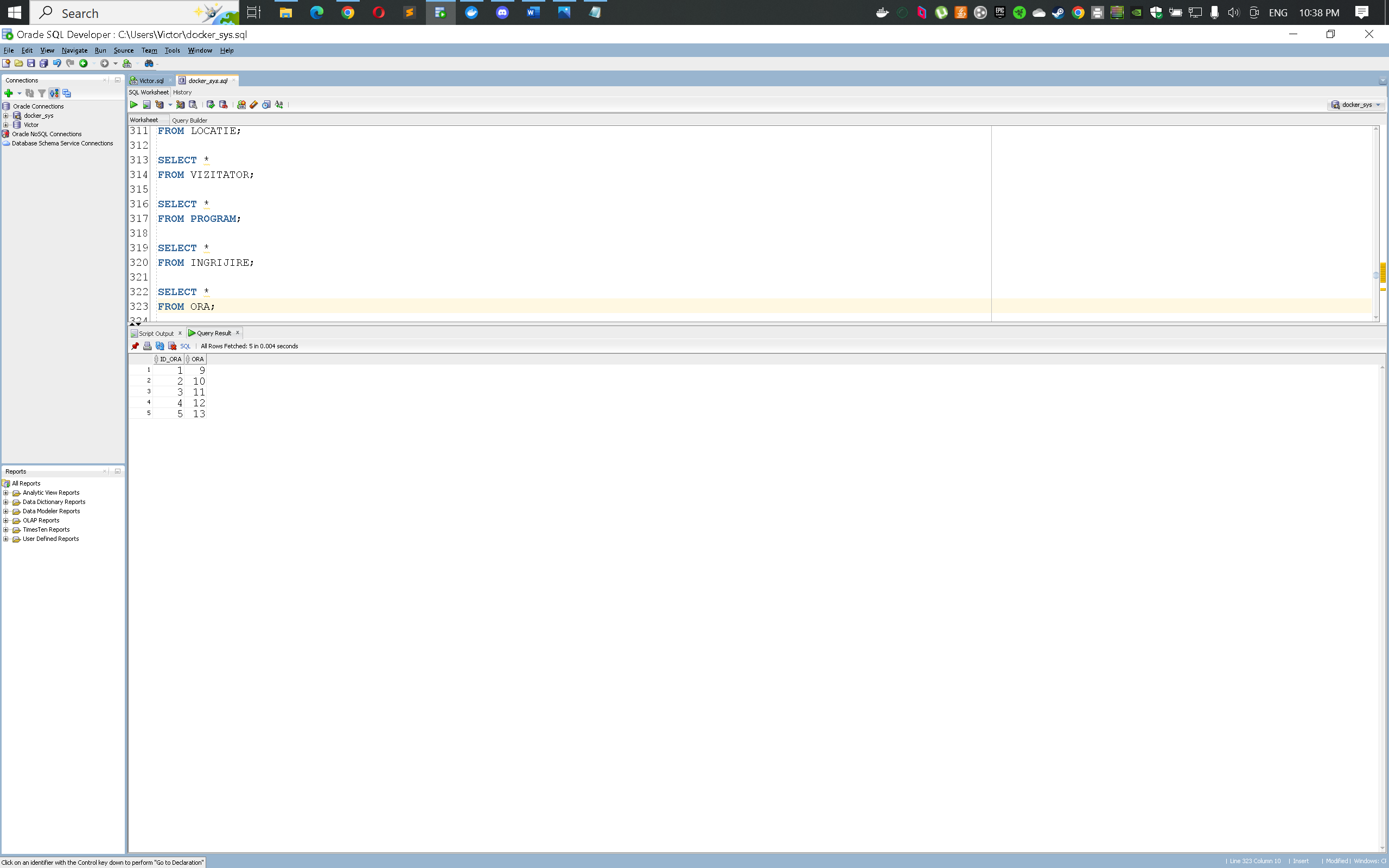












CREATE TABLE GRADINA\_ZOO(

ID\_GRADINA\_ZOO NUMBER(9) PRIMARY KEY,

MARIME NUMBER(7),

ORAS VARCHAR(30),

ID\_ANIMAL NUMBER(9),

ID\_ANGAJAT NUMBER(9),

ID\_VIZITATOR NUMBER(9),

ID\_PROGRAM NUMBER(9),

FOREIGN KEY (ID\_ANIMAL) REFERENCES ANIMAL(ID\_ANIMAL),

FOREIGN KEY (ID\_ANGAJAT) REFERENCES ANGAJAT(ID\_ANGAJAT),

FOREIGN KEY (ID\_VIZITATOR) REFERENCES VIZITATOR(ID\_VIZITATOR),

FOREIGN KEY (ID\_PROGRAM) REFERENCES PROGRAM(ID\_PROGRAM)

);

CREATE TABLE ANGAJAT(

ID\_ANGAJAT NUMBER(9) PRIMARY KEY,

NUME VARCHAR(20) NOT NULL,

PRENUME VARCHAR(20)NOT NULL,

VARSTA NUMBER(3),

GEN VARCHAR(20),

SALARIU NUMBER(9)

);

CREATE TABLE ANIMAL(

ID\_ANIMAL NUMBER(9) PRIMARY KEY,

CATEGORIE VARCHAR(20) NOT NULL,

NR\_ANIMALE NUMBER(5),

NUME\_ANIMAL VARCHAR(50) NOT NULL,

ID\_SPECIE NUMBER(9),

FOREIGN KEY (ID\_SPECIE) REFERENCES SPECIE(ID\_SPECIE)

);

CREATE TABLE BILET(

ID\_BILET NUMBER(9) PRIMARY KEY,

PRET NUMBER(4) NOT NULL,

CATEGORIE\_BILET VARCHAR(20) NOT NULL

);

CREATE TABLE SPECTACOL(

ID\_SPECTACOL NUMBER(9) PRIMARY KEY,

NR\_MAX\_LOCURI NUMBER(6) NOT NULL,

ID\_ORA NUMBER(9),

ID\_BILET NUMBER(9),

FOREIGN KEY (ID\_BILET) REFERENCES BILET(ID\_BILET),

FOREIGN KEY (ID\_ORA) REFERENCES ORA(ID\_ORA)

);

CREATE TABLE LEGATURA(

ID\_ANIMAL NUMBER(9),

ID\_SPECTACOL NUMBER(9),

PRIMARY KEY (ID\_ANIMAL, ID\_SPECTACOL),

FOREIGN KEY (ID\_ANIMAL) REFERENCES ANIMAL(ID\_ANIMAL),

FOREIGN KEY (ID\_SPECTACOL) REFERENCES SPECTACOL(ID\_SPECTACOL)

);

CREATE TABLE SPECIE(

ID\_SPECIE NUMBER(9) PRIMARY KEY,

RASA VARCHAR(50) NOT NULL,

NR\_FAMILII NUMBER(5)

);

CREATE TABLE CUSCA(

ID\_CUSCA NUMBER(9) PRIMARY KEY,

MATERIAL VARCHAR(20) NOT NULL,

DIMENSIUNE NUMBER(5),

ID\_LOCATIE NUMBER(9),

FOREIGN KEY (ID\_LOCATIE) REFERENCES LOCATIE(ID\_LOCATIE)

);

CREATE TABLE LOCATIE(

ID\_LOCATIE NUMBER(9) PRIMARY KEY,

POZITIE VARCHAR(20),

DISTANTA NUMBER(4)

);

CREATE TABLE VIZITATOR(

ID\_VIZITATOR NUMBER(9) PRIMARY KEY,

CATEGORIE\_VARSTA VARCHAR(20)

);

CREATE TABLE PROGRAM(

ID\_PROGRAM NUMBER(9) PRIMARY KEY,

ZIUA\_SAPTAMANII VARCHAR(20),

ORA\_DESCHIDERII NUMBER(3),

ORA\_INCHIDERII NUMBER(3)

);

CREATE TABLE INGRIJIRE(

ID\_ANIMAL NUMBER(9),

ID\_ANGAJAT NUMBER(9),

ID\_CUSCA NUMBER(9),

PRIMARY KEY (ID\_ANIMAL, ID\_ANGAJAT, ID\_CUSCA),

FOREIGN KEY(ID\_ANIMAL) REFERENCES ANIMAL(ID\_ANIMAL),

FOREIGN KEY(ID\_ANGAJAT) REFERENCES ANGAJAT(ID\_ANGAJAT),

FOREIGN KEY(ID\_CUSCA) REFERENCES CUSCA(ID\_CUSCA)

);

CREATE TABLE ORA(

ID\_ORA NUMBER(3) PRIMARY KEY,

ORA NUMBER(3) NOT NULL

);

INSERT INTO GRADINA\_ZOO (ID\_GRADINA\_ZOO, MARIME, ORAS, ID\_ANIMAL, ID\_ANGAJAT, ID\_VIZITATOR, ID\_PROGRAM)

VALUES (GRADINA\_ZOO\_SEQ.NEXTVAL, 5000, 'Bucuresti', 1, 1, 1, 1);

INSERT INTO GRADINA\_ZOO (ID\_GRADINA\_ZOO, MARIME, ORAS, ID\_ANIMAL, ID\_ANGAJAT, ID\_VIZITATOR, ID\_PROGRAM)

VALUES (GRADINA\_ZOO\_SEQ.NEXTVAL, 3000, 'Cluj-Napoca', 2, 2, 2, 2);

INSERT INTO GRADINA\_ZOO (ID\_GRADINA\_ZOO, MARIME, ORAS, ID\_ANIMAL, ID\_ANGAJAT, ID\_VIZITATOR, ID\_PROGRAM)

VALUES (GRADINA\_ZOO\_SEQ.NEXTVAL, 2000, 'Timisoara', 3, 3, 3, 3);

INSERT INTO GRADINA\_ZOO (ID\_GRADINA\_ZOO, MARIME, ORAS, ID\_ANIMAL, ID\_ANGAJAT, ID\_VIZITATOR, ID\_PROGRAM)

VALUES (GRADINA\_ZOO\_SEQ.NEXTVAL, 4000, 'Iasi', 4, 4, 4, 4);

INSERT INTO GRADINA\_ZOO (ID\_GRADINA\_ZOO, MARIME, ORAS, ID\_ANIMAL, ID\_ANGAJAT, ID\_VIZITATOR, ID\_PROGRAM)

VALUES (GRADINA\_ZOO\_SEQ.NEXTVAL, 2500, 'Constanta', 5, 5, 5, 5);

INSERT INTO ANGAJAT (ID\_ANGAJAT, NUME, PRENUME, VARSTA, GEN, SALARIU)

VALUES (1, 'Popescu', 'Ion', 30, 'M', 3000);

INSERT INTO ANGAJAT (ID\_ANGAJAT, NUME, PRENUME, VARSTA, GEN, SALARIU)

VALUES (2, 'Ionescu', 'Maria', 25, 'F', 2500);

INSERT INTO ANGAJAT (ID\_ANGAJAT, NUME, PRENUME, VARSTA, GEN, SALARIU)

VALUES (3, 'Radu', 'Andrei', 35, 'M', 3500);

INSERT INTO ANGAJAT (ID\_ANGAJAT, NUME, PRENUME, VARSTA, GEN, SALARIU)

VALUES (4, 'Popa', 'Elena', 28, 'F', 2800);

INSERT INTO ANGAJAT (ID\_ANGAJAT, NUME, PRENUME, VARSTA, GEN, SALARIU)

VALUES (5, 'Georgescu', 'Mihai', 32, 'M', 3200);

INSERT INTO ANIMAL (ID\_ANIMAL, CATEGORIE, NR\_ANIMALE, NUME\_ANIMAL, ID\_SPECIE)

VALUES (1, 'Mamifer', 10, 'Leu', 1);

INSERT INTO ANIMAL (ID\_ANIMAL, CATEGORIE, NR\_ANIMALE, NUME\_ANIMAL, ID\_SPECIE)

VALUES (2, 'Pasare', 20, 'Papagal', 2);

INSERT INTO ANIMAL (ID\_ANIMAL, CATEGORIE, NR\_ANIMALE, NUME\_ANIMAL, ID\_SPECIE)

VALUES (3, 'Reptila', 5, 'Testoasa', 3);

INSERT INTO ANIMAL (ID\_ANIMAL, CATEGORIE, NR\_ANIMALE, NUME\_ANIMAL, ID\_SPECIE)

VALUES (4, 'Peste', 30, 'Rechin', 4);

INSERT INTO ANIMAL (ID\_ANIMAL, CATEGORIE, NR\_ANIMALE, NUME\_ANIMAL, ID\_SPECIE)

VALUES (5, 'Insecta', 100, 'Albina', 5);

INSERT INTO BILET (ID\_BILET, PRET, CATEGORIE\_BILET)

VALUES (1, 20, 'Adulti');

INSERT INTO BILET (ID\_BILET, PRET, CATEGORIE\_BILET)

VALUES (2, 10, 'Copii');

INSERT INTO BILET (ID\_BILET, PRET, CATEGORIE\_BILET)

VALUES (3, 15, 'Elevi');

INSERT INTO BILET (ID\_BILET, PRET, CATEGORIE\_BILET)

VALUES (4, 25, 'Pensionari');

INSERT INTO BILET (ID\_BILET, PRET, CATEGORIE\_BILET)

VALUES (5, 30, 'Grupuri');

INSERT INTO SPECTACOL (ID\_SPECTACOL, NR\_MAX\_LOCURI, ID\_ORA, ID\_BILET)

VALUES (1, 1000, 1, 1);

INSERT INTO SPECTACOL (ID\_SPECTACOL, NR\_MAX\_LOCURI, ID\_ORA, ID\_BILET)

VALUES (2, 800, 2, 2);

INSERT INTO SPECTACOL (ID\_SPECTACOL, NR\_MAX\_LOCURI, ID\_ORA, ID\_BILET)

VALUES (3, 500, 3, 3);

INSERT INTO SPECTACOL (ID\_SPECTACOL, NR\_MAX\_LOCURI, ID\_ORA, ID\_BILET)

VALUES (4, 1200, 4, 4);

INSERT INTO SPECTACOL (ID\_SPECTACOL, NR\_MAX\_LOCURI, ID\_ORA, ID\_BILET)

VALUES (5, 1500, 5, 5);

INSERT INTO LEGATURA (ID\_ANIMAL, ID\_SPECTACOL)

VALUES (1, 1);

INSERT INTO LEGATURA (ID\_ANIMAL, ID\_SPECTACOL)

VALUES (1, 2);

INSERT INTO LEGATURA (ID\_ANIMAL, ID\_SPECTACOL)

VALUES (2, 3);

INSERT INTO LEGATURA (ID\_ANIMAL, ID\_SPECTACOL)

VALUES (2, 4);

INSERT INTO LEGATURA (ID\_ANIMAL, ID\_SPECTACOL)

VALUES (3, 5);

INSERT INTO LEGATURA (ID\_ANIMAL, ID\_SPECTACOL)

VALUES (3, 1);

INSERT INTO LEGATURA (ID\_ANIMAL, ID\_SPECTACOL)

VALUES (4, 2);

INSERT INTO LEGATURA (ID\_ANIMAL, ID\_SPECTACOL)

VALUES (4, 3);

INSERT INTO LEGATURA (ID\_ANIMAL, ID\_SPECTACOL)

VALUES (5, 4);

INSERT INTO LEGATURA (ID\_ANIMAL, ID\_SPECTACOL)

VALUES (5, 5);

INSERT INTO SPECIE (ID\_SPECIE, RASA, NR\_FAMILII)

VALUES (1, 'Leu African', 3);

INSERT INTO SPECIE (ID\_SPECIE, RASA, NR\_FAMILII)

VALUES (2, 'Ara', 2);

INSERT INTO SPECIE (ID\_SPECIE, RASA, NR\_FAMILII)

VALUES (3, 'Testoasa de uscat', 1);

INSERT INTO SPECIE (ID\_SPECIE, RASA, NR\_FAMILII)

VALUES (4, 'Rechin Alb', 4);

INSERT INTO SPECIE (ID\_SPECIE, RASA, NR\_FAMILII)

VALUES (5, 'Apis mellifera', 10);

INSERT INTO CUSCA (ID\_CUSCA, MATERIAL, DIMENSIUNE, ID\_LOCATIE)

VALUES (1, 'Metal', 3, 1);

INSERT INTO CUSCA (ID\_CUSCA, MATERIAL, DIMENSIUNE, ID\_LOCATIE)

VALUES (2, 'Lemn', 2, 2);

INSERT INTO CUSCA (ID\_CUSCA, MATERIAL, DIMENSIUNE, ID\_LOCATIE)

VALUES (3, 'Sticla', 4, 3);

INSERT INTO CUSCA (ID\_CUSCA, MATERIAL, DIMENSIUNE, ID\_LOCATIE)

VALUES (4, 'Plastic', 5, 4);

INSERT INTO CUSCA (ID\_CUSCA, MATERIAL, DIMENSIUNE, ID\_LOCATIE)

VALUES (5, 'Metal', 3, 5);

INSERT INTO LOCATIE (ID\_LOCATIE, POZITIE, DISTANTA)

VALUES (1, 'Sector A', 100);

INSERT INTO LOCATIE (ID\_LOCATIE, POZITIE, DISTANTA)

VALUES (2, 'Sector B', 200);

INSERT INTO LOCATIE (ID\_LOCATIE, POZITIE, DISTANTA)

VALUES (3, 'Sector C', 150);

INSERT INTO LOCATIE (ID\_LOCATIE, POZITIE, DISTANTA)

VALUES (4, 'Sector D', 300);

INSERT INTO LOCATIE (ID\_LOCATIE, POZITIE, DISTANTA)

VALUES (5, 'Sector E', 250);

INSERT INTO VIZITATOR (ID\_VIZITATOR, CATEGORIE\_VARSTA)

VALUES (1, 'Adult');

INSERT INTO VIZITATOR (ID\_VIZITATOR, CATEGORIE\_VARSTA)

VALUES (2, 'Copil');

INSERT INTO VIZITATOR (ID\_VIZITATOR, CATEGORIE\_VARSTA)

VALUES (3, 'Elev');

INSERT INTO VIZITATOR (ID\_VIZITATOR, CATEGORIE\_VARSTA)

VALUES (4, 'Pensionar');

INSERT INTO VIZITATOR (ID\_VIZITATOR, CATEGORIE\_VARSTA)

VALUES (5, 'Adult');

INSERT INTO PROGRAM (ID\_PROGRAM, ZIUA\_SAPTAMANII, ORA\_DESCHIDERII, ORA\_INCHIDERII)

VALUES (1, 'Luni', 9, 18);

INSERT INTO PROGRAM (ID\_PROGRAM, ZIUA\_SAPTAMANII, ORA\_DESCHIDERII, ORA\_INCHIDERII)

VALUES (2, 'Marti', 10, 17);

INSERT INTO PROGRAM (ID\_PROGRAM, ZIUA\_SAPTAMANII, ORA\_DESCHIDERII, ORA\_INCHIDERII)

VALUES (3, 'Miercuri', 9, 18);

INSERT INTO PROGRAM (ID\_PROGRAM, ZIUA\_SAPTAMANII, ORA\_DESCHIDERII, ORA\_INCHIDERII)

VALUES (4, 'Joi', 10, 17);

INSERT INTO PROGRAM (ID\_PROGRAM, ZIUA\_SAPTAMANII, ORA\_DESCHIDERII, ORA\_INCHIDERII)

VALUES (5, 'Vineri', 9, 18);

INSERT INTO INGRIJIRE (ID\_ANIMAL, ID\_ANGAJAT, ID\_CUSCA)

VALUES (1, 1, 1);

INSERT INTO INGRIJIRE (ID\_ANIMAL, ID\_ANGAJAT, ID\_CUSCA)

VALUES (2, 2, 2);

INSERT INTO INGRIJIRE (ID\_ANIMAL, ID\_ANGAJAT, ID\_CUSCA)

VALUES (3, 3, 3);

INSERT INTO INGRIJIRE (ID\_ANIMAL, ID\_ANGAJAT, ID\_CUSCA)

VALUES (4, 4, 4);

INSERT INTO INGRIJIRE (ID\_ANIMAL, ID\_ANGAJAT, ID\_CUSCA)

VALUES (5, 5, 5);

INSERT INTO INGRIJIRE (ID\_ANIMAL, ID\_ANGAJAT, ID\_CUSCA)

VALUES (2, 1, 1);

INSERT INTO INGRIJIRE (ID\_ANIMAL, ID\_ANGAJAT, ID\_CUSCA)

VALUES (3, 2, 2);

INSERT INTO INGRIJIRE (ID\_ANIMAL, ID\_ANGAJAT, ID\_CUSCA)

VALUES (4, 3, 3);

INSERT INTO INGRIJIRE (ID\_ANIMAL, ID\_ANGAJAT, ID\_CUSCA)

VALUES (5, 4, 4);

INSERT INTO INGRIJIRE (ID\_ANIMAL, ID\_ANGAJAT, ID\_CUSCA)

VALUES (1, 5, 5);

INSERT INTO ORA (ID\_ORA, ORA)

VALUES (1, 9);

INSERT INTO ORA (ID\_ORA, ORA)

VALUES (2, 10);

INSERT INTO ORA (ID\_ORA, ORA)

VALUES (3, 11);

INSERT INTO ORA (ID\_ORA, ORA)

VALUES (4, 12);

INSERT INTO ORA (ID\_ORA, ORA)

VALUES (5, 13);

SELECT \*

FROM GRADINA\_ZOO;

SELECT \*

FROM ANGAJAT;

SELECT \*

FROM ANIMAL;

SELECT \*

FROM BILET;

SELECT \*

FROM SPECTACOL;

SELECT \*

FROM LEGATURA;

SELECT \*

FROM SPECIE;

SELECT \*

FROM CUSCA;

SELECT \*

FROM LOCATIE;

SELECT \*

FROM VIZITATOR;

SELECT \*

FROM PROGRAM;

SELECT \*

FROM INGRIJIRE;

SELECT \*

FROM ORA;

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

• subcereri sincronizate în care intervin cel puțin 3 tabele

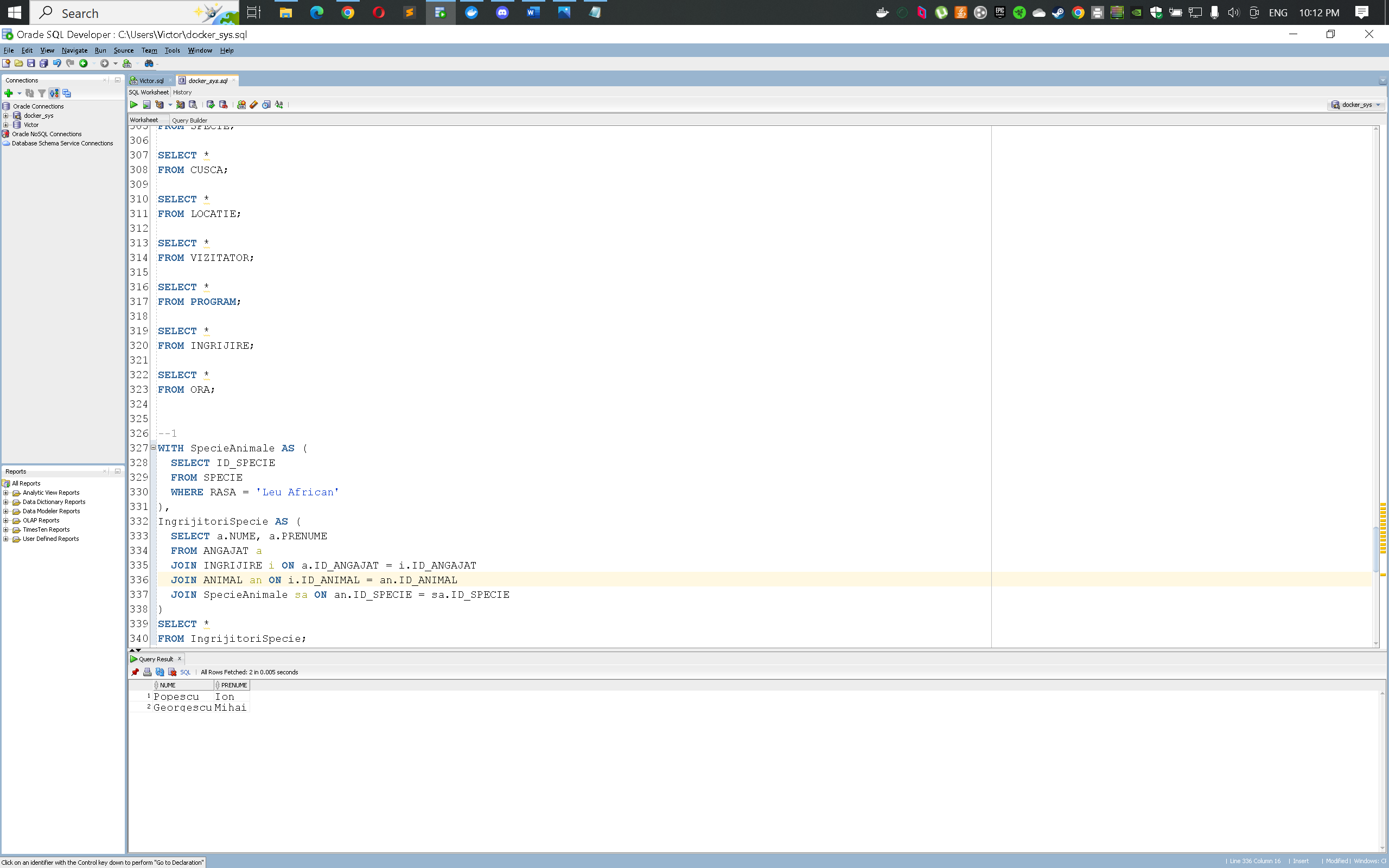
• subcereri nesincronizate în clauza FROM

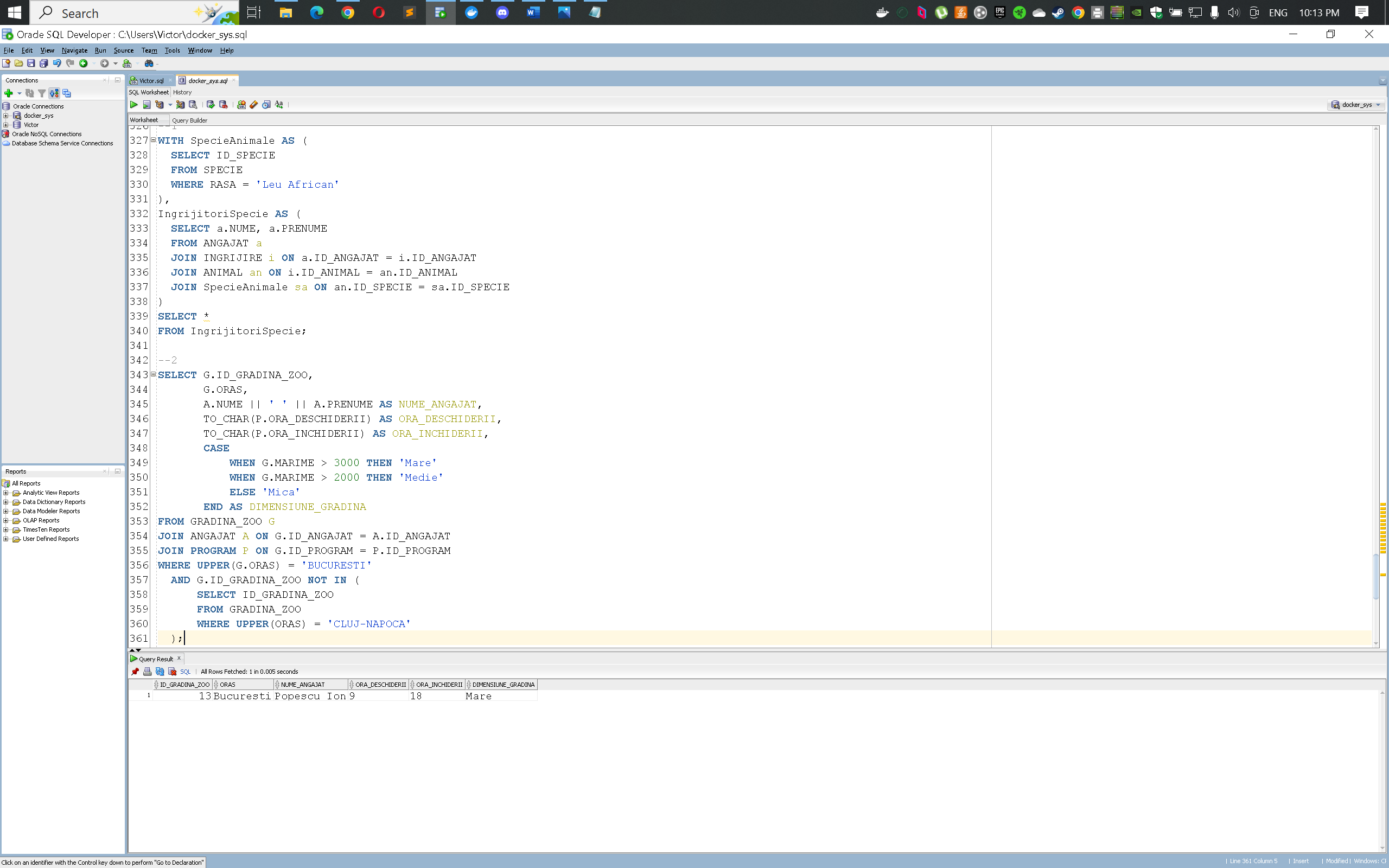
• grupări de date cu subcereri nesincronizate in care intervin cel putin 3 tabele, funcții grup, filtrare la nivel de grupuri (in cadrul aceleiasi cereri)

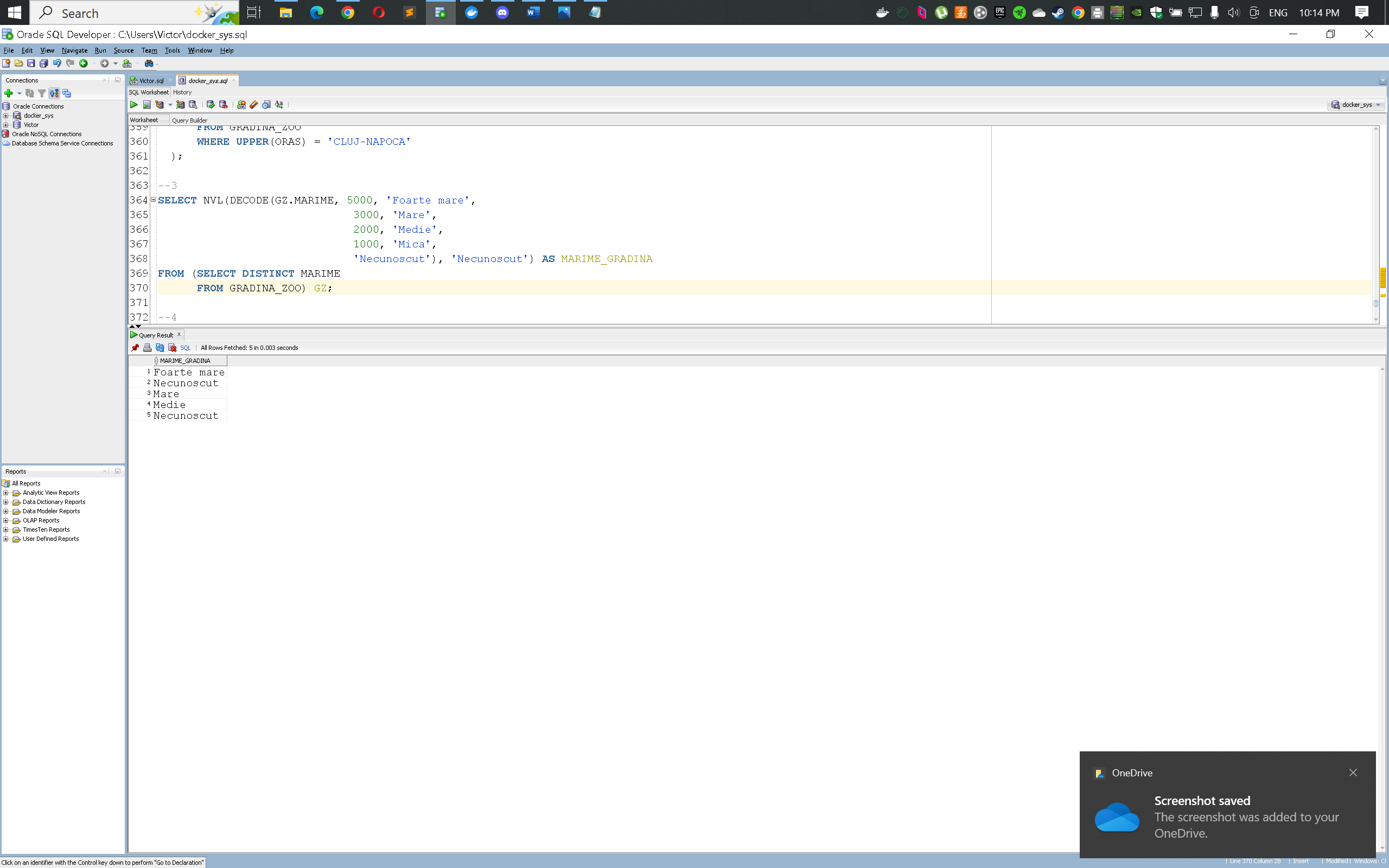
• ordonări si utilizarea funcțiilor NVL și DECODE (in cadrul aceleiasi cereri)

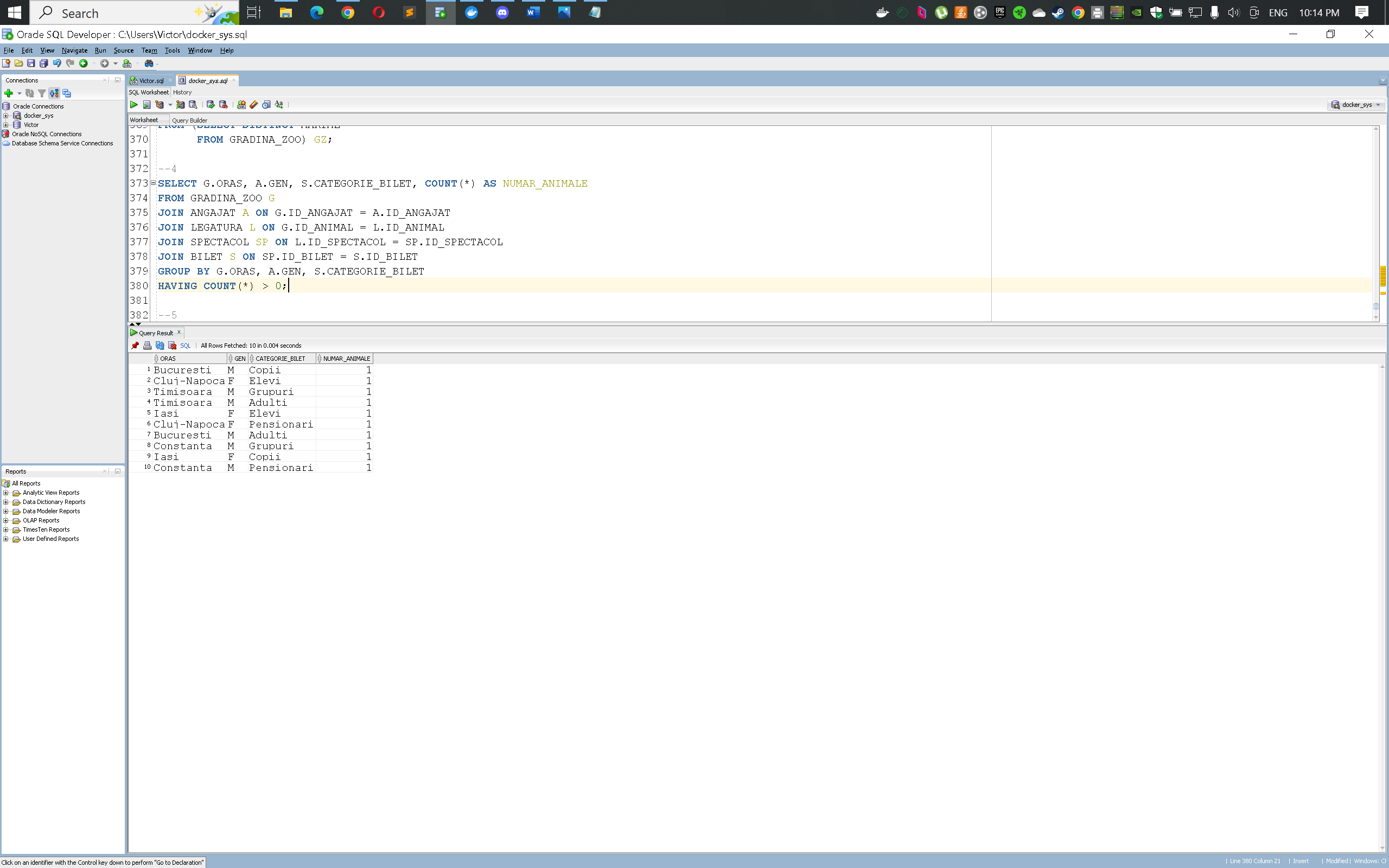
• utilizarea a cel puțin 2 funcții pe șiruri de caractere, 2 funcții pe date calendaristice, a cel puțin unei expresii CASE

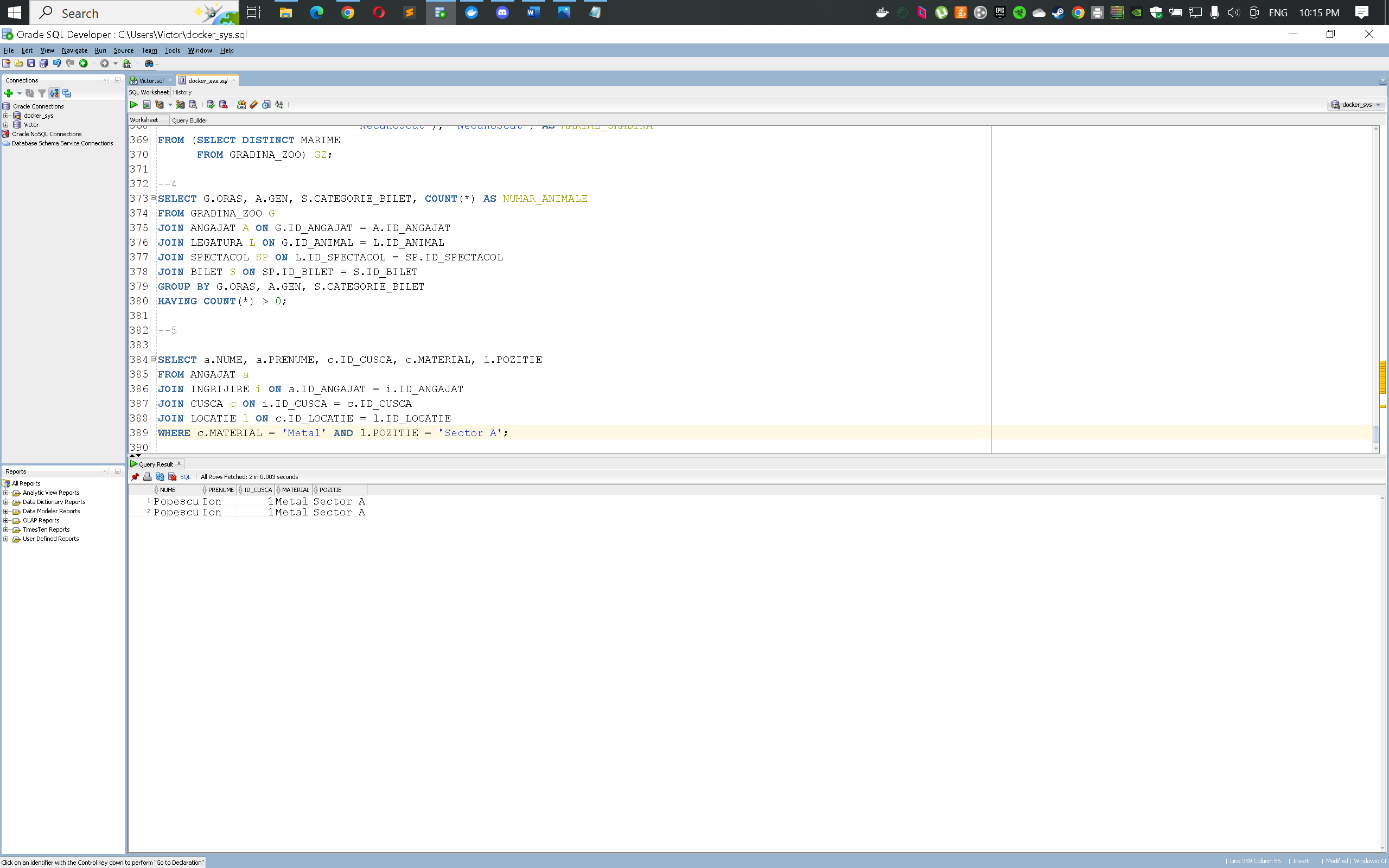
• utilizarea a cel puțin 1 bloc de cerere (clauza WITH)











--1

WITH SpecieAnimale AS (

SELECT ID\_SPECIE

FROM SPECIE

WHERE RASA = 'Leu African'

),

IngrijitoriSpecie AS (

SELECT a.NUME, a.PRENUME

FROM ANGAJAT a

JOIN INGRIJIRE i ON a.ID\_ANGAJAT = i.ID\_ANGAJAT

JOIN ANIMAL an ON i.ID\_ANIMAL = an.ID\_ANIMAL

JOIN SpecieAnimale sa ON an.ID\_SPECIE = sa.ID\_SPECIE

)

SELECT \*

FROM IngrijitoriSpecie;

--2

SELECT G.ID\_GRADINA\_ZOO,

G.ORAS,

A.NUME || ' ' || A.PRENUME AS NUME\_ANGAJAT,

TO\_CHAR(P.ORA\_DESCHIDERII) AS ORA\_DESCHIDERII,

TO\_CHAR(P.ORA\_INCHIDERII) AS ORA\_INCHIDERII,

CASE

WHEN G.MARIME > 3000 THEN 'Mare'

WHEN G.MARIME > 2000 THEN 'Medie'

ELSE 'Mica'

END AS DIMENSIUNE\_GRADINA

FROM GRADINA\_ZOO G

JOIN ANGAJAT A ON G.ID\_ANGAJAT = A.ID\_ANGAJAT

JOIN PROGRAM P ON G.ID\_PROGRAM = P.ID\_PROGRAM

WHERE UPPER(G.ORAS) = 'BUCURESTI'

AND G.ID\_GRADINA\_ZOO NOT IN (

SELECT ID\_GRADINA\_ZOO

FROM GRADINA\_ZOO

WHERE UPPER(ORAS) = 'CLUJ-NAPOCA'

);

--3

SELECT NVL(DECODE(GZ.MARIME, 5000, 'Foarte mare',

3000, 'Mare',

2000, 'Medie',

1000, 'Mica',

'Necunoscut'), 'Necunoscut') AS MARIME\_GRADINA

FROM (SELECT DISTINCT MARIME

FROM GRADINA\_ZOO) GZ;

--4

SELECT G.ORAS, A.GEN, S.CATEGORIE\_BILET, COUNT(\*) AS NUMAR\_ANIMALE

FROM GRADINA\_ZOO G

JOIN ANGAJAT A ON G.ID\_ANGAJAT = A.ID\_ANGAJAT

JOIN LEGATURA L ON G.ID\_ANIMAL = L.ID\_ANIMAL

JOIN SPECTACOL SP ON L.ID\_SPECTACOL = SP.ID\_SPECTACOL

JOIN BILET S ON SP.ID\_BILET = S.ID\_BILET

GROUP BY G.ORAS, A.GEN, S.CATEGORIE\_BILET

HAVING COUNT(\*) > 0;

--5

SELECT a.NUME, a.PRENUME, c.ID\_CUSCA, c.MATERIAL, l.POZITIE

FROM ANGAJAT a

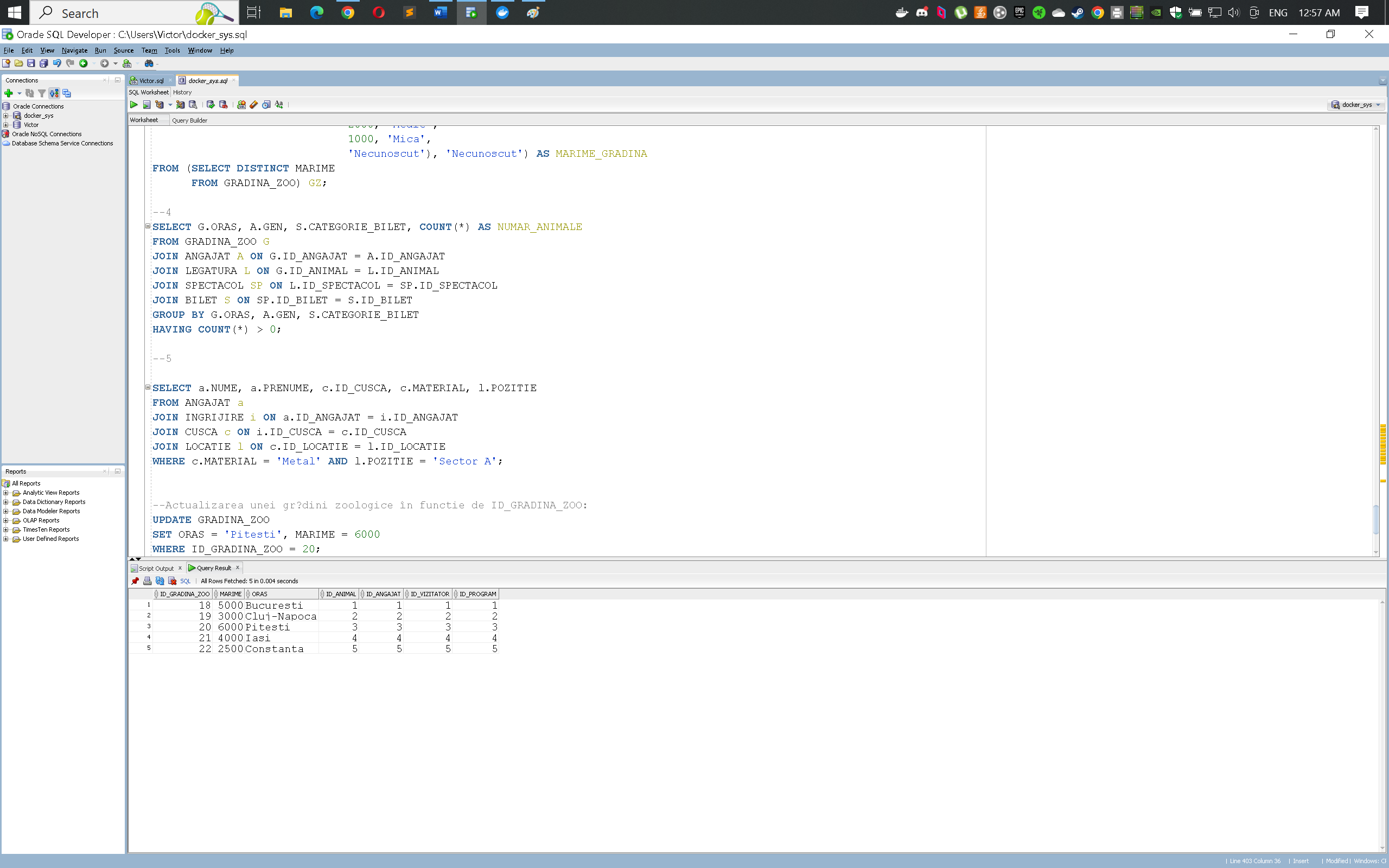
JOIN INGRIJIRE i ON a.ID\_ANGAJAT = i.ID\_ANGAJAT

JOIN CUSCA c ON i.ID\_CUSCA = c.ID\_CUSCA

JOIN LOCATIE l ON c.ID\_LOCATIE = l.ID\_LOCATIE

WHERE c.MATERIAL = 'Metal' AND l.POZITIE = 'Sector A';

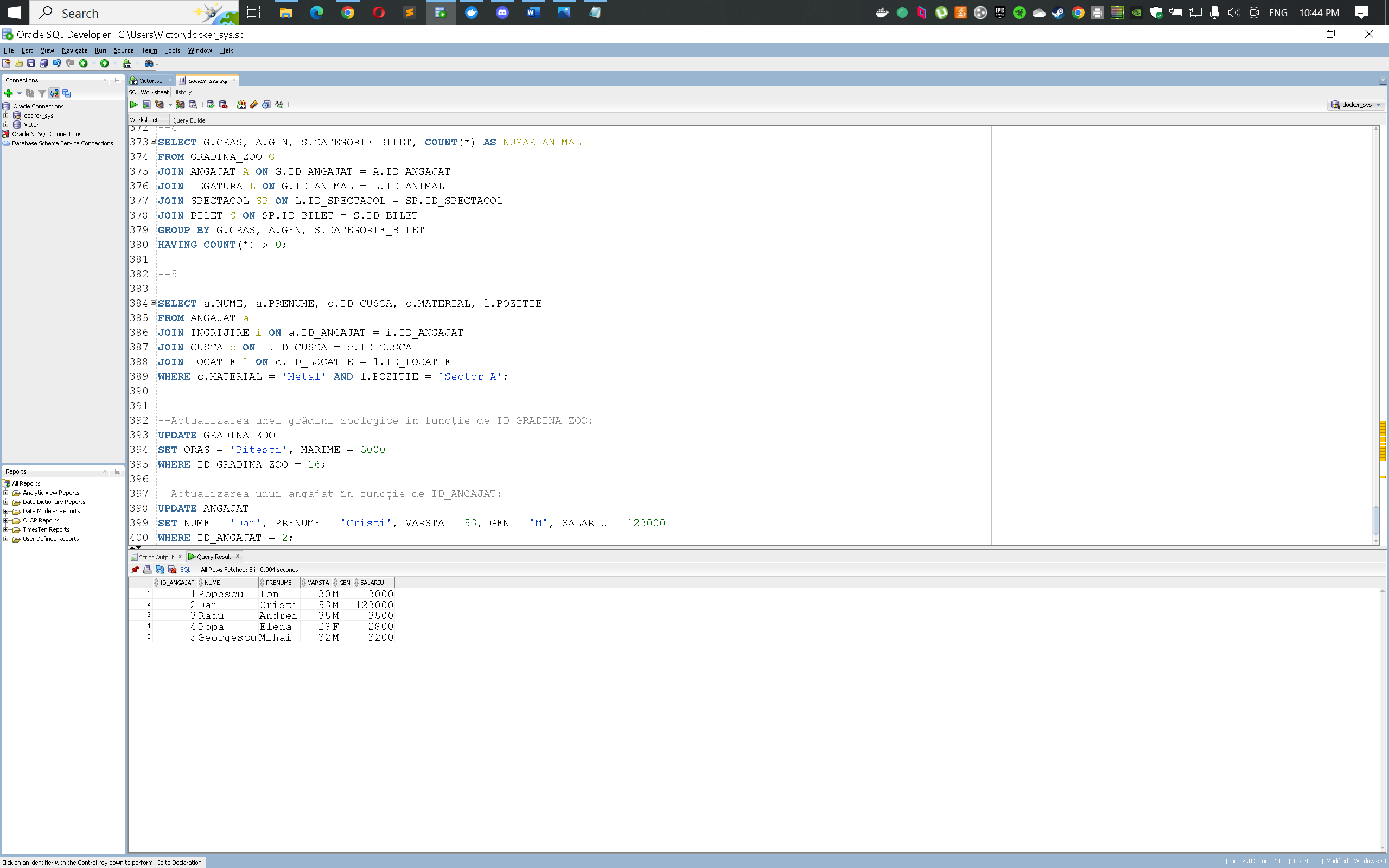
13. Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri.



UPDATE GRADINA\_ZOO

SET ORAS = 'Pitesti', MARIME = 6000

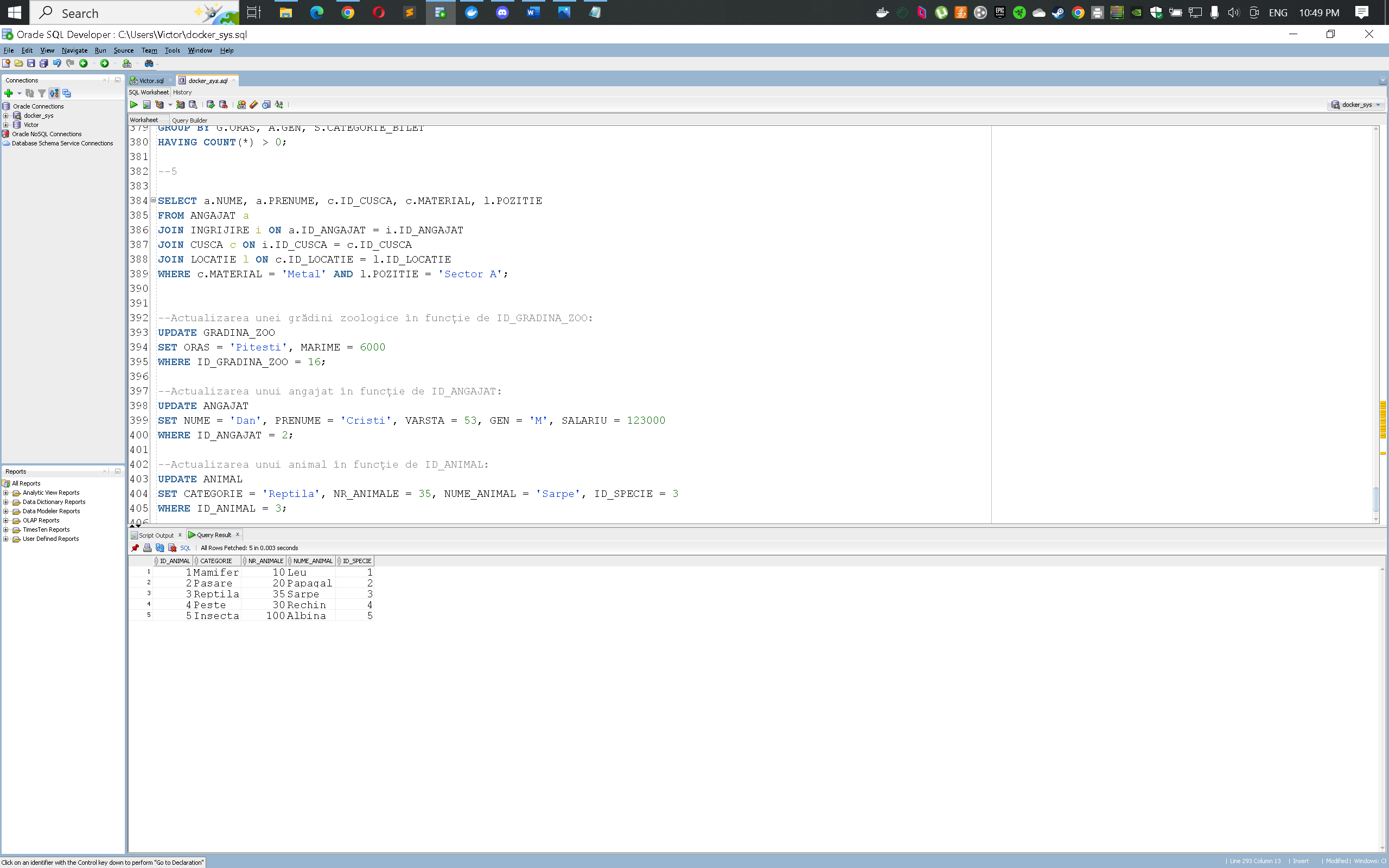
WHERE ID\_GRADINA\_ZOO = 20;



UPDATE ANGAJAT

SET NUME = 'Dan', PRENUME = 'Cristi', VARSTA = 53, GEN = 'M', SALARIU = 123000

WHERE ID\_ANGAJAT = 2;



UPDATE ANIMAL

SET CATEGORIE = 'Reptila', NR\_ANIMALE = 35, NUME\_ANIMAL = 'Sarpe', ID\_SPECIE = 3

WHERE ID\_ANIMAL = 3;