**PROIECT BAZE DE DATE**

**Adapost de Animale**

Prof. Laborator: Natalia Gabriela Ozunu

Prof. Curs: Letiția Ana Marin

**MOCANU ALEXANDRA**

**Grupa 232**

Contents

[1. Descrierea modelului real, utilitatii acestuia si a regulilor de functionare. 3](#_Toc167728903)

[2. Prezentarea constragerilor (restrictii, reguli) impuse asupra modelului. 3](#_Toc167728904)

[3. Descrierea entităților, incluzând precizarea cheii primare. 4](#_Toc167728905)

[4. Descrierea relațiilor, incluzând precizarea cardinalității acestora. 4](#_Toc167728906)

[5. Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicite, valori posibile ale atributelor. 4](#_Toc167728907)

[6. Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5 4](#_Toc167728908)

[7. Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6. 4](#_Toc167728909)

[8. Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale proiectate la punctul 7. 4](#_Toc167728910)

[9. Realizarea normalizării până la forma normală 3(FN1-FN3). 4](#_Toc167728911)

[10. Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 11). 4](#_Toc167728912)

[11. Crearea tabelelor in SQL si inserarea de date coerente in fiecare dintre acestea. 4](#_Toc167728913)

[12. Formulați în limbaj natural și implementați 5 cereri SQL complexe ce vor utiliza, în ansamblul lor, următoarele elemente: 4](#_Toc167728914)

[13. Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri. 4](#_Toc167728915)

[14. Formulați în limbaj natural și implementați în SQL. 4](#_Toc167728916)

[15. Optimizarea unei cereri, aplicând regulile de optimizare ce derivă din proprietățile operatorilor algebrei relaționale. Cererea va fi exprimatăprin expresie algebrică, arbore algebric și limbaj (SQL), atât anterior cât și ulterior optimizării. 5](#_Toc167728917)

[16. 5](#_Toc167728918)

[17. Codul intreg: 6](#_Toc167728919)

**PROIECT BAZE DE DATE - ADAPOST DE ANIMALE**

# Descrierea modelului real, utilitatii acestuia si a regulilor de functionare.

Acest proiect urmareste realizarea design-ului unei baze de date a unui adapost de animale. Utilitatea acestui design este aceea ca se poate tine cu usurinta evidenta angajatilor (veterinari, directori) din adapost, dar si a unor consumabile (medicamente, hrana), sau documente (cereri de adoptie).

In ceea ce priveste organizarea, adapostul este impartit in sectii, iar fiecare sectie exista sectoare in care se afla custile animalelor adapostite. Veterinarii pot face consultatii si noteaza in carnetul veterinary al fiecarui animal toate datele necesare.

# Prezentarea constragerilor (restrictii, reguli) impuse asupra modelului.

Constrângerile din codul furnizat includ: chei primare, chei externe, restricții de referință, chei primare compuse, restricții de unicitate și restricții de integritate referențială. Acestea asigură unicitatea valorilor, relațiile între tabele, respectarea valorilor cheilor primare și integritatea datelor.

Cheie primară: O constrângere de cheie primară asigură că o coloană sau un set de coloane dintr-o tabelă au valori unice și identifică în mod unic fiecare înregistrare. De exemplu, o constrângere de cheie primară ar putea fi aplicată coloanei "id" dintr-o tabelă "Angajat", astfel încât fiecare angajatsă aibă un ID unic.

Cheie externă: O constrângere de cheie externă definește relații între tabele, asigurând integritatea referențială. De exemplu, o constrângere de cheie externă ar putea fi aplicată coloanei "id\_angajat" dintr-o tabelă "Contract", referindu-se la coloana "id" din tabela "Angajat", pentru a asigura că fiecare comandă este asociată cu un utilizator valid.

Restricție de unicitate: O restricție de unicitate asigură că o coloană sau un set de coloane dintr-o tabelă nu conține valori duplicate. De exemplu, o restricție de unicitate ar putea fi aplicată coloanei "CNP" dintr-o tabelă "Vizitator", astfel încât fiecare vizitator să aibe un CNP unic.

# Descrierea entităților, incluzând precizarea cheii primare.

Tabela CONTRACT:

Cheia primară: NUMAR (de tip INT)

Tabela ANGAJAT:

Cheia primară: ID\_ANGAJAT (de tip INT)

Tabela DIRECTOR:

Cheia primară: ID\_DIRECTOR (de tip INT)

Tabela SECTIE:

Cheia primară: ID\_SECTIE (de tip INT)

Tabela SECTOR:

Cheia primară: ID\_SECTOR (de tip INT)

Tabela SECTIE\_SECTOR:

Cheile primare: ID\_SECTIE și ID\_SECTOR (ambele de tip INT)

Tabela FARMACIE:

Cheia primară: ID\_FARMACIE (de tip INT)

Tabela MEDICAMENT:

Cheia primară: NUMAR (de tip INT)

Tabela VIZITATOR:

Cheia primară: CNP (de tip VARCHAR(13))

Tabela CERERE\_ADOPTIE:

Cheia primară: SERIE, ID\_ANIMAL, ID\_VIZITATOR (toate de tip VARCHAR(255), INT, VARCHAR(13) respectiv)

Tabela ANIMAL:

Cheia primară: ID\_ANIMAL (de tip INT)

Tabela CUSCA:

Cheia primară: ID\_CUSCA (de tip INT)

Tabela VETERINAR:

Cheia primară: CNP (de tip VARCHAR(13))

Tabela MAGAZIN:

Cheia primară: ID\_MAGAZIN (de tip INT)

Tabela HRANA:

Cheia primară: NUMAR (de tip INT)

Tabela CARTE\_DE\_SANATATE:

Cheia primară: ID\_ANIMAL (de tip INT)

Tabela CONSULTATIE:

Cheia primară: ID\_CONSULTATIE (de tip INT)

Tabela CLINICA\_VETERINARA:

Cheia primară: ID\_CLINICA (de tip INT)

Vizualizarea v\_angajati\_sectii:

Această vizualizare nu are o cheie primară, deoarece este o vizualizare rezultată din interogarea datelor din mai multe tabele.

# Descrierea relațiilor, incluzând precizarea cardinalității acestora.

Tabelul CONTRACT are o relație de tip 1 la N cu tabelul ANGAJAT. Un contract poate fi asociat unui singur angajat, iar un angajat poate avea asociat unul sau mai multe contracte.

Tabelul ANGAJAT are o relație de tip 1 la 1 cu tabelul CONTRACT. Fiecare angajat este asociat cu un singur contract.

Tabelul DIRECTOR are o relație de tip 1 la 1 cu tabelul CONTRACT. Fiecare director este asociat cu un singur contract.

Tabelul SECTIE are o relatie cu tabela Angajat si una cu tabela SECTIE\_SECTOR.

Tabelul SECTOR are relatie atat cu tablea SECTIE\_SECTOR cat si cu tabela CUSCA.

Tabelul SECTIE\_SECTOR are o relație de tip N la N cu tabelele SECTIE și SECTOR. O sectie poate fi asociata cu mai multe sectoare, iar un sector poate fi asociat cu mai multe sectii.

Tabelul FARMACIE are relațiE cu tabelele CONTRACT si FARMACIE.

Tabelul MEDICAMENT are o relație de tip 1 la N cu tabelul FARMACIE. Un medicament poate fi asociat unei singure farmacii, iar o farmacie poate avea asociate unul sau mai multe medicamente.

Tabelul VIZITATOR nu are relații cu alte tabele.

Tabelul CERERE\_ADOPTIE are o relație de tip N la N cu tabelele ANIMAL și VIZITATOR. O cerere de adoptie poate fi asociata cu mai multe animale si vizitatori, iar un animal si un vizitator pot fi asociati cu mai multe cereri de adoptie.

Tabelul ANIMAL are o relație de tip 1 la 1 cu tabelul CUSCA. Fiecare animal este asociat cu o singură cusca.

Tabelul CUSCA are relații cu tabelele SECTOR si ANIMAL.

Tabelul VETERINAR are o relație de tip 1 la N cu tabelul ANIMAL. Un veterinar poate fi asociat cu unul sau mai multe animale, iar un animal poate fi asociat cu un singur veterinar.

Tabelul MAGAZIN are relații cu tabelea HRANA si tabela CONTRACT.

Tabelul HRANA are o relație de tip 1 la N cu tabelul MAGAZIN. Un produs de hrana poate fi asociat unui singur magazin, iar un magazin poate avea asociate unul sau mai multe produse de hrana.

Tabelul CARTE\_DE\_SANATATE are o relație de tip 1 la 1 cu tabelul ANIMAL. Fiecare animal are asociată o singură carte de sănătate.

Tabelul CONSULTATIE are relatie cu tabela ANIMALl si cu tabela VETERINAR.

Tabelul CLINICA\_VETERINARA are relatie cu tabelele VETERINAR si tabela CONTRACT.

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

Tabelul CONTRACT:

Atribute: NUMAR (cheie primară), DATA\_EMITERE, DURATA, TIP, id\_angajat (cheie externă referențiază ANGAJAT.ID\_ANGAJAT).

Tabelul ANGAJAT:

Atribute: ID\_ANGAJAT (cheie primară), NUME, PRENUME, CNP, ADRESA, TELEFON, EMAIL, ID\_CONTRACT (cheie externă referențiază CONTRACT.NUMAR).

Tabelul DIRECTOR:

Atribute: ID\_DIRECTOR (cheie primară), ID\_CONTRACT (cheie externă referențiază CONTRACT.NUMAR).

Tabelul SECTIE:

Atribute: ID\_SECTIE (cheie primară), NUME.

Tabelul SECTOR:

Atribute: ID\_SECTOR (cheie primară), TIP\_NEVOIE, NUMAR\_LOCURI.

Tabelul SECTIE\_SECTOR:

Atribute: ID\_SECTIE, ID\_SECTOR (cheie primară compusă), ID\_SECTIE (cheie externă referențiază SECTIE.ID\_SECTIE), ID\_SECTOR (cheie externă referențiază SECTOR.ID\_SECTOR).

Tabelul FARMACIE:

Atribute: ID\_FARMACIE (cheie primară), NUME, TELEFON, PROGRAM.

Tabelul MEDICAMENT:

Atribute: NUMAR (cheie primară), DENUMIRE\_PRODUS, CANTITATE, ID\_FARMACIE (cheie externă referențiază FARMACIE.ID\_FARMACIE).

Tabelul VIZITATOR:

Atribute: CNP (cheie primară), NUME, PRENUME, TELEFON, EMAIL.

Tabelul CERERE\_ADOPTIE:

Atribute: SERIE, ID\_ANIMAL, ID\_VIZITATOR, DATA, APROBATA\_DE\_DIRECTOR, APROBATA\_DE\_ASOCIATIE (cheie primară compusă), ID\_ANIMAL (cheie externă referențiază ANIMAL.ID\_ANIMAL), ID\_VIZITATOR (cheie externă referențiază VIZITATOR.CNP).

Tabelul ANIMAL:

Atribute: ID\_ANIMAL (cheie primară), NUME, RASA, VARSTA, SEX, DATA\_SOSIRE, ID\_CUSCA (cheie externă referențiază CUSCA.ID\_CUSCA), ID\_VETERINAR (cheie externă referențiază VETERINAR.CNP).

Tabelul CUSCA:

Atribute: ID\_CUSCA (cheie primară), NUMAR\_LOCURI.

Tabelul VETERINAR:

Atribute: CNP (cheie primară), NUME, PRENUME, ID\_ANIMAL (cheie externă referențiază ANIMAL.ID\_ANIMAL).

Tabelul MAGAZIN:

Atribute: ID\_MAGAZIN (cheie primară), NUME, TELEFON, PROGRAM.

Tabelul HRANA:

Atribute: ID\_HRANA (cheie primară), DENUMIRE, CANTITATE, ID\_MAGAZIN (cheie externă referențiază MAGAZIN.ID\_MAGAZIN).

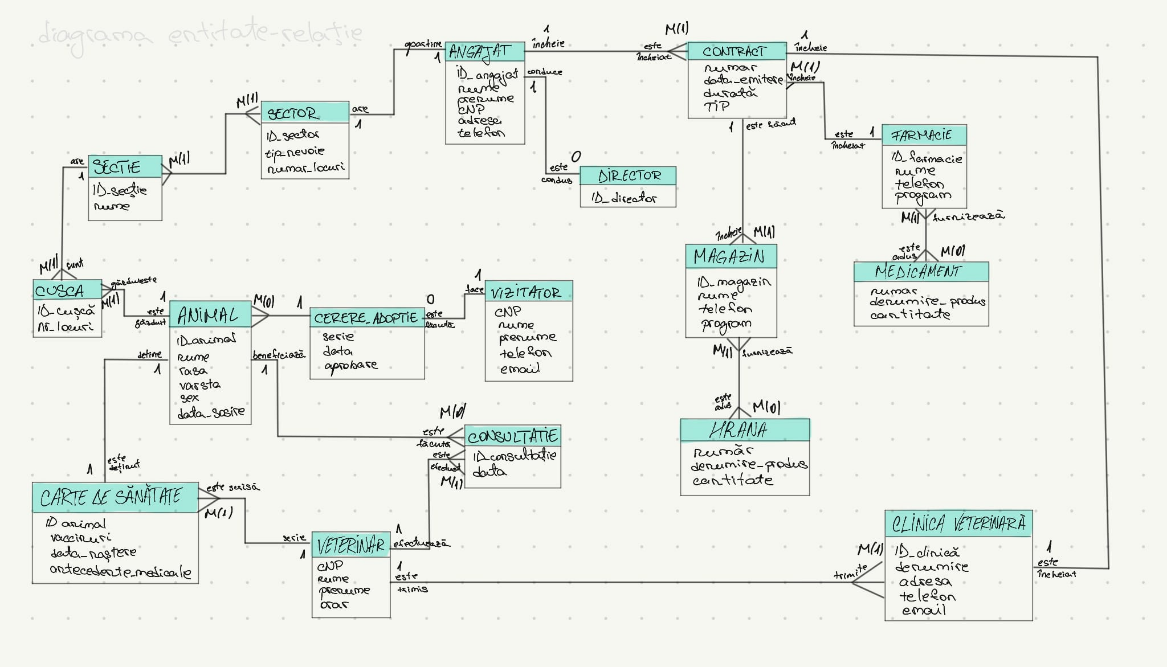
Tabelul COMANDA:

Atribute: ID\_COMANDA (cheie primară), DATA, SUMA\_TOTALA, ID\_MAGAZIN (cheie externă referențiază MAGAZIN.ID\_MAGAZIN).

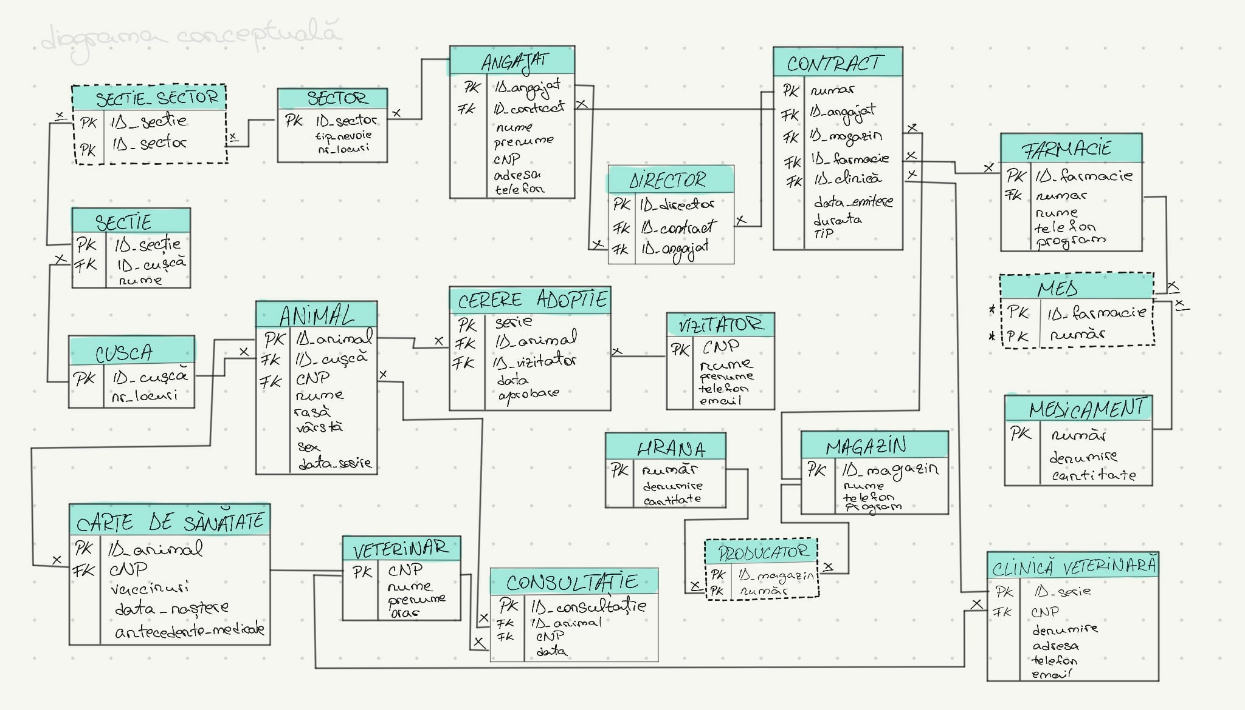
Tabelul DETALII\_COMANDA:

Atribute: ID\_COMANDA, ID\_HRANA (cheie primară compusă), ID\_COMANDA (cheie externă referențiază COMANDA.ID\_COMANDA), ID\_HRANA (cheie externă referențiază HRANA.ID\_HRANA).

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



# Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6.



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

* Contract (numar, data\_emitere, durata, id\_angajat)
* Angajat (id\_angajat, nume, prenume, cnp, adresa, telefon, email, id\_contract)
* Director (id\_director, id\_contract)
* Sectie (id\_sectie, nume)
* Sector (id\_sector, tip\_nevoie, numar\_locuri)
* Sectie\_sector (id\_sectie, id\_sector)
* Farmacie (id\_farmacie, nume, telefon, program)
* Medicament (numar, denumire\_produs, cantitate, id\_farmacie)
* Vizitator (cnp, nume, prenume, telefon, email)
* Cerere\_adoptie (serie, id\_animal, id\_vizitator, data, aprobata\_de\_director, aprobata\_de\_asociatie)
* Animal (id\_animal, nume, rasa, varsta, sex, data\_sosire, id\_cusca, id\_veterinar)
* Cusca (id\_cusca, numar\_locuri)
* Veterinar (cnp, nume, prenume, id\_animal, id\_veterinar)
* Magazin (id\_magazin, nume, telefon, program)
* Hrana (numar, denumire\_produs, cantitate, id\_magazin)

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

Forma normală de la 1 (FN1) implică eliminarea valorilor repetate și atomizarea atributelor într-o singură valoare. În codul tău, FN1 este respectată prin eliminarea duplicatelor și atomizarea datelor în tabelele tale. De exemplu, în tabela ANGAJAT, fiecare angajat are o valoare unică pentru ID\_ANGAJAT și nu există duplicări ale acestui atribut.

Forma normală de la 2 (FN2) implică eliminarea dependențelor funcționale parțiale și asigurarea dependențelor funcționale complete prin adăugarea cheilor primare în tabelele de referință. În codul tău, FN2 este respectată prin utilizarea cheilor primare și străine pentru a stabili relații între tabele. De exemplu, tabela ANGAJAT are o cheie străină ID\_CONTRACT care face referire la tabela CONTRACT, asigurând astfel dependența funcțională completă între aceste tabele.

Forma normală de la 3 (FN3) implică eliminarea dependențelor tranzitive și asigurarea dependențelor funcționale directe între toate atributele. În codul tău, FN3 pare să fie respectată, deoarece nu există dependențe tranzitive evidente și relațiile între tabelele tale par să fie bine definite.

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



A picture containing text, software, computer icon, web page

Description automatically generatedA screenshot of a computer

Description automatically generated

# Crearea tabelelor in SQL si inserarea de date coerente in fiecare dintre acestea.



A screenshot of a computer

Description automatically generated

# 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 dat ecu subcereri nesincronizate in care intervin cel putin 3 tabele, funcții grup, filtrare la nivel degrupuri(in cadrul aceleiasi cereri)
* ordonărisi 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)Observație:Într-o cerere se vor regăsi mai multe elemente dintre cele enumerate mai sus, astfel încât cele 5 cereri să le cuprindă pe toate.

a) Subcereri sincronizate în care intervin cel puțin 3 tabele

Această interogare selectează informații despre animale, veterinari și vaccinuri pentru animalele care au primit vaccinuri împotriva rabiei. Folosește tabelele ANIMAL, VETERINAR și CARTE\_DE\_SANATATE.

c) Grupări de date, funcții grup, filtrare la nivel de grupuri cu subcereri

Interogarea calculează numărul de animale pentru fiecare tip de nevoie din sectoare și filtrează grupurile care au un număr de animale peste media numărului de animale din alte sectoare. Utilizează tabelele SECTOR, SECTIE\_SECTOR, SECTIE și ANIMAL.

d) Ordonări și utilizarea funcțiilor NVL și DECODE

Interogarea extrage informații despre vizitatori și numărul de cereri de adopție, aplicând ajustări pe baza tipului de cerere (urgent sau nu). Utilizează tabelele VIZITATOR și CERERE\_ADOPTIE.

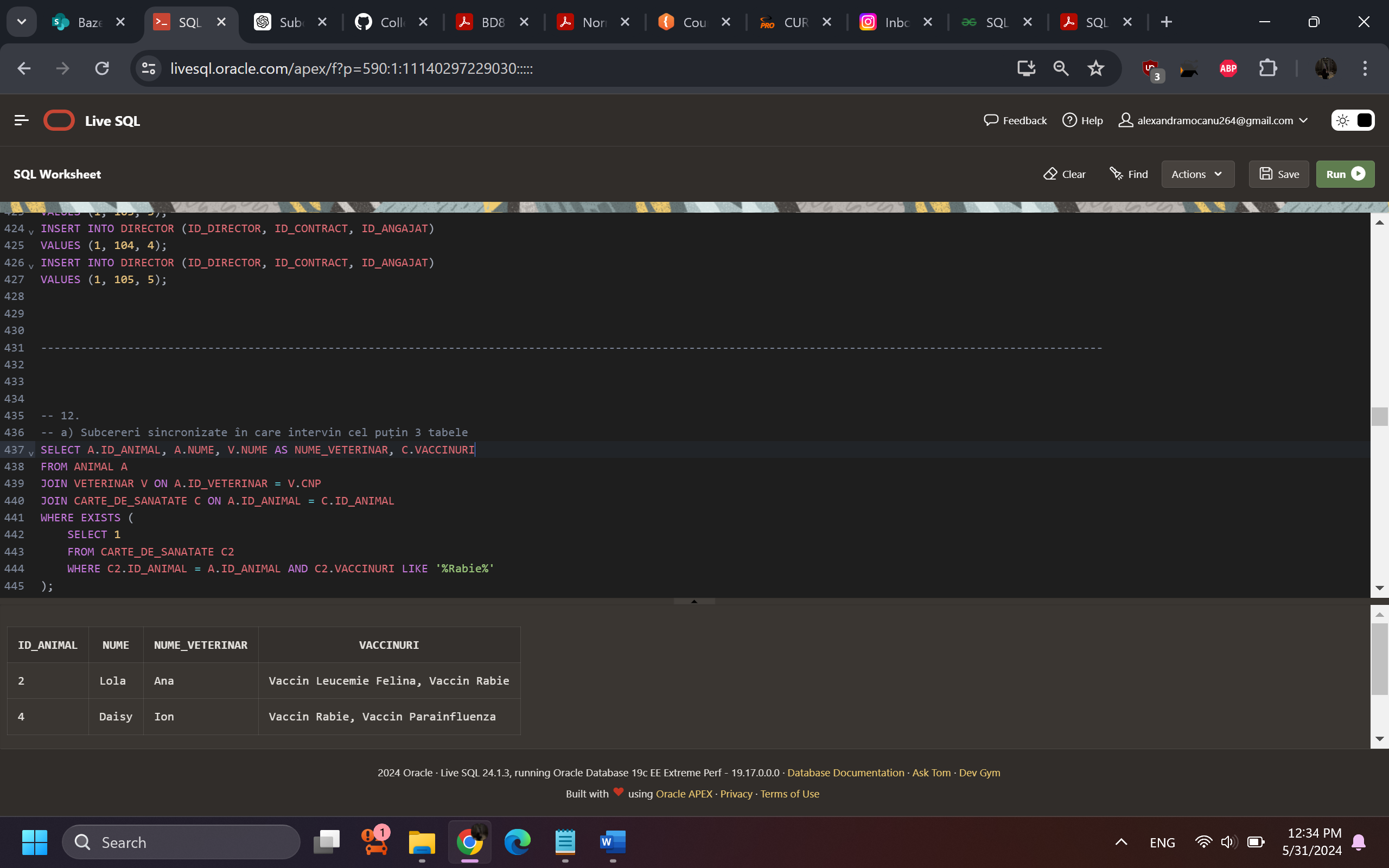
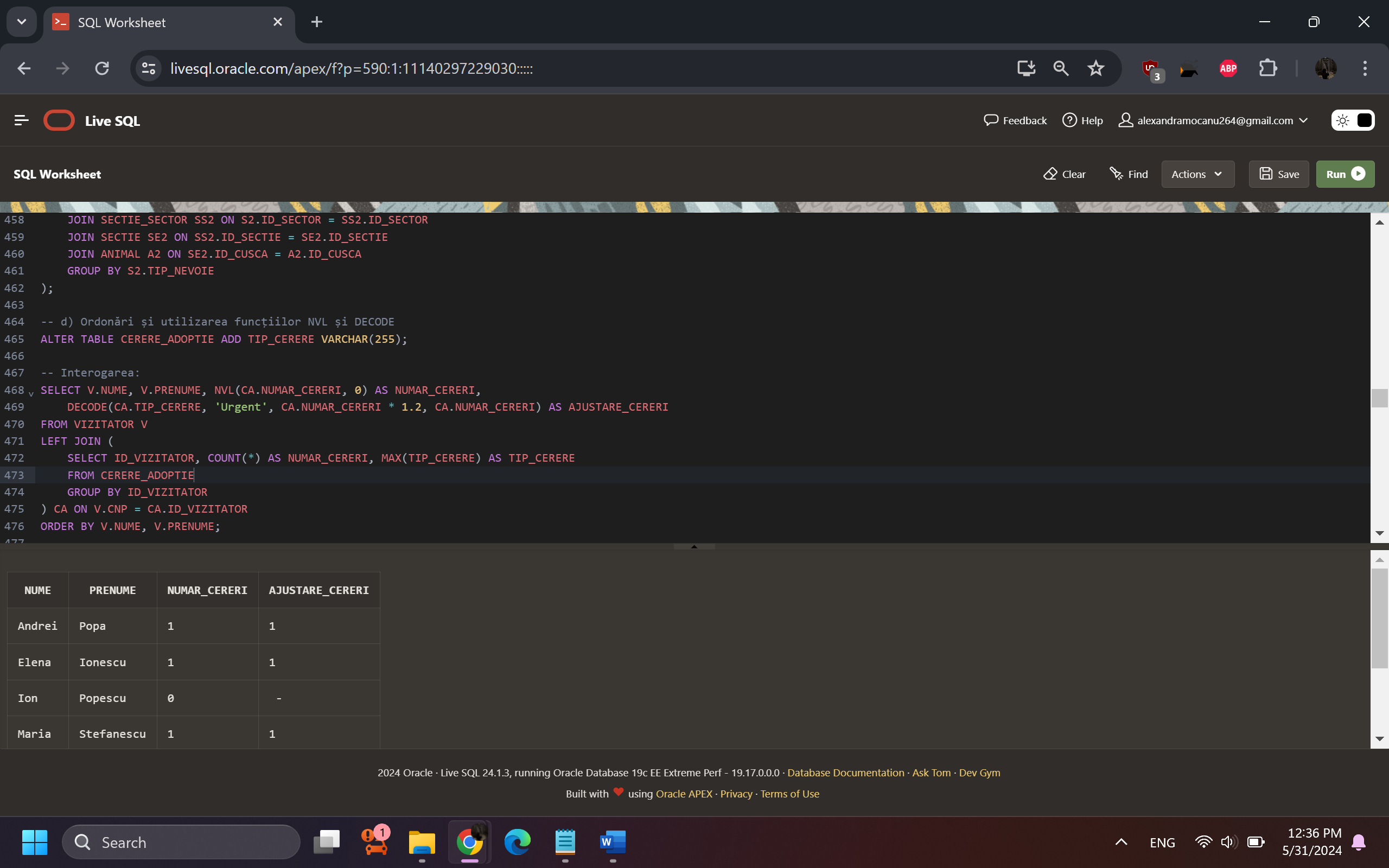
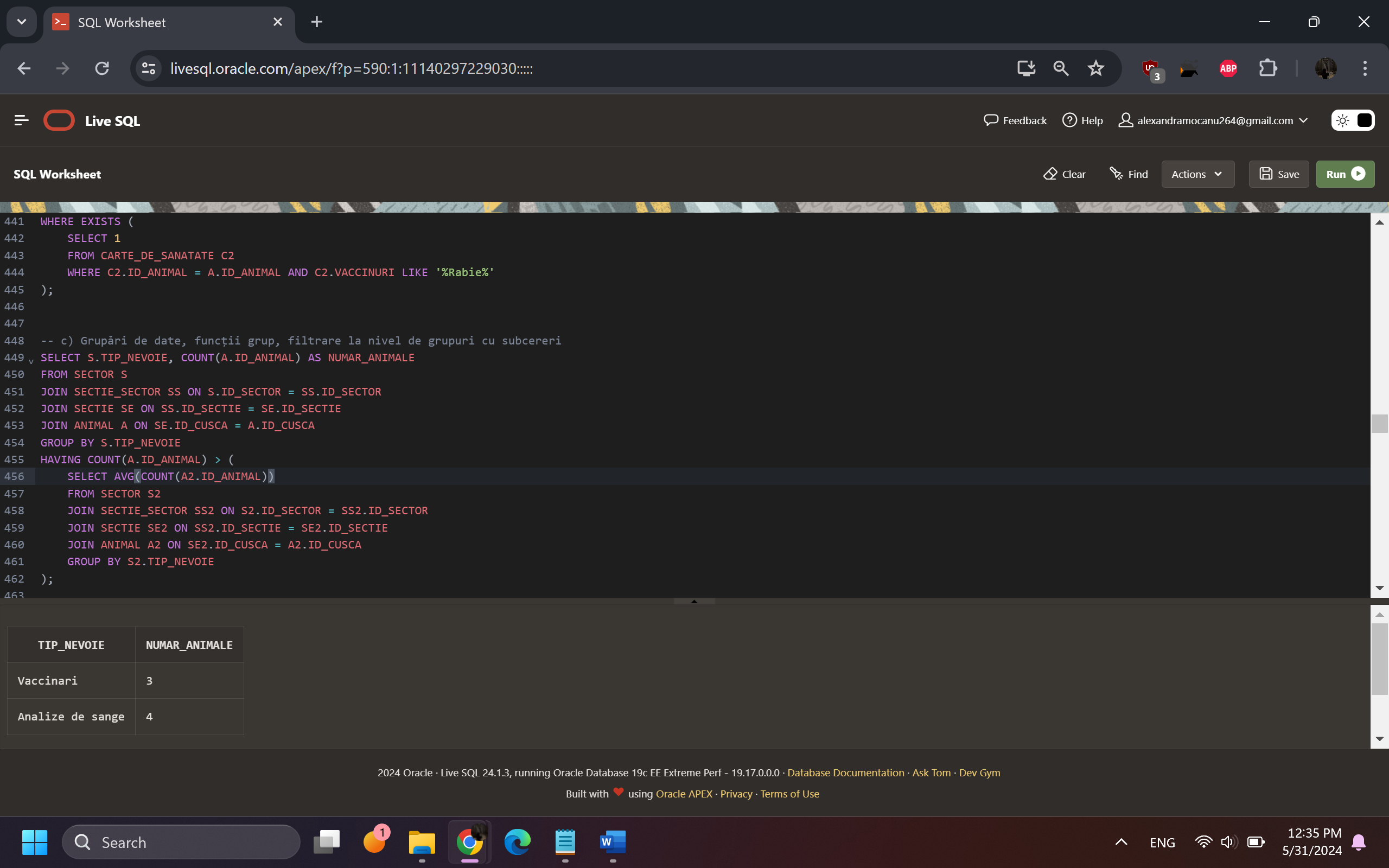
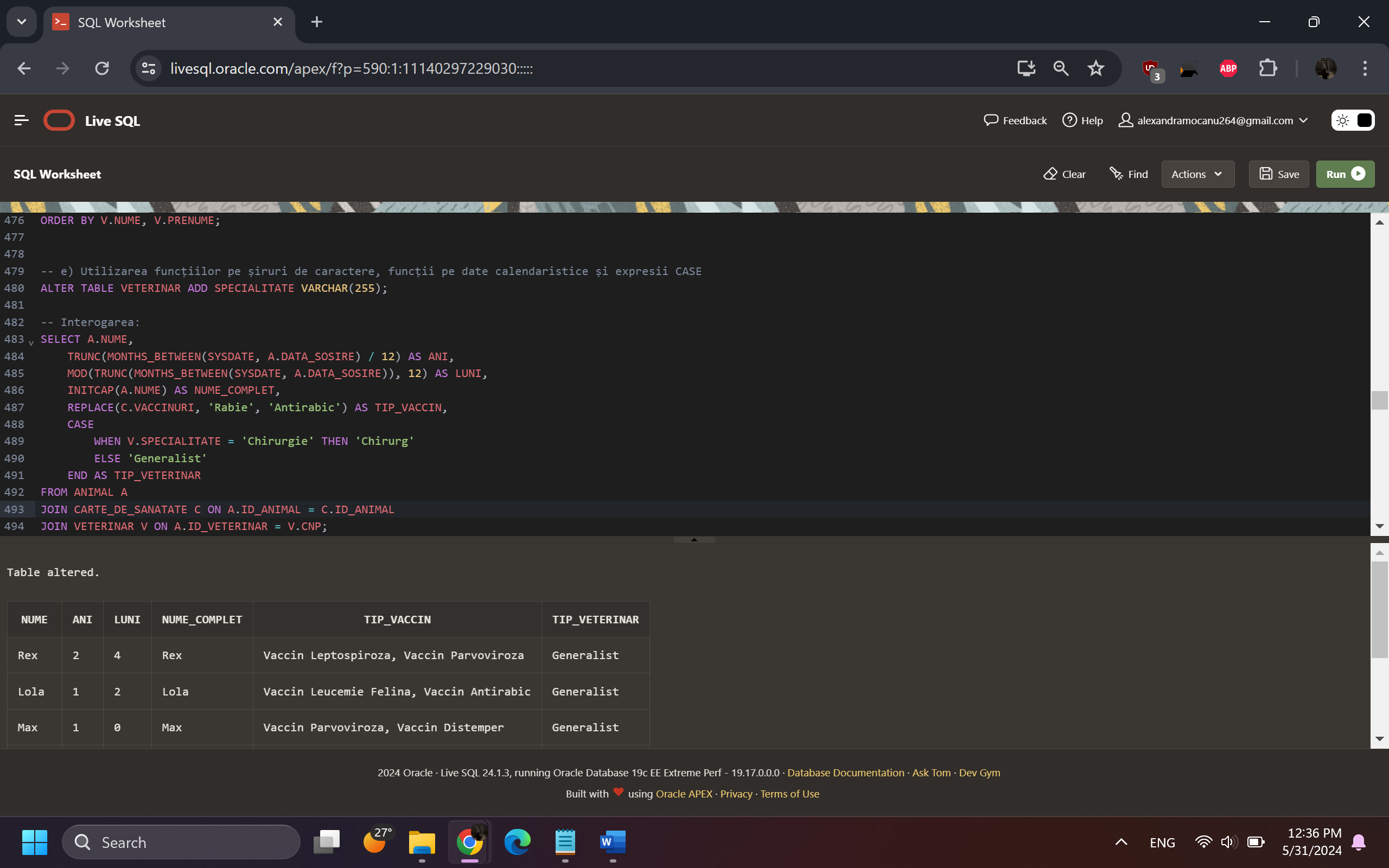
e) Funcții pe șiruri de caractere, funcții pe date calendaristice și expresii CASE

Interogarea selectează informații despre animale, durata șederii lor și tipul de vaccinuri, și determină specializarea veterinarilor. Utilizează tabelele ANIMAL, CARTE\_DE\_SANATATE și VETERINAR.

b și f) Subcereri nesincronizate în clauza FROM și utilizarea a cel puțin 1 bloc de cerere (clauza WITH)

Interogarea calculează numărul de animale pe fiecare secție, utilizând subcereri și clauza WITH. Folosește tabelele SECTIE, CUSCA și ANIMAL.





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

a) Operații de actualizare (UPDATE) utilizând subcereri

Actualizează datele din tabele în funcție de anumite condiții. De exemplu, actualizarea numărului de locuri din cuști în funcție de numărul de animale alocate.

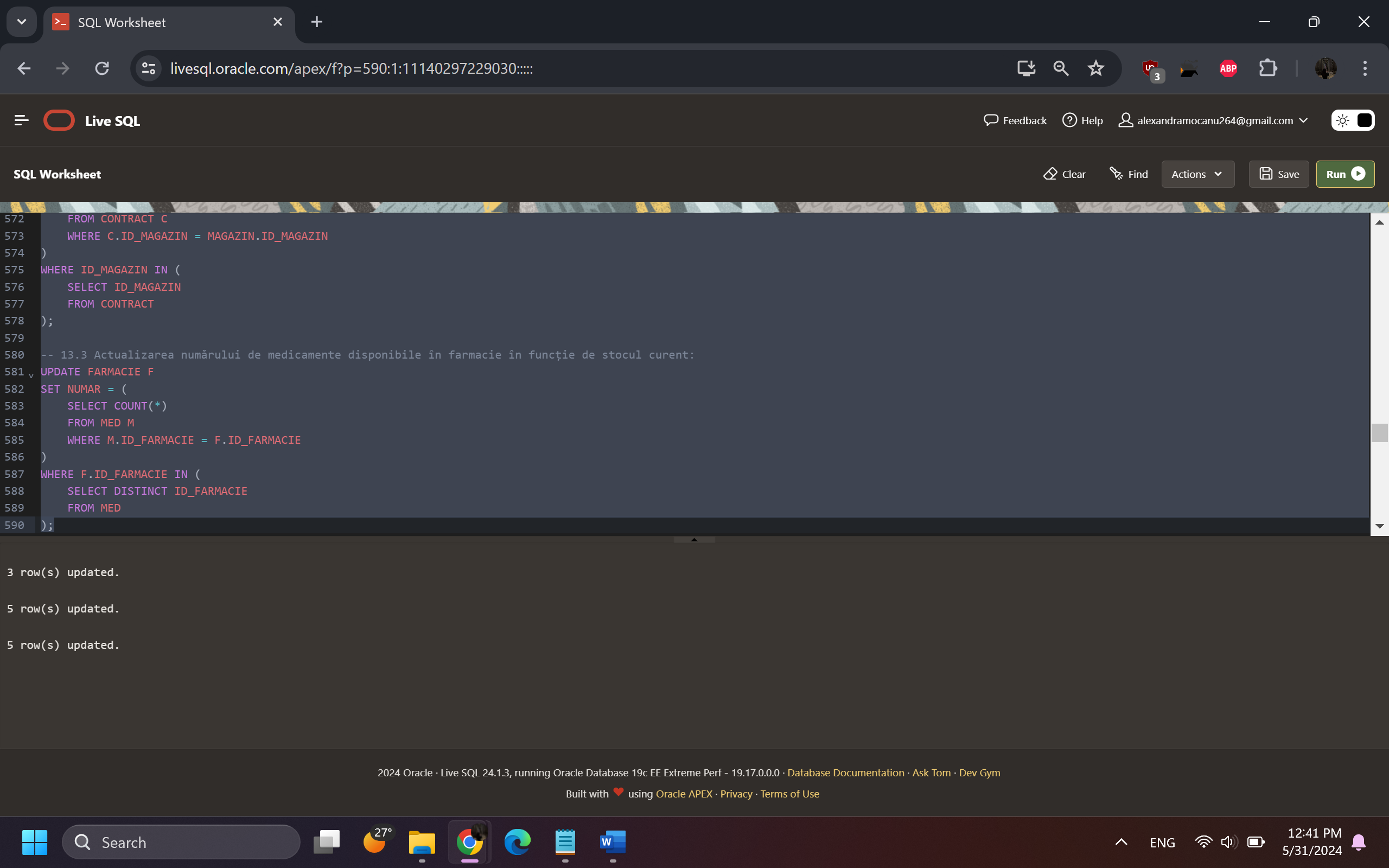
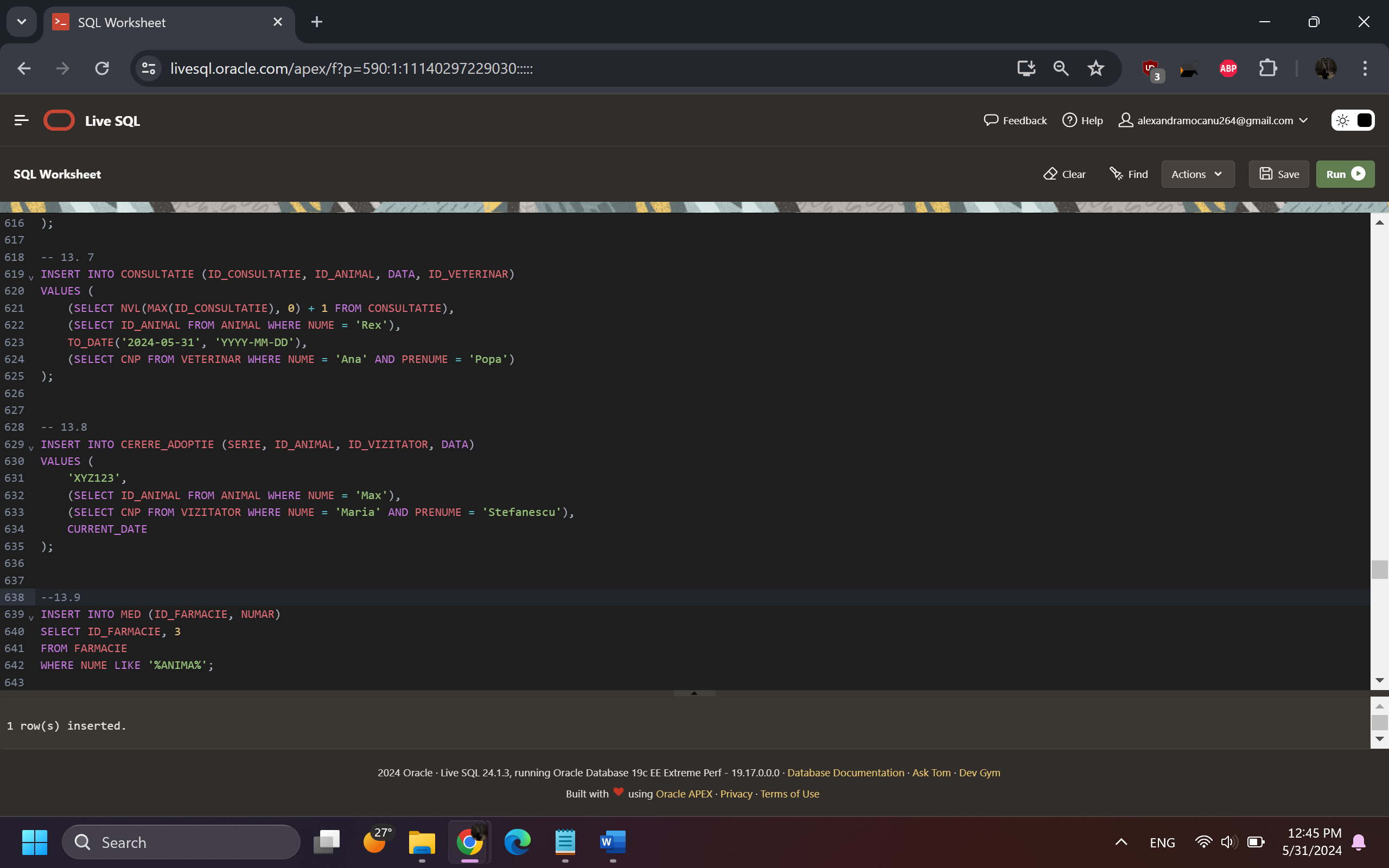
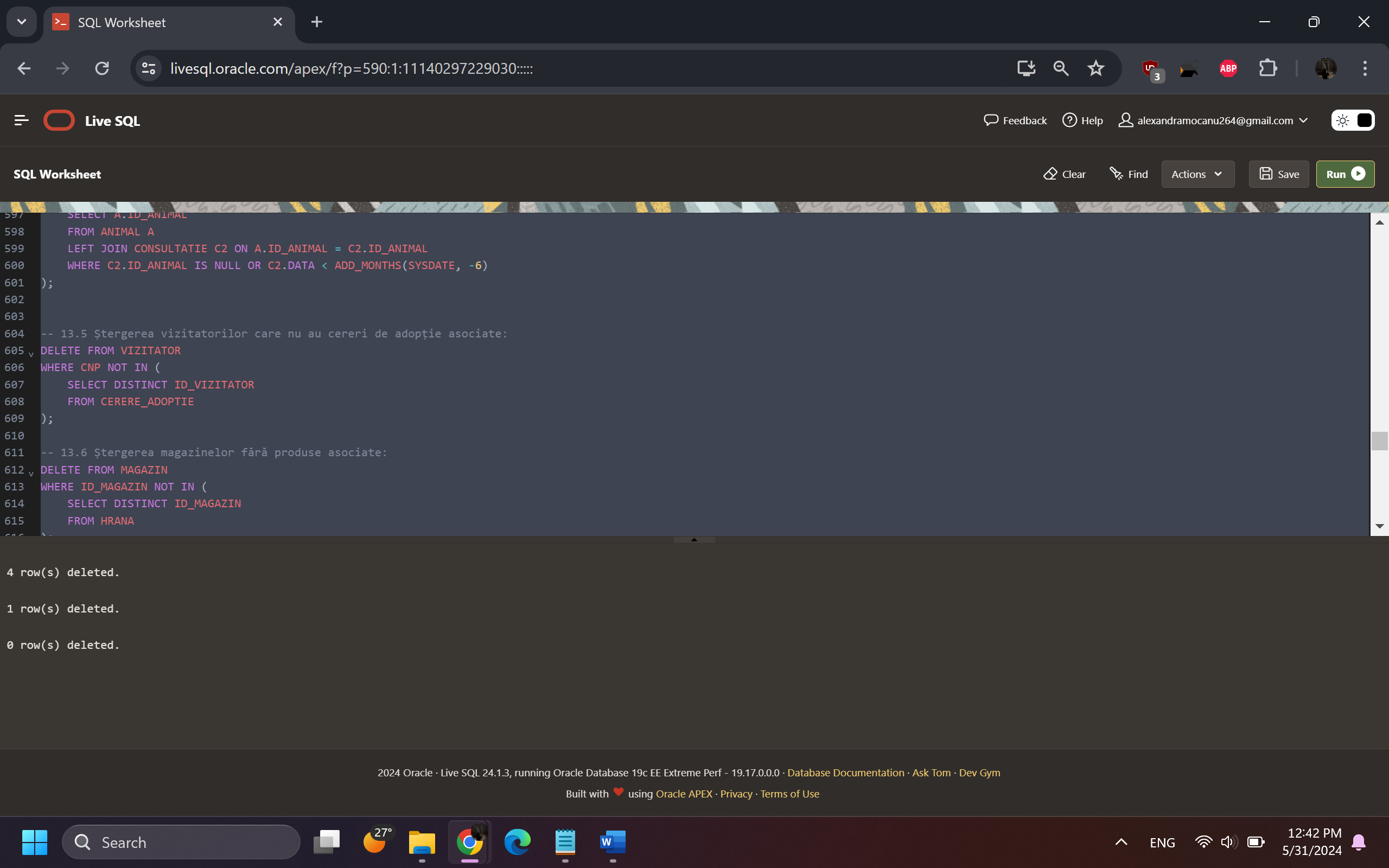
b) Operații de suprimare (DELETE) utilizând subcereri

Șterge înregistrări din tabele bazându-se pe anumite condiții. De exemplu, ștergerea animalelor care nu au fost vizitate în ultimele 6 luni.

c) Operații de inserare (INSERT) utilizând subcereri

Introduce noi înregistrări în tabele, folosind subcereri pentru a determina valorile corecte pentru câmpurile respective.

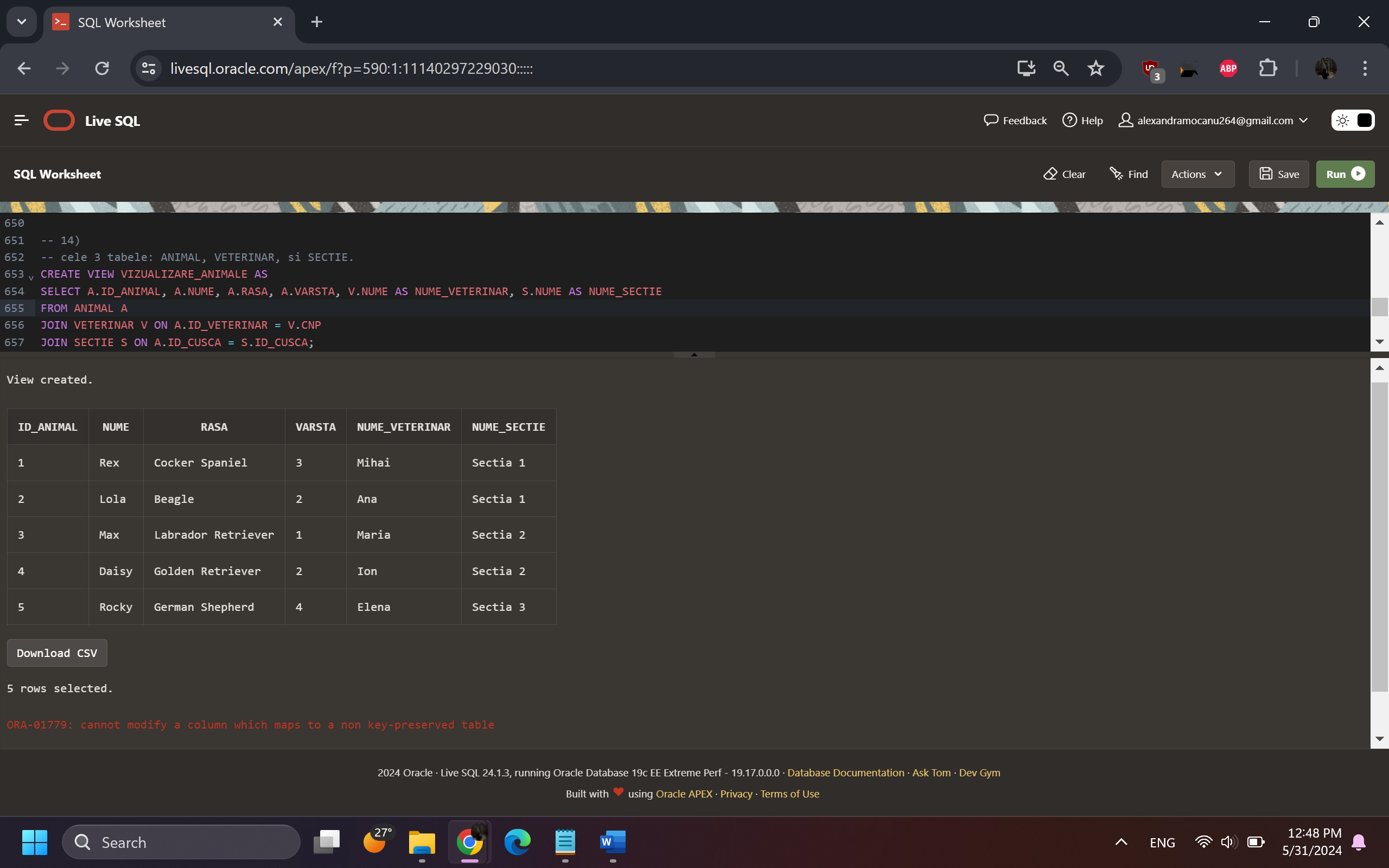


# Crearea unei vizualizari complexe

Creează o vizualizare care combină date din tabelele ANIMAL, VETERINAR și SECTIE, oferind o modalitate de a vizualiza datele combinate. Sunt exemplificate operații permise și nepermise pe această vizualizare.





# o cerere ce utilizează operația outer join pe minimum 4 tabele, o cerere ce utilizează operația division si o cerere care implementeaza analiza top-n

Operația outer join pe minimum 4 tabele

Selectează informații despre animale, secții, sectoare și vaccinuri folosind operații de tip outer join pentru a include toate animalele, chiar dacă nu au date asociate în toate tabelele.

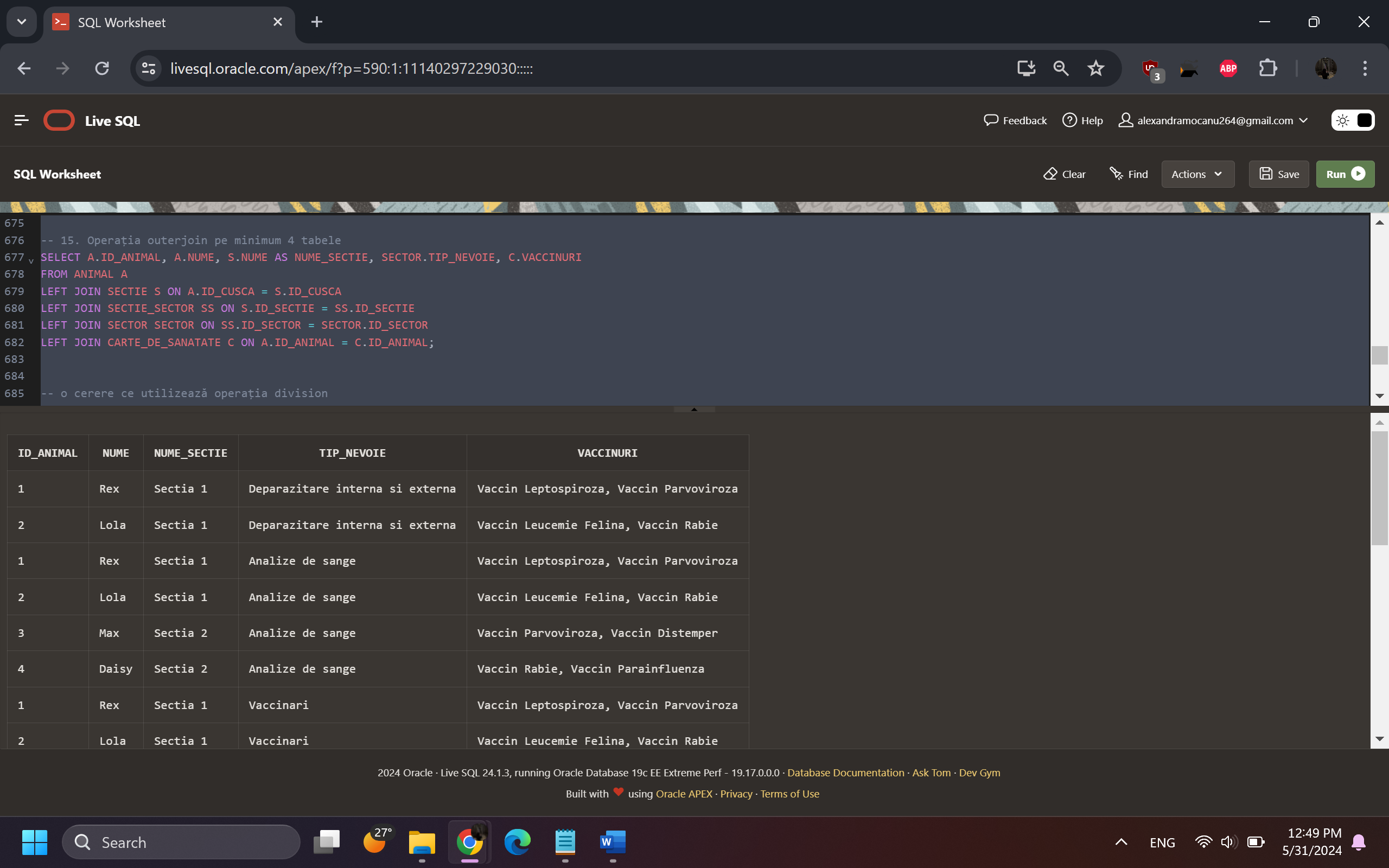
Cererea care utilizează operația division

Selectează numele animalelor care sunt prezente în toate sectoarele, utilizând o operație de tip division.

Analiza top-n

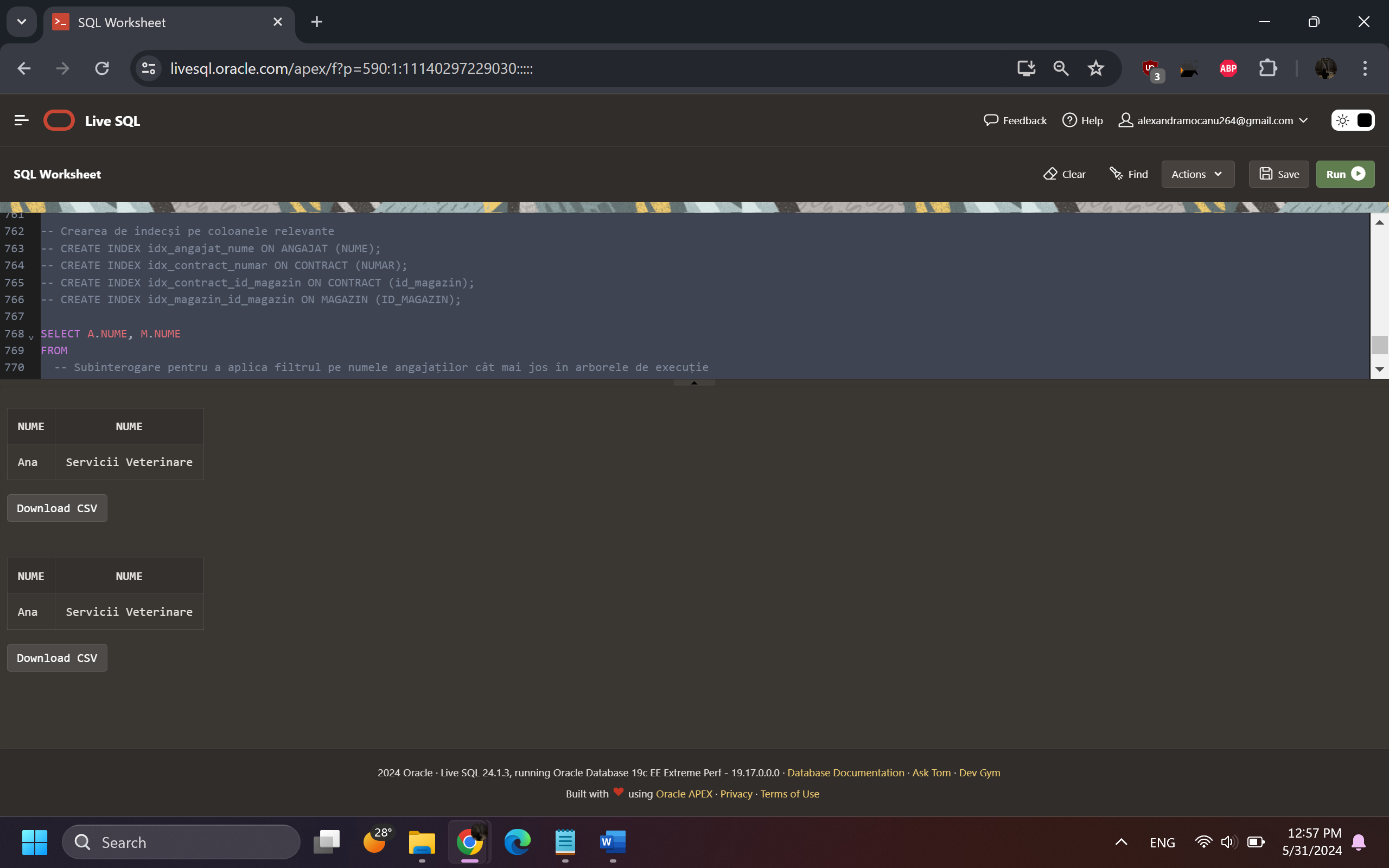
Selectează primele trei animale în funcție de vârstă, utilizând analiza top-n.





# 16. Optimizarea unei cereri





# a. Realizarea normalizării BCNF, FN4, FN5.

# Aplicarea denormalizării, justificând necesitatea acesteia



# Codul Intreg

