PROIECT BAZE DE DATE

ADMINISTRAREA UNUI LANT DE SALI DE FITNESS

134 PETCU MIRCEA

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

Proiectul are ca scop administrare unui lant de sali de fitness ce inglobeaza tot ce tine de acest domeniu: clienti, instructori,aparatura etc. Utilitatea acestei baze de date o reprezinta o adiministrare mult mai facila si mai rapida a salilor de fintness aflate in conglomerat ajutand, astfel, la cresterea profitului dar si a satisfactiei clientilor si urmareste dezvoltarea afacerii. Dintr-un lant de fac parte mai multe sali aflate in diferite locatii. Aceste sali, pentru a functiona au nevoie de angajati (intructori, receptioneri, oameni pentru curatenie etc.), de aparatura(de cardio, de fitness etc.), alte produse pentru vanzare (suplimente) si nu in ultimul rand, de clineti. Clientii isi fac abonamnete ce sunt de mai multe tipuri(standard sau premium) ce duce la cresterea incasarilor dar si capacitate mai mare de clienti pe care o sala le are este direct proportionala cu cheltuielile salii. Capacitate unei sali nu este oferita doar de locurile disponibile din vestiar ci si de numarul de aparate disponibile in sala, de numarul de intructori si de suprafata efectiva a salii.

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

Un client se poate numi client al sălii de fitness dacă a avut cel puțin un abonament la una dintre sălile de fitness de-a lungul timpului. Un departament sa nu aibe niciun angajat. Un departament poate sa nu aparțină niciunei liste de departamente. O sala de fitness pentru a funcționa, trebuie sa aibe o lista de departamente, o gama de aparate și o gama de suplimente dar poate sa nu aibe o lista de clienți (sa nu aibe clienți). Un abonament poate să fie deținut de mai mulți clienți.

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

Entitatiile sunt:

SALA\_DE\_FITNESS:

* cheia primara este sala\_id;
* este principalul tabel al bazei de date

ANGAJAT:

* cheia primara este angajat\_id
* reprezintă angajații ce lucrează în cadrul sălii din care fac parte

DEPARTAMENT:

* cheia primara este departament\_id
* reprezintă departamentele în care sunt impartiti angajații(instructori,conducere,curatenie etc.)

SUPLIMENT:

* cheia primara este supliment\_id
* reprezintă un produs ce poate fi vândut la recepție pentru un plus de profit

ABONAMENT:

* cheia primara este abonament\_id
* reprezint subscriptia prin care se face accesul în sala
* este împărțit în 2 subentitati(abonament\_standard, abonament\_premium)

APARAT\_FITNESS:

* cheia primara este aparat\_fitness\_id
* reprezintă aparatura cu care este dotata salile(gantere, aparat pentru hipertrofie, cardio etc.)

LOCATIE:

* cheia primara este locatie\_id
* reprezintă locația (țara, judetul,orasul) în care este localizata sala

ADRESA:

* cheia primara este adresa\_id
* reprezint locația exactă din orașul în care este localizata sala

CLIENTI:

* cheia primara este client\_id
* reprezintă persoana ce beneficiaza de serviciile și produsele oferite de sala în schimbul unei sume de bani

LISTA\_CLIENTI

* chei primara este lista\_clienti\_id
* rerezinta lista tuturor clientilor care apartin de o sala

GAMA\_APARATE

* cheia primara este gama\_aparate\_id
* reprezint gama de aparate de fitness care le deține o sala

GAMA\_SUPLIMNETE

* cheia primara este gama\_suplimente\_id
* reprezintă gama de suplimente pe care o sala o deține

LISTA\_DEPARTAMENTE

* chiea primara este lista\_dep\_id#
* reprezintă lista de departamente din cadrul unei sali de fitness

ABONAMENT\_STANDARD

* cheia primara este abonament\_id#
* reprezint o subentitate a enititatii ABONAMENT, un tip de abonament de baza

ABONAMENT\_PREMIUM

* cheia primara este abonament\_id#
* reprezint o subentitate a enititatii ABONAMENT, un tip de abonament exclusist

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

SALA\_DE\_FITNESS – LOCATIE:

* relație binara
* verb: este localizata
* mai multe sali pot fi localizate într-o locatie
* nicio sala nu trebuie să fie localizata într-o locatie
* o sala poate fi localizata în cel mult o locatie
* o sala trebuie să fie localizata într-o locatie
* 1 – M(0)

LOCATIE – ADRESA

* relație binara
* verb: are
* o locatie poate sa aibe o adresa
* o locatie trebuie sa aibe o adresa
* o adresa poate sa aibe o locatie
* o adresa poate trebuie sa aibe o locatie
* 1 – 1

ANGAJAT – DEPARTAMENTE

* relație binara
* verb: lucreaza
* un angajat poate sa lucreze într-un departament
* un angajat trebuie sa lucreze într-un departament
* intr-un departamet poate sa lucreze mai mulți angajați
* intr-un departamet nu trebuie sa lucreze niciun angajat
* M(0) – 1

CLIENTI – ABONAMENT

* relație binara
* verb: deține
* un client poate deține mai multe abonamente
* un client trebuie sa dețina un abonament
* un abonament poate să fie deținut de mai mulți clienti
* un abonament trebuie să fie deținut de niciun client
* M(0) – M(1)

SALA\_DE\_FITNESS – LISTA\_CLIENTI

* relație binara
* verb: are
* o sala de fitness poate sa aibe o lista de clienți
* o sala de fitness nu trebuie sa aibe nicio lista de clienți
* o lista de clienți poate sa aparțină de o singura sala
* o lista de clienți trebuie sa aparțină de o singura de sala
* 1 - 1(0)

LISTA\_CLIENTI – CLIENTI

* relație binara
* verb: aparține
* de o lista de clienți poate sa apatina mai mulți clienți
* de o lista de clienți trebuie sa aparțină
* un client poate sa aparțină de o lista de clienți
* un client trebuie sa aparțină de o lista de clienți
* 1 – M(0)

SALA\_DE\_FITNESS – GAMA\_APARATE

* relație binara
* o sala de fitness poate sa aibe o gama de aparate
* o sala de fitness trebuie sa aibe cel puțin o gama de aparate
* o gama de aparate poate sa aparțină mai multe sali
* o gama de aparate trebuie sa aparțină de nicio sala
* 1 – M(0)

GAMA\_APARATE – APARAT\_FITNESS

* relație binara
* verb: include
* o gama de aparata poate sa includa mai multe aparate de fitness
* o gama de aparate trebuie sa includa niciun aparat
* un aparat poate să fie inclus într-o gama de aparate
* un aparat de fitness trebuie să fie inclus în nicio gama de aparate
* 1(0) – M(0)

SALA\_DE\_FITNESS – GAMA\_SUPLIMENTE

* relație binara
* o sala de fitness poate sa aibe o gama de aparate
* o sala de fitness trebuie sa aibe cel puțin o gama de aparate
* o gama de aparate poate sa aparțină de mai multe sali
* o gama de aparate trebuie sa aparțină de nicio sala
* 1 – M(0)

GAMA\_SUPLIMENTE – SUPLIMENT

* relație binara
* verb: include
* o gama de suplimente poate sa includa mai multe suplimente
* o gama de suplimente trebuie sa includa niciun supliment
* un supliment poate să fie inclus într-o gama de suplimente
* un supliment trebuie să fie inclus în nicio gama de suplimente
* 1(0) – M(0)

SALA\_DE\_FITNESS – LISTA\_DEPARTAMENTE

* relație binara
* verb: include
* o sala de fitness poate sa includa o singura lista de departamente
* o sala de fitness poate sa includa o singura lista de departamente
* o lista de departamente poate să fie inclusa în mai multe sali
* o lista de departamente trebuie să fie inclusa în nicio sala
* 1 – M(0)

LISTA\_DEPARTAMENTE – DEPARTAMENT

* relație binara
* verb: include
* o lista poate sa include mai multe departamente
* o lista trebuie sa includa niciun departament
* un departament poate să fie inclus în mai multe liste
* un departament trebuie să fie inclus în nicio lista
* M(0) - M(0)

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

SALA\_DE\_FITNESS

* sala\_id – number(6), cheie primara
* denumire- varchar2(30), cum se numește sala
* email – varchar2(30),unic, adresa unde clientii pot discuta cu conducerea sălii de eventuale plângeri, optimizari ale sălii etc., nenula
* locatie\_id – number(6), cheie externa referentiata la tabela LOCATIE
* gama\_aparate\_id – number(6), cheie externa referntiata la tabela GAMA\_APARATE
* gama\_suplimente\_id – number(6), cheie externa referentiata la tabela GAMA\_SUPLIMENTE

LOCATIE

* locatie\_id – number(6), cheie primara
* țara – varchar2(30), țara în care este localizata sala
* regiune – varchar2(30), regiunea în care este localizata sala
* oras – varchar2(30), orasul în care este localizata sala

ADRESA

* adresa\_id – number(6),cheie primara
* localie\_id – number(6),cheie primara
* strada- varchar2(30), numele strazii
* numar – number(4), numărul strazii
* cod-postal – number(5), codul postal

ANGAJAT

* angajat\_id – number(6), cheie primara
* nume – varchar2(30), numele angajatului
* prenume – varchar2(30), prenumele angajatului
* email – varchar2(30), ,unic,email unde poate fi contactat angajatul
* salariu- number(7,2), mai mare decât 1500, salariul angajatului
* departament\_id – number(6), cheie externa referentiata la tabela DEPARTAMENT

DEPARTAMENT

* departament\_id – number(6), cheie primara
* denumire – varchar(30), denumirea departamentului

SUPLIMENT

* supliment\_id – number(6), cheie primara
* denumire – varchar(30), denumirea suplimentului
* pret- number(6,2), nenul, prețul suplimentului
* cantitate – number(4,2), nenul, cantitatea efectiva a suplimentului
* gama\_suplimente\_id -number(6),nenula, cheie externa referentiata la tabela GAMA\_SUPLIMENTE

CLIENTI

* client\_id – number(6), cheie primara
* nume – varchar2(30), numele clientului
* prenume – varchar2(30), prenumele clientului
* email – varchar2(30),unic, adresa unde se poate contacta clientului
* lista\_clienti\_id – number(6), nenula,cheie externa referentiata la tabela LISTA\_CLIENTI

ABONAMENT

* abonament\_id – number(6), cheie primara
* data\_incepere – date,nenula, data când a fost creat abonamentul
* data\_sfarsit – date,nenula, data când se termina abonamentul

ABONAMENT\_STANDARD

* abonament\_id –number(6), cheie primara
* data\_incepere – date,nenula, data când a fost creat abonamentul
* data\_sfarsit – date,nenula, data când se termina abonamentul
* pret\_standard – number(5,2), prețul abonamentului standard

ABONAMENT\_PREMIUM

* abonament\_id – number(6), cheie primara
* data\_incepere – date,nenula, data când a fost creat abonamentul
* data\_sfarsit – date,nenula, data când se termina abonamentul
* pret\_premium – number(6,2), prețul abonamentului premium

APARAT\_FITNESS

* aparat\_fitness\_id – number(6),cheie primara
* denumire – varchar2(30), denumirea aparatului
* preț – number(10,2), prețul aparatului
* gama\_aparate\_id – number(6), nenula,cheie externa referentiata la tabela GAMA\_APARATE

GAMA\_APARATE

* gama\_aparate\_id – number(6),cheie primara

GAMA\_SUPLIMENTE

* gama\_suplimente\_id – number(6), cheie primara

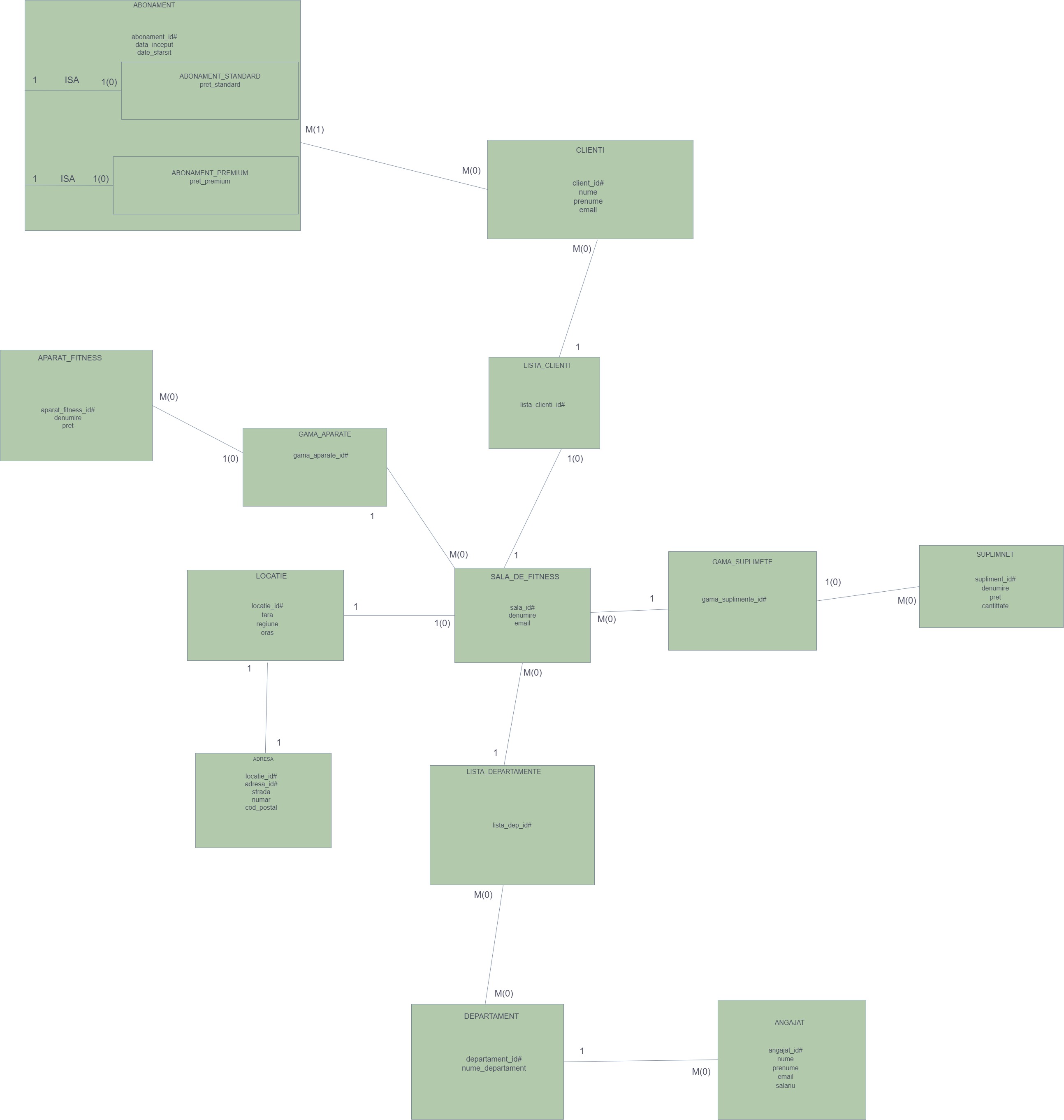
LISTA\_DEPARTAMENTE

* lista\_departamente\_id – number(6), cheie primara

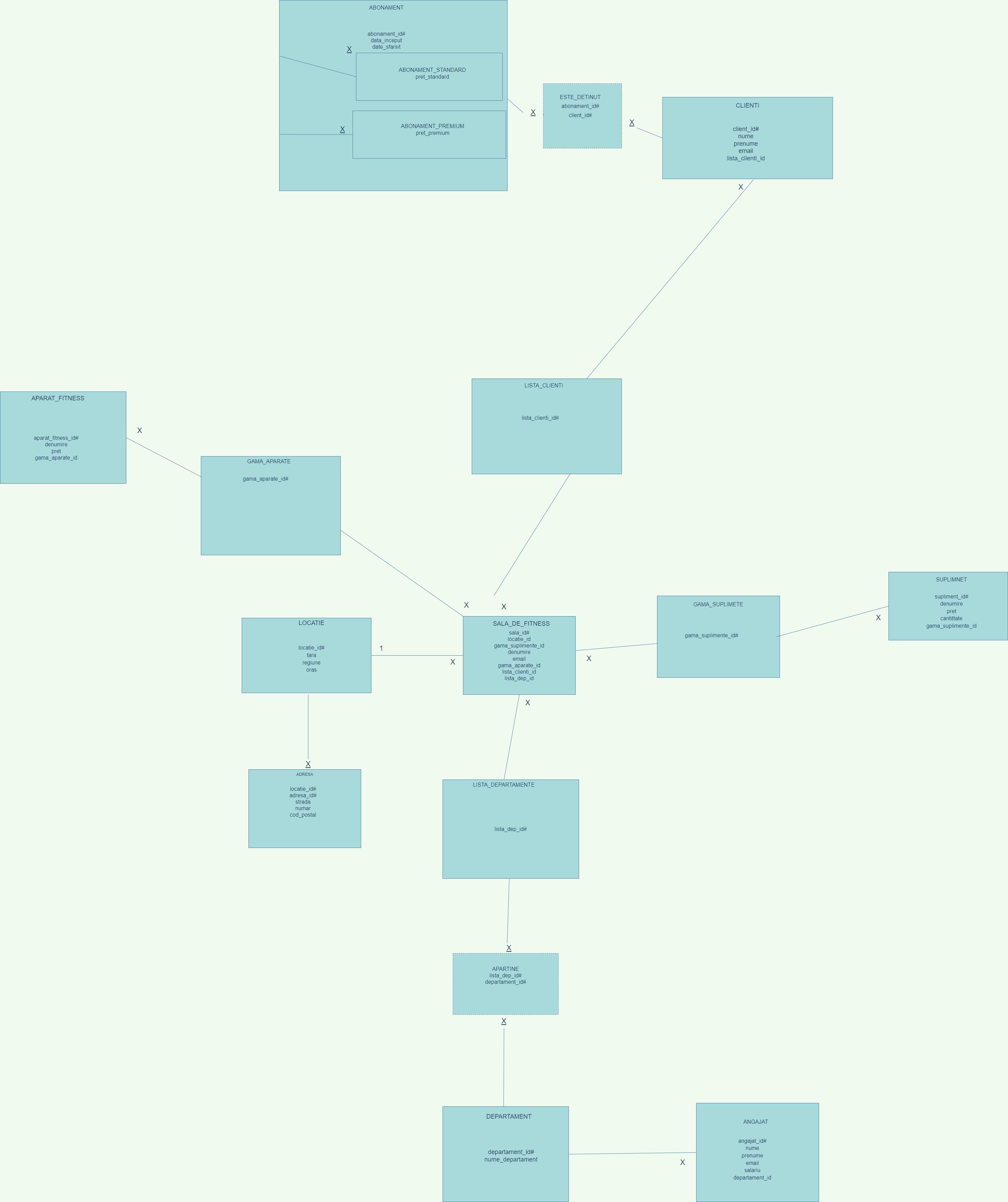
LISTA\_CLIENTI

* lista\_clienti\_id – number(6), cheie primara

**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.**

SALA\_DE\_FITNESS(sala\_id#,denumire,email,locatie\_id,gama\_suplimente\_id,gama\_aparate\_id,lista\_clienti\_id,lista\_dep\_id)

LOCATIE (locatie\_id,tara,regiune,oras)

ADRESA (adresa\_id#,locatie\_id#,strada, numar, cod\_postal)

LISTA\_DEPARTAMENTE(lista\_dep\_id#)

APARTINE(lista\_dep\_id#,departament\_id#)

DEPARTAMENT (departament\_id#,nume\_departament)

ANGAJAT (angajat\_id#,nume,prenume,email,salariu,departament\_id)

GAMA\_APARATE (gama\_aparate\_id#)

APARAT\_FITNESS(aparat\_fitness\_id#,denumire,pret,gama\_aparat\_id)

GAMA\_SUPLIMENTE (gama\_suplimente\_id#)

SUPLIMENT (supliment\_id#,denumire,pret,cantitate,gama\_suplimente\_id)

LISTA\_CLIENTI (lista\_clienti\_id#)

CLIENT (client\_id#,nume,prenume,email,lista\_clienti\_id)

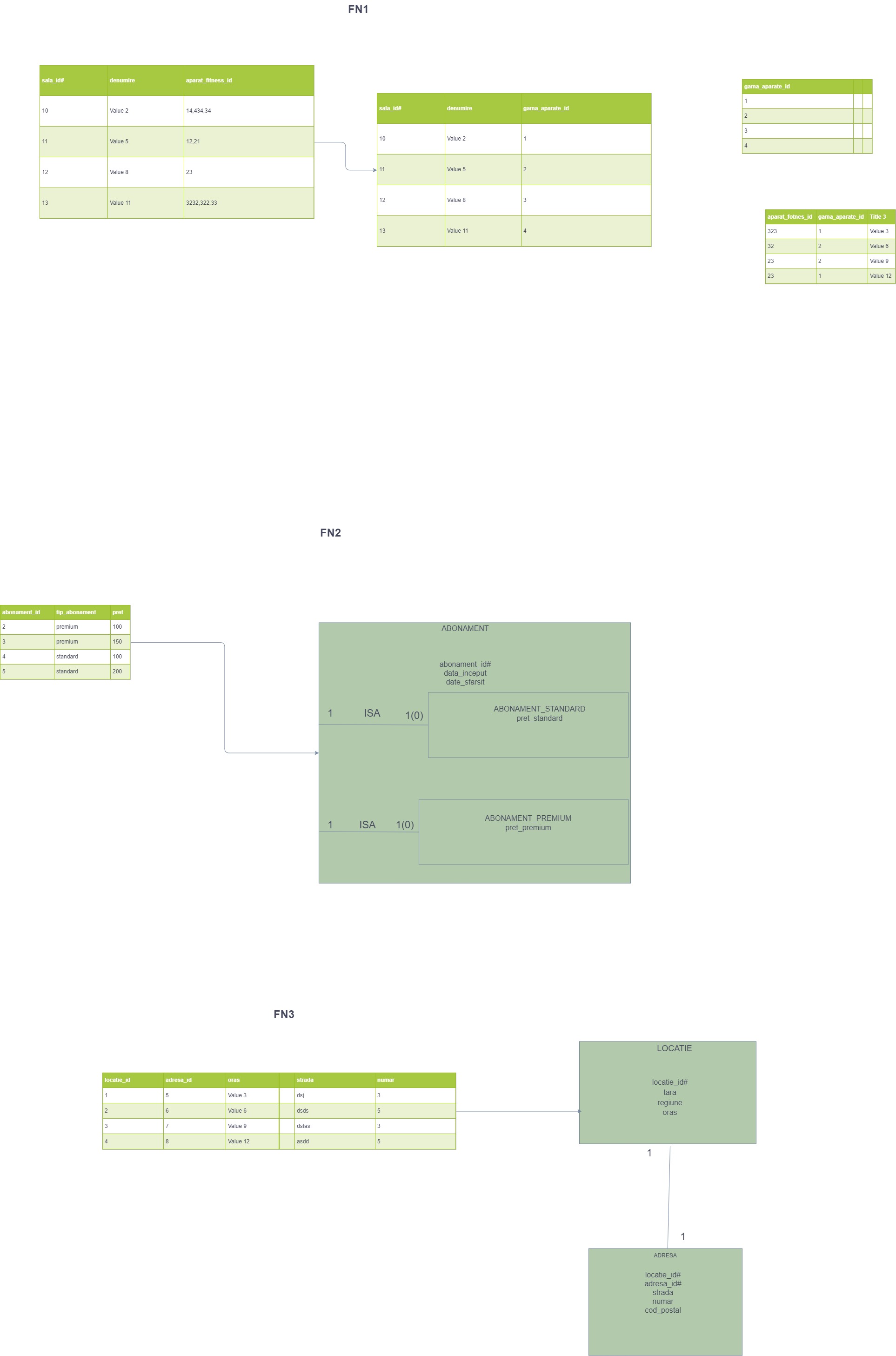
ABONAMENT (abonament\_id#,data\_inceput,data\_sfarsit)

ESTE\_DETINUT (abonament\_id#,client\_id#)

ABONAMENT\_STANDARD (abonament\_id#, data\_inceput, data\_sfarsit, pret\_standard)

ABONAMENT\_PREMIUM (abonament\_id#, data\_inceput, data\_sfarsit,pret\_premium)

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

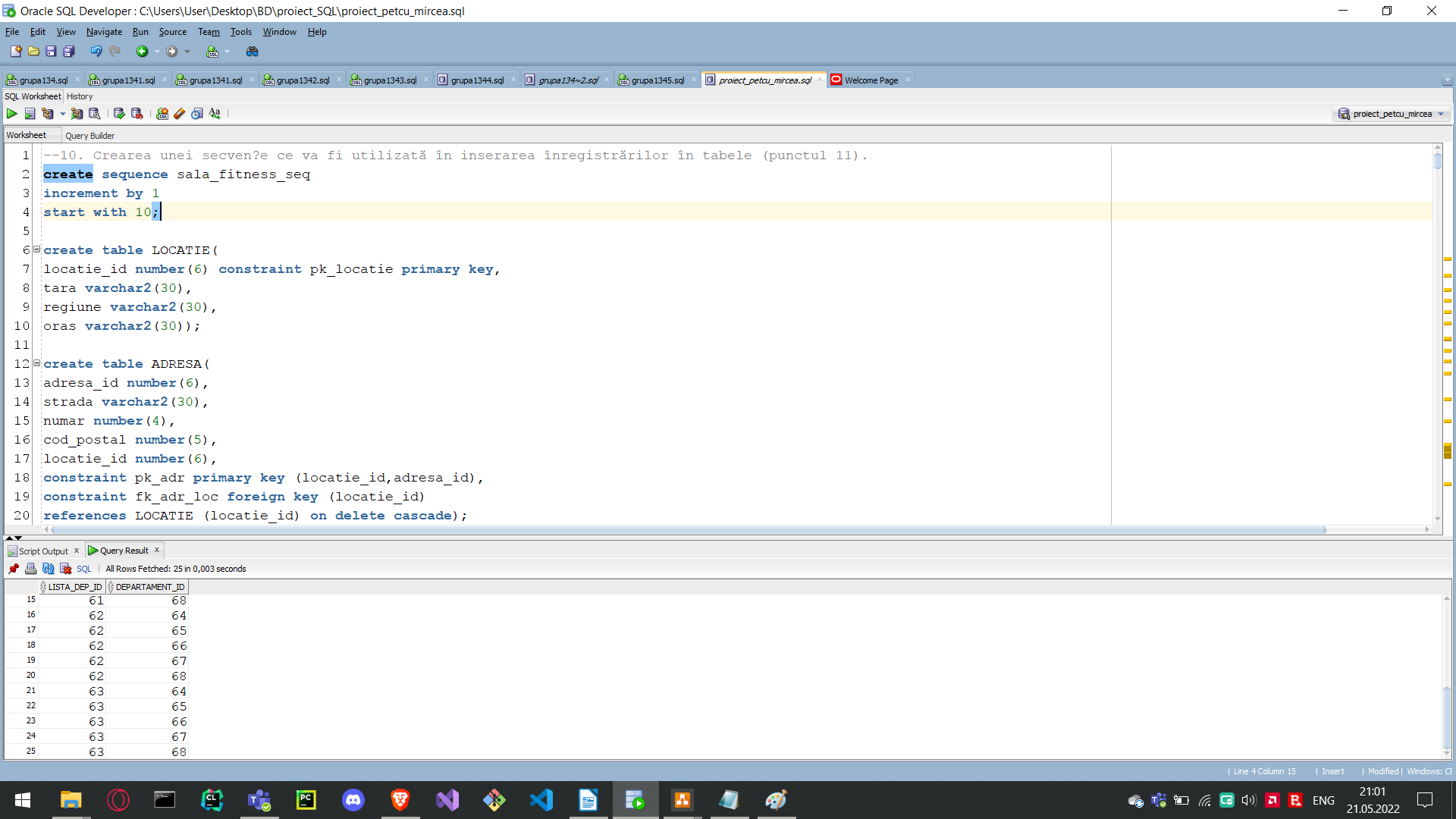
****

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

create sequence sala\_fitness\_seq

increment by 1

start with 10;



**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 LOCATIE(

locatie\_id number(6) constraint pk\_locatie primary key,

tara varchar2(30),

regiune varchar2(30),

oras varchar2(30));

create table ADRESA(

adresa\_id number(6),

strada varchar2(30),

numar number(4),

cod\_postal number(5),

locatie\_id number(6),

constraint pk\_adr primary key (locatie\_id,adresa\_id),

constraint fk\_adr\_loc foreign key (locatie\_id)

references LOCATIE (locatie\_id) on delete cascade);

drop table adresa;

create table GAMA\_SUPLIMENTE(

gama\_suplimente\_id number(6) constraint pk\_gama\_supl primary key);

create table GAMA\_APARATE(

gama\_aparate\_id number(6) constraint pk\_gama\_ap primary key);

create table SUPLIMENT(

supliment\_id number(6) constraint pk\_supliment primary key,

denumire varchar2(30),

pret number(6,2),

cantitate number(4,2),

gama\_suplimente\_id number(6),

constraint fk\_gam\_sup foreign key (gama\_suplimente\_id)

references GAMA\_SUPLIMENTE (gama\_suplimente\_id) on delete cascade);

create table APARAT\_FITNESS(

aparat\_fitness\_id number(6) constraint pk\_aparat primary key,

denumire varchar2(30),

pret number(10,2),

gama\_aparate\_id number(6),

constraint fk\_gam\_ap foreign key (gama\_aparate\_id)

references GAMA\_APARATE (gama\_aparate\_id) on delete cascade);

create table DEPARTAMENT(

departament\_id number(6) constraint pk\_dep primary key,

nume\_departament VARCHAR2(30));

create table ANGAJAT(

angajat\_id number(6) constraint pk\_ang primary key,

nume varchar2(30),

prenume varchar2(30),

email varchar2(30) constraint uniq\_em\_ang unique,

salariu number(7,2),

departament\_id number(6),

constraint ck\_sal check(salariu >1500),

constraint fk\_ang\_dep foreign key (departament\_id)

references DEPARTAMENT (departament\_id) on delete cascade);

create table LISTA\_DEPARTAMENTE(

lista\_dep\_id number(6) constraint pk\_list\_dep primary key);

create table LISTA\_CLIENTI(

lista\_clienti\_id number(6) constraint pk\_list\_clienti primary key);

create table CLIENTI(

client\_id number(6) constraint pk\_client primary key,

nume varchar2(30),

prenume varchar2(30),

email varchar2(30),

lista\_clienti\_id number(6) constraint nn\_cl not null,

constraint fk\_list\_cli foreign key (lista\_clienti\_id)

references LISTA\_CLIENTI (lista\_clienti\_id) on delete cascade);

create table ABONAMENT(

abonament\_id number(6) constraint pk\_abon primary key,

data\_inceput date,

data\_sfarsit date);

create table ABONAMENT\_STANDARD(

abonament\_id number(6) constraint pk\_standard primary key,

data\_inceput date,

data\_sfarsit date,

pret\_standard number(5,2),

constraint fk\_standard foreign key (abonament\_id)

references ABONAMENT (abonament\_id) on delete cascade

);

create table ABONAMENT\_PREMIUM(

abonament\_id number(6) constraint pk\_premium primary key,

data\_inceput date,

data\_sfarsit date,

pret\_premium number(6,2),

constraint fk\_premium foreign key (abonament\_id)

references ABONAMENT (abonament\_id) on delete cascade

);

create table ESTE\_DETINUT(

abonament\_id number(6),

client\_id number(6),

constraint pk\_detinut primary key (abonament\_id,client\_id),

constraint fk\_detinut1 foreign key (abonament\_id) references ABONAMENT (abonament\_id),

constraint fk\_detinut2 foreign key (client\_id) references CLIENTI (client\_id));

create table APARTINE(

lista\_dep\_id number(6),

departament\_id number(6),

constraint pk\_apartine primary key (lista\_dep\_id,departament\_id),

constraint fk\_apartine1 foreign key (lista\_dep\_id) references LISTA\_DEPARTAMENTE (lista\_dep\_id),

constraint fk\_apartine2 foreign key (departament\_id) references DEPARTAMENT (departament\_id));

create table SALA\_DE\_FITNESS(

sala\_id number(6) constraint pk\_sala\_pm primary key,

denumire varchar2(30),

lista\_clienti\_id number(6),

email varchar2(40) constraint uniq\_den\_sala unique,

locatie\_id number(6) constraint null\_locatie not null,

gama\_aparate\_id number(6) constraint null\_aparate not null,

gama\_suplimente\_id number(6) constraint null\_suplimente not null,

lista\_dep\_id number(6),

constraint fk\_sala1 foreign key (lista\_clienti\_id) references LISTA\_CLIENTI (lista\_clienti\_id),

constraint fk\_sala2 foreign key (lista\_dep\_id) references LISTA\_DEPARTAMENTE (lista\_dep\_id),

constraint fk\_sala3 foreign key (gama\_aparate\_id) references GAMA\_APARATE (gama\_aparate\_id) ,

constraint fk\_sala4 foreign key (gama\_suplimente\_id) references GAMA\_SUPLIMENTE (gama\_suplimente\_id) ,

constraint fk\_sala5 foreign key (locatie\_id) references LOCATIE (locatie\_id)

);

insert into GAMA\_APARATE (gama\_aparate\_id)

values (10);

insert into GAMA\_APARATE (gama\_aparate\_id)

values (SALA\_FITNESS\_SEQ.nextval);

select \* from gama\_suplimente;

insert into aparat\_fitness

values (17,'squat\_rack',2000.5,12);

delete from aparat\_fitness where aparat\_fitness\_id = 17;

insert into aparat\_fitness

values (sala\_fitness\_seq.nextval,'presa\_picioare',5000.45,13);

insert into aparat\_fitness

values (sala\_fitness\_seq.nextval,'aparat\_ramat',2440.5,14);

insert into aparat\_fitness

values (sala\_fitness\_seq.nextval,'banca\_smith',8000,15);

insert into aparat\_fitness

values (sala\_fitness\_seq.nextval,'gantere\_30kg',3000.5,16);

insert into aparat\_fitness

values (sala\_fitness\_seq.nextval,'extensii\_cvadriceps',30660.5,10);

insert into aparat\_fitness

values (sala\_fitness\_seq.nextval,'helcometru',40660.5,16);

insert into aparat\_fitness

values (sala\_fitness\_seq.nextval,'bara\_olimpica',50660.5,12);

SELECT \*

from aparat\_fitness;

insert into gama\_suplimente

values (sala\_fitness\_seq.nextval);

insert into supliment

values(sala\_fitness\_seq.nextval,'proteina\_din\_zer',250,5,28);

insert into supliment

values(sala\_fitness\_seq.nextval,'proteina\_din\_soia',200,5,27);

insert into supliment

values(sala\_fitness\_seq.nextval,'beta-alanina',100,1,26);

insert into supliment

values(sala\_fitness\_seq.nextval,'creatina',80,1,25);

insert into supliment

values(sala\_fitness\_seq.nextval,'citrulina',120,1,30);

insert into supliment

values(sala\_fitness\_seq.nextval,'taurina',110,1.2,30);

select \*from supliment;

--locatie

insert into locatie

values (sala\_fitness\_seq.nextval,'Romania','Ilfov','Voluntari');

insert into locatie

values (sala\_fitness\_seq.nextval,'Romania','Bucuresti','Bucuesti');

insert into locatie

values (sala\_fitness\_seq.nextval,'Romania','Cluj','Cluj-Napoca');

insert into locatie

values (sala\_fitness\_seq.nextval,'Romania','Constanta','Constanta');

insert into locatie

values (sala\_fitness\_seq.nextval,'Romania','Iasi','Iasi');

select \* from locatie;

--adresa

insert into adresa

values(sala\_fitness\_seq.nextval,'Copou',33,5582,49);

insert into adresa

values(sala\_fitness\_seq.nextval,'Suceava',3,5322,46);

insert into adresa

values(sala\_fitness\_seq.nextval,'Calea Victorie',333,4322,47);

insert into adresa

values(sala\_fitness\_seq.nextval,'Buna ziua',43,4542,48);

insert into adresa

values(sala\_fitness\_seq.nextval,'Ion Creanga',4,2542,45);

select \* from adresa;

--lista\_departamente

insert into lista\_departamente

values (sala\_fitness\_seq.nextval);

insert into lista\_departamente

values (sala\_fitness\_seq.nextval);

insert into lista\_departamente

values (sala\_fitness\_seq.nextval);

insert into lista\_departamente

values (sala\_fitness\_seq.nextval);

insert into lista\_departamente

values (sala\_fitness\_seq.nextval);

select \* from lista\_departamente;

--departamente

insert into departament

values (sala\_fitness\_seq.nextval, 'receptie');

insert into departament

values (sala\_fitness\_seq.nextval, 'instructori');

insert into departament

values (sala\_fitness\_seq.nextval, 'curatenie');

insert into departament

values (sala\_fitness\_seq.nextval, 'contabilitate');

insert into departament

values (sala\_fitness\_seq.nextval, 'paza');

--59-63

--64-68

--apartine - am schimbat doar cifrele si am inserat de 5\*5 = 25 de ori

insert into apartine

values(63,68);

select \* from lista\_departamente;

select \* from departament;

--angajat

insert into angajat

values(sala\_fitness\_seq.nextval,'Maria','Andreea','ma@gmail.com',9300,68);

insert into angajat

values(sala\_fitness\_seq.nextval,'Petcu Mircea', 'mp@gmail.com', 5000,65);

insert into angajat

values(sala\_fitness\_seq.nextval,'Petcu Gabriel', 'gp@gmail.com', 6800, 64);

insert into angajat

values(sala\_fitness\_seq.nextval,'Ion Andrei' ,'ia@gmail.com', 8800, 67);

insert into angajat

values(sala\_fitness\_seq.nextval,'Iona Andreea', 'iaa@gmail.com', 9800, 66);

select \* from angajat;

--lista\_clienti

insert into lista\_clienti

values (sala\_fitness\_seq.nextval);

insert into lista\_clienti

values (sala\_fitness\_seq.nextval);

insert into lista\_clienti

values (sala\_fitness\_seq.nextval);

insert into lista\_clienti

values (sala\_fitness\_seq.nextval);

insert into lista\_clienti

values (sala\_fitness\_seq.nextval);

select \* from lista\_clienti;

--75-78

--clienti

insert into clienti

values (sala\_fitness\_seq.nextval,'Alex Ion', 'ai@gmail.com', 74);

insert into clienti

values (sala\_fitness\_seq.nextval,'Alexa Ionel', 'aaii@gmail.com', 76);

insert into clienti

values (sala\_fitness\_seq.nextval,'Andrei Ioana','aai@gmail.com', 78);

insert into clienti

values (sala\_fitness\_seq.nextval,'Ciot Ioana','ci@gmail.com', 77);

insert into clienti

values (sala\_fitness\_seq.nextval,'Ciotan','Maria','cm@gmail.com',75);

select \* from clienti;

--abonament

insert into abonament

values (sala\_fitness\_seq.nextval,TO\_DATE('17/11/2021', 'DD/MM/YYYY'),TO\_DATE('17/12/2021', 'DD/MM/YYYY'));

insert into abonament

values (sala\_fitness\_seq.nextval,TO\_DATE('17/11/2022', 'DD/MM/YYYY'),TO\_DATE('17/12/2022', 'DD/MM/YYYY'));

insert into abonament

values (sala\_fitness\_seq.nextval,TO\_DATE('17/09/2021', 'DD/MM/YYYY'),TO\_DATE('17/10/2021', 'DD/MM/YYYY'));

insert into abonament

values (sala\_fitness\_seq.nextval,TO\_DATE('17/08/2021', 'DD/MM/YYYY'),TO\_DATE('17/09/2021', 'DD/MM/YYYY'));

insert into abonament

values (sala\_fitness\_seq.nextval,TO\_DATE('17/07/2021', 'DD/MM/YYYY'),TO\_DATE('17/08/2021', 'DD/MM/YYYY'));

insert into abonament

--

values (sala\_fitness\_seq.nextval,TO\_DATE('13/07/2021', 'DD/MM/YYYY'),TO\_DATE('13/08/2021', 'DD/MM/YYYY'));

insert into abonament

values (sala\_fitness\_seq.nextval,TO\_DATE('10/07/2021', 'DD/MM/YYYY'),TO\_DATE('10/08/2021', 'DD/MM/YYYY'));

insert into abonament

values (sala\_fitness\_seq.nextval,TO\_DATE('15/07/2021', 'DD/MM/YYYY'),TO\_DATE('15/08/2021', 'DD/MM/YYYY'));

insert into abonament

values (sala\_fitness\_seq.nextval,TO\_DATE('15/07/2020', 'DD/MM/YYYY'),TO\_DATE('15/08/2020', 'DD/MM/YYYY'));

insert into abonament

values (sala\_fitness\_seq.nextval,TO\_DATE('12/07/2020', 'DD/MM/YYYY'),TO\_DATE('12/08/2020', 'DD/MM/YYYY'));

insert into abonament

values (sala\_fitness\_seq.nextval,TO\_DATE('12/07/2019', 'DD/MM/YYYY'),TO\_DATE('12/08/2019', 'DD/MM/YYYY'));

insert into abonament

values (sala\_fitness\_seq.nextval,null,null);

insert into abonament

values (sala\_fitness\_seq.nextval,null,null);

select \* from abonament;

--abonament\_standard

insert into abonament\_standard

values (89,TO\_DATE('17/11/2021', 'DD/MM/YYYY'),TO\_DATE('17/12/2021', 'DD/MM/YYYY'),100);

insert into abonament\_standard

values (90,TO\_DATE('17/11/2022', 'DD/MM/YYYY'),TO\_DATE('17/12/2022', 'DD/MM/YYYY'),100);

insert into abonament\_standard

values (91,TO\_DATE('17/09/2021', 'DD/MM/YYYY'),TO\_DATE('17/10/2021', 'DD/MM/YYYY'),100);

insert into abonament\_standard

values (92,TO\_DATE('17/07/2021', 'DD/MM/YYYY'),TO\_DATE('17/08/2021', 'DD/MM/YYYY'),100);

insert into abonament\_standard

values (93,TO\_DATE('17/07/2021', 'DD/MM/YYYY'),TO\_DATE('17/08/2021', 'DD/MM/YYYY'),100);

--abonament\_premium

insert into abonament\_premium

values (99,TO\_DATE('12/07/2019', 'DD/MM/YYYY'),TO\_DATE('12/08/2019', 'DD/MM/YYYY'),200);

insert into abonament\_premium

values (95,TO\_DATE('10/07/2021', 'DD/MM/YYYY'),TO\_DATE('10/08/2021', 'DD/MM/YYYY'),200);

insert into abonament\_premium

values (96,TO\_DATE('15/07/2021', 'DD/MM/YYYY'),TO\_DATE('15/08/2021', 'DD/MM/YYYY'),200);

insert into abonament\_premium

values (97,TO\_DATE('15/07/2020', 'DD/MM/YYYY'),TO\_DATE('15/08/2020', 'DD/MM/YYYY'),200);

insert into abonament\_premium

values (98,TO\_DATE('12/07/2020', 'DD/MM/YYYY'),TO\_DATE('12/08/2020', 'DD/MM/YYYY'),200);

select \* from abonament\_premium;

--79-83

--95-99

--89 -93

--este\_detinut

insert into este\_detinut

values (89,79);

insert into este\_detinut

values (90,80);

insert into este\_detinut

values (91,81);

insert into este\_detinut

values (92,82);

insert into este\_detinut

values (93,83);

insert into este\_detinut

values (95,79);

insert into este\_detinut

values (96,80);

insert into este\_detinut

values (97,81);

insert into este\_detinut

values (98,82);

insert into este\_detinut

values (99,83);

select \* from lista\_clienti;

select \* from locatie;

select \* from gama\_aparate;

select \* from gama\_suplimente;

select \* from lista\_departamente;

--sala\_de\_fitness

desc sala\_de\_fitness;

insert into sala\_de\_fitness

values (sala\_fitness\_seq.nextval,'Spartan Gym',74,'spartan\_gym@gmail.com',45,16,25,59);

insert into sala\_de\_fitness

values (sala\_fitness\_seq.nextval,'Fit Gym',75,'fit\_gym@gmail.com',46,10,26,60);

insert into sala\_de\_fitness

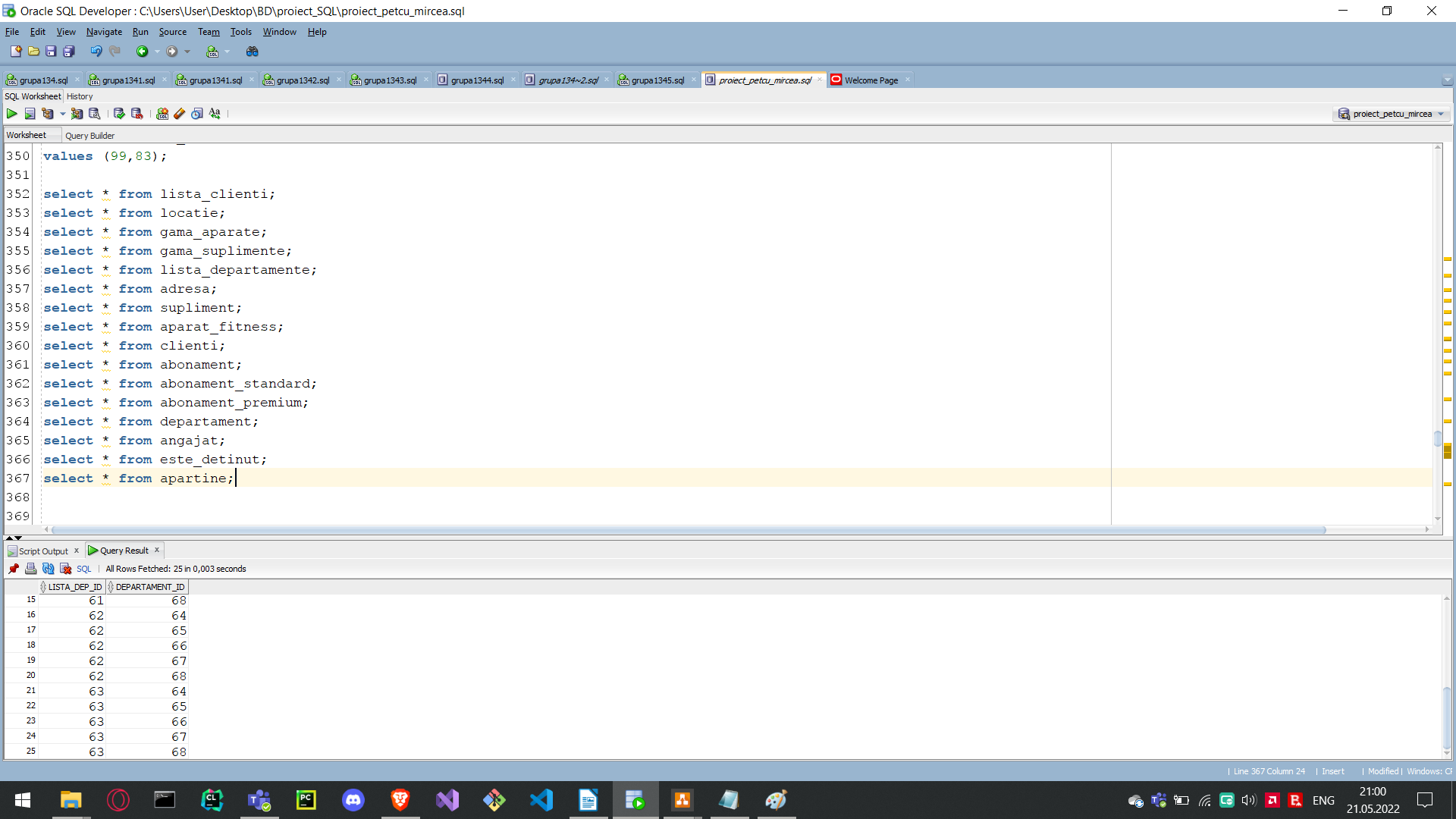
values (sala\_fitness\_seq.nextval,'Total Gym',76,'total\_gym@gmail.com',47,12,27,61);

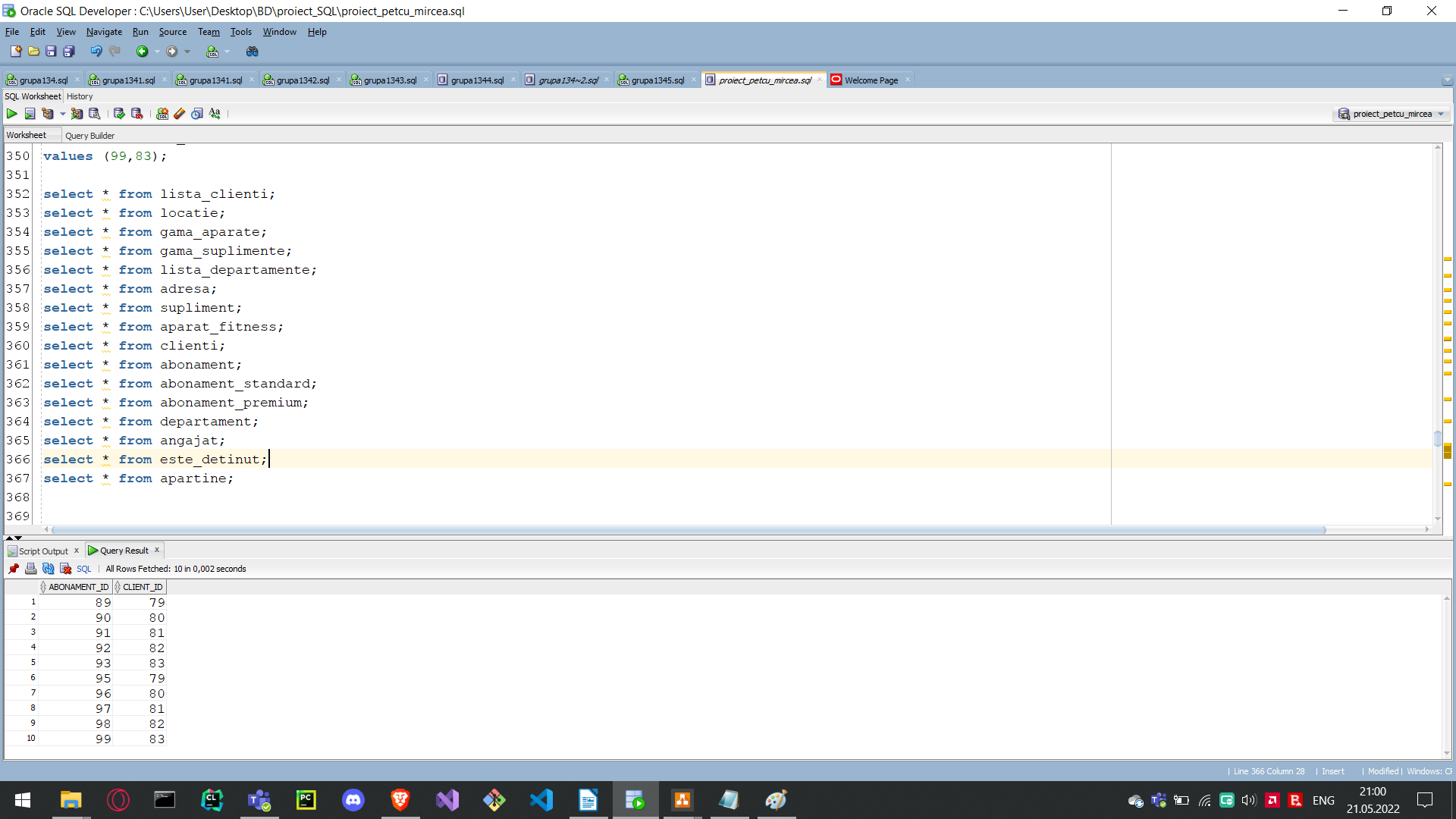
insert into sala\_de\_fitness

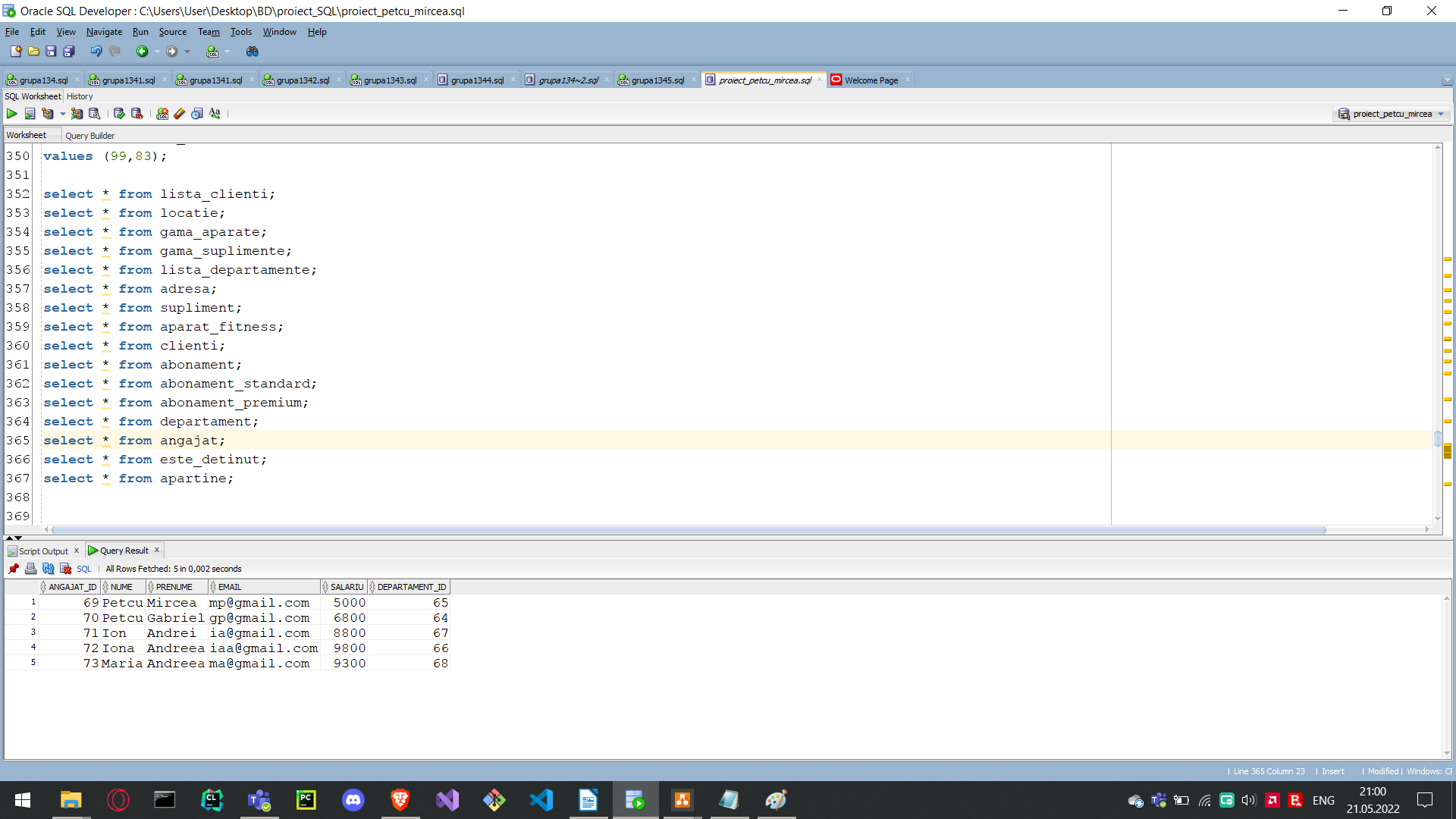
values (sala\_fitness\_seq.nextval,'Power Gym',77,'power\_gym@gmail.com',48,13,28,62);

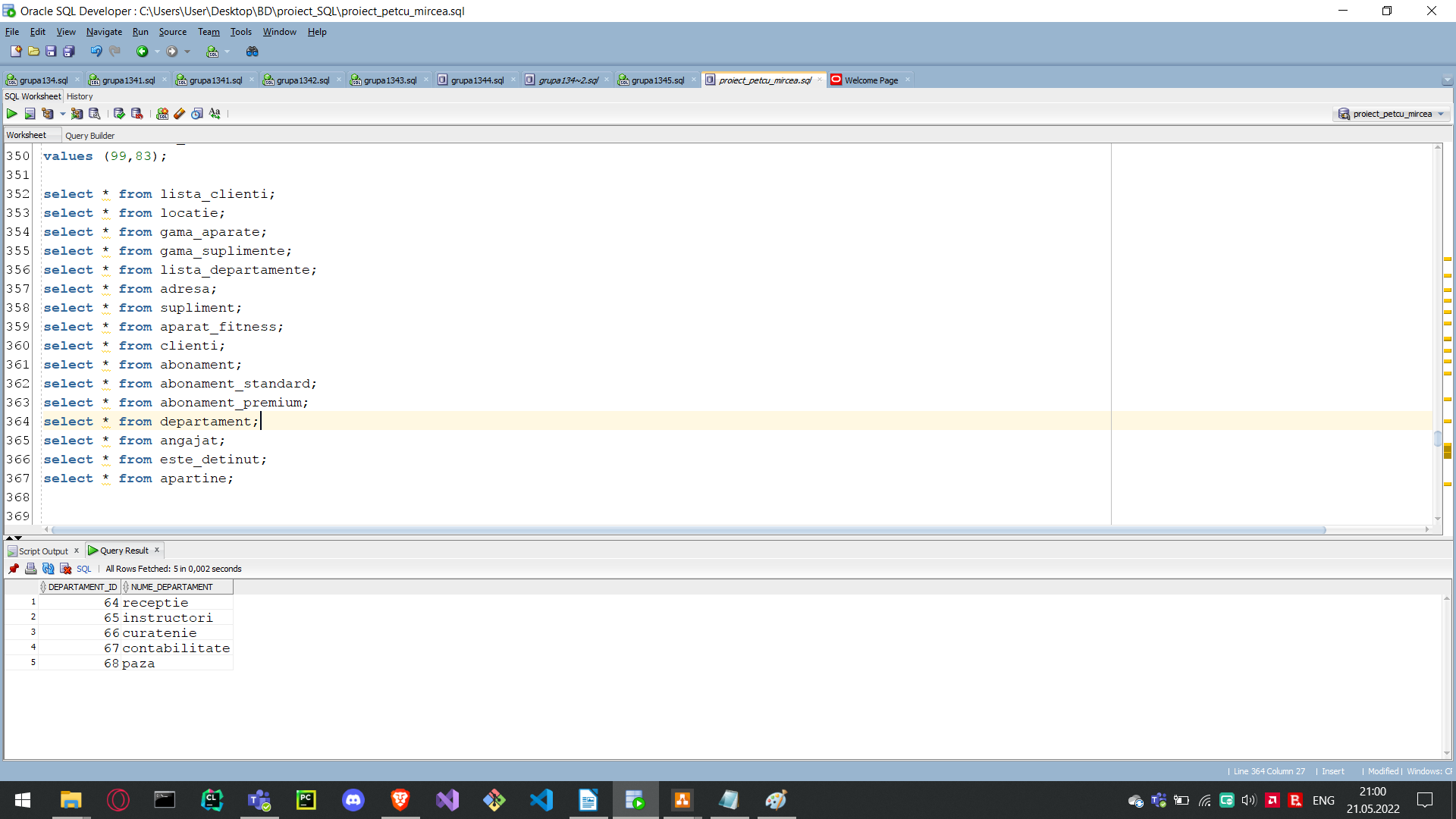
insert into sala\_de\_fitness

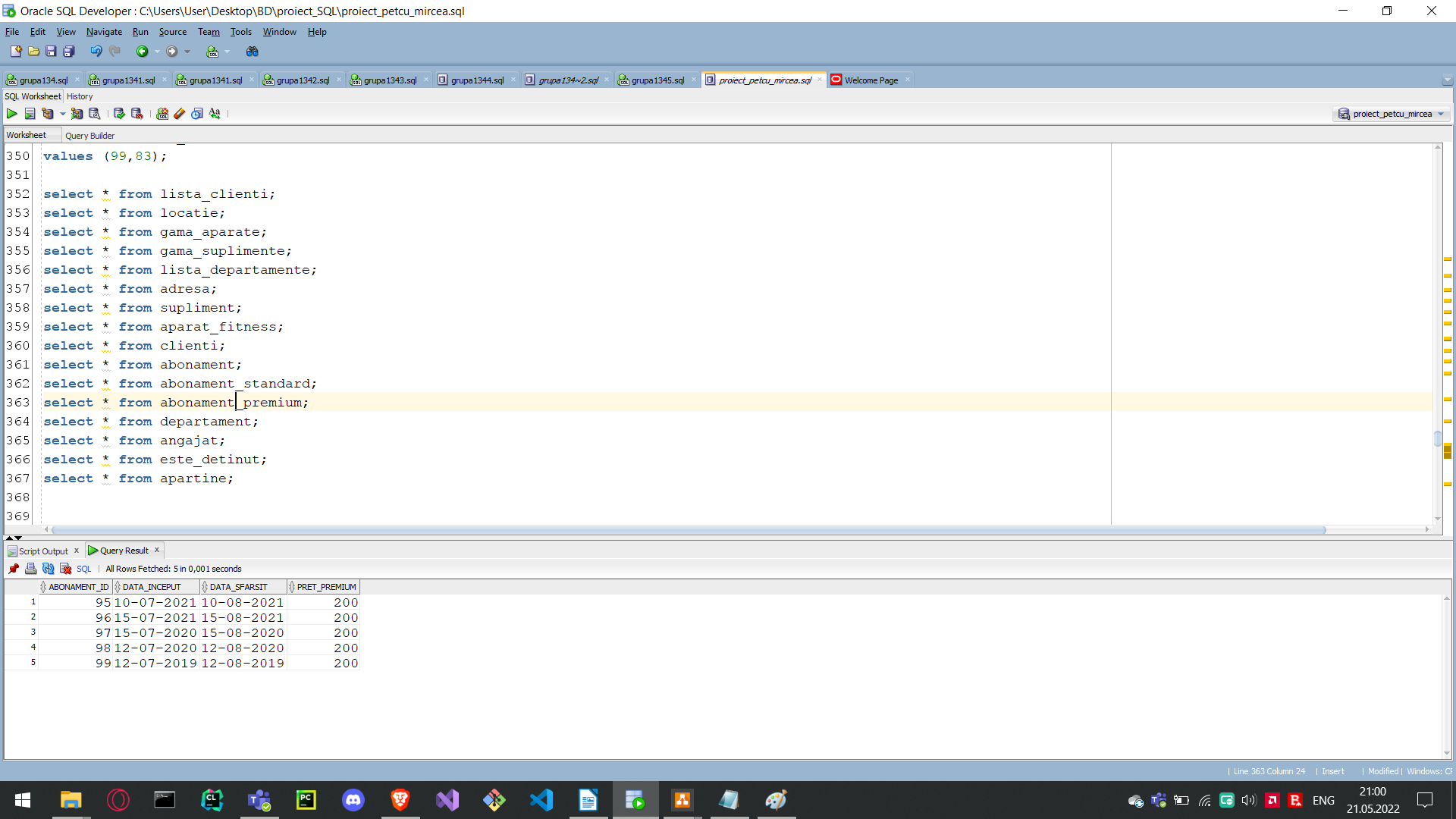
values (sala\_fitness\_seq.nextval,'Strong Gym',78,['strong\_gym@gmail.com](mailto:'strong_gym@gmail.com)',49,14,30,63);

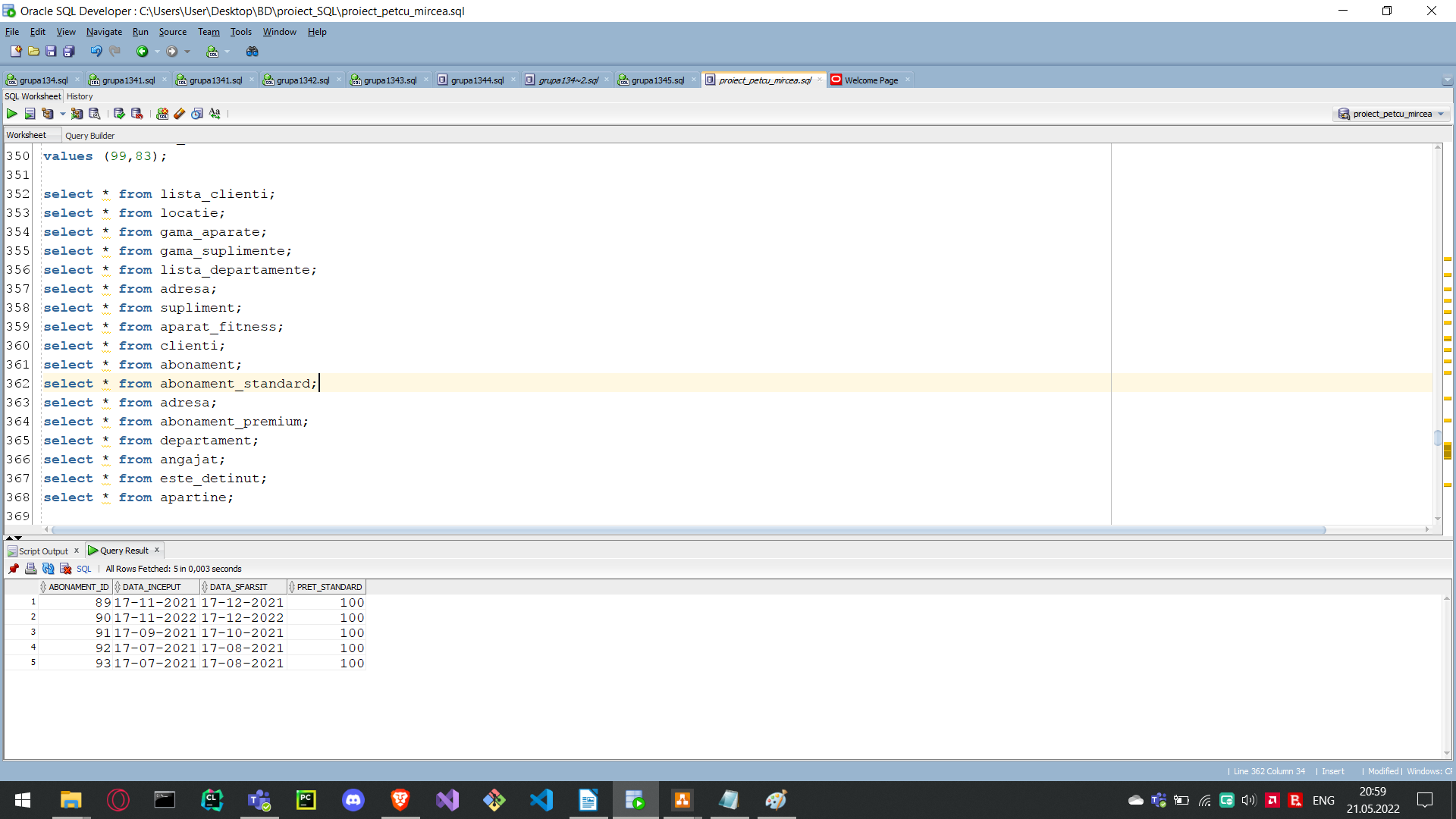


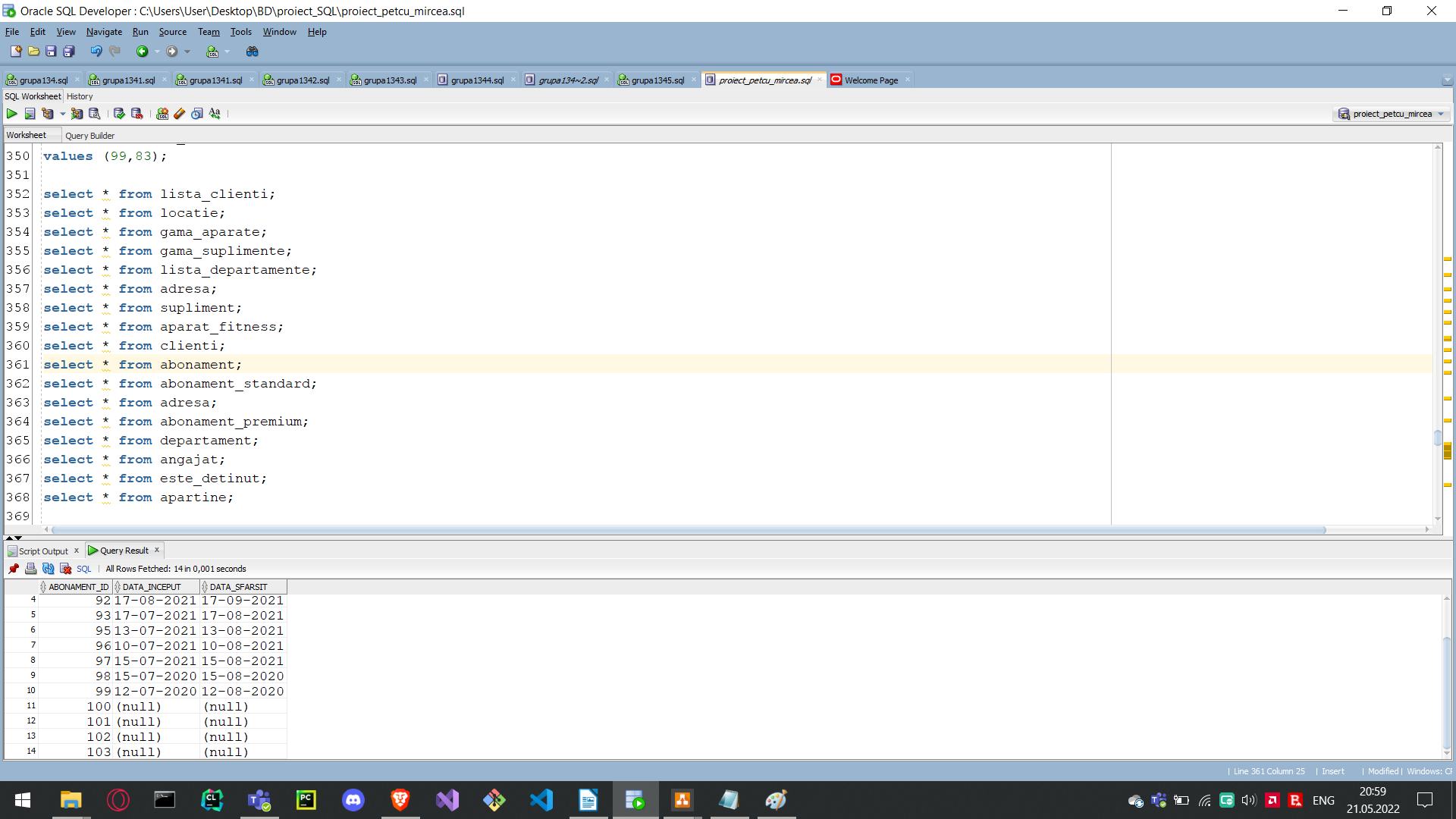


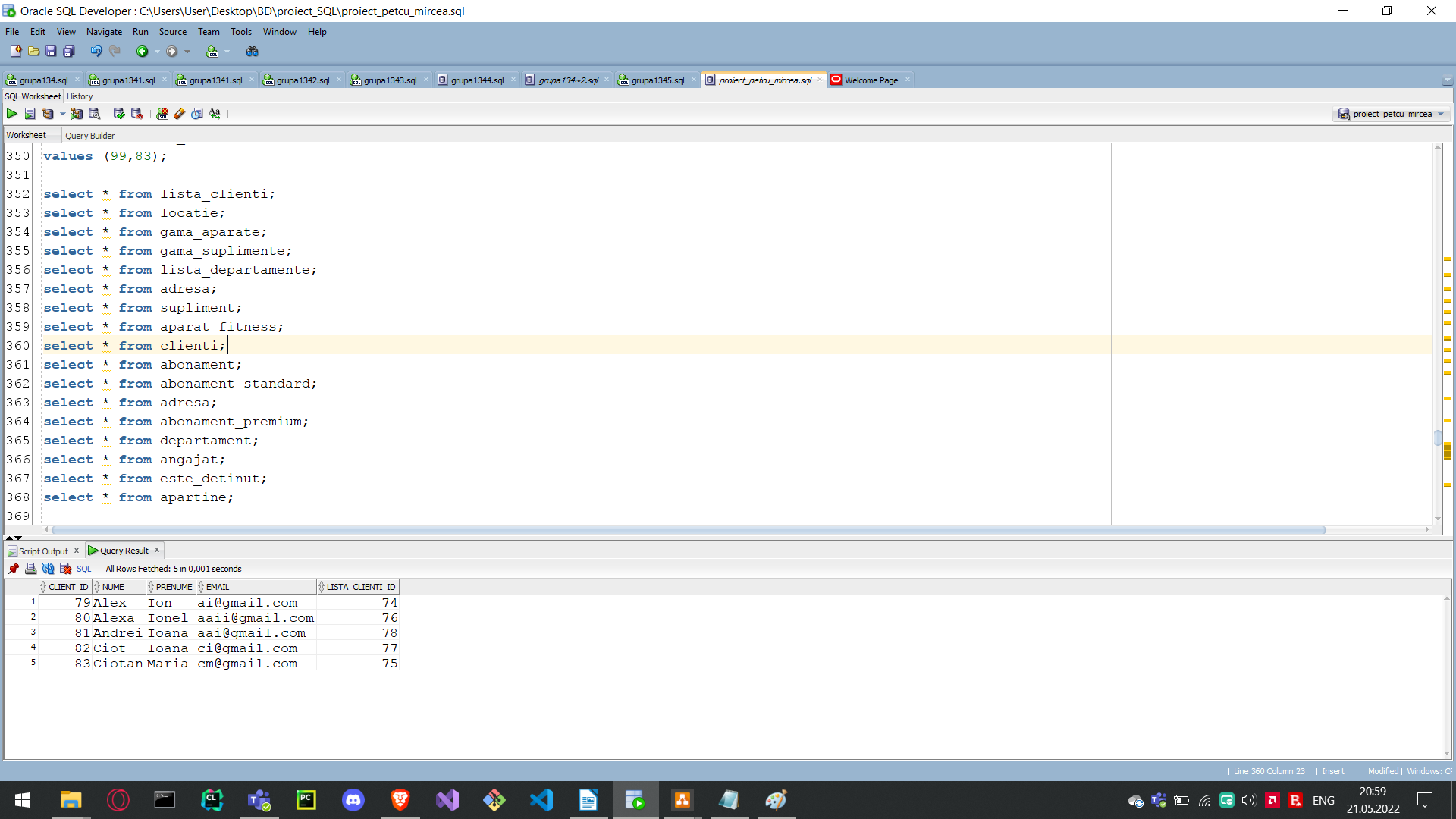


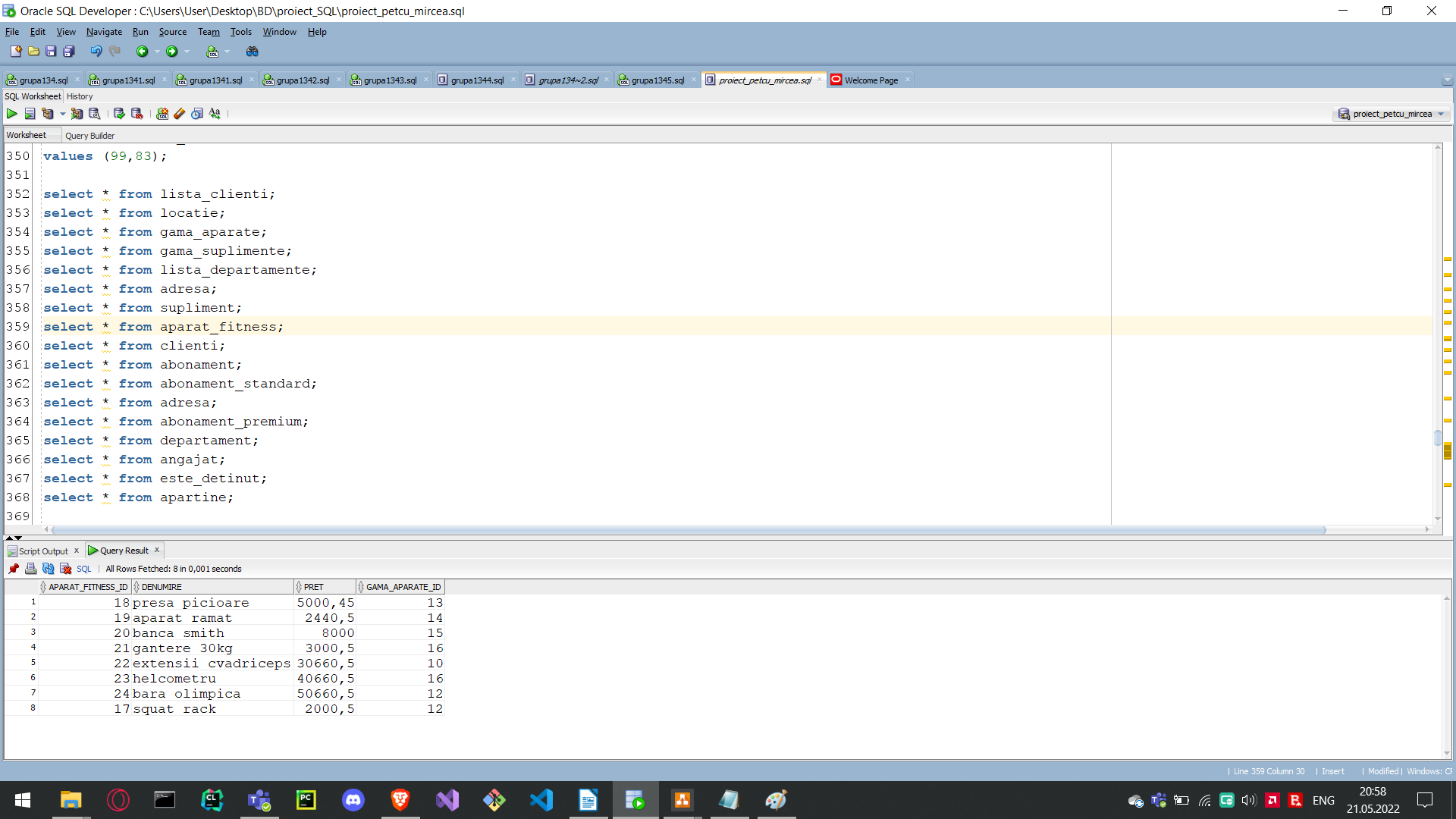


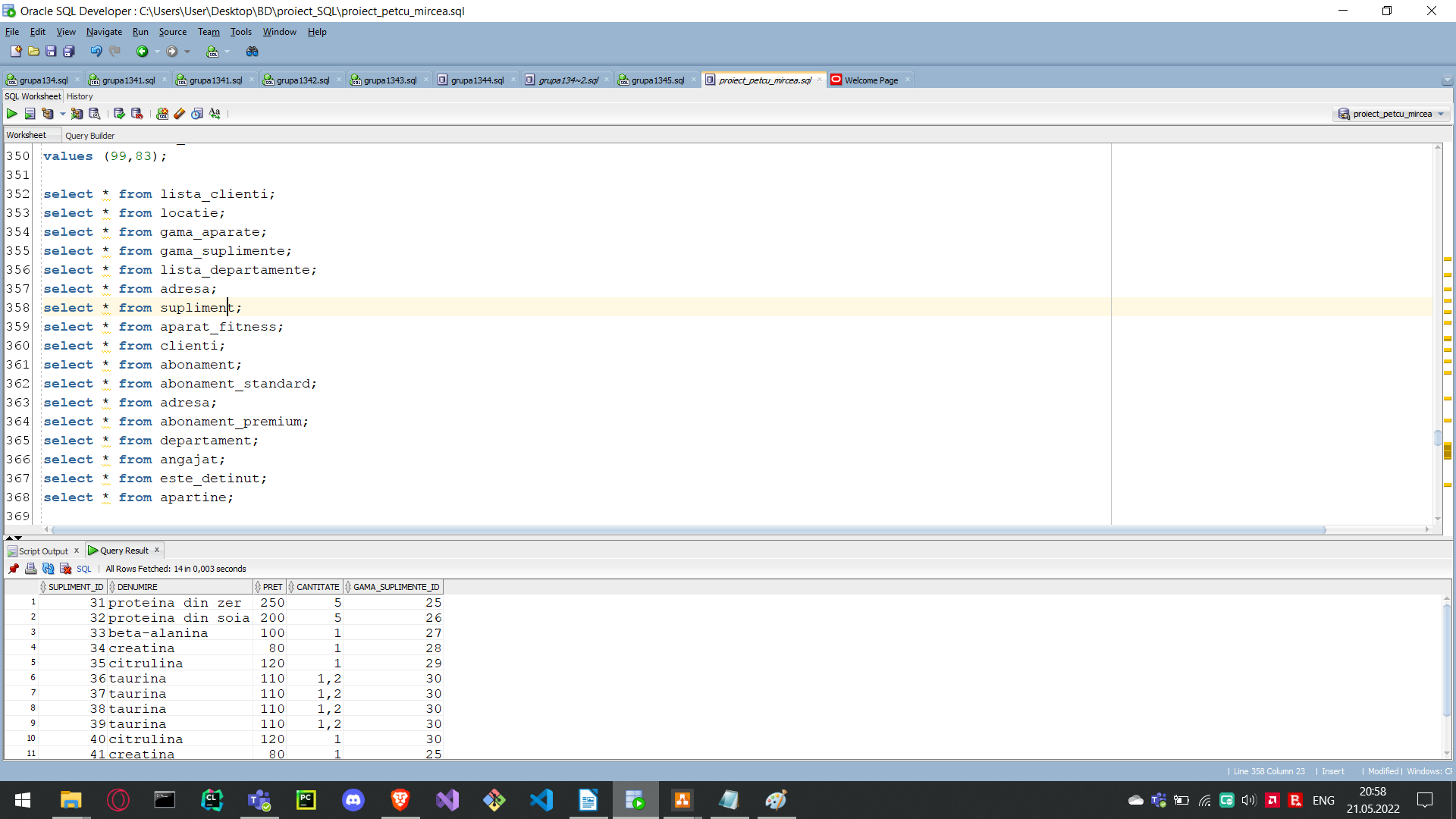


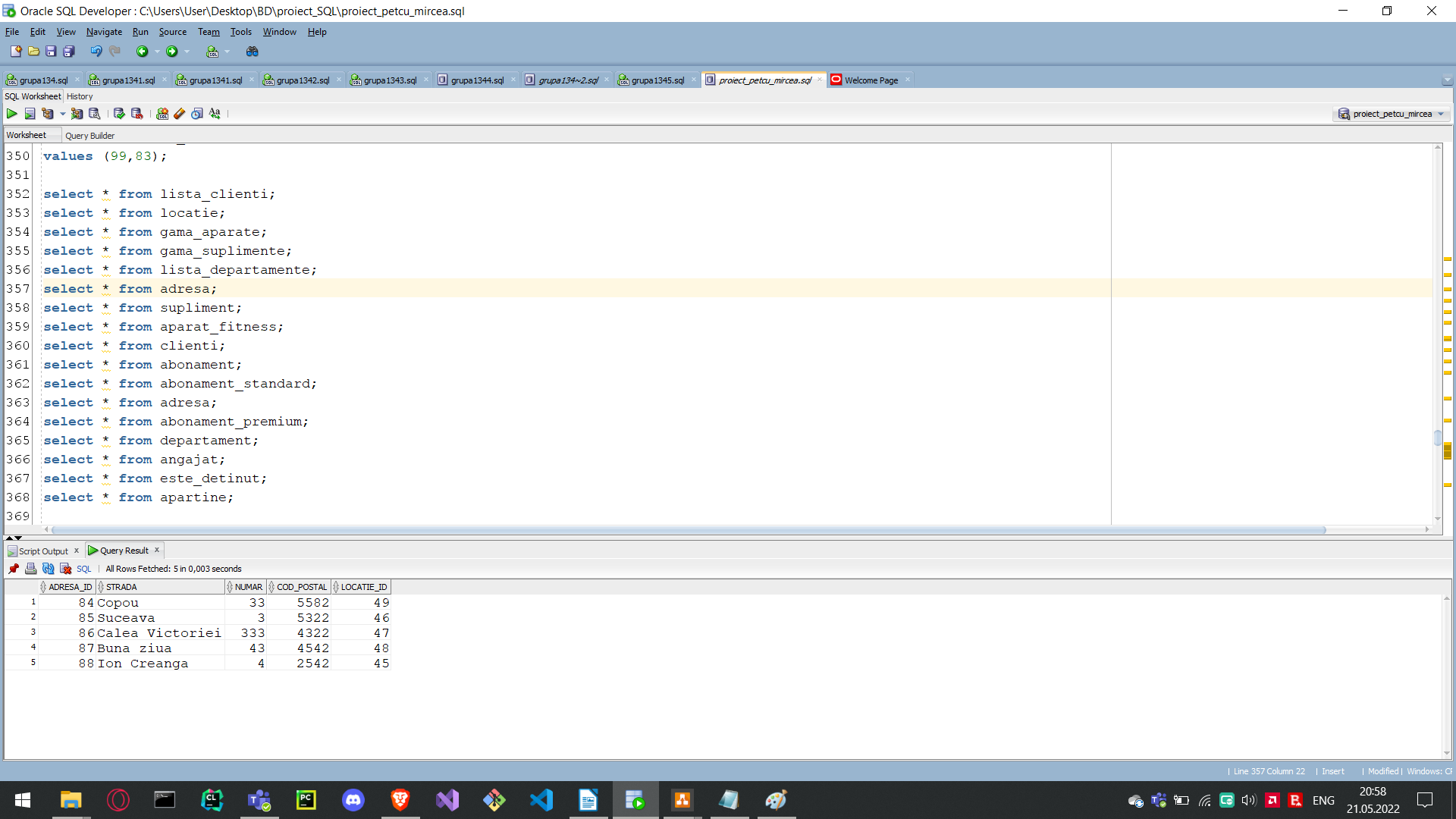


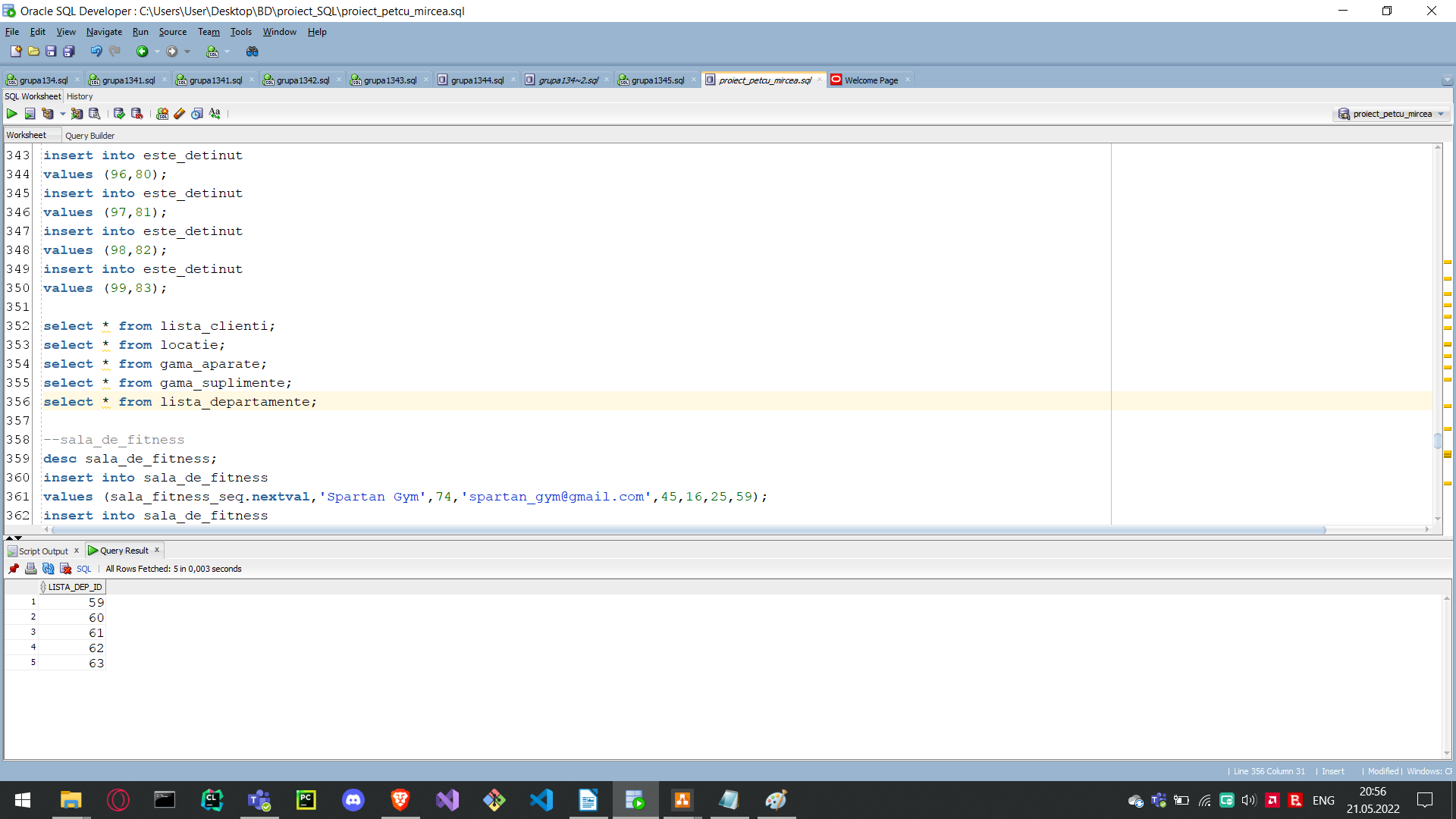


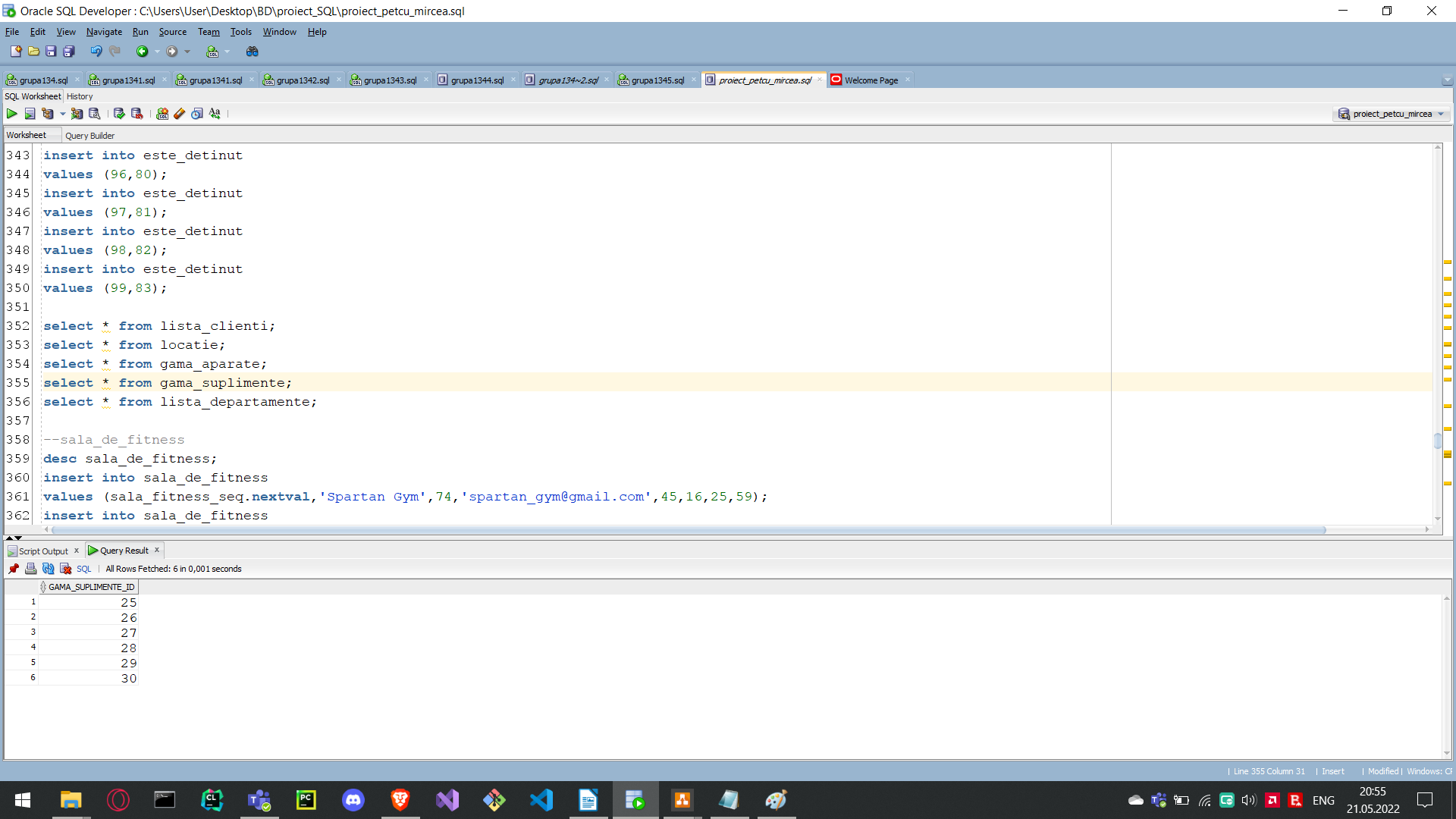


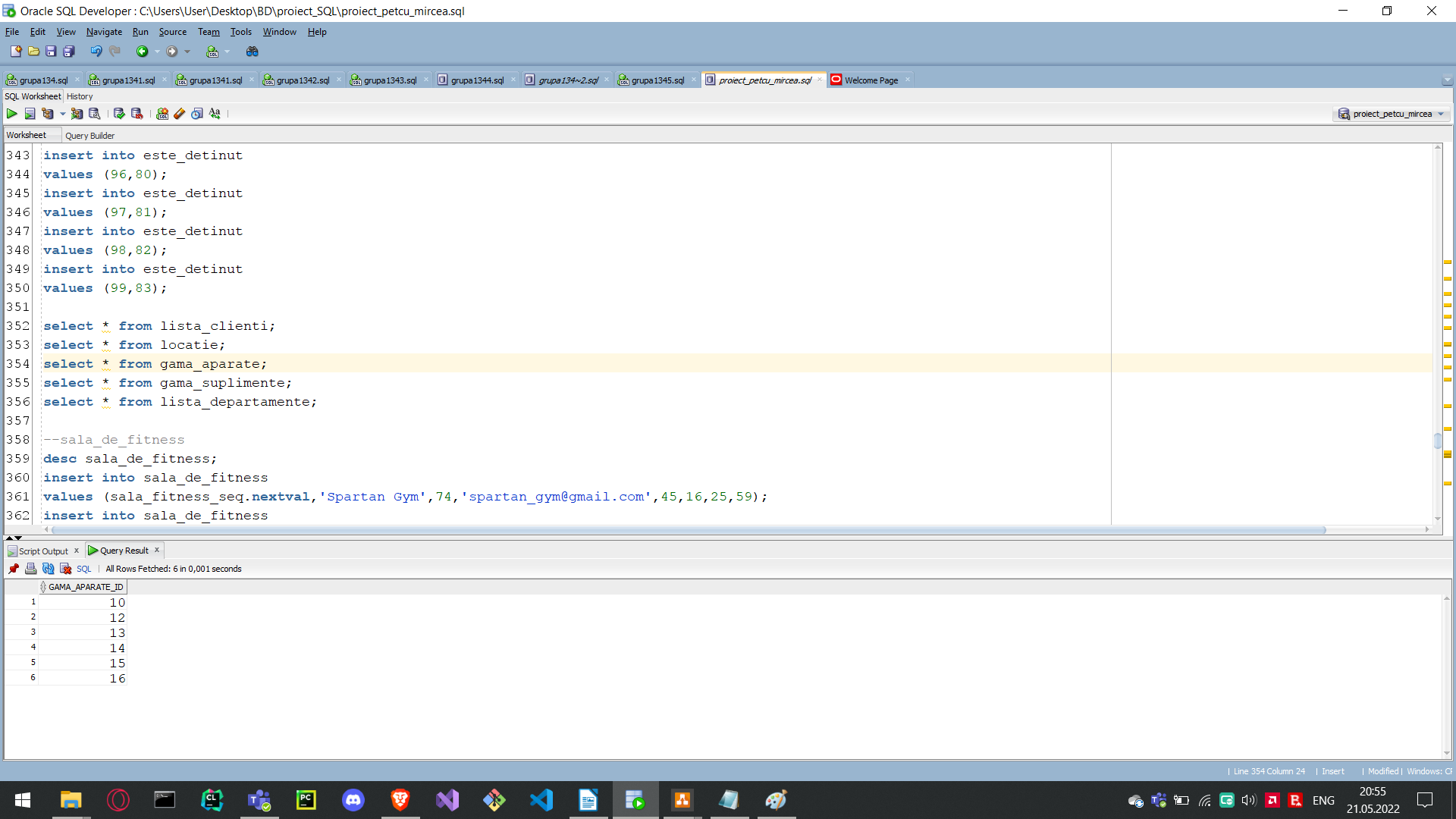


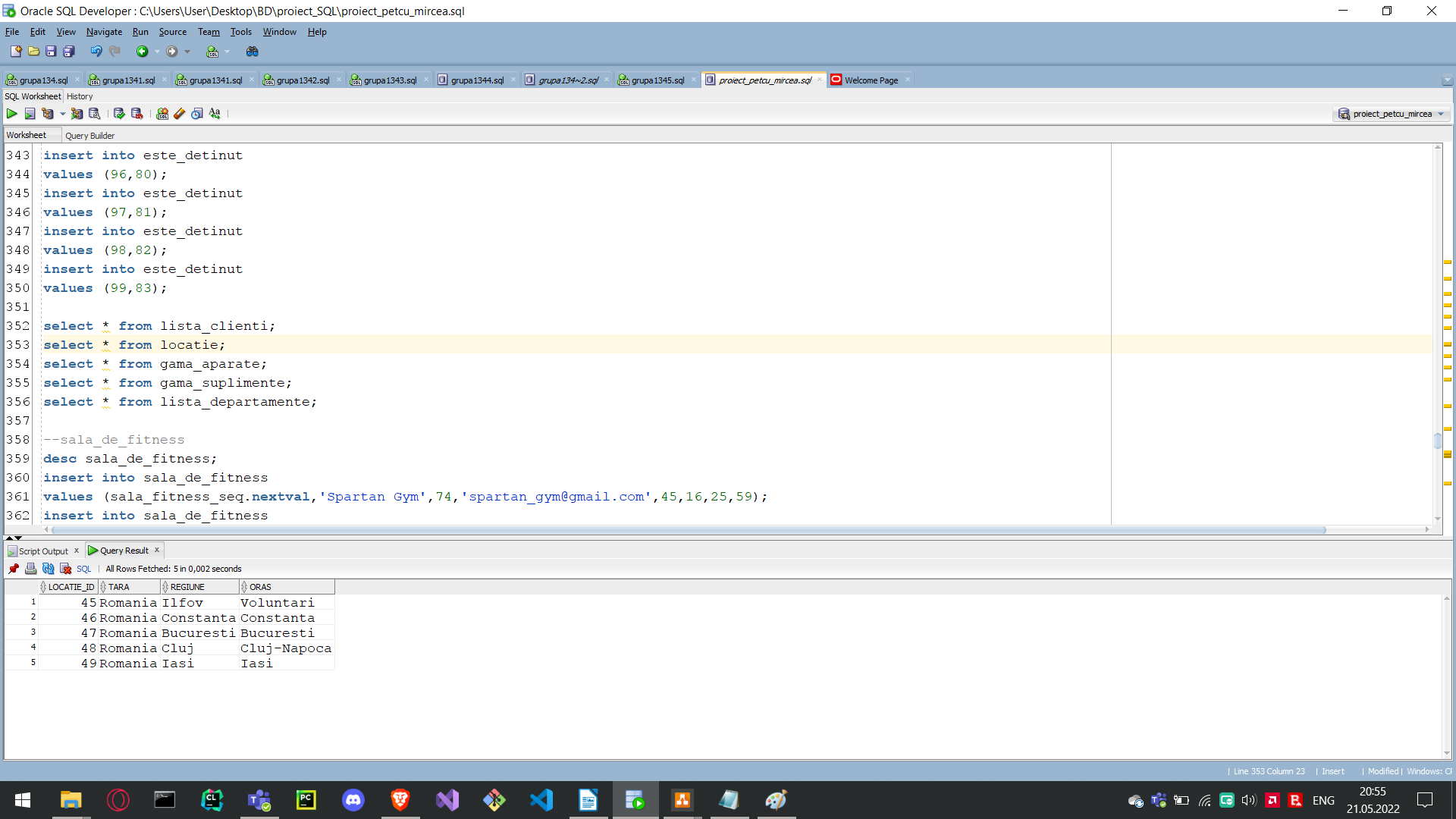


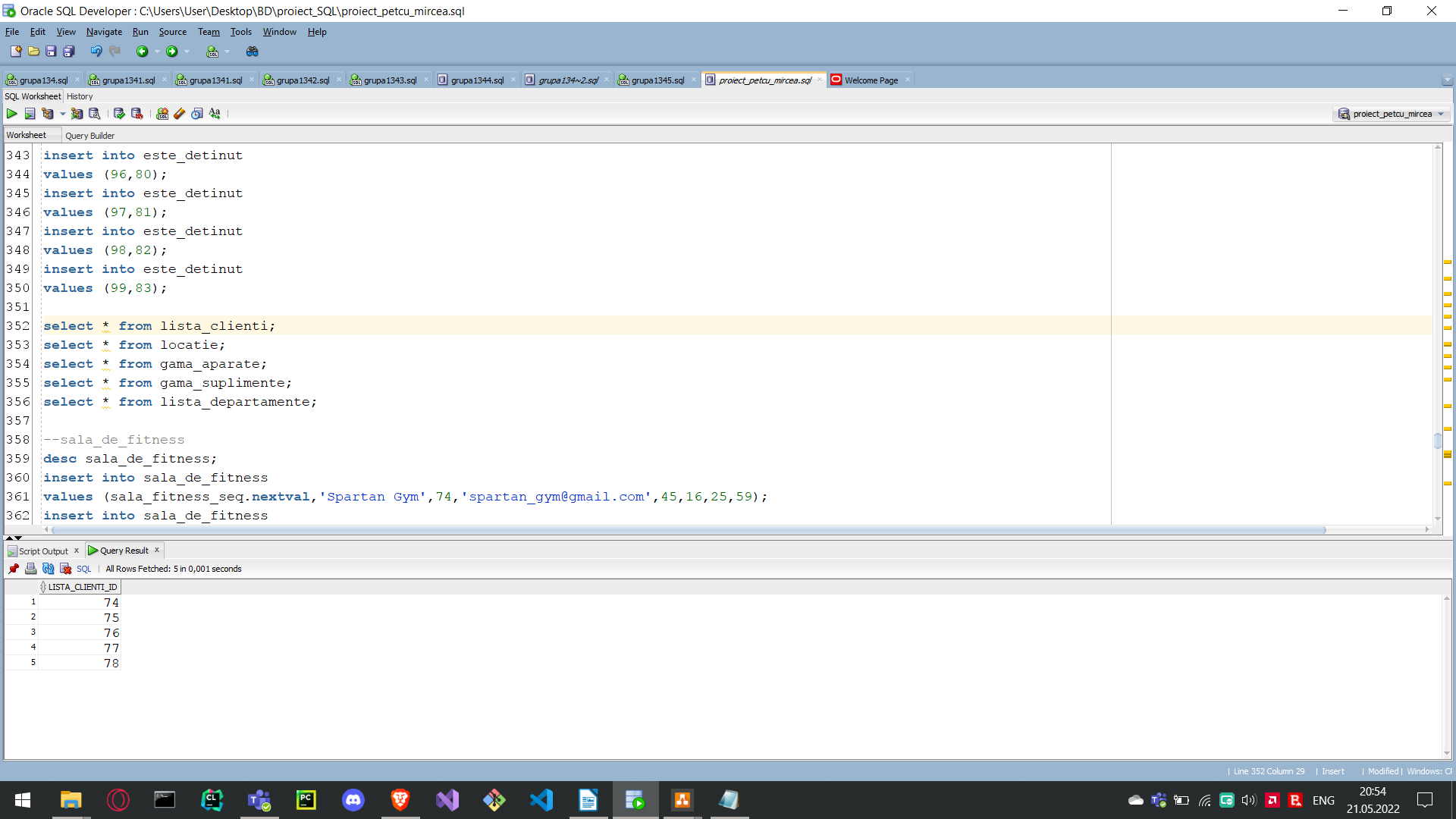










**12. Formulați în limbaj natural și implementați 5 cereri SQL complexe ce vor utiliza, în ansamblul lor, următoarele elemente: • operație join pe cel puțin 4 tabele • filtrare la nivel de linii • subcereri sincronizate în care intervin cel puțin 3 tabele • subcereri nesincronizate în care intervin cel 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) .**

--1) Afisati toate salilie de fitness ce au clienti cu abonamente premium

--si numarul strazii din adresa sali este egal numarul sali cu numarul strazii din adresa salii unde exista cel putin un angajat prenumele "Mircea

select sf.\*

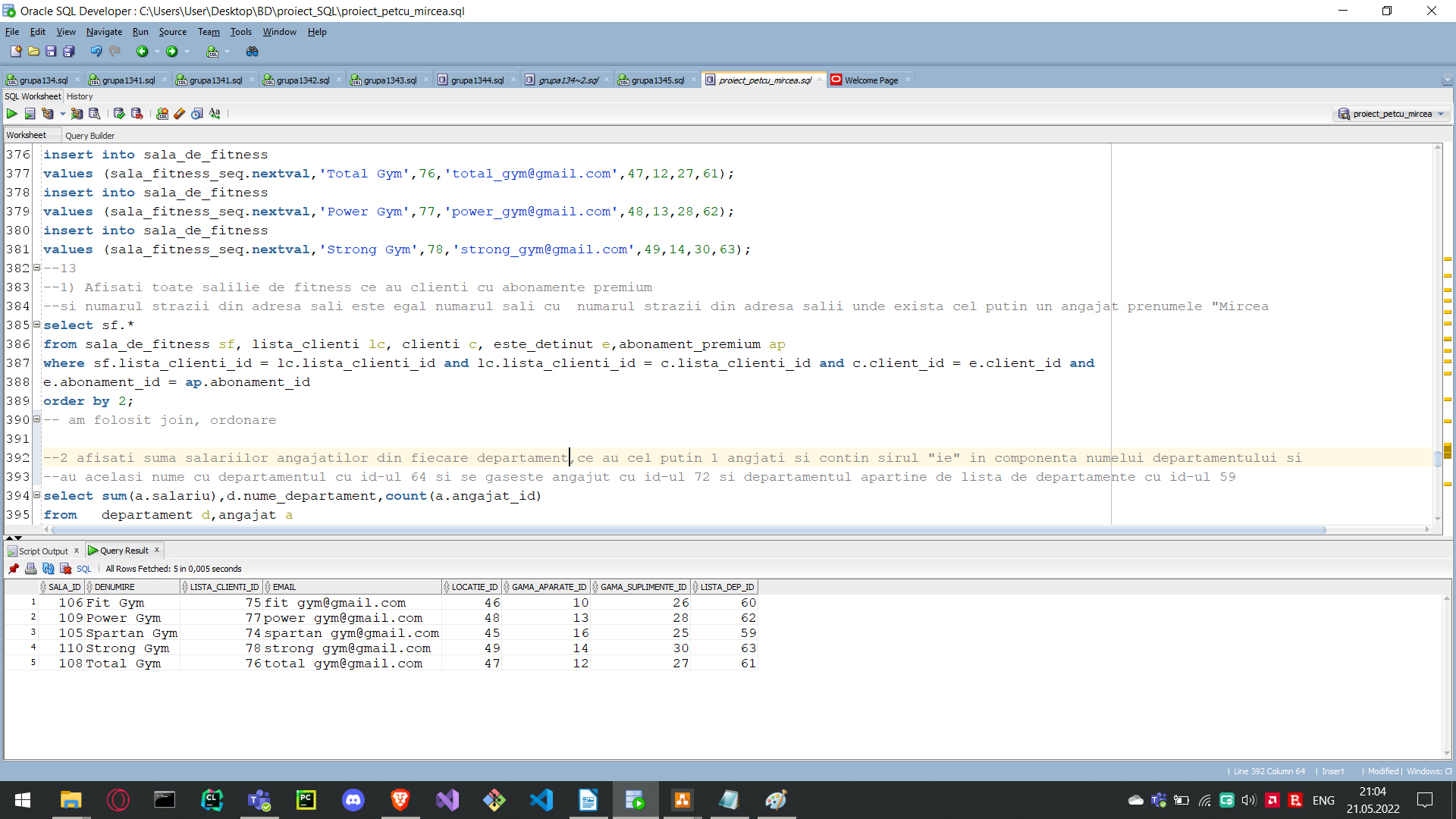
from sala\_de\_fitness sf, lista\_clienti lc, clienti c, este\_detinut e,abonament\_premium ap

where sf.lista\_clienti\_id = lc.lista\_clienti\_id and lc.lista\_clienti\_id = c.lista\_clienti\_id and c.client\_id = e.client\_id and

e.abonament\_id = ap.abonament\_id

order by 2;

-- am folosit join, ordonare

--2 afisati suma salariilor angajatilor din fiecare departament,ce au cel putin 1 angjati si contin sirul "ie" in componenta numelui departamentului si

--au acelasi nume cu departamentul cu id-ul 64 si se gaseste angajut cu id-ul 72 si departamentul apartine de lista de departamente cu id-ul 59

select sum(a.salariu),d.nume\_departament,count(a.angajat\_id)

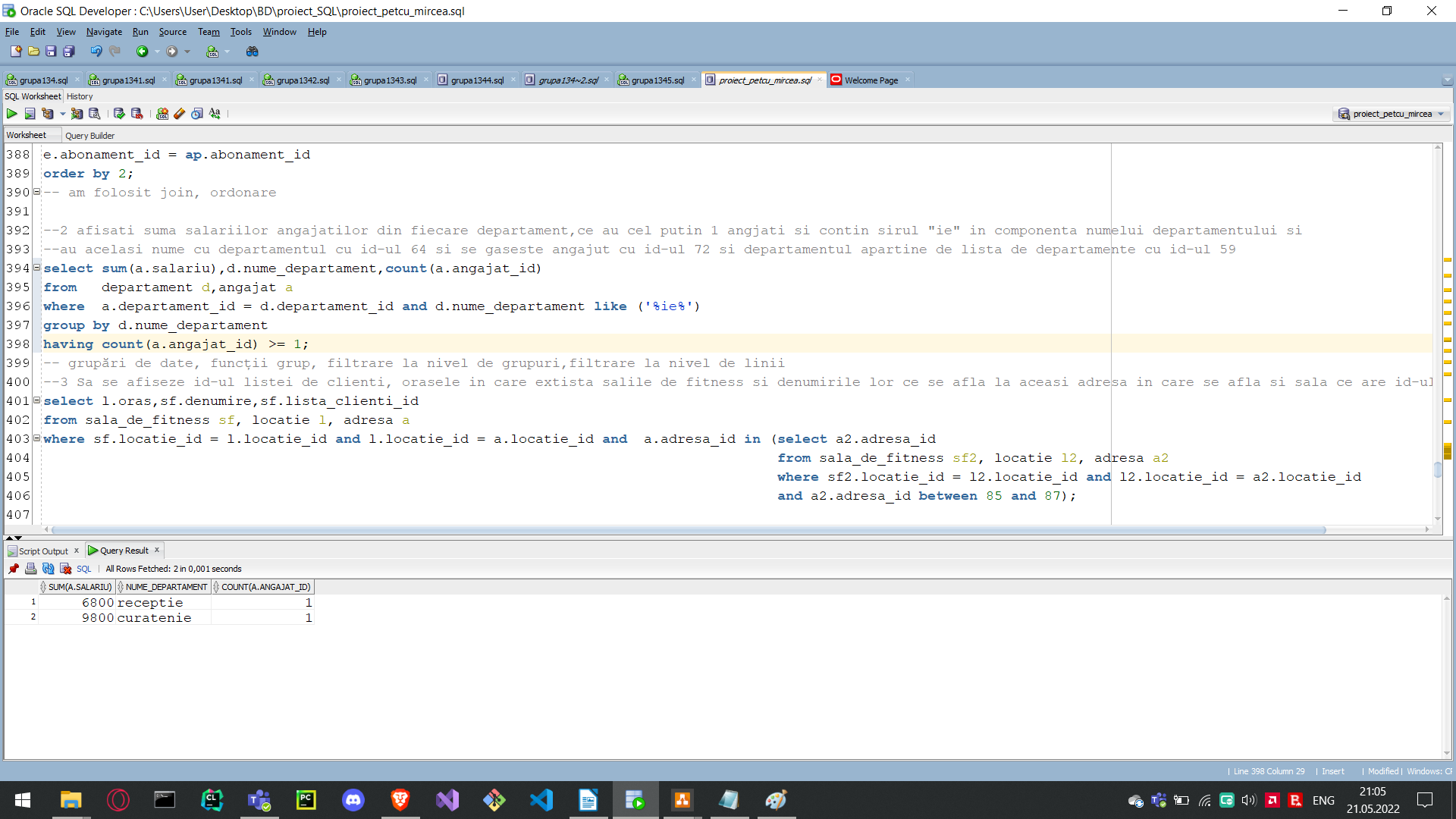
from departament d,angajat a

where a.departament\_id = d.departament\_id and d.nume\_departament like ('%ie%')

group by d.nume\_departament

having count(a.angajat\_id) >= 1;

-- grupări de date, funcții grup, filtrare la nivel de grupuri,filtrare la nivel de linii

--3 Sa se afiseze id-ul listei de clienti, orasele in care extista salile de fitness si denumirile lor ce se afla la aceasi adresa in care se afla si sala ce are id-ul adresei 84

select l.oras,sf.denumire,sf.lista\_clienti\_id

from sala\_de\_fitness sf, locatie l, adresa a

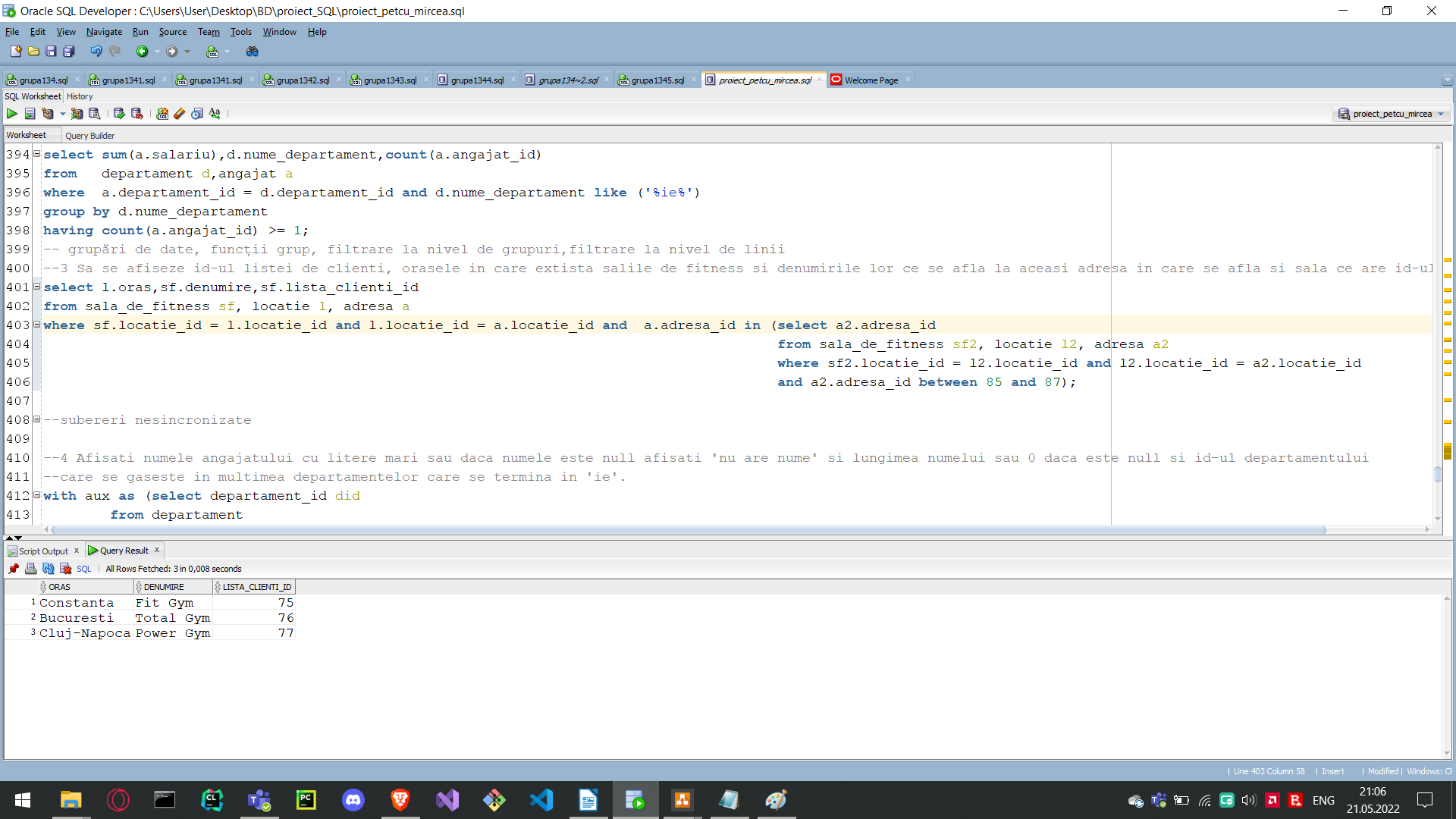
where sf.locatie\_id = l.locatie\_id and l.locatie\_id = a.locatie\_id and a.adresa\_id in (select a2.adresa\_id

from sala\_de\_fitness sf2, locatie l2, adresa a2

where sf2.locatie\_id = l2.locatie\_id and l2.locatie\_id = a2.locatie\_id

and a2.adresa\_id between 85 and 87);

--subereri nesincronizate

--4 Afisati numele angajatului cu litere mari sau daca numele este null afisati 'nu are nume' si lungimea numelui sau 0 daca este null si id-ul departamentului

--care se gaseste in multimea departamentelor care se termina in 'ie'.

with aux as (select departament\_id did

from departament

where nume\_departament like('%ie'))

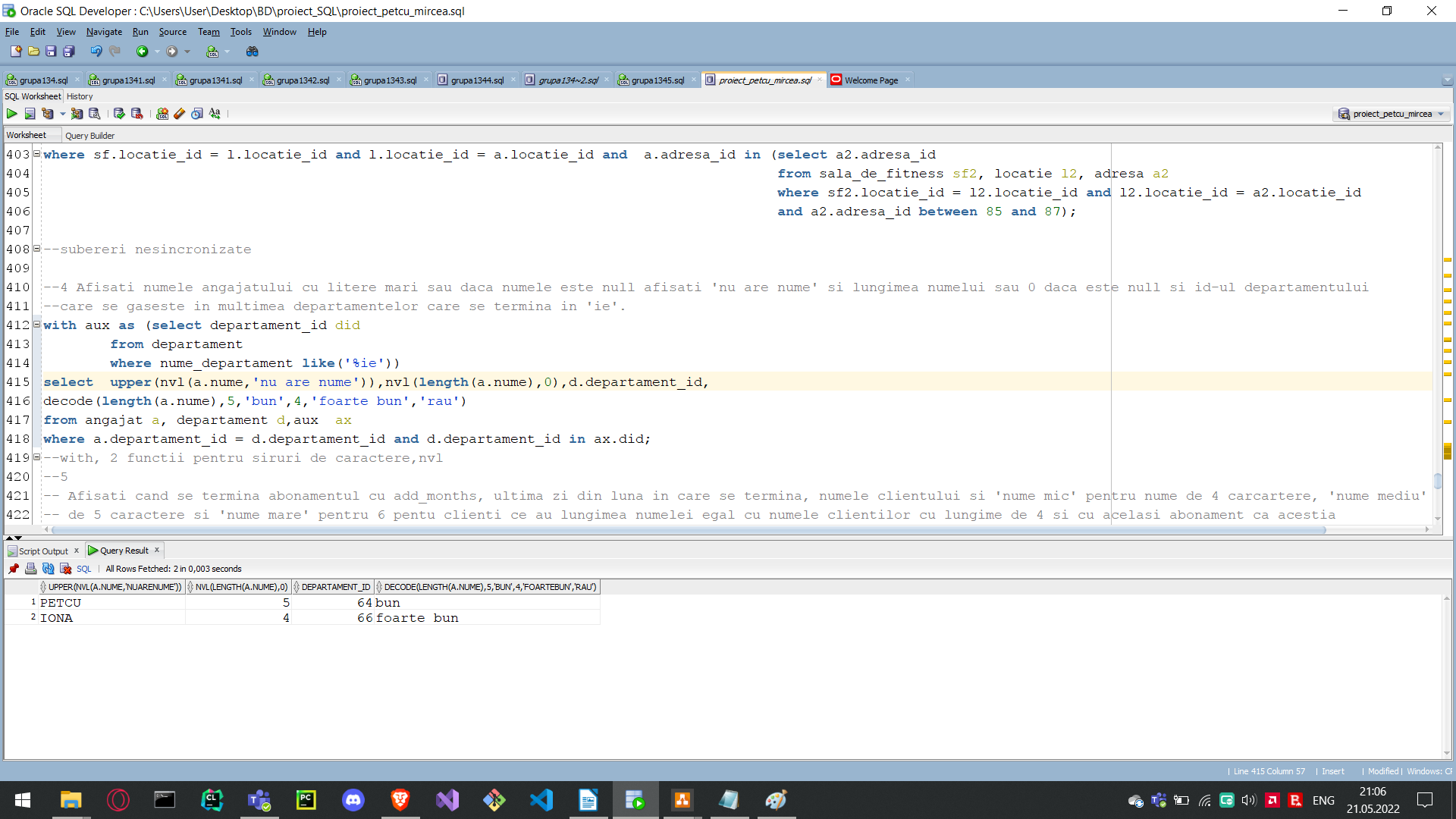
select upper(nvl(a.nume,'nu are nume')),nvl(length(a.nume),0),d.departament\_id,

decode(length(a.nume),5,'bun',4,'foarte bun','rau')

from angajat a, departament d,aux ax

where a.departament\_id = d.departament\_id and d.departament\_id in ax.did;

--with, 2 functii pentru siruri de caractere,nvl

--5

-- Afisati cand se termina abonamentul cu add\_months, ultima zi din luna in care se termina, numele clientului si 'nume mic' pentru nume de 4 carcartere, 'nume mediu' pentru nume

-- de 5 caractere si 'nume mare' pentru 6 pentu clienti ce au lungimea numelei egal cu numele clientilor cu lungime de 4 si cu acelasi abonament ca acestia

select add\_months(ab.data\_inceput,1),last\_day(ab.data\_sfarsit),c.nume,

case length(c.nume)

when 4 then 'nume mic'

when 5 then 'nume mediu'

when 6 then 'nume mare'

end

from clienti c, este\_detinut ed, abonament ab

where c.client\_id = ed.client\_id and ab.abonament\_id = ed.abonament\_id and c.client\_id in (select c2.client\_id

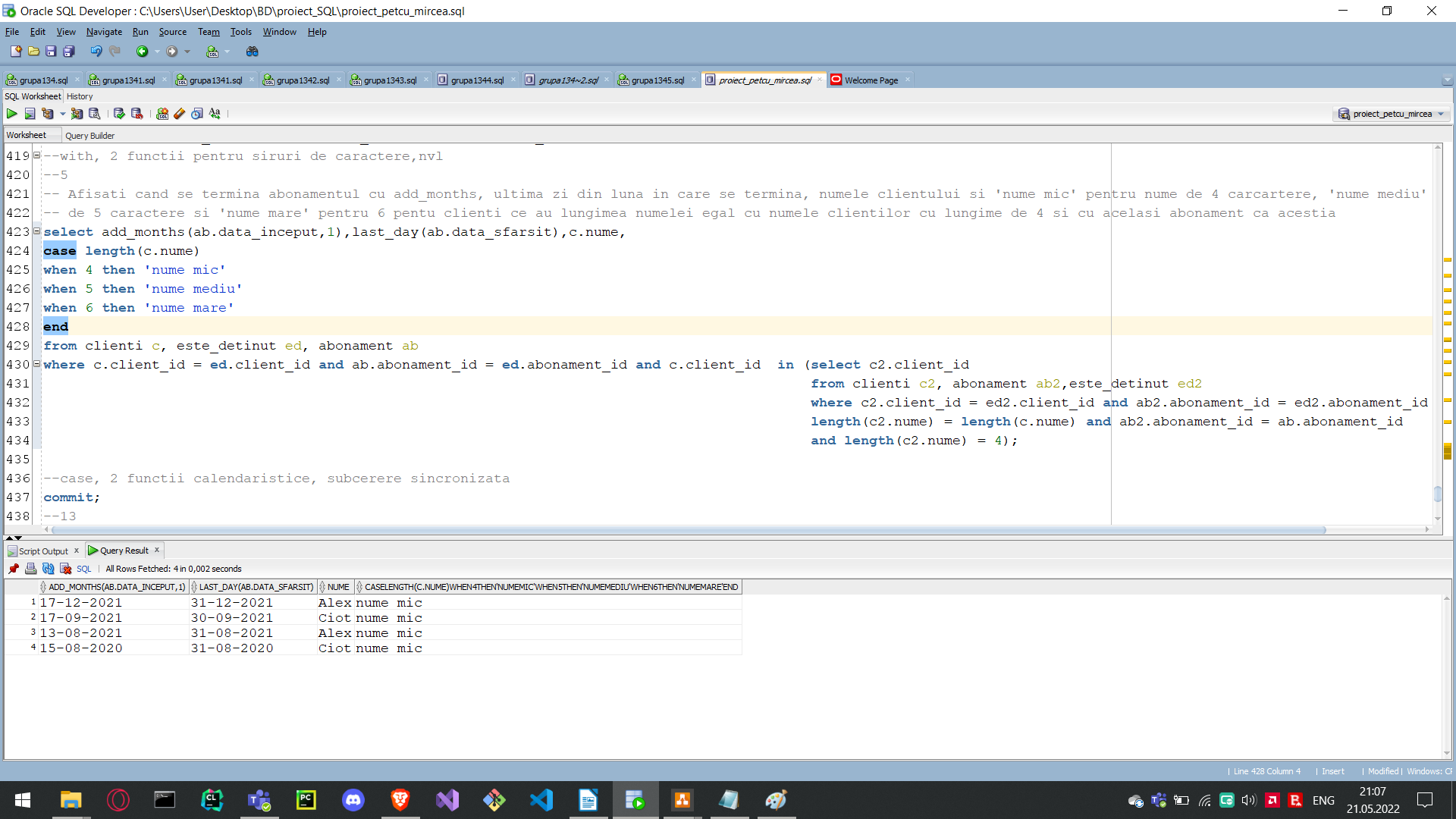
from clienti c2, abonament ab2,este\_detinut ed2

where c2.client\_id = ed2.client\_id and ab2.abonament\_id = ed2.abonament\_id and

length(c2.nume) = length(c.nume) and ab2.abonament\_id = ab.abonament\_id

and length(c2.nume) = 4);

--case, 2 functii calendaristice, subcerere sincronizata

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

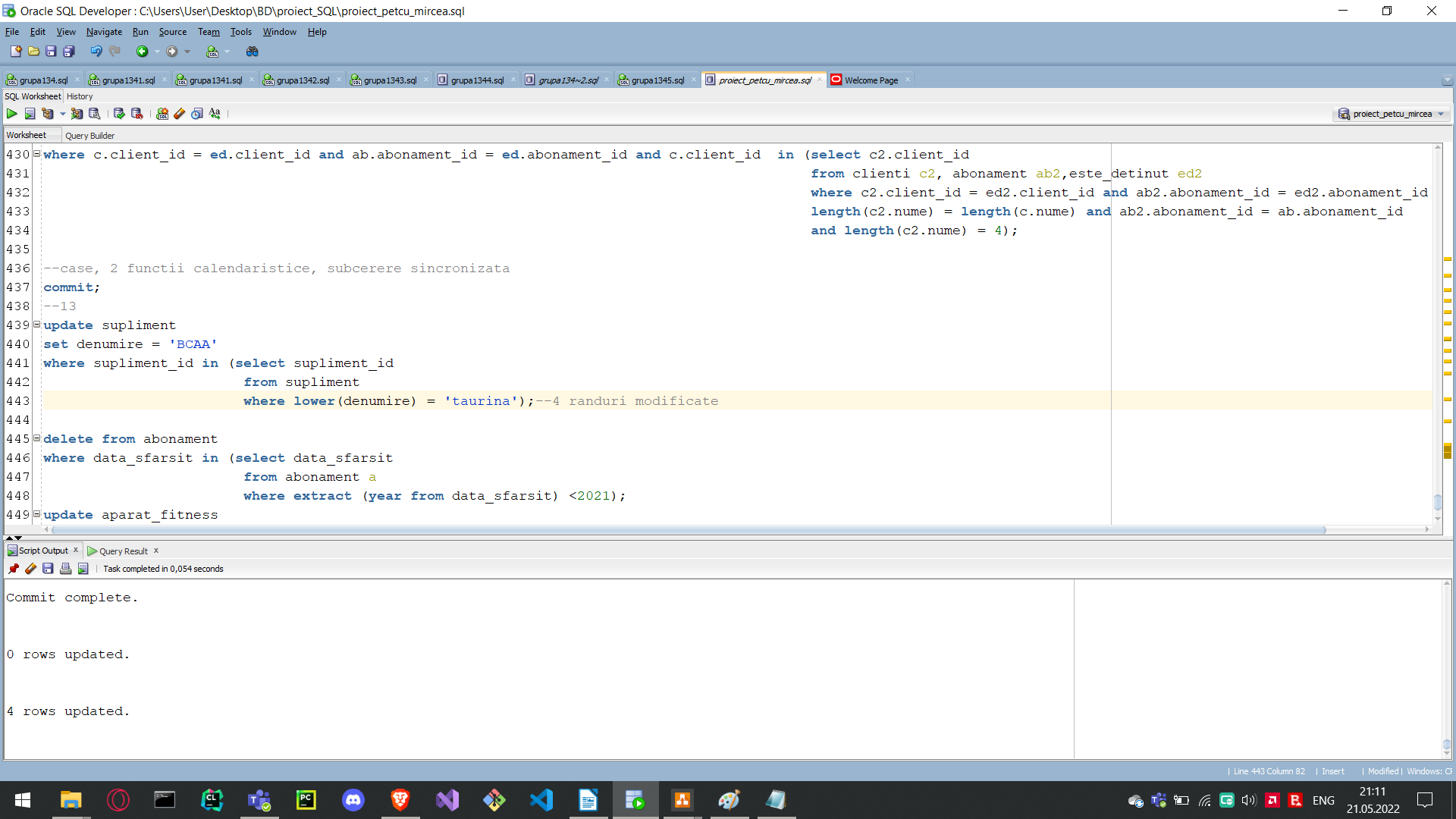
update supliment

set denumire = 'BCAA'

where supliment\_id in (select supliment\_id

from supliment

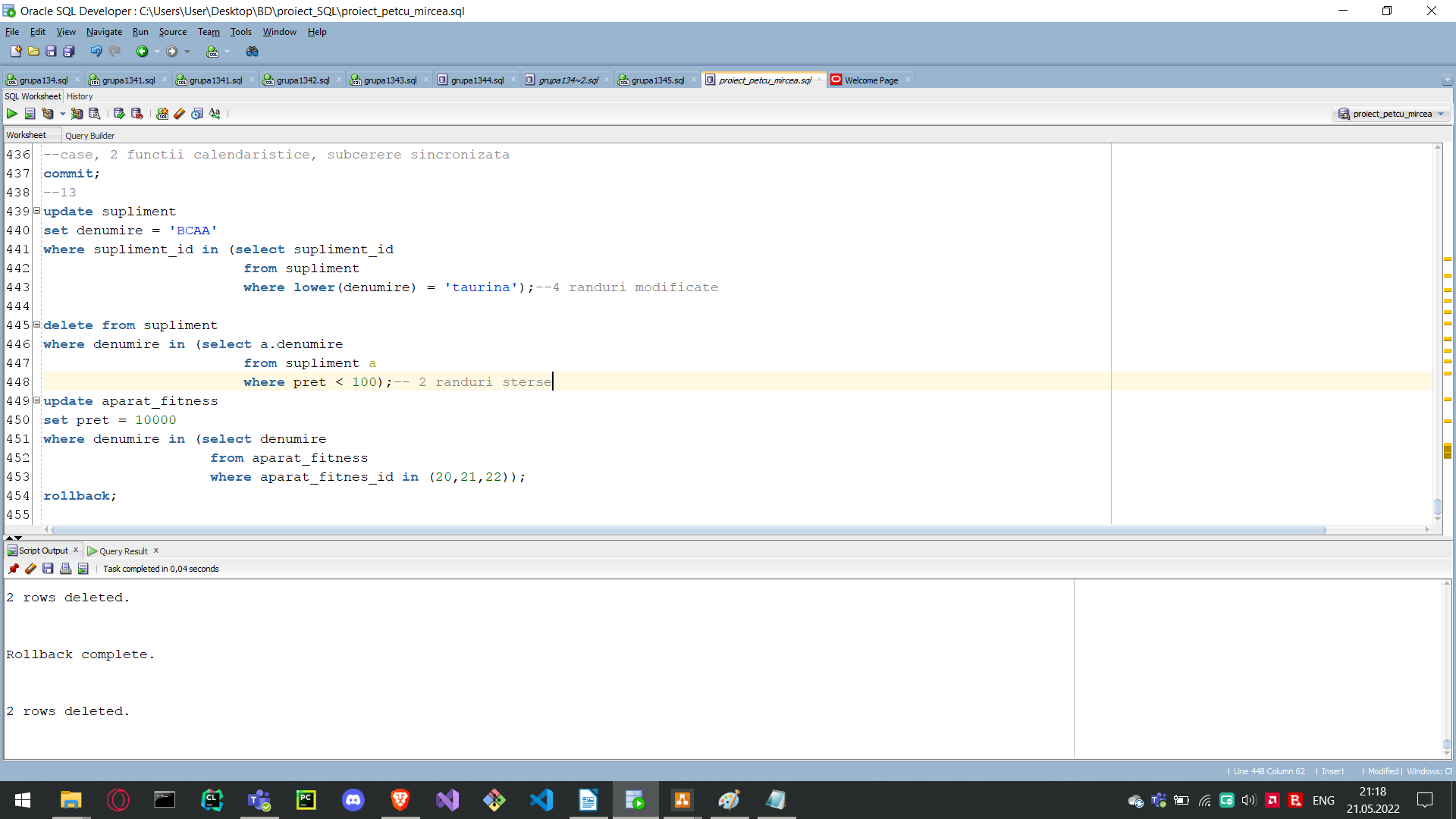
where lower(denumire) = 'taurina');

delete from supliment

where denumire in (select a.denumire

from supliment a

where pret < 100);-- 2 randuri sterse

update aparat\_fitness

set pret = 10000

where denumire in (select denumire

from aparat\_fitness

where aparat\_fitness\_id in (20,21,22));-- 3 randuri modificate

rollback;

