Gestionarea unei stupine

-Dogaru Mihail Danut

-grupa 132

Cuprins

[1.Descrierea modelului real 3](#_Toc136037842)

[2.Constrângerile impuse asupra modelului 3](#_Toc136037843)

[3.Descrierea entităților 4](#_Toc136037844)

[4.Descrierea relațiilor 5](#_Toc136037845)

[5. Descrierea atributelor 6](#_Toc136037846)

[6.Diagrama entitate-relație(după normalizare) 13](#_Toc136037847)

[7.Diagrama conceptuală(normalizată) 14](#_Toc136037848)

[8.Schemele relaționale 14](#_Toc136037849)

[9. Realizarea normalizării până la forma normală 3(FN1-FN3). 15](#_Toc136037850)

[10.Crearea unei secvențe de incrementare 17](#_Toc136037851)

[11.Crearea tabelelor și inserarea de date coerente 17](#_Toc136037852)

[Creare tabele 22](#_Toc136037853)

[Creare secvențe 25](#_Toc136037854)

[Inserturi 26](#_Toc136037855)

[12.Cereri SQL 37](#_Toc136037856)

[13.Operații de actualizare și suprimare 42](#_Toc136037857)

[15 44](#_Toc136037858)

# 1.Descrierea modelului real

Modelul de date va gestiona informații despre o stupină de multiplicare. Afacerea este gestionată de mai mulții apicultori licențiați ce își au stupii în mai multe locații pentru a acoperi o suprafață mai vastă, ajutând la producția de miere si diversitatea genetică. Fiecare apicultor poate merge în oricare dintre locație și poate verifica orice stup, stupii putând fi mutați dintr-o locație în alta.

Apicultorii pot angaja lucrători pe diverse posturi, fiind asignați uneia dintre locații. Afacerea trebuie sa aibe întotdeauna un contabil, iar în fiecare locație cel puțin un paznic pentru ca stupii să nu fie furați. Angajații pot lucra o singură dată pentru firmă, concedierea fiind permanentă.

Pot exista stupi fără matcă, aceasta fiind adusă ulterior, dar poate fi și înlocuită cu una nouă, însă trebuie sa fie mereu într-o familie de albine. Mătcile sunt marcate respectând convenția de marcare,iar stupii sunt colorați pentru a ajuta la orientarea albinelor,dar sunt și numerotați pentru a facilita identificarea lor, însă au și un cod unic de înregistrare în cadrul județului în care este licențiată stupina.

Fiecare stup poate produce o varietate de produse apicole specifice consumului, dar și multiplicării și pot fi achiziționate de orice client.

# 2.Constrângerile impuse asupra modelului

* Apicultorii pot vizita simultan mai multe locații
* Un stup se poate afla într-o singura locație la un moment,dar poate fi mutat,dar după ce este mutat trebuie verificat
* Fiecare apicultor poate decide subiectiv ce calificative de producție, roire și blândețe să dea unei familii de albine
* Un stup poate să nu conțină nici o matcă sau una singură
* O matcă trebuie să aparțină unei familii de albine
* Rasa mătcii trebuie sa fie “carpatină”, fiind rasa pe care este certificată stupina
* Un angajat poate lucra într-o singură locație la un moment,dar poate fi mutat
* Fiecare locație trebuie sa aibă cel puțin un angajat (paznic)
* Clienții pot să nu fi cumpărat încă nici un produs
* Produsele pot fi produse de mai mulți stupi
* 1 <= producție\_miere <= 5 ,unde 5 este cea mai bună valoare (stupul are o cantitate ridicată de miere adusă)
* 1 <= instinct\_roire <= 5 , unde 5 este cea mai buna valoare (familia de albine nu tinde să roiască)
* 1 <= blândețe <= 5 ,unde 5 este cea mai bună valoare (albinele sunt cât mai blânde)
* Culoarea marcajului mătcii poate fi doar: roșu, albastru, galben, alb, verde, respectând standardul internațional

# 3.Descrierea entităților

APICULTOR = persoană fizică a cărei activitate constă în creșterea familiilor de albine și gestionarea stupinei.

* entitate independentă
* cheie primară: id\_apicultor

LOCAȚIE = spațiul unde sunt ținuți o parte din stupi

* entitate independentă
* cheie primară: cod\_locatie

FAMILIE\_DE\_ALBINE = totalitatea albinelor dintr-un stup, alcătuită din albine lucrătoare, matcă și trântori

* entitate independentă
* cheie primară : id\_stup

MATCA = femela albina, unică în stup, care are rolul de a depune ouă din care vor ecluziona noi generații de albine, numită și “regina albină”

* entitate dependentă de FAMILIE\_DE\_ALBINE
* cheie primară: id\_matca
* cheie străină: id\_stup

ANGAJAT = persoană fizică, angajată într-una din locațiile pe care se desfășoară activitatea stupinei

* entitate dependentă de LOCATIE
* cheie primară: id\_angajat
* cheie străină: cod\_locatie

CLIENT = persoană fizică ce poate comanda produse apicole de la stupină

* entitate independentă
* cheie primară: id\_client

PRODUS = materialul produs de apicultori prin exploatarea familiilor de albine, pentru a fi puse în vânzare

* entitate independentă
* cheie primară: cod\_produs

# 4.Descrierea relațiilor

CLIENT\_cumpara\_PRODUSE = relație ce leagă entitățile CLIENT și PRODUSE (ce produse a cumpărat fiecare client).

* Cardinalitate minimă: 0:0
* Cardinalitate maximă: n:m

FAMILIE\_DE\_ALBINE\_produce\_PRODUS = relație ce leagă entitățile FAMILIE\_DE\_ALBINE și PRODUS (ce produse au fost recoltate de la o familie de albine).

* Cardinalitate minimă: 1:0
* Cardinalitate maximă: n:m

APICULTOR\_angajeaza\_ANGAJAT = relație ce leagă entitățile APICULTOR și ANGAJAT ( de către ce apicultor a fost angajat fiecare lucrător al firmei).

* Cardinalitate minimă: 1:0
* Cardinalitate maximă: n:m

Intr-o\_LOCATIE\_lucreaza\_ANGAJAT = relație ce leagă entitățile ANGAJAT și LOCATIE (ce angajați lucrează intr-una din locațiile pe care se desfășoară activitatea stupinei).

* Cardinalitate minimă: 1:1
* Cardinalitate maximă: 1:n

FAMILIE\_DE\_ALBINE\_contine\_MATCA = relație ce leagă entitățile FAMILIE\_DE\_ALBINE și MATCA (mătcile conținute in stupi).

* Cardinalitate minimă: 1:0
* Cardinalitate maximă: 1:1

APICULTOR\_viziteaza\_LOCATIE = relație ce leagă entitățile APICULTOR și LOCATIE (ce apicultor a fost să verifice una dintre locații)

* Cardinalitate minimă: 1:1
* Cardinalitate maximă: n:m

APICULTOR\_verifica\_FAMILIE\_DE\_ALBINE\_amplasata\_intr-o\_LOCATIE = relație de tip 3 ce leagă entitățile APICULTOR, FAMILIE\_DE\_ALBINE și LOCATIE (ce stup a fost verificat de către apicultor, dintr-o locație).

* Cardinalitate minimă: 1:1:1
* Cardinalitate maximă: n:m:p

Denumirea relației: *verifica*

# 5. Descrierea atributelor

Entitatea independentă APICULTOR

* Id\_apicultor = codul unic de înregistrare al apicultorului
  + Tip: number
  + Dimensiune/Precizie: 6,0
  + NOT NULL
  + PK
* nume\_apicultor = numele apicultorului
  + Tip: varchar2
  + Dimensiune: 20
  + NOT NULL
* prenume\_apicultor = prenumele apicultorului
  + Tip: varchar2
  + Dimensiune: 20
  + NOT NULL
* data\_inregistrare = data la care apicultorul și-a primit legitimarea
  + Tip: date
  + NOT NULL

Entitatea independentă CLIENT

* id\_client = codul unic de client din evidența firmei
  + Tip: number
  + Dimensiune/Precizie: 6,0
  + NOT NULL
  + PK
* nume\_client = numele clientului
  + Tip: varchar2
  + Dimensiune: 20
  + NOT NULL
* prenume\_client = prenumele clientului
  + Tip: varchar2
  + Dimensiune: 20
* multiplicator = daca clientul este multiplicator sau nu
  + Tip: number
  + Dimensiune: 1
  + Valori posibile: 0 , 1
* nr\_telefon = numărul de telefon al clientului
  + Tip: varchar2
  + Dimensiune: 10

Entitatea independentă PRODUS

* cod\_produs = codul unic al
  + Tip: number
  + Dimensiune/Precizie: 13,0
  + NOT NULL
  + PK
* preț = prețul produsului
  + Tip: number
  + Dimensiune/Precizie: 5,2
* nume\_produs = numele produsului
  + Tip: varchar2
  + Dimensiune: 30
* categorie = categoria produsului
  + Tip: varchar2
  + Dimensiune: 20
  + Valori posibile: NULL, biologic, consumabil

Entitatea dependentă ANGAJAT

* id\_angajat = codul unic de angajat din cadrul firmei
  + Tip: number
  + Dimensiune/Precizie: 3,0
  + NOT NULL
  + PK
* cod\_locatie = locul în care lucrează angajatul
  + Tip: number
  + Dimensiune/Precizie: 2,0
  + NOT NULL
  + FK din tabela LOCATIE
* nume\_angajat = numele angajatului
  + Tip:varchar2
  + Dimensiune:20
* prenume\_angajat = prenumele angajatului
  + Tip:varchar2
  + Dimensiune: 20

Entitatea independentă LOCATIE

* cod\_locatie = codul unic al locației unde se află stupii
  + Tip: number
  + Dimensiune/Precizie: 2,0
  + NOT NULL
  + PK
* nume\_loc = numele locului unde se află stupii
  + Tip: varchar2
  + Dimensiune: 20

Entitatea dependentă FAMILIE\_DE\_ALBINE

* id\_stup = numărul unic de înmatriculare al stupului
  + Tip: varchar2
  + Dimensiune: 8
  + NOT NULL
  + PK
* culoare = culoarea cu care este vopsit stupul pentru a fi distinctiv albinelor
  + Tip: varchar2
  + Dimensiune: 15
* numar = indexul stupului din cadrul locației
  + Tip:number
  + Dimensiune/Precizie: 3,0

Entitatea dependentă MATCA

* id\_matca = codul unic al mătcii
  + Tip: varchar2
  + Dimensiune:3
  + NOT NULL
  + PK
* id\_stup = stupul de care aparține matca
  + Tip:varchar2
  + Dimensiune 6
  + NOT NULL
  + FK din tabela FAMILIE\_DE\_ALBINE
* rasa = rasa mătcii
  + Tip:varchar2
  + Dimensiune: 20
  + Valori posibile: “carpatina”
* culoare\_marcaj = culoare cu care a fost însemnată în anul împerecherii
  + Tip: varchar2
  + Dimensiune: 1
  + Valori posibile: a(alb), g(galben), s(albastru), r(roșu), v(verde)
* data\_inlocuire = data la care matca a fost schimbată, dacă aceasta este NULL înseamnă că încă se află în stup
  + Tip:date

Relația CLIENT\_cumpara\_PRODUSE

* cod\_vanzare = codul unic al înregistrării tranzacției
  + Tip: number
  + Dimensiune/Precizie:8,0
  + NOT NULL
  + PK
* id\_client = id-ul clientului care cumpără produsul
  + Tip:number
  + Dimensiune/Precizie: 6,0
  + FK din tabela CLIENT
* cod\_produs = codul produsului achiziționat
  + Tip: number
  + Dimensiune/Precizie:13,0
  + FK din tabela PRODUS
* cantitate = cantitatea din care clientul a cumpărat din fiecare produs
  + Tip:number
  + Dimensiune/Precizie:4,0
* data\_vanzare = data la care a fost plătită comanda
  + Tip:date

Relația FAMILIE\_DE\_ALBINE\_produce\_PRODUS

* Id\_recoltare = id-ul recoltării
  + Tip:number
  + Dimensiune/Precizie: 7,0
  + NOT NULL
  + PK
* id\_stup = id-ul stupului din care provine produsul
  + Tip: varchar2
  + Dimensiune: 8
  + FK din tabela FAMILIE\_DE\_ALBINE
* cod\_produs = codul produsului recoltat
  + Tip:number
  + Dimensiune/Precizie: 13,0
  + FK din tabela PRODUS
* cantitate = cantitatea recoltată de produs dintr-un stup
  + Tip:number
  + Dimensiune/Precizie: 4,0

Relația APICULTOR\_angajeaza\_ANGAJAT

* id\_apicultor = id-ul apicultorului care a aprobat angajarea
  + Tip:number
  + Dimensiune/Precizie: 6,0
  + FK din tabela APICULTOR
* id\_angajat = id-ul angajatului
  + Tip:number
  + Dimensiune/Precizie: 3,0
  + FK din tabela ANGAJAT
* data\_angajare = data la care a fost angajat clientul de către apicultor
  + Tip:date
* data\_eliminare = data la care angajatul a părăsit firma
  + Tip:date
* post = poziția pe care este/a fost angajat lucrătorul
  + Tip:varchar2
  + Dimensiune: 20
* salariu = salariul lunar pe care îl are/l-a avut lucrătorul în RON
  + Tip:number
  + Dimensiune/Precizie: 5,0

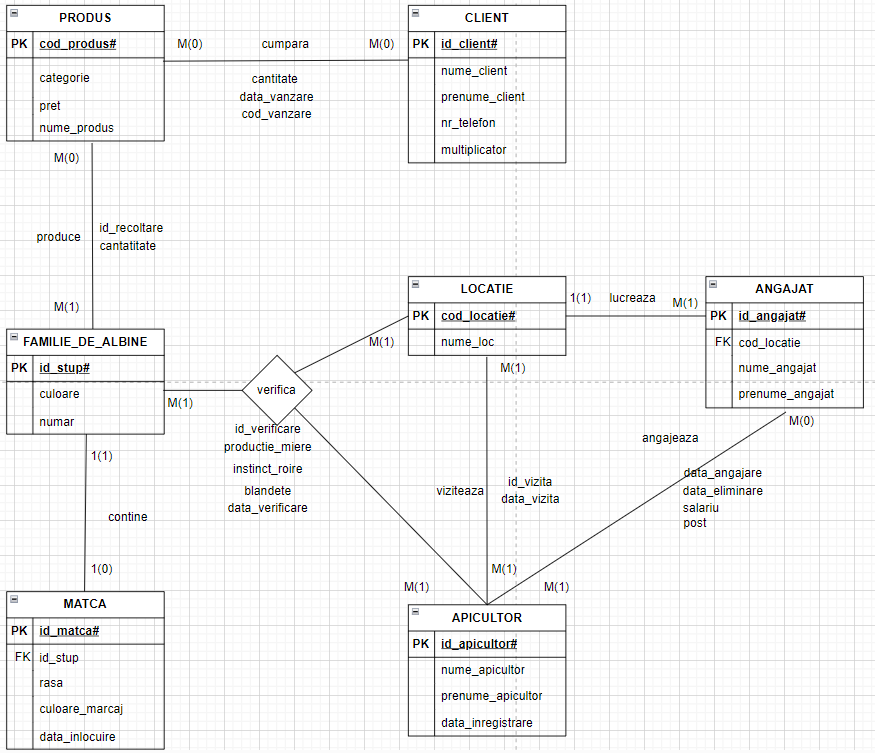
Relația APICULTOR\_viziteaza\_LOCATIE

* Id\_vizita = id-ul vizitei
  + Tip:number
  + Dimensiune/Precizie: 6,0
  + NOT NULL
  + PK
* id\_apicultor = id-ul apicultorului ce a realizat vizita
  + Tip:number
  + Dimensiune/Precizie: 6,0
  + FK din tabela APICULTOR
* cod\_locatie = codul locației vizitate
  + Tip:number
  + Dimensiune/Precizie:2,0
  + FK din tabela LOCATIE
* data\_vizita = data la care a fost vizitată locația
  + Tip:date

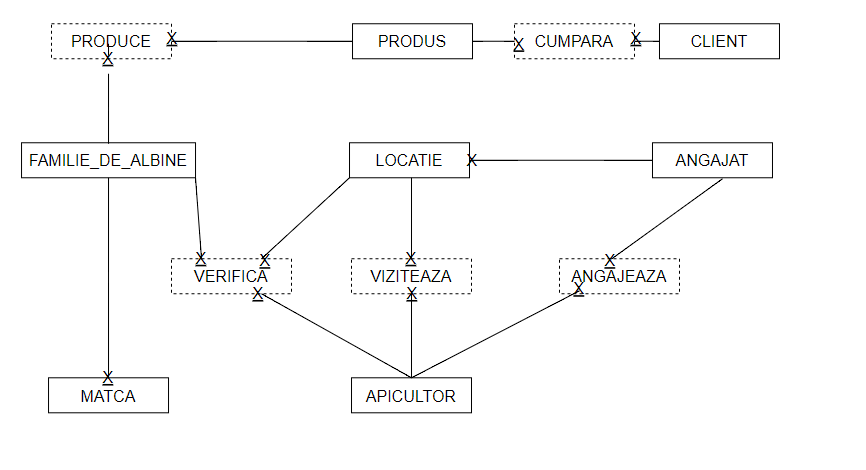
Relația APICULTOR\_verifica\_FAMILIE\_DE\_ALBINE\_amplasata\_intr-o\_LOCATIE

* Id\_verificare = id-ul verificării
  + Tip: number
  + Dimensiune/Precizie: 7,0
  + NOT NULL
  + PK
* id\_apicultor = id-ul apicultorului care face verificarea
  + Tip: number
  + Dimensiune/Precizie: 6,0
  + FK din tabela APICULTOR
* cod\_locatie = codul locației vizitate
  + Tip:number
  + Dimensiune/Precizie: 2,0
  + FK din tabela LOCATIE
* id\_stup = id-ul stupului verificat
  + Tip:varchar2
  + Dimensiune: 6
  + FK din tabela FAMILIE\_DE\_ALBINE
* productie\_miere = calificativul familiei de albine primit pentru cantitatea de miere pe care a recoltat-o
  + Tip:number
  + Dimensiune/Precizie: 1,0
  + Valori posibile: 1-5
* blandete = calificativul familiei de albine primit pentru gradul de agresivitate
  + Tip:number
  + Dimensiune/Precizie: 1,0
  + Valori posibile: 1-5
* instinct\_roire = calificativul familiei de albine primit pentru cât de mult tinde să roiască
  + Tip:number
  + Dimensiune/Precizie: 1,0
  + Valori posibile: 1-5
* data\_verficare = data la care a fost verificată familia de albine
  + Tip:date

6.Diagrama entitate-relație(după normalizare)



# 7.Diagrama conceptuală(normalizată)



# 8.Schemele relaționale

APICULTOR(id\_apicultor#, nume\_apicultor, prenume\_apicultor, data\_inregistrare)

CLIENT(id\_client#, nume\_client, prenume\_client, nr\_telefon, multiplicator)

PRODUS(cod\_produs#, categorie, pret, nume\_produs)

ANGAJAT(id\_angajat#, nume\_angajat, prenume\_angajat)

LOCATIE(cod\_locatie#, nume\_loc, nr\_stupi)

FAMILIE\_DE\_ALBINE(id\_stup#, culoare, numar, regina)

MATCA(id\_matca#, id\_stup, rasa, culoare\_marcaj)

VERIFICA(id\_verificare#,id\_apicultor,id\_stup,cod\_locatie,productie\_miere,instinct\_roire,blandete, data\_verificare)

VIZITEAZA(id\_vizita#,id\_apicultor, cod\_locatie, comentariu\_vizita, data\_vizita)

ANGAJEAZA(id\_apicultor#, id\_angajat#, data\_angajare, data\_eliminare, salariu, post)

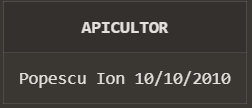
CUMPARA(cod\_vanzare#,id\_client, cod\_produs, cantitate, data\_vanzare)

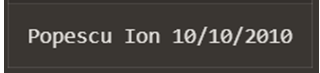
PRODUCE(id\_recoltare#,id\_stup, cod\_produs, nume\_produs, cantitate)

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

-exemplu FN1:

* entiatea APICULTOR este despărțită în atributele atomice: *nume\_apicultor,prenume\_apicultor,data\_inregistrare* pentru ca pe coloane să fie înregistrate doar câte o valoare
* fiecărei înregistrări îi este atribuită o cheie primară unică pentru identificarea înregistrării
* tabelele nu conțin duplicate

Ex non-FN1:



-exemplu FN2:

* tabela ANGAJEAZA este deja în FN1
* nu există dependență parțială,atributele care nu fac parte din cheia primară sunt dependente de cheia primară



Ex non-FN2:

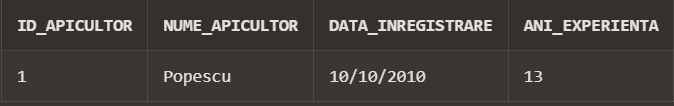


\*Unde cheia cheia primară a tabelei este compusa din atributele ID\_apicultor și ID\_angajat

-exemplu FN3

* tabela este deja in FN2
* nu există dependențe tranzitive, atribute care să depindă de valorile altor atribute

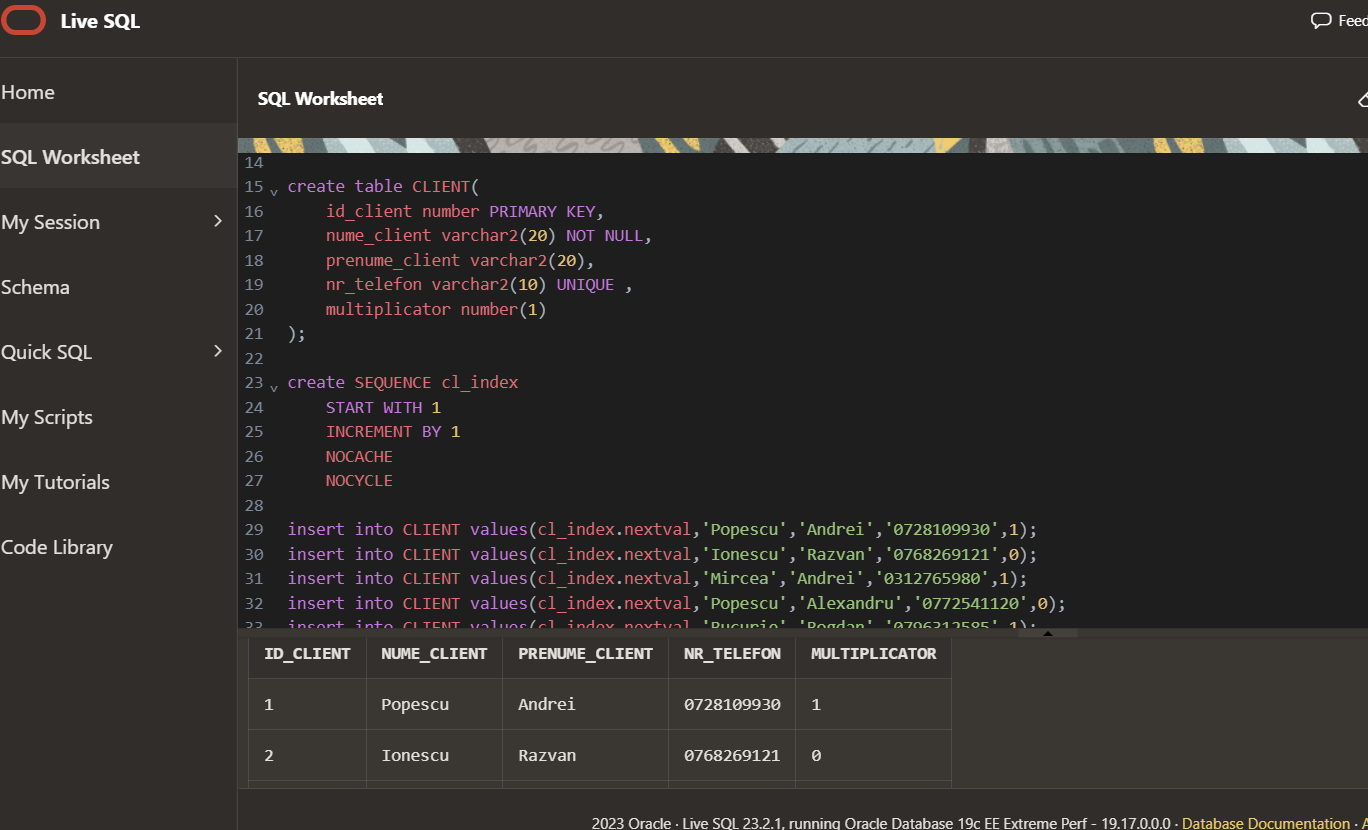
EX non-FN3:



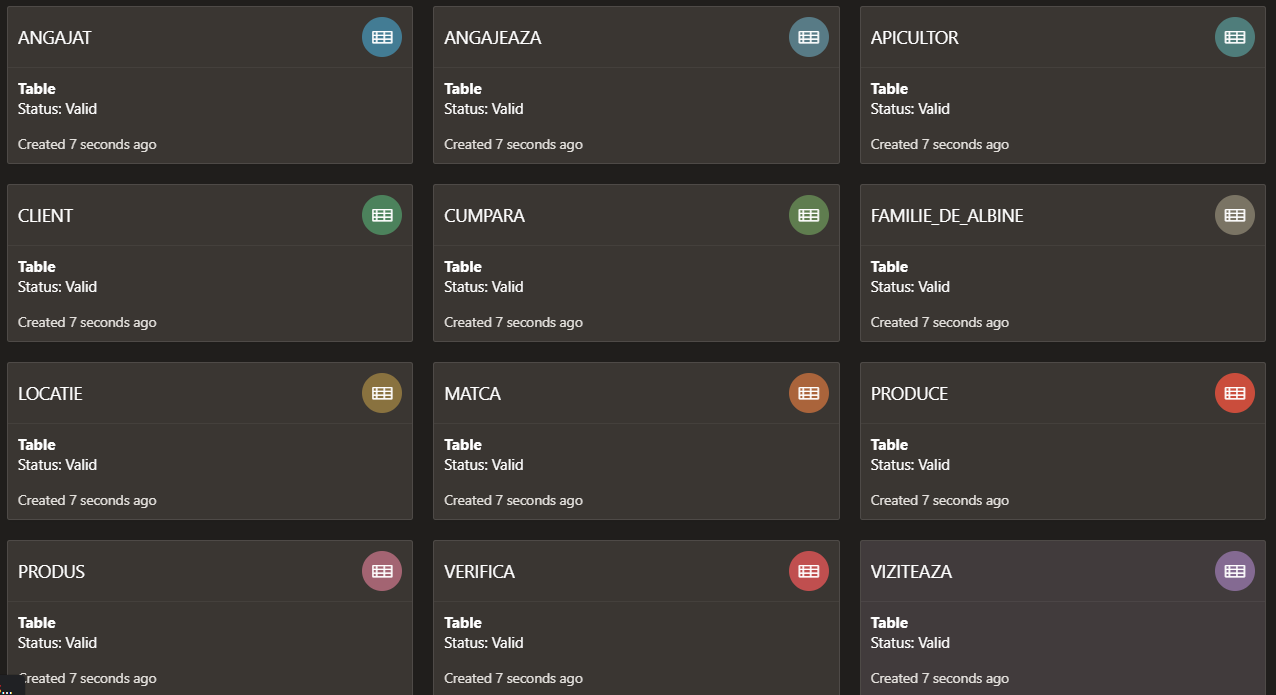
{Id\_apicultor(PK)} -> {data\_inregistrare} -> {ani\_experienta}

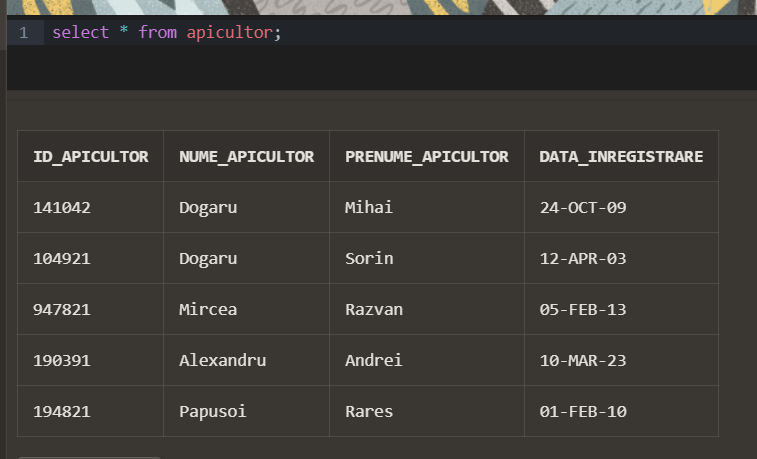
\*La updatarea coloanei *data\_inregistrare*  , coalana  *ani\_experienta* trebuie updatată la rândul ei ,conform datei de înregistrare

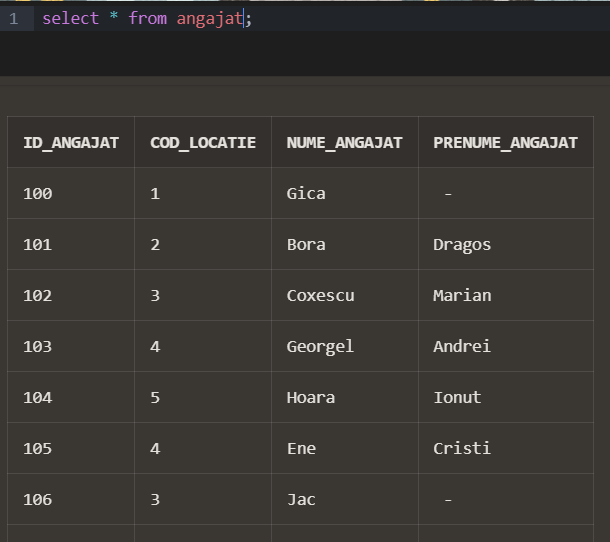
# 10.Crearea unei secvențe de incrementare



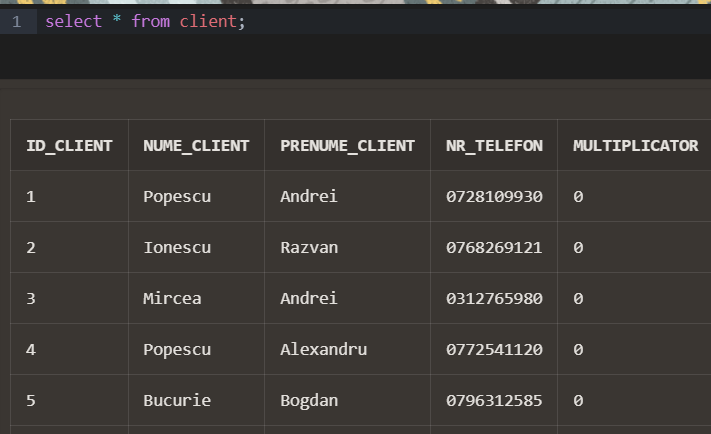
# 11.Crearea tabelelor și inserarea de date coerente

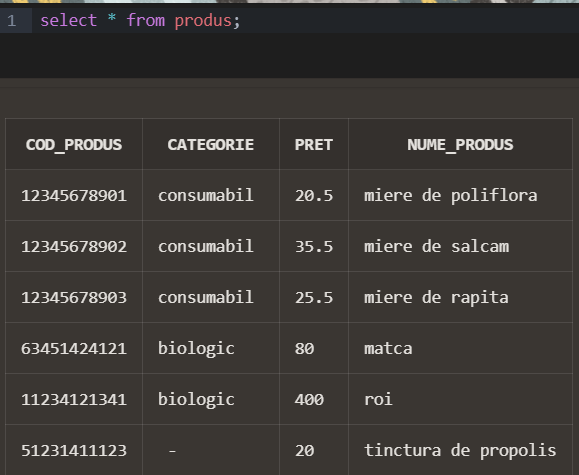


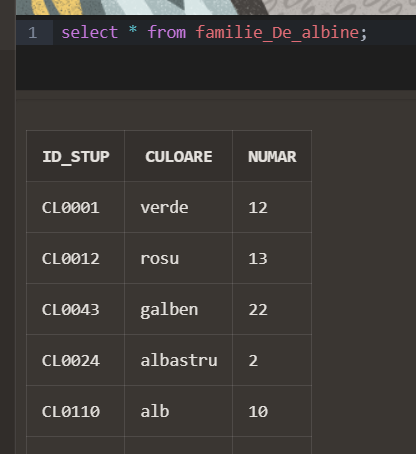


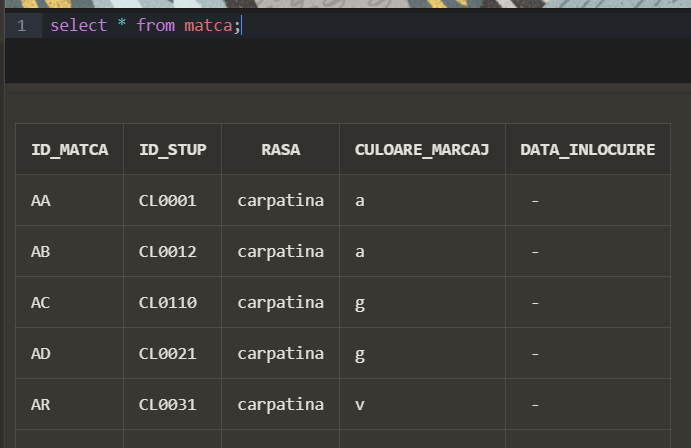
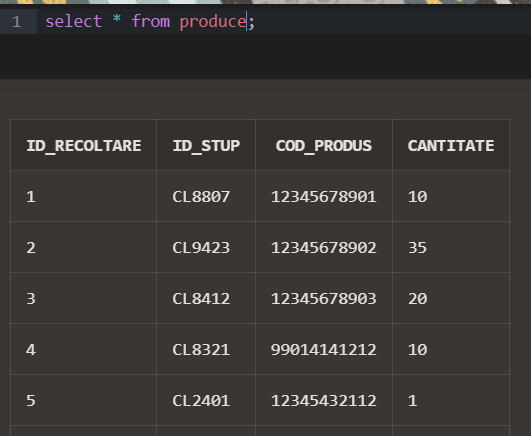


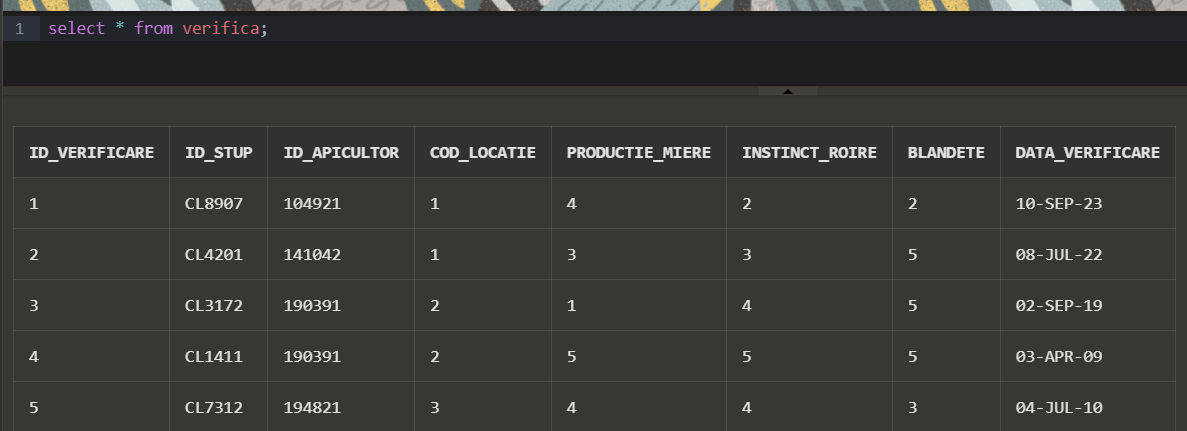




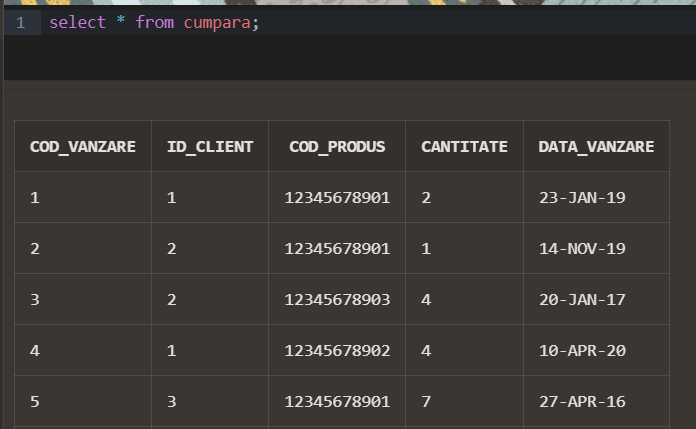
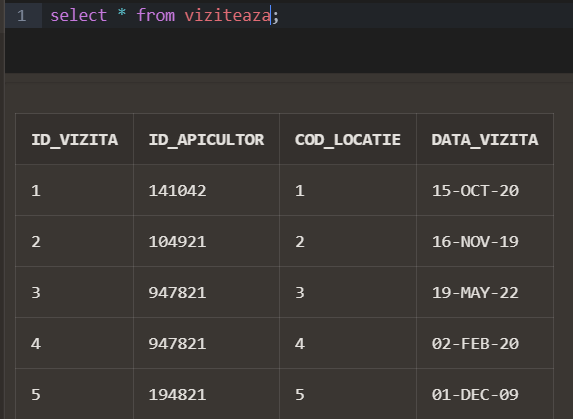








# Creare tabele

create table APICULTOR(

id\_apicultor number(6) PRIMARY KEY,

nume\_apicultor varchar2(20) NOT NULL,

prenume\_apicultor varchar2(20) NOT NULL,

data\_inregistrare date NOT NULL

);

create table CLIENT(

id\_client number(6) PRIMARY KEY,

nume\_client varchar2(20) NOT NULL,

prenume\_client varchar2(20),

nr\_telefon varchar2(10) UNIQUE ,

multiplicator number(1)

);

create table PRODUS(

cod\_produs number(13) PRIMARY KEY,

categorie varchar2(20),

pret number(5,2),

nume\_produs varchar2(30) UNIQUE

);

create table LOCATIE(

cod\_locatie number(2) PRIMARY KEY,

nume\_loc varchar2(20) UNIQUE

);

create table ANGAJAT(

id\_angajat number(3) PRIMARY KEY,

cod\_locatie number(2) NOT NULL,

nume\_angajat varchar2(20),

prenume\_angajat varchar2(20)

);

create table FAMILIE\_DE\_ALBINE(

id\_stup varchar2(8) PRIMARY KEY,

culoare varchar2(15),

numar number(3)

);

create table MATCA(

id\_matca varchar2(3) PRIMARY KEY,

id\_stup varchar2(6),

rasa varchar2(20) CHECK( rasa = 'carpatina'),

culoare\_marcaj varchar2(1) CHECK(culoare\_marcaj = 'a' or culoare\_marcaj = 's' or culoare\_marcaj = 'r' or culoare\_marcaj = 'g' or culoare\_marcaj = 'v'),

data\_inlocuire date

);

create table CUMPARA(

cod\_vanzare number(8) PRIMARY KEY,

id\_client number(6),

cod\_produs number(13),

cantitate number(4),

data\_vanzare date

);

create table VIZITEAZA(

id\_vizita number(6) PRIMARY KEY,

id\_apicultor number(6),

cod\_locatie number(2),

data\_vizita date

);

create table ANGAJEAZA(

id\_apicultor number(6),

id\_angajat number(3),

data\_angajare date,

data\_eliminare date,

post varchar2(20),

salariu number(5) CHECK (salariu >1000)

);

create table PRODUCE(

id\_recoltare number(7) PRIMARY KEY,

id\_stup varchar2(6),

cod\_produs number(13),

cantitate number(4) CHECK(cantitate > 0)

);

create table VERIFICA(

id\_verificare number(7) PRIMARY KEY,

id\_stup varchar2(6),

id\_apicultor number(6),

cod\_locatie number(2),

productie\_miere number(1) CHECK(productie\_miere > 0 and productie\_miere < 6 ),

instinct\_roire number(1) CHECK(instinct\_roire > 0 and instinct\_roire < 6),

blandete number(1) CHECK(blandete > 0 and blandete < 6),

data\_verificare date

);

alter table ANGAJAT

add constraint fk\_cod\_locatie FOREIGN KEY(cod\_locatie)

REFERENCES LOCATIE(cod\_locatie);

alter table MATCA

add constraint fk\_id\_stup FOREIGN KEY(id\_stup)

references FAMILIE\_DE\_ALBINE(id\_stup);

alter table PRODUCE

add(constraint fk\_prod\_id\_stup FOREIGN KEY(id\_stup) references FAMILIE\_DE\_ALBINE(id\_stup),

constraint fk\_prod\_cod\_produs FOREIGN KEY(cod\_produs) references PRODUS(cod\_produs));

alter table VERIFICA

add(constraint fk\_ver\_id\_stup FOREIGN KEY(id\_stup) references FAMILIE\_DE\_ALBINE(id\_stup),

constraint fk\_ver\_id\_apicultor FOREIGN KEY(id\_apicultor) references APICULTOR(id\_apicultor),

constraint fk\_ver\_cod\_locatie FOREIGN KEY(cod\_locatie) references LOCATIE(cod\_locatie));

alter table ANGAJEAZA

add (constraint pk\_angajeaza PRIMARY KEY(id\_apicultor,id\_angajat),

constraint fk\_ang\_id\_apicultor FOREIGN KEY(id\_apicultor) references APICULTOR(id\_apicultor),

constraint fk\_ang\_id\_angajat FOREIGN KEY(id\_angajat) references ANGAJAT(id\_angajat));

alter table VIZITEAZA

add(constraint fk\_viz\_id\_apicultor FOREIGN KEY(id\_apicultor) references APICULTOR(id\_apicultor),

constraint fk\_viz\_cod\_locatie FOREIGN KEY (cod\_locatie) references LOCATIE(cod\_locatie));

alter table CUMPARA

add( constraint fk\_cump\_id\_client FOREIGN KEY(id\_client) references CLIENT(id\_client),

constraint fk\_cump\_cod\_produs FOREIGN KEY(cod\_produs) references PRODUS(cod\_produs));

# Creare secvențe

create SEQUENCE index\_ang

START WITH 100

INCREMENT BY 1

MAXVALUE 999

NOCACHE

NOCYCLE;

create SEQUENCE cl\_index

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;

create SEQUENCE index\_loc

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;

create SEQUENCE index\_vanzare

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;

create SEQUENCE index\_vizita

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;

create SEQUENCE index\_recoltare

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;

create SEQUENCE index\_verificare

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;

# Inserturi

insert into APICULTOR values (141042, 'Dogaru','Mihai',to\_date('24.10.2009','dd.mm.yyyy'));

insert into APICULTOR values (104921 , 'Dogaru','Sorin',to\_date('12.04.2003','dd.mm.yyyy'));

insert into APICULTOR values (947821 , 'Mircea','Razvan',to\_date('05.02.2013','dd.mm.yyyy'));

insert into APICULTOR values (190391 , 'Alexandru','Andrei',to\_date('10.03.2023','dd.mm.yyyy'));

insert into APICULTOR values (194821, 'Papusoi','Rares',to\_date('01.02.2010','dd.mm.yyyy'));

--select \* from APICULTOR;

insert into CLIENT values(cl\_index.nextval,'Popescu','Andrei','0728109930',0);

insert into CLIENT values(cl\_index.nextval,'Ionescu','Razvan','0768269121',0);

insert into CLIENT values(cl\_index.nextval,'Mircea','Andrei','0312765980',0);

insert into CLIENT values(cl\_index.nextval,'Popescu','Alexandru','0772541120',0);

insert into CLIENT values(cl\_index.nextval,'Bucurie','Bogdan','0796312585',0);

insert into CLIENT values(cl\_index.nextval,'Iuliu','Daniel','0787654312',0);

insert into CLIENT values(cl\_index.nextval,'Anghel','Rares','0798765432',0);

insert into CLIENT values(cl\_index.nextval,'Paun','Andrei','0712311111',0);

insert into CLIENT values(cl\_index.nextval,'Ghergu','Cezar','0387645674',0);

insert into CLIENT values(cl\_index.nextval,'Grigore','Vasile','0700122332',0);

insert into client values (cl\_index.nextval, 'Coroama','Cristian',NULL,0);

insert into client values (cl\_index.nextval, 'Tirila','Patric',NULL,0);

insert into client values (cl\_index.nextval, 'Gavrila','Alexandru',NULL,0);

insert into client values (cl\_index.nextval, 'Codarcea','Alexandru',NULL,0);

--select \* from CLIENT;

insert into PRODUS values(12345678901,'consumabil',20.5 ,'miere de poliflora');

insert into PRODUS values(12345678902,'consumabil',35.5 ,'miere de salcam');

insert into PRODUS values(12345678903,'consumabil',25.5 ,'miere de rapita');

insert into PRODUS values(63451424121,'biologic', 80,'matca');

insert into PRODUS values(11234121341,'biologic',400 ,'roi');

insert into PRODUS values(51231411123,NULL,20 ,'tinctura de propolis');

insert into PRODUS values(99014141212,'consumabil',25,'laptisor de matca');

insert into produs values(14252641314,'biologic','25','botca');

insert into produs values('12345432112','consumbabil',50,'polen');

--select \* from PRODUS;

insert into LOCATIE values(index\_loc.nextval,'padure ciornuleasa');

insert into LOCATIE values(index\_loc.nextval,'mamaia');

insert into LOCATIE values(index\_loc.nextval,'camp');

insert into LOCATIE values(index\_loc.nextval,'acasa');

insert into LOCATIE values(index\_loc.nextval,'padure curcani');

--select \* from LOCATIE;

insert into ANGAJAT values(index\_ang.nextval,1,'Gica',NULL);

insert into ANGAJAT values(index\_ang.nextval,2,'Bora','Dragos');

insert into ANGAJAT values(index\_ang.nextval,3,'Coxescu','Marian');

insert into ANGAJAT values(index\_ang.nextval,4,'Georgel','Andrei');

insert into ANGAJAT values(index\_ang.nextval,5,'Hoara','Ionut');

insert into ANGAJAT values(index\_ang.nextval,4,'Ene','Cristi');

insert into ANGAJAT values(index\_ang.nextval,3,'Jac',NULL);

insert into ANGAJAT values(index\_ang.nextval,4,'Bobita',NULL);

insert into ANGAJAT values(index\_ang.nextval,1,'Cristi','Cristi');

insert into ANGAJAT values(index\_ang.nextval,4,'Ionut','Ionut');

--select \* from ANGAJAT;

--loc1

insert into FAMILIE\_DE\_ALBINE values('CL0001','verde',12);

insert into FAMILIE\_DE\_ALBINE values('CL0012','rosu',13);

insert into FAMILIE\_DE\_ALBINE values('CL0043','galben',22);

insert into FAMILIE\_DE\_ALBINE values('CL0024','albastru',2);

insert into FAMILIE\_DE\_ALBINE values('CL0110','alb',10);

insert into FAMILIE\_DE\_ALBINE values('CL0021','verde',17);

insert into FAMILIE\_DE\_ALBINE values('CL0031','albastru',56);

insert into FAMILIE\_DE\_ALBINE values('CL0041','rosu',101);

insert into FAMILIE\_DE\_ALBINE values('CL0054','alb',345);

insert into FAMILIE\_DE\_ALBINE values('CL0087','rosu',19);

--loc2

insert into FAMILIE\_DE\_ALBINE values('CL0056','verde',1);

insert into FAMILIE\_DE\_ALBINE values('CL0023','rosu',77);

insert into FAMILIE\_DE\_ALBINE values('CL0112','alb',102);

insert into FAMILIE\_DE\_ALBINE values('CL0101','alb',100);

insert into FAMILIE\_DE\_ALBINE values('CL0201','albastru',71);

--loc3

insert into FAMILIE\_DE\_ALBINE values('CL0090','verde',321);

insert into FAMILIE\_DE\_ALBINE values('CL0009','rosu',118);

insert into FAMILIE\_DE\_ALBINE values('CL1111','alb',512);

insert into FAMILIE\_DE\_ALBINE values('CL3101','alb',24);

insert into FAMILIE\_DE\_ALBINE values('CL4201','albastru',171);

insert into FAMILIE\_DE\_ALBINE values('CL5656','verde',906);

insert into FAMILIE\_DE\_ALBINE values('CL1002','rosu',471);

insert into FAMILIE\_DE\_ALBINE values('CL3172','alb',903);

insert into FAMILIE\_DE\_ALBINE values('CL3501','alb',156);

insert into FAMILIE\_DE\_ALBINE values('CL9109','albastru',781);

insert into FAMILIE\_DE\_ALBINE values('CL7456','verde',822);

insert into FAMILIE\_DE\_ALBINE values('CL1223','rosu',773);

insert into FAMILIE\_DE\_ALBINE values('CL7410','alb',404);

insert into FAMILIE\_DE\_ALBINE values('CL0381','alb',828);

insert into FAMILIE\_DE\_ALBINE values('CL1411','albastru',971);

insert into FAMILIE\_DE\_ALBINE values('CL8806','verde',991);

insert into FAMILIE\_DE\_ALBINE values('CL7323','rosu',797);

insert into FAMILIE\_DE\_ALBINE values('CL6312','alb',717);

insert into FAMILIE\_DE\_ALBINE values('CL6221','alb',530);

insert into FAMILIE\_DE\_ALBINE values('CL9301','albastru',978);

--loc4

insert into FAMILIE\_DE\_ALBINE values('CL9806','verde',900);

insert into FAMILIE\_DE\_ALBINE values('CL8323','rosu',700);

insert into FAMILIE\_DE\_ALBINE values('CL7312','alb',117);

insert into FAMILIE\_DE\_ALBINE values('CL7221','alb',500);

insert into FAMILIE\_DE\_ALBINE values('CL1301','albastru',970);

insert into FAMILIE\_DE\_ALBINE values('CL9807','rosu',904);

insert into FAMILIE\_DE\_ALBINE values('CL8324','rosu',710);

insert into FAMILIE\_DE\_ALBINE values('CL7313','alb',515);

insert into FAMILIE\_DE\_ALBINE values('CL7022','verde',487);

insert into FAMILIE\_DE\_ALBINE values('CL5302','bej',877);

--loc5

insert into FAMILIE\_DE\_ALBINE values('CL8807','verde',999);

insert into FAMILIE\_DE\_ALBINE values('CL9423','rosu',799);

insert into FAMILIE\_DE\_ALBINE values('CL8412','alb',199);

insert into FAMILIE\_DE\_ALBINE values('CL8321','alb',599);

insert into FAMILIE\_DE\_ALBINE values('CL2401','albastru',222);

insert into FAMILIE\_DE\_ALBINE values('CL8907','rosu',333);

insert into FAMILIE\_DE\_ALBINE values('CL1824','rosu',444);

insert into FAMILIE\_DE\_ALBINE values('CL7553','alb',151);

insert into FAMILIE\_DE\_ALBINE values('CL7442','verde',666);

insert into FAMILIE\_DE\_ALBINE values('CL5444','alb',777);

--select \* from FAMILIE\_DE\_ALBINE;

insert into MATCA values('AA','CL0001','carpatina','a',NULL);

insert into MATCA values('AB','CL0012','carpatina','a',NULL);

insert into MATCA values('AC','CL0110','carpatina','g',NULL);

insert into MATCA values('AD','CL0021','carpatina','g',NULL);

insert into MATCA values('AR','CL0031','carpatina','v',NULL);

insert into MATCA values('AT','CL0041','carpatina','r',NULL);

insert into MATCA values('AY','CL0087','carpatina','r',NULL);

insert into MATCA values('AU','CL0023','carpatina','s',NULL);

insert into MATCA values('AI','CL0056','carpatina','a',NULL);

insert into MATCA values('AO','CL0101','carpatina','g',NULL);

insert into MATCA values('BT','CL0090','carpatina','r',NULL);

insert into MATCA values('BY','CL0009','carpatina','s',NULL);

insert into MATCA values('BU','CL3101','carpatina','s',NULL);

insert into MATCA values('BI','CL5656','carpatina','a',NULL);

insert into MATCA values('BR','CL9806','carpatina','v',NULL);

insert into MATCA values('BE','CL5444','carpatina','a',NULL);

insert into MATCA values('CY','CL1223','carpatina','r',NULL);

insert into MATCA values('CU','CL0381','carpatina','v',NULL);

insert into MATCA values('CI','CL8806','carpatina','a',NULL);

insert into MATCA values('WO','CL7323','carpatina','g',NULL);

insert into MATCA values('EE','CL7022','carpatina','v',NULL);

insert into MATCA values('QW','CL9423','carpatina','a',NULL);

--select \* from MATCA

insert into cumpara values(index\_vanzare.nextval, 1,12345678901,2,to\_date('23.01.2019','dd.mm.yyyy'));

insert into cumpara values(index\_vanzare.nextval, 2,12345678901,1,to\_date('14.11.2019','dd.mm.yyyy'));

insert into cumpara values(index\_vanzare.nextval, 2,12345678903,4,to\_date('20.01.2017','dd.mm.yyyy'));

insert into cumpara values(index\_vanzare.nextval, 1,12345678902 ,4,to\_date('10.04.2020','dd.mm.yyyy'));

insert into cumpara values(index\_vanzare.nextval, 3,12345678901,7,to\_date('27.04.2016','dd.mm.yyyy'));

insert into cumpara values(index\_vanzare.nextval, 1,99014141212,10,to\_date('13.07.2018','dd.mm.yyyy'));

insert into cumpara values(index\_vanzare.nextval, 10,12345678901,3,to\_date('20.02.2022','dd.mm.yyyy'));

insert into cumpara values(index\_vanzare.nextval, 5,63451424121,23,to\_date('15.11.2023','dd.mm.yyyy'));

insert into cumpara values(index\_vanzare.nextval, 3,11234121341,5,to\_date('20.05.2020','dd.mm.yyyy'));

insert into cumpara values(index\_vanzare.nextval, 4,14252641314,10,to\_date('23.01.2019','dd.mm.yyyy'));

insert into cumpara values(index\_vanzare.nextval, 9,12345678901,1,to\_date('21.11.2009','dd.mm.yyyy'));

insert into cumpara values(index\_vanzare.nextval, 8,63451424121,12,to\_date('24.05.2018','dd.mm.yyyy'));

insert into cumpara values(index\_vanzare.nextval, 7,12345678901,11,to\_date('20.03.2020','dd.mm.yyyy'));

insert into cumpara values(index\_vanzare.nextval, 6,11234121341,2,to\_date('03.01.2023','dd.mm.yyyy'));

--select \* from CUMPARA;

insert into viziteaza values(index\_vizita.nextval,141042,1,to\_date('15.10.2020','dd.mm.yyyy'));

insert into viziteaza values(index\_vizita.nextval,104921,2,to\_date('16.11.2019','dd.mm.yyyy'));

insert into viziteaza values(index\_vizita.nextval,947821,3,to\_date('19.05.2022','dd.mm.yyyy'));

insert into viziteaza values(index\_vizita.nextval,947821,4,to\_date('02.02.2020','dd.mm.yyyy'));

insert into viziteaza values(index\_vizita.nextval,194821,5,to\_date('01.12.2009','dd.mm.yyyy'));

insert into viziteaza values(index\_vizita.nextval,104921,2,to\_date('26.04.2010','dd.mm.yyyy'));

insert into viziteaza values(index\_vizita.nextval,104921,3,to\_date('30.11.2019','dd.mm.yyyy'));

insert into viziteaza values(index\_vizita.nextval,104921,1,to\_date('15.06.2022','dd.mm.yyyy'));

insert into viziteaza values(index\_vizita.nextval,141042,5,to\_date('14.05.2018','dd.mm.yyyy'));

insert into viziteaza values(index\_vizita.nextval,947821,5,to\_date('15.10.2019','dd.mm.yyyy'));

--select \* from VIZITEAZA;

insert into angajeaza values (141042,100 ,to\_date('04.07.2017','dd.mm.yyyy'),NULL,'paznic','1700');

insert into angajeaza values (104921,101 ,to\_date('12.03.2018','dd.mm.yyyy'),NULL,'paznic','1650');

insert into angajeaza values (141042,102 ,to\_date('14.04.2020','dd.mm.yyyy'),to\_date('11.12.2009','dd.mm.yyyy'),'sofer','1400');

insert into angajeaza values (141042,103 ,to\_date('10.12.2009','dd.mm.yyyy'),to\_date('11.12.2009','dd.mm.yyyy'),'asistent','1800');

insert into angajeaza values (104921,104 ,to\_date('10.12.2009','dd.mm.yyyy'),NULL,'paznic','1700');

insert into angajeaza values (104921,105 ,to\_date('15.05.2010','dd.mm.yyyy'),NULL,'contabil','2500');

insert into angajeaza values (194821,106 ,to\_date('14.01.2019','dd.mm.yyyy'),NULL,'paznic','1800');

insert into angajeaza values (104921,107 ,to\_date('10.12.2009','dd.mm.yyyy'),NULL,'paznic','1700');

insert into angajeaza values (194821,108 ,to\_date('24.09.2011','dd.mm.yyyy'),NULL,'asistent','1800');

insert into angajeaza values (104921,109 ,to\_date('07.04.2020','dd.mm.yyyy'),NULL,'ambalator','1700');

--select \* from ANGAJEAZA;

insert into produce values(index\_recoltare.nextval,'CL8807',12345678901,10);

insert into produce values(index\_recoltare.nextval,'CL9423',12345678902,35);

insert into produce values(index\_recoltare.nextval,'CL8412',12345678903,20);

insert into produce values(index\_recoltare.nextval,'CL8321',99014141212,10);

insert into produce values(index\_recoltare.nextval,'CL2401',12345432112,1);

insert into produce values(index\_recoltare.nextval,'CL8907',12345678901,10);

insert into produce values(index\_recoltare.nextval,'CL1824',12345678901,14);

insert into produce values(index\_recoltare.nextval,'CL0054',12345678901,26);

insert into produce values(index\_recoltare.nextval,'CL0056',12345678901,21);

insert into produce values(index\_recoltare.nextval,'CL0101',12345678901,20);

insert into produce values(index\_recoltare.nextval,'CL0201',12345678902,23);

insert into produce values(index\_recoltare.nextval,'CL3101',12345678901,13);

insert into produce values(index\_recoltare.nextval,'CL4201',12345678901,24);

insert into produce values(index\_recoltare.nextval,'CL5656',12345678902,17);

insert into produce values(index\_recoltare.nextval,'CL9109',12345678901,35);

insert into produce values(index\_recoltare.nextval,'CL1301',12345678902,25);

insert into produce values(index\_recoltare.nextval,'CL7312',63451424121,1);

--select \* from PRODUCE;

insert into verifica values(index\_verificare.nextval,'CL8907',104921,1,4,2,2,to\_date('10.09.2023','dd.mm.yyyy'));

insert into verifica values(index\_verificare.nextval,'CL4201',141042,1,3,3,5,to\_date('08.07.2022','dd.mm.yyyy'));

insert into verifica values(index\_verificare.nextval,'CL3172',190391,2,1,4,5,to\_date('02.09.2019','dd.mm.yyyy'));

insert into verifica values(index\_verificare.nextval,'CL1411',190391,2,5,5,5,to\_date('03.04.2009','dd.mm.yyyy'));

insert into verifica values(index\_verificare.nextval,'CL7312',194821,3,4,4,3,to\_date('04.07.2010','dd.mm.yyyy'));

insert into verifica values(index\_verificare.nextval,'CL1301',947821,4,2,3,4,to\_date('02.06.2012','dd.mm.yyyy'));

insert into verifica values(index\_verificare.nextval,'CL7221',947821,3,1,4,5,to\_date('11.09.2014','dd.mm.yyyy'));

insert into verifica values(index\_verificare.nextval,'CL7312',104921,3,4,4,5,to\_date('12.07.2014','dd.mm.yyyy'));

insert into verifica values(index\_verificare.nextval,'CL0001',104921,5,4,4,2,to\_date('14.05.2016','dd.mm.yyyy'));

insert into verifica values(index\_verificare.nextval,'CL8412',141042,5,1,2,3,to\_date('16.04.2015','dd.mm.yyyy'));

insert into verifica values(index\_verificare.nextval,'CL7221',141042,1,4,1,1,to\_date('17.06.2020','dd.mm.yyyy'));

insert into verifica values(index\_verificare.nextval,'CL7312',947821,4,2,3,3,to\_date('10.03.2021','dd.mm.yyyy'));

insert into verifica values(index\_verificare.nextval,'CL0001',141042,4,4,4,2,to\_date('11.07.2022','dd.mm.yyyy'));

insert into verifica values(index\_verificare.nextval,'CL8412',141042,4,5,5,5,to\_date('26.08.2023','dd.mm.yyyy'));

insert into verifica values(index\_verificare.nextval,'CL3172',194821,2,3,2,4,to\_date('30.06.2023','dd.mm.yyyy'));

insert into verifica values(index\_verificare.nextval,'CL1411',190391,4,2,3,4,to\_date('10.04.2018','dd.mm.yyyy'));

insert into verifica values(index\_verificare.nextval,'CL7312',194821,4,2,3,3,to\_date('18.06.2017','dd.mm.yyyy'));

insert into verifica values(index\_verificare.nextval,'CL1301',104921,4,1,2,2,to\_date('25.07.2014','dd.mm.yyyy'));

insert into verifica values(index\_verificare.nextval,'CL7221',947821,2,5,2,5,to\_date('27.08.2013','dd.mm.yyyy'));

--select \* from VERIFICA;

# 12.Cereri SQL

* să se afișeze numele si prenumele apicultorilor care sunt înregistrați de înainte de 2011 si au angajat cel puțin 3 angajați

select concat(concat(nume\_apicultor,' '),prenume\_apicultor) numeApicultor, count(a.id\_angajat) nrAngajati

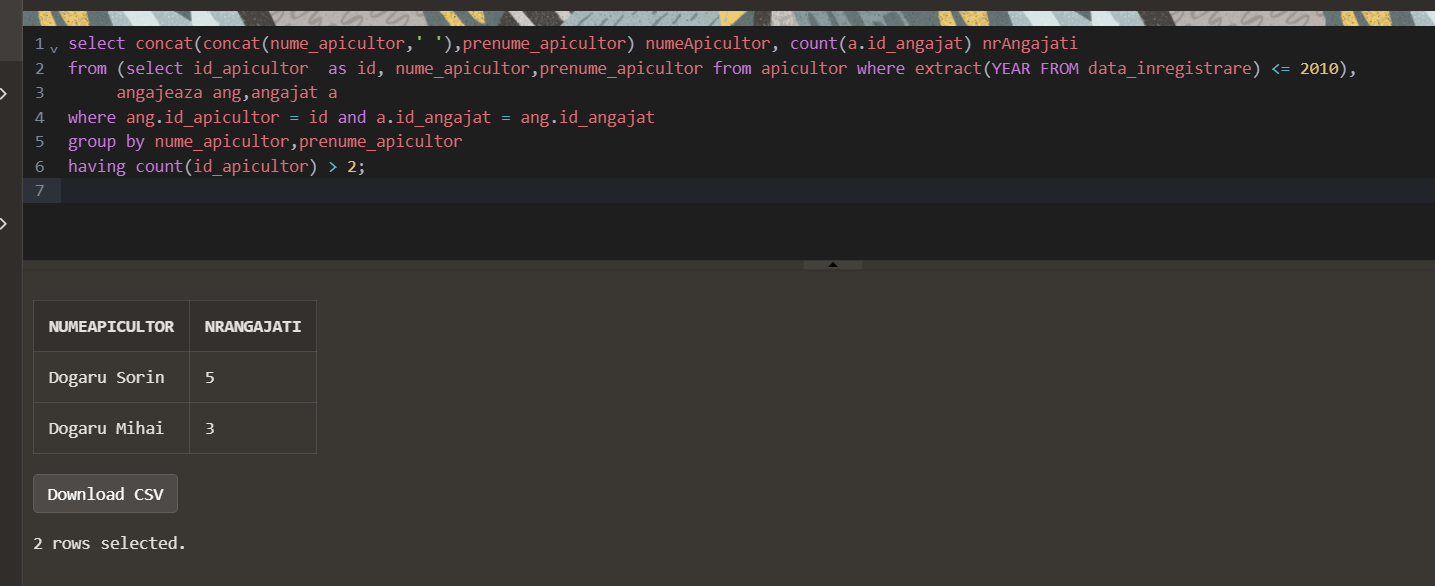
from (select id\_apicultor as id, nume\_apicultor,prenume\_apicultor from apicultor where extract(YEAR FROM data\_inregistrare) <= 2010),

angajeaza ang,angajat a

where ang.id\_apicultor = id and a.id\_angajat = ang.id\_angajat

group by nume\_apicultor,prenume\_apicultor

having count(id\_apicultor) > 2;



Elemente utilizate:

* funcție pe șiruri de caractere: concat()
* funcție pe date calendaristice: extract()
* cerere nesincronizată în clauza FROM
* funcția de agregare: count()
* culoarea(explicit) fiecărei mătci care are id-ul conținând A, iar stupul din care face parte are o medie de producție de miere >= 3

with stupiBuni as(

select v.id\_stup id\_Stup,avg(v.productie\_miere) miere

from verifica v

group by v.id\_Stup

having avg(v.productie\_miere) >= 3

)

select s.id\_stup,s.miere,m.id\_matca,case m.culoare\_marcaj

when 'a' then 'alb'

when 's' then 'albastru'

when 'g' then 'galben'

when 'v' then 'verde'

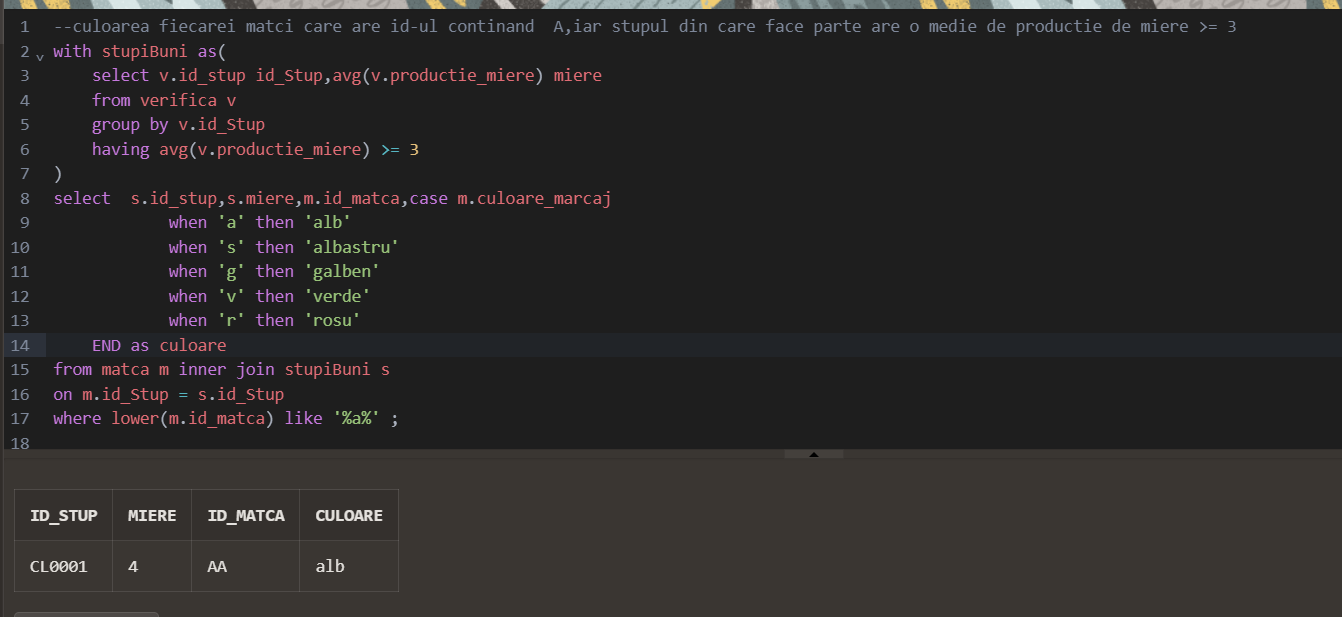
when 'r' then 'rosu'

END as culoare

from matca m inner join stupiBuni s

on m.id\_Stup = s.id\_Stup

where lower(m.id\_matca) like '%a%' ;



Elemente utilizate:

* bloc de cerere WITH
* expresie CASE
* funcție pe șiruri de caractere: lower()
* functia de agregare: avg()
* să se afișeze numele , postul, anii de lucru la stupină și angajatorul angajaților cu salariul mai mare decât media

select (select a.nume\_angajat||' '||a.prenume\_angajat from angajat a

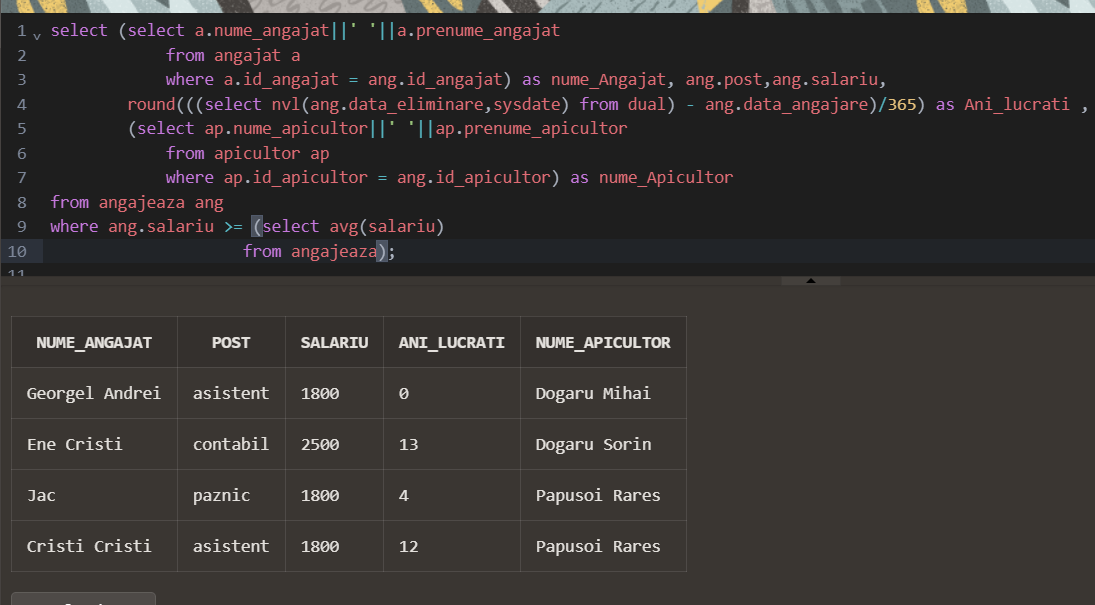
where a.id\_angajat = ang.id\_angajat) as nume\_Angajat, ang.post,ang.salariu,round(((select nvl(ang.data\_eliminare,sysdate) from dual) - ang.data\_angajare)/365) as Ani\_lucrati ,(select ap.nume\_apicultor||' '||ap.prenume\_apicultor

from apicultor ap

where ap.id\_apicultor = ang.id\_apicultor) as nume\_Apicultor

from angajeaza ang

where ang.salariu >= (select avg(salariu) from angajeaza);



Elemente utilizate:

* subcereri sincronizate în care intervin 3 tabele
* operatorul de concatenare
* funcția de agregare: avg()
* funcția de date: sysdate
* funcția NVL
* să se afișeze codul apicultorilor și numărul de stupi verificați daca este mai mare decât 1 și stupii au numărul < 500

select id\_apicultor,count(id\_verificare) as nr

from verifica

where id\_stup in (select id\_stup

from verifica

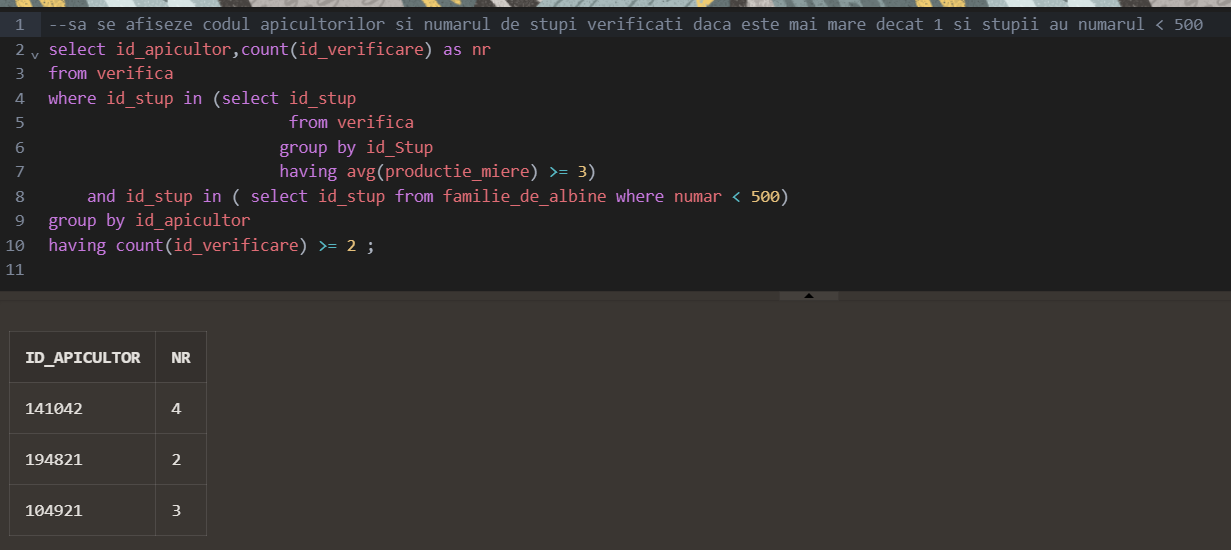
group by id\_Stup

having avg(productie\_miere) >= 3)

and id\_stup in ( select id\_stup from familie\_de\_albine where numar < 500)

group by id\_apicultor

having count(id\_verificare) >= 2 ;



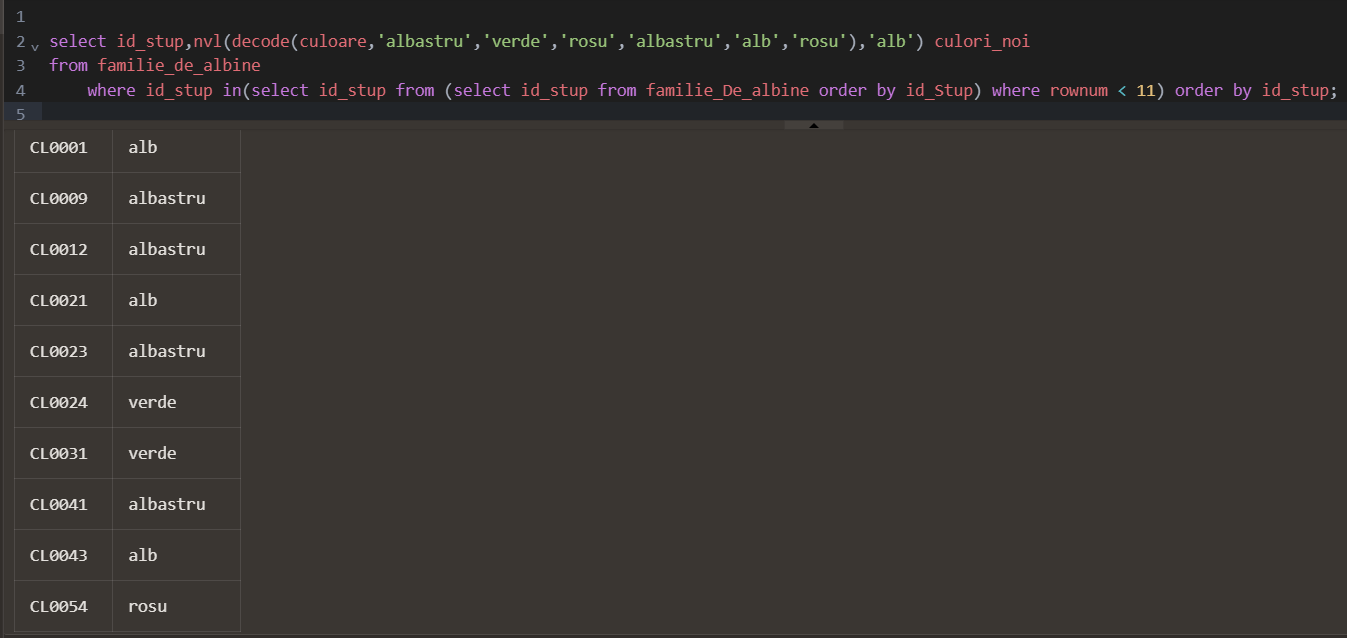
Elemente utilizate:

* subcereri corelate în care intervin 3 tabele
* funcții de agregare: count(), avg()
* filtrări de date la nivel de grup
* se ia în considerare ca primii 10 stupi să fie revopsiți, stupii albaștri vor deveni verzi, cei roșii vor fi albaștrii, iar cei albi vor fi roșii,toți ceilalți vor fi vopsiți cu alb. Cum vor arăta ulterior?

select id\_stup,nvl(decode(culoare,'albastru','verde','rosu','albastru','alb','rosu'),'alb') culori\_noi

from familie\_de\_albine

where id\_stup in(select id\_stup from (select id\_stup from familie\_De\_albine order by id\_Stup) where rownum < 11) order by id\_stup;



Elemente utilizate:

* funcțiile NVL și DECODE
* analiză top-n
* subcereri necorelate

# 13.Operații de actualizare și suprimare

* sa se mărească salariul tuturor angajaților de la paza cu 10% si lucrează intr-o pădure

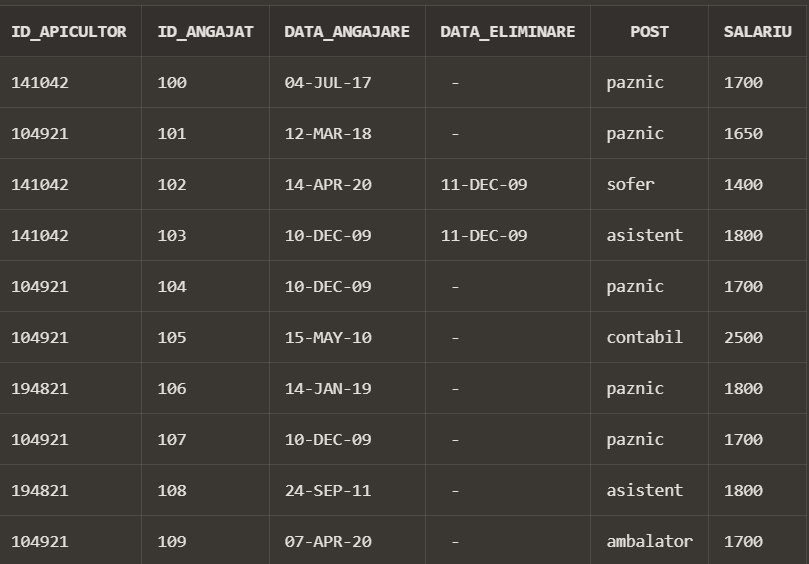
update angajeaza

set salariu = salariu \* 1.1

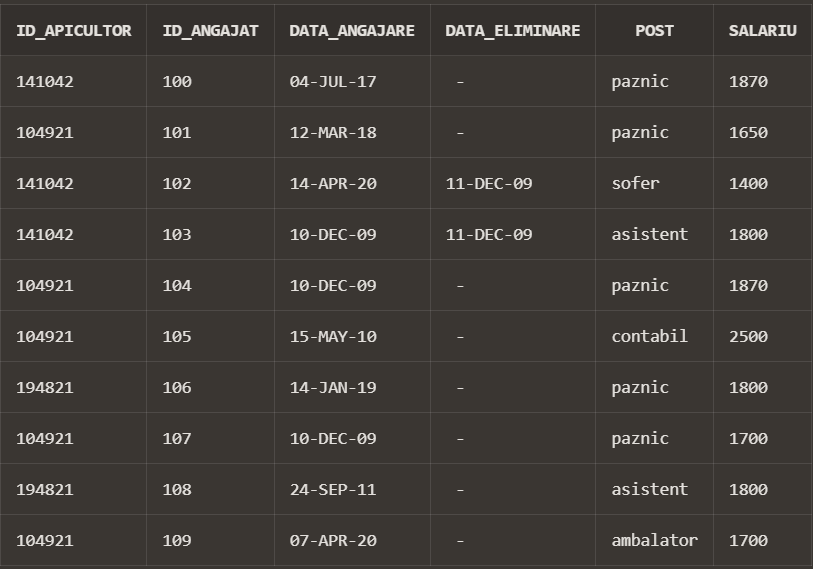
where post = 'paznic' and data\_eliminare is null

and id\_angajat in (select id\_angajat from angajat where cod\_locatie in (select l.cod\_locatie from locatie l where lower(l.nume\_loc) like '%padure%'));

INIȚIAL:



ULTERIOR:

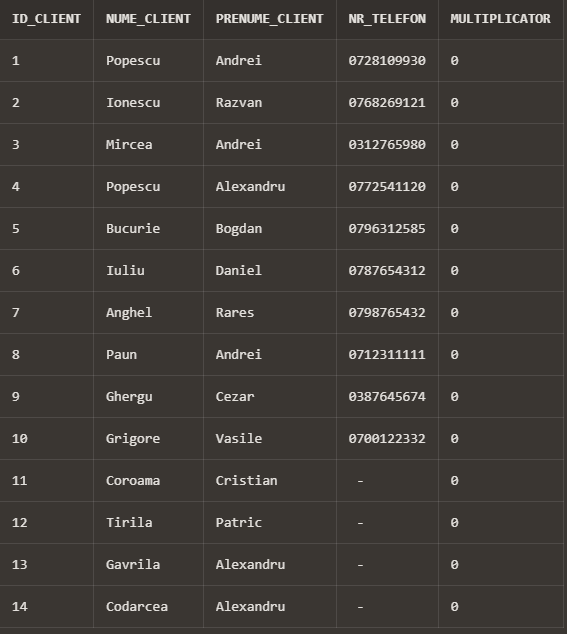
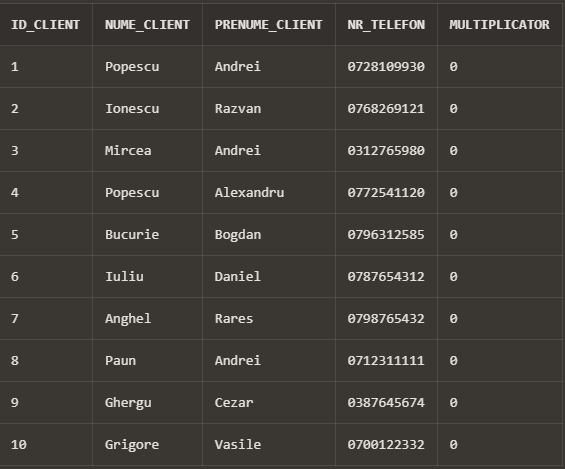


* sa se șteargă din clienții care nu au cumpărat nimic

delete from client

where id\_client not in (select unique id\_client from cumpara);

INIȚIAL: ULTERIOR:



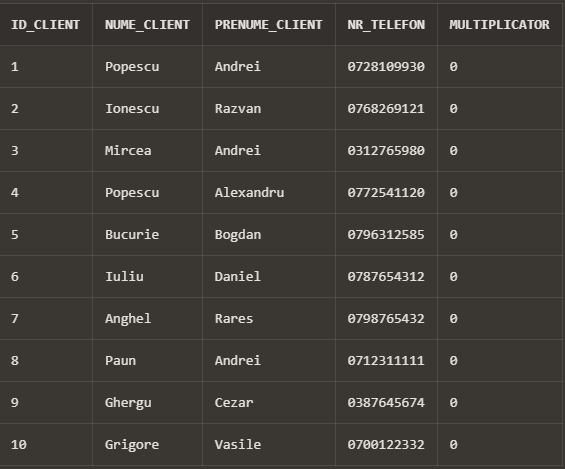
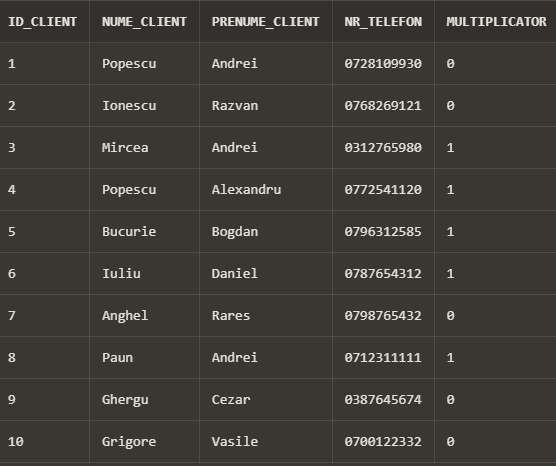
* să se dea valoarea 1 in câmpul de 'multiplicator' tuturor clienților care au cumpărat material biologic

update client set multiplicator = 1

where id\_client in (select unique c.id\_client

from client c,cumpara cum,produs p

where c.id\_client = cum.id\_client and p.cod\_produs = cum.cod\_produs and p.categorie = 'biologic');

INIȚIAL: ULTERIOR:

15.

* Analiza top-n : Să se afișeze top 3 produse vandute,codul și cantitatea

select \* from(

select p.nume\_produs as nume,c.cod\_produs as cod, sum(c.cantitate) as cantiate

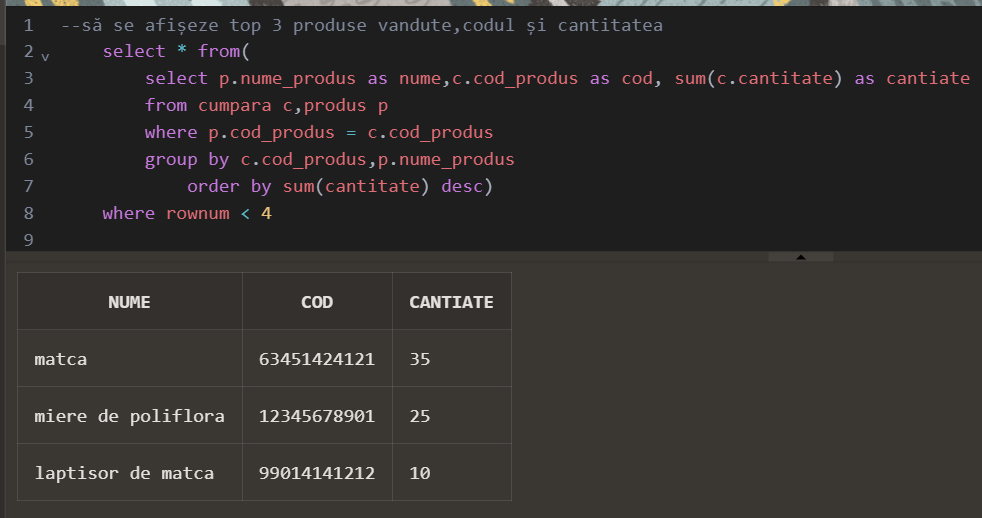
from cumpara c,produs p

where p.cod\_produs = c.cod\_produs

group by c.cod\_produs,p.nume\_produs

order by sum(cantitate) desc)

where rownum < 4



* Division: Să se găsească toți stupii care au fost verificați de toți apicultorii cu numele ‘Dogaru’

select distinct id\_stup

from verifica

MINUS

select id\_stup from (select distinct id\_Stup ,id\_apicultor

from (select distinct id\_stup from verifica),

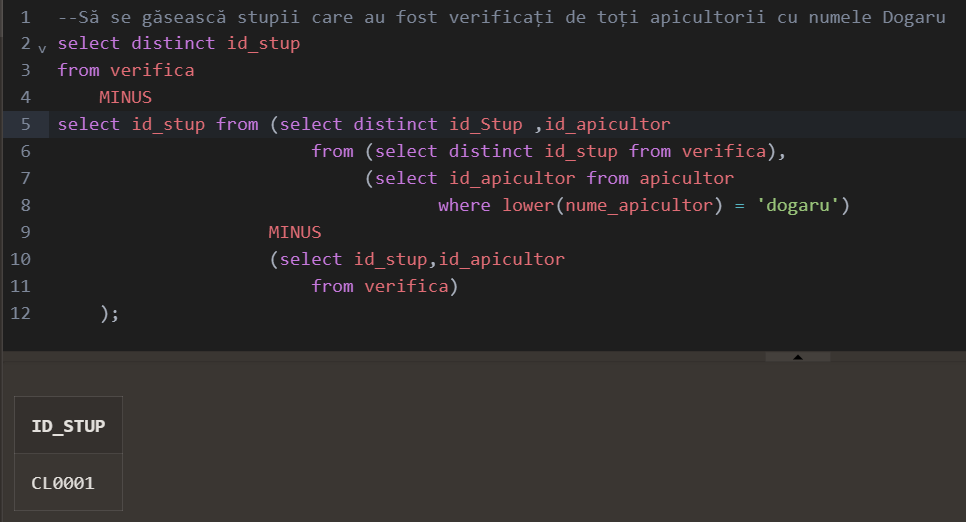
(select id\_apicultor from apicultor

where lower(nume\_apicultor) = 'dogaru')

MINUS

(select id\_stup,id\_apicultor from verifica)

);



* Outer-join: Câte produse a produs fiecare stup, ce culoare are matca sa(dacă are), când a fost verificat ultima dată(dacă a fost)

select fam.id\_Stup,

count(pro.id\_recoltare) as Nr\_prod\_Recoltate,

decode(ma.culoare\_marcaj,'a','alb','g','galben','r','rosu','v','verde','s','albastru','Nu are matca :(') as culoare\_matca,

nvl(to\_char(max(vf.data\_verificare)),'Neverificat') as ultima\_data

from familie\_De\_albine fam,produce pro,matca ma,verifica vf

where fam.id\_stup = pro.id\_Stup(+)

and fam.id\_Stup = ma.id\_stup(+)

and fam.id\_stup = vf.id\_stup(+)

group by fam.id\_Stup,decode(ma.culoare\_marcaj,'a','alb','g','galben','r','rosu','v','verde','s','albastru','Nu are matca :(');



|  |  |  |  |
| --- | --- | --- | --- |
| ID\_STUP | NR\_PROD\_RECOLTATE | CULOARE\_MATCA | ULTIMA\_DATA |
| CL7312 | 4 | Nu are matca :( | 10-Mar-21 |
| CL0056 | 1 | a | Neverificat |
| CL5302 | 0 | Nu are matca :( | Neverificat |
| CL0054 | 1 | Nu are matca :( | Neverificat |
| CL0110 | 0 | g | Neverificat |
| CL3501 | 0 | Nu are matca :( | Neverificat |
| CL0101 | 1 | g | Neverificat |
| CL8321 | 1 | Nu are matca :( | Neverificat |
| CL0201 | 1 | Nu are matca :( | Neverificat |
| CL6312 | 0 | Nu are matca :( | Neverificat |
| CL8806 | 0 | a | Neverificat |
| CL2401 | 1 | Nu are matca :( | Neverificat |
| CL0009 | 0 | s | Neverificat |
| CL0023 | 0 | s | Neverificat |
| CL1411 | 0 | Nu are matca :( | 10-Apr-18 |
| CL7221 | 0 | Nu are matca :( | 17-Jun-20 |
| CL0043 | 0 | Nu are matca :( | Neverificat |
| CL3101 | 1 | s | Neverificat |
| CL0381 | 0 | v | Neverificat |
| CL7410 | 0 | Nu are matca :( | Neverificat |
| CL9301 | 0 | Nu are matca :( | Neverificat |
| CL0090 | 0 | r | Neverificat |
| CL7553 | 0 | Nu are matca :( | Neverificat |
| CL7456 | 0 | Nu are matca :( | Neverificat |
| CL1223 | 0 | r | Neverificat |
| CL1002 | 0 | Nu are matca :( | Neverificat |
| CL8323 | 0 | Nu are matca :( | Neverificat |
| CL9807 | 0 | Nu are matca :( | Neverificat |
| CL6221 | 0 | Nu are matca :( | Neverificat |
| CL3172 | 0 | Nu are matca :( | 30-Jun-23 |
| CL9806 | 0 | v | Neverificat |
| CL1111 | 0 | Nu are matca :( | Neverificat |
| CL5444 | 0 | a | Neverificat |
| CL0041 | 0 | r | Neverificat |
| CL8907 | 1 | Nu are matca :( | 10-Sep-23 |
| CL0021 | 0 | g | Neverificat |
| CL7022 | 0 | v | Neverificat |
| CL1824 | 1 | Nu are matca :( | Neverificat |
| CL7442 | 0 | Nu are matca :( | Neverificat |
| CL0024 | 0 | Nu are matca :( | Neverificat |
| CL1301 | 2 | Nu are matca :( | 25-Jul-14 |
| CL0112 | 0 | Nu are matca :( | Neverificat |
| CL9423 | 1 | a | Neverificat |
| CL7323 | 0 | g | Neverificat |
| CL9109 | 1 | Nu are matca :( | Neverificat |
| CL0031 | 0 | v | Neverificat |
| CL0087 | 0 | r | Neverificat |
| CL0012 | 0 | a | Neverificat |
| CL7313 | 0 | Nu are matca :( | Neverificat |
| CL4201 | 1 | Nu are matca :( | 8-Jul-22 |