Proiect Baze de Date

Danciu Maryo-Razvan

Grupa 133

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

Modelul de date gestioneaza informatii legate de organizarea si buna functionare a unui Lant de Adaposturi pentru Caini. Consider ca acest model are o aplicabilitate reala si ar ajuta la optimizarea functionarii unei companii cu activitate similara, rezolvand anumite probleme ce tin de gestionarea cantitatilor mari de informatii. Acest lant are 5 locatii diferite in 5 orase diferite. Fiecare locatie are in grija un numar de caini, un numar de produse date spre vanzare si un numar de angajati. Fiecare locatie are un manager care supervizeaza intreaga activitate din zona respectiva. Angajatii au diferite functii care sunt impartite pe departamente. La fel ca locatiile, fiecare departament are un manager care are in evidenta progresul angajatilor subordonati. De altfel, produsele date spre vanzare care sunt de acelasi fel sunt grupate in alta tabela care contine informatii despre pretul, brand-ul sau numarul de produse disponibile din stoc.

1. **Prezentarea constrangerilor (restrictii, reguli) impuse asupra modelului.**

* O locatie are doar un singur adapost de caini, iar un adapost se afla intr-o singura locatie
* Un angajat poate lucra intr-o singura locatie, dar o locatie poate avea mai multi angajati
* Un adapost poate contine mai multi caini, dar un caine se poate afla intr-o singura locatie
* Un produs poate fi vandut dintr-o singura locatie, dar o locatie poate vinde mai multe produse
* Fiecare departament are un singur manager
* Fiecare locatie are un singur manager
* Un angajat poate avea un singur job

1. **Descrierea entitatilor**

Pentru modelul de date care gestioneaza un Lant de Adaposturi pentru Caini, structurile CAINI, ANGAJATI, DEPARTAMENTE, LOCATII, JOBS, PRODUSE si PRODUS reprezinta entitati.

Toate entitatile enumerate mai sus sunt entitati independente.

CAINI = contine informatii despre cainii care sunt in grija adposturilor. Cheia primara este #dog\_id.

ANGAJATI = contine informatii despre persoanele angajate din fiecare locatie. Cheia primara este #ang\_id.

DEPARTAMENTE = contine infromatii despre departamentele existente. Fiecare job apartine unui departament, astfel, o persoana cu un anumit job va face parte din departamentul respectiv acelui job. Cheia primara este #dep\_id.

JOBS = contine informatii despre joburile existente in acest Lant de Adaposturi. Cheia primara este #job\_id.

PRODUSE = contine informatii despre un grup de produse de acelasi tip: pretul acestora, brand-ul sau numarul de produse disponibil pe stoc.

PRODUS = contine informatii despre fiecare produs existent in parte: locatia in care poate fi vandut, data fabricarii, data expirarii si culoarea.

CLIENTI = contine informatii despre fiecare client in parte. Cheia primara este #cl\_id.

TRANZACTII = contine informatii despre fiecare tranzactie in parte: ce produs a fost cumparat, de cine a fost cumparat, unde si cine l-a vandut. Cheia primara este #tr\_id.

ADOPTII = similar cu tabela TRANZACTII, dar in loc de produse, avem cainii care sunt adoptati. Cheia primara este #ad\_id

1. **Descrierea relatiilor**

CAINE\_SE\_AFLA\_IN\_LOCATIE = relatie care leaga entitatile LOCATII de CAINI si reda legatura dintre acestea(pentru fiecare locatie, exista un numar de caini care sunt ingrijiti). Relatia are cardinalitate minima 1:0 si maxima 1:n.

PRODUS\_SE\_AFLA\_IN\_LOCATIE = relatie care leaga entitatile LOCATII de PRODUS si reda legatura dintre acestea(pentru fiecare locatie, exista un numar de produse date spre vanzare). Relatia are cardinalitate 1:0 minima si maxima 1:n.

TRANZACTIE\_ESTE\_FACUTA\_LA\_LOCATIE = relatie care leaga entitatile LOCATII de TRANZACTII si reda legatura dintre acestea (pentru fiecare tranzactie, exita o locatie unde aceasta a avut loc). Relatia are cardinalitate 1:0 si maxima 1:n.

ADOPTIE\_ESTE\_FACUTA\_LA\_LOCATIE = relatie care leaga entitatile LOCATII de ADOPTII si reda legatura dintre acestea (pentru fiecare adoptie, exista o locatie unde aceasta a avut loc). Relatia are cardinalitate 1:0 si maxima 1:n.

CLIENT\_CUMPARA\_DIN\_LOCATIE = relatie care leaga entitatile LOCATII de CLIENTI si reda legatura dintre acestea (pentru fiecare client, exista o locatie de unde aceasta cumpara mereu). Relatia are cardinalitate 1:0 si maxima 1:n.

ANGAJAT\_LUCREAZA\_IN\_LOCATIE = relatie care leaga entitatile LOCATII de ANGAJAT si reda legatura dintre acestea(pentru fiecare locatie, exista un numar de angajati). Relatia are cardinalitate minima 1:0 si maxima 0:n.

ANGAJAT\_ARE\_JOB = relatie care leaga entitatile ANGAJATI de JOBS si reda legatura dintre acestea(fiecare angajat are un job). Relatia are cardinalitate minima 1:1 si maxima 1:1

ANGAJAT\_LUCREAZA\_IN\_DEPARTAMENT = relatie care leaga entitatile ANGAJATI de DEPARTAMENTE si reda legatura dintre acestea (fiecare angajat face parte dintr-un departament). Relatia are cardinalitate minima 1:1 si maxima 1:1.

ANGAJAT\_FACE\_TRANZACTIE = relatie care leaga entitatile ANGAJATI de TRANZACTII si reda legatura dintre acestea(fiecare tranzactie este intermediata de un angajat). Relatia are cardinalitate minima 1:1 si maxima 1:1

ANGAJAT\_FACE\_ADOPTIE = relatie care leaga entitatile ANGAJATI de ADOPTII si reda legatura dintre acestea(fiecare adoptie este intermediata de un angajat). Relatia are cardinalitate minima 1:1 si maxima 1:1

CLIENT\_FACE\_TRANZACTIE = relatie care leaga entitatile CLIENTI de TRANZACTII si reda legatura dintre acestea(fiecare tranzactie este facuta de un client). Relatia are cardinalitate minima 1:n si maxima 1:n

CLIENT\_FACE\_ADOPTIE = relatie care leaga entitatile CLIENTI de ADOPTII si reda legatura dintre acestea(fiecare adoptie este facuta de un client). Relatia are cardinalitate minima 1:n si maxima 1:n

CAINE\_ESTE\_IMPLICAT\_IN\_ADOPTIE = relatie care leaga entitatile CAINI de ADOPTII si reda legauta dintre acestea (fiecare adoptie are un caine implicat). Relatia are cardinalitate minima 1:1 si maxima 1:1

PRODUS\_ESTE\_IMPLICAT\_IN\_TRANZACTIE = relatie care leaga entitatile PRODUSE de TRANZACTII si reda legauta dintre acestea (fiecare tranzactie are un produs implicat). Relatia are cardinalitate minima 1:1 si maxima 1:n

MANAGER\_CONDUCE\_DEPARTAMENT = relatie care leaga entitatile ANGAJATI de DEPARTAMENTE si reda legatura dintre acestea (pentru fiecare department exista un singur manager). Relatia are cardinalitate minima 1:1 si maxima 1:1.

MANAGER\_CONDUCE\_LOCATIE = relatie care leaga entitatile ANGAJATI de DEPARTAMENTE si reda legatura dintre acestea (pentru fiecare locartie exista un singur manager). Relatia are cardinalitate minima 1:1 si maxima 1:1.

PRODUS\_FACE\_PARTE\_DIN\_PRODUSE = relatie care leaga entitatile PRODUS de PRODUSE si reda legatura dintre acestea (fiecare produs face parte dintr-un grup de produse la fel ca acesta). Relatia are cardinalitate minima 1:1 si maxima 1:1.

1. **Descrierea atributelor**

Entitatea CAINI are ca attribute:

* dog\_id: variabila de tip intreg, de lungima maxima 4, cheie primara
* name: variabila de tip caracter, de lungime maxima 20, constrangere not null, reprezinta numele cainelui
* sex: variabila de tip caracter, de lungime maxima 20, constrangere not null, reprezinta sexul cainelui
* age: variabila de tip intreg, de lungime maxima 4, constrangere not null, reprezinta varsta cainelui
* join\_date: variabila de tip date, reprezinta data la care cainele a fost luat in ingrijire de catre un adapost
* location\_id: variabila de tip intreg, de lungime maxima 4, constrangere not null, reprezinta id-ul locatiei adapostului in care este un caine ingrijit
* rasa: variabila de tip caracter, de lungime maxima 20, constrangere not null, reprezinta rasa cainelui

Entitatea ANGAJATI are ca attribute:

* ang\_id: variabila de tip intreg, de lungime maxima 4, cheie primara
* last\_name: variabila de tip caracter, de lungime maxima 20, constrangere not null, reprezinta numele de familie al angajatului
* first\_name: variabila de tip caracter, de lungime maxima 20, constrangere not null, reprezinta prenumele angajatului
* email: variabila de tip caracter, de lungime maxima 50, reprezinta email-ul angajatului
* phone\_number: variabila de tip caracter, de lungime maxima 20, reprezinta numarul de telefon al angajatului
* hire\_date: variabila de tip date, reprezinta data la care a fost angajata o persoana
* job\_id: variabila de tip intreg, de lungime maxima 4, reprezinta id-ul job-ului pe care il are angajatul
* dep\_id: variabila de tip intreg, de lungime maxima 4, reprezinta id-ul departamentului din care face parte angajatul
* salary: variabila de tip intreg, de lungime maxima 4, reprezinta salariul pe care il are angajatul
* location\_id: variabila de tip intreg, de lungime maxima 4, reprezinta id-ul locatiei in care lucreaza angajatul
* boss\_id: variabila de tip intreg, de lungime maxima 4, reprezinta id-ul managerului din deparamentul unde lucreaza angajatul

Entitatea DEPARTAMENTE are ca atribute:

* dep\_id: variabila de tip intreg, de lungime maxima 4, cheie primara
* dep\_name: variabila de tip caracter, de lungime maxima 20, constrangere not null, reprezinta numele departamentului
* manager\_id: variabila de tip intreg, de lungime maxima 4, constrangere not null, reprezinta id-ul managerului departamentului
* found\_date: variabila de tip data, constrangere not null, reprezinta data fondarii departamentului

Entitatea LOCATII are ca attribute:

* location\_id: variabila de tip intreg, de lungime maxima 4, cheie primara
* street\_name: variabila de tip caracter, de lungime maxima 20, reprezinta numele strazii locatiei
* city: variabila de tip caracter, de lungime maxima 20, reprezinta numele orasului locatiei
* phone\_numer: variabila de tip caracter, de lungime maxima 20, reprezinta numarul de telefon al locatiei
* email: variabila de tip caracter, de lungime maxima 50, reprezinta email-ul locatiei
* manager\_id: variabila de tip intreg, de lungime maxima 4, reprezinta id-ul managerului locatiei

Entitatea JOBS are ca attribute:

* job\_id: variabila de tip intreg, de lungime maxima 4, cheie primara
* job\_title: variabila de tip caracter, de lungime maxima 20, constrangere not null, reprezinta numele job-ului
* min\_salary: variabila de tip intreg, de lungime maxima 4, constrangere not null, reprezinta salariul minim pe care il poate avea un angajat cu jobul respective
* max\_salary: variabila de tip intreg, de lungime maxima 4, reprezinta salariul maxim pe care il poate avea un angajat cu jobul respective

Entitatea PRODUSE are ca attribute:

* group\_id: variabila de tip intreg, de lungime maxima 4, cheie primara
* prods\_name: variabila de tip caracter, de lungime maxima 20, reprezinta numele grupului de produse
* pret: variabila de tip intreg, de lungime maxima 4, reprezinta pretul grupului de produse
* nr\_stoc: variabila de tip intreg, de lungime maxima 4, constrangere not null, reprezinta numarul de produse disponibile pe stoc din acel grup
* brand\_name: variabila de tip caracter, de lungime maxima 4, reprezinta numele brandului grupului de produse

Entitatea PRODUS are ca attribute:

* prod\_id: variabila de tip intreg, de lungime maxima 4, cheie primara
* group\_id: variabila de tip intreg, de lungime maxima 4, reprezinta id-ul grupuilui de produse din care face produsul
* location\_id: variabila de tip intreg, de lungime maxima 4, reprezinta id-ul locatiei in care se vinde produsul
* fab\_date: variabila de tip date, constrangere not null, reprezinta data fabricarii produsului
* exp\_date: variabila de tip date, reprezinta data expirarii produsului
* color: variabila de tip caracter, de lungime maxima 20, reprezinta culoarea produsului

Entitatea TRANZACTII are ca attribute:

* tr\_id: variabila de tip intreg, de lungime maxima 4, cheie primara
* cl\_id: variabila de tip intreg, de lungime maxima 4, reprezinta id-ul clientului care face tranzactia
* ang\_id: variabila de tip intreg, de lungime maxima 4, reprezinta id-ul anagajatului care intermediaza tranzactia
* location\_id: variabila de tip intreg, de lungime maxima 4, reprezinta id-ul locatiei unde se face tranzactia
* group\_id: variabila de tip intreg, de lungime maxima 4, reprezinta id-ul grupului de produse care este vandut in tranzactie
* purchase\_date: variabila de tip date, reprezinta data la care este facuta tranzactia
* quantity: variabila de tip intreg, de lungime maxima 4, reprezinta cantitatea de produse cumparate

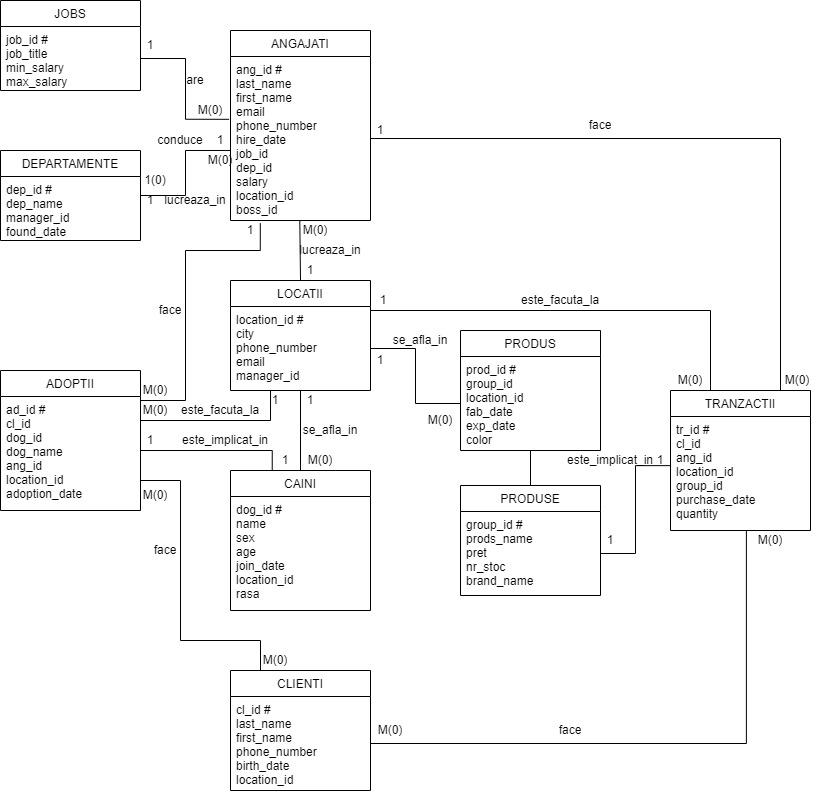
Entitatea CLIENTI are ca attribute:

* cl\_id: variabila de tip intreg, de lungime maxima 4, cheie primara
* last\_name: variabila de tip caracter, reprezinta numele de familie al clientului
* first\_name: variabila de tip caracter, reprezinta prenumele clientului
* phone\_number: variabila de tip caracter, reprezinta numarul de telefon al clientului
* birth\_date: variabila de tip date, reprezinta data de nastere a clientului
* location\_id: variabila de tip intreg, de lungime maxima 4, reprezinta locatia de unde clientul face cumparaturi

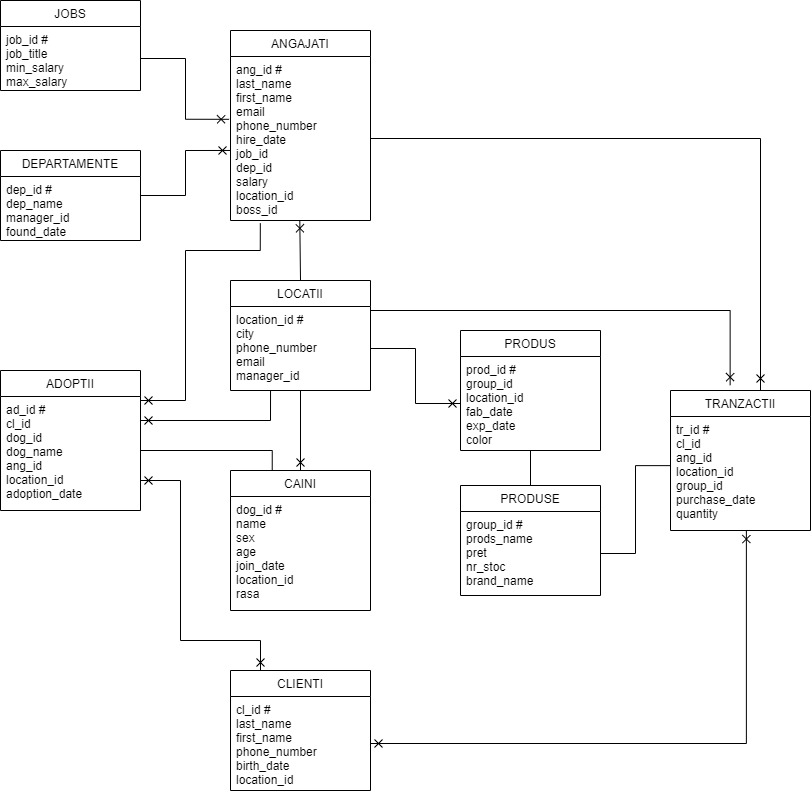
Entitatea ADOPTII are ca attribute:

* ad\_id: variabila de tip intreg, de lungime maxima 4, cheie primara
* cl\_id: variabila de tip intreg, de lungime maxima 4, reprezinta id-ul clientului care face adoptia
* dog\_id: variabila de tip intreg, de lungime maxima 4, reprezinta id-ul cainelui care este implicat in adoptie
* dog\_name: variabila de tip caracter, reprezinta numele cainelui care este implicat in adoptie
* ang\_id: variabila de tip intreg, de lungime maxima 4, reprezinta id-ul anagajatului care intermediaza adoptia
* location\_id: variabila de tip intreg, de lungime maxima 4, reprezinta id-ul locatiei unde se face adoptia
* adoption\_date: variabila de tip date, reprezinta data la care este facuta adoptia

1. **Diagrama E/R**

****

1. **Diagrama conceptuala**

****

1. **Schemele relationale**

Schemele relationale corespunzatoare diagramei conceptuale sunt urmatoarele:

CAINI(dog\_id#, name, sex, age, join\_date, location\_id)

ANGAJATI(ang\_id#, first\_name, last\_name, email, hire\_date, job\_id, dep\_id, salary, location\_id)

DEPARTAMENTE(dep\_id#, dep\_name, found\_date, manager\_id)

LOCATII(location\_id#, street\_name, city)

JOBS(job\_id#, job\_title, min\_salary, max\_salary)

PRODUSE(group\_id#, nr\_stoc, brand\_id)

PRODUS(prod\_id#, group\_id, location\_id, exp\_date, fab\_date, color)

CLIENTI(cl\_id#, last\_name, first\_name, phone\_number, birth\_date, location\_id)

TRANZACTII(tr\_id#, cl\_id, ang\_id, location\_id, group\_id, purchase\_date, quantity)

ADOPTII(ad\_id#, cl\_id, dog\_id, dog\_name, ang\_id, location\_id, adoption\_date)

1. **Normalizarea(la final)**
2. **Crearea unei secvente**

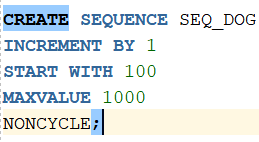
CREATE SEQUENCE SEQ\_DOG

INCREMENT BY 1

START WITH 100

MAXVALUE 1000

NONCYCLE;



insert into caini (dog\_id, name, sex, age, join\_date, location\_id, rasa)

values (seq\_dog.nextval, 'Buzz', 'M', 1, to\_date('15.07.2020','DD.MM.YYYY'), 201, 'Bulldog');

insert into caini (dog\_id, name, sex, age, join\_date, location\_id, rasa)

values (seq\_dog.nextval, 'Maggie', 'f', 10, to\_date('18.05.2014','DD.MM.YYYY'), 202, 'Golden Retriever');

insert into caini (dog\_id, name, sex, age, join\_date, location\_id, rasa)

values (seq\_dog.nextval, 'Jumbo', 'M', 8, to\_date('27.04.2019','DD.MM.YYYY'), 203, 'Pug');

1. **Crearea tabelelor si inserarea datelor**
2. **CAINI – inserare cu secventa**

create table caini(

dog\_id number(4),

name varchar(20) constraint null\_nume not null,

sex varchar(20) constraint null\_sex not null,

age number(4) constraint null\_age not null,

join\_date date default sysdate,

location\_id number(4) constraint null\_loc not null,

rasa varchar(20) constraint null\_rasa not null,

constraint pk\_caini primary key (dog\_id)

);

insert into caini (dog\_id, name, sex, age, join\_date, location\_id, rasa)

values (SEQ\_DOG.nextval, 'Buzz', 'M', 1, to\_date('15.07.2020','DD.MM.YYYY'), 201, 'Bulldog');

insert into caini (dog\_id, name, sex, age, join\_date, location\_id, rasa)

values (SEQ\_DOG.nextval, 'Maggie', 'f', 10, to\_date('18.05.2014','DD.MM.YYYY'), 202, 'Golden Retriever');

insert into caini (dog\_id, name, sex, age, join\_date, location\_id, rasa)

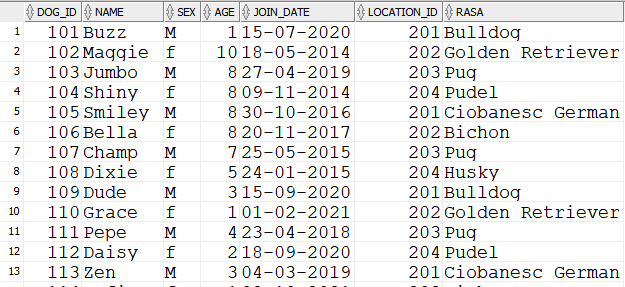
values (SEQ\_DOG.nextval, 'Jumbo', 'M', 8, to\_date('27.04.2019','DD.MM.YYYY'), 203, 'Pug');

insert into caini (dog\_id, name, sex, age, join\_date, location\_id, rasa)

values (SEQ\_DOG.nextval, 'Shiny', 'f', 8, to\_date('09.11.2014','DD.MM.YYYY'), 204, 'Pudel');

insert into caini (dog\_id, name, sex, age, join\_date, location\_id, rasa)

values (SEQ\_DOG.nextval, 'Smiley', 'M', 8, to\_date('30.10.2016','DD.MM.YYYY'), 201, 'Ciobanesc German');

****

1. **ANGAJATI**

create table angajati(

ang\_id number(4),

last\_name varchar(20) constraint null\_last\_name not null,

first\_name varchar(20) constraint null\_first\_name not null,

email varchar(50),

phone\_number varchar(20),

hire\_date date default sysdate,

job\_id number(4),

dep\_id number(4),

salary number(4) constraint null\_sal not null,

location\_id number(4),

boss\_id number(4),

constraint pk\_ang primary key (ang\_id)

);

insert into angajati

(ang\_id, last\_name, first\_name, email, phone\_number, hire\_date, job\_id, dep\_id, salary, location\_id, boss\_id)

values (301, 'Bîrău', 'Victoria', 'vic.bir@dog.ro', '0762182783', to\_date('06.03.2014','DD.MM.YYYY'), 502, 401, 9500, 201, 310);

insert into angajati

(ang\_id, last\_name, first\_name, email, phone\_number, hire\_date, job\_id, dep\_id, salary, boss\_id)

values (302, 'Teodorescu', 'Adrian', 'ad.teo@dog.ro', '0744707212', to\_date('19.08.2014','DD.MM.YYYY'), 502, 401, 6500, 310);

insert into angajati

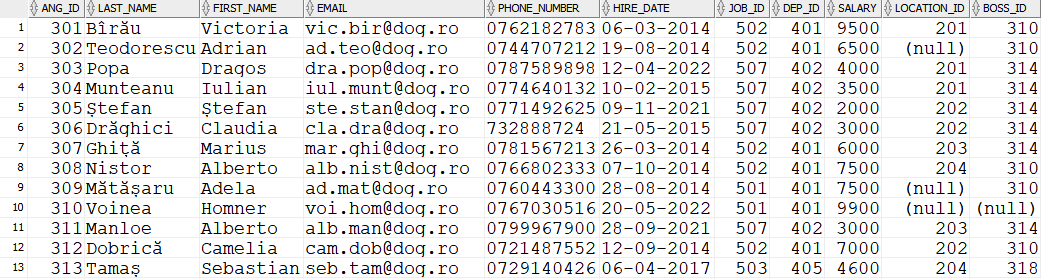
(ang\_id, last\_name, first\_name, email, phone\_number, hire\_date, job\_id, dep\_id, salary, location\_id, boss\_id)

values (303, 'Popa', 'Dragos', 'dra.pop@dog.ro', '0787589898', to\_date('12.04.2022','DD.MM.YYYY'), 507, 402, 4000, 201, 314);

insert into angajati

(ang\_id, last\_name, first\_name, email, phone\_number, hire\_date, job\_id, dep\_id, salary, location\_id, boss\_id)

values (304, 'Munteanu', 'Iulian', 'iul.munt@dog.ro', '0774640132', to\_date('10.02.2015','DD.MM.YYYY'), 507, 402, 3500, 201, 314);



1. **DEPARTAMENTE**

create table departamente(

dep\_id number(4),

dep\_name varchar(20) constraint null\_dep\_name not null,

manager\_id number(4) constraint null\_location\_id not null,

found\_date date default sysdate constraint null\_found not null,

constraint pk\_dep primary key (dep\_id)

);

insert into departamente (dep\_id, dep\_name, manager\_id, found\_date)

values (401, 'administrativ', 310, to\_date('01.02.2014','DD.MM.YYYY'));

insert into departamente (dep\_id, dep\_name, manager\_id, found\_date)

values (402, 'medical', 314, to\_date('10.04.2014','DD.MM.YYYY'));

insert into departamente (dep\_id, dep\_name, manager\_id, found\_date)

values (403, 'IT', 302, to\_date('18.02.2015','DD.MM.YYYY'));

insert into departamente (dep\_id, dep\_name, manager\_id, found\_date)

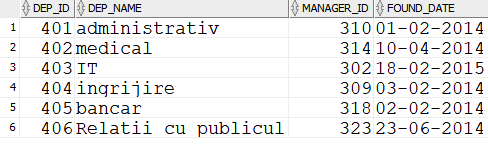
values (404, 'ingrijire', 309, to\_date('03.02.2014','DD.MM.YYYY'));

insert into departamente (dep\_id, dep\_name, manager\_id, found\_date)

values (405, 'bancar', 318, to\_date('02.02.2014','DD.MM.YYYY'));

insert into departamente (dep\_id, dep\_name, manager\_id, found\_date )

values (406, 'Relatii cu publicul', 323, to\_date('23.06.2014','DD.MM.YYYY'));



1. **LOCATII**

create table locatii(

location\_id number(4),

street\_name varchar(20),

city varchar(20),

phone\_number varchar(20),

email varchar(50),

manager\_id number(4),

constraint pk\_loaction primary key (location\_id)

);

insert into locatii (location\_id, street\_name, city, phone\_number, email, manager\_id)

values (201, 'Muncii', 'Foc?ani', '0762182783', 'dog.focsani@gmail.com', 301);

insert into locatii (location\_id, street\_name, city, phone\_number, email, manager\_id)

values (202, 'Decebal', 'Gala?i', '0721487552', 'dog.galati@gmail.com', 312);

insert into locatii (location\_id, street\_name, city, phone\_number, email, manager\_id)

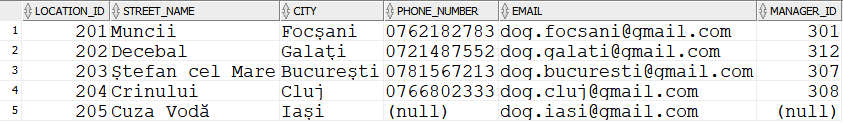
values (203, '?tefan cel Mare', 'Bucure?ti', '0781567213', 'dog.bucuresti@gmail.com', 307);

insert into locatii (location\_id, street\_name, city, phone\_number, email, manager\_id)

values (204, 'Crinului', 'Cluj', '0766802333', 'dog.cluj@gmail.com', 308);

insert into locatii (location\_id, street\_name, city, email)

values (205, 'Cuza Vodă', 'Ia?i', 'dog.iasi@gmail.com');



1. **JOBS**

create table jobs(

job\_id number(4),

job\_title varchar(20) constraint null\_job not null,

min\_salary number(4) constraint null\_min not null,

max\_salary number(4),

constraint pk\_job primary key (job\_id)

);

insert into jobs (job\_id, job\_title, min\_salary)

values (501, 'manager departamente', 4000);

insert into jobs (job\_id, job\_title, min\_salary)

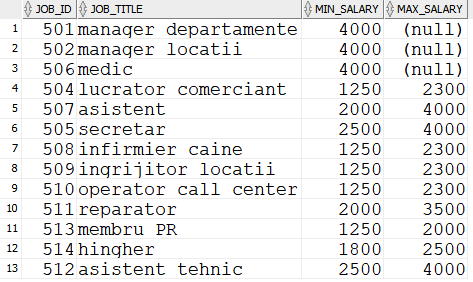
values (502, 'manager locatii', 4000);

insert into jobs (job\_id, job\_title, min\_salary)

values (506, 'medic', 4000);

insert into jobs (job\_id, job\_title, min\_salary, max\_salary)

values (503, 'contabil', 3000, 6000);



1. **PRODUSE**

create table produse(

group\_id number(4),

prods\_name varchar(20),

pret number(4),

nr\_stoc number(4) constraint null\_nr not null,

brand\_name varchar(20),

constraint pk\_group primary key (group\_id)

);

insert into produse

(group\_id, prods\_name, pret, nr\_stoc, brand\_name)

values (601, 'Pasta de dinti', 25, 100, 'Beaphar');

insert into produse

(group\_id, prods\_name, pret, nr\_stoc, brand\_name)

values (602, 'Masina de tuns', 200, 52, 'Moser');

insert into produse

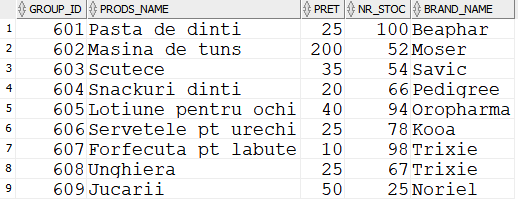
(group\_id, prods\_name, pret, nr\_stoc, brand\_name)

values (603, 'Scutece', 35, 54, 'Savic');

insert into produse

(group\_id, prods\_name, pret, nr\_stoc, brand\_name)

values (604, 'Snackuri dinti', 20, 66, 'Pedigree');



1. **PRODUS**

create table produs(

prod\_id number(4),

group\_id number(4),

location\_id number(4),

fab\_date date default sysdate constraint null\_fab not null,

exp\_date date default sysdate,

color varchar(20),

constraint pk\_prod primary key (prod\_id)

);

insert into produse

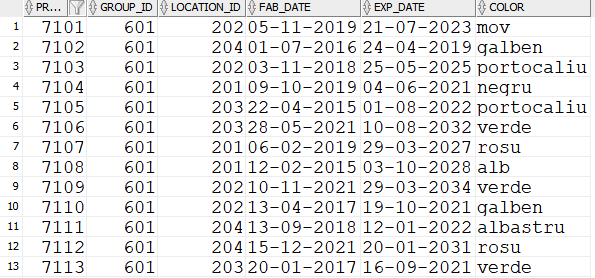
(group\_id, prods\_name, pret, nr\_stoc, brand\_name)

values (601, 'Pasta de dinti', 25, 100, 'Beaphar');

insert into produse

(group\_id, prods\_name, pret, nr\_stoc, brand\_name)

values (602, 'Masina de tuns', 200, 52, 'Moser');

****

1. **CLIENTI**

create table clienti(

cl\_id number(4),

last\_name varchar(20),

first\_name varchar(20),

phone\_number varchar(20),

birth\_date date default sysdate,

location\_id number(4),

constraint pk\_client primary key (cl\_id)

);

insert into clienti

(cl\_id, last\_name, first\_name, phone\_number, birth\_date, location\_id)

values (801, 'Podaru', 'Elena', '0751423621', to\_date('1984-10-14','YYYY-MM-DD'), 201);

insert into clienti

(cl\_id, last\_name, first\_name, phone\_number, birth\_date, location\_id)

values (802, 'Pasclaru', 'Miruna', '0712488233', to\_date('1990-01-06','YYYY-MM-DD'), 202);

insert into clienti

(cl\_id, last\_name, first\_name, phone\_number, birth\_date, location\_id)

values (803, 'Popescu', 'Ana-Maria', '0724615469', to\_date('1993-02-12','YYYY-MM-DD'), 203);

****

1. **Tranzactii**

create table tranzactii(

tr\_id number(4),

cl\_id number(4),

ang\_id number(4),

location\_id number(4),

group\_id number(4),

purchase\_date date default sysdate,

quantity number(4),

constraint pk\_tranz primary key (tr\_id)

);

insert into tranzactii

(tr\_id, cl\_id, ang\_id, location\_id, group\_id, purchase\_date, quantity)

values (901, 813, 328, 201, 609, to\_date('2020-10-14','YYYY-MM-DD'), 1);

insert into tranzactii

(tr\_id, cl\_id, ang\_id, location\_id, group\_id, purchase\_date, quantity)

values (902, 822, 329, 202, 605, to\_date('2016-01-06','YYYY-MM-DD'), 1);

insert into tranzactii

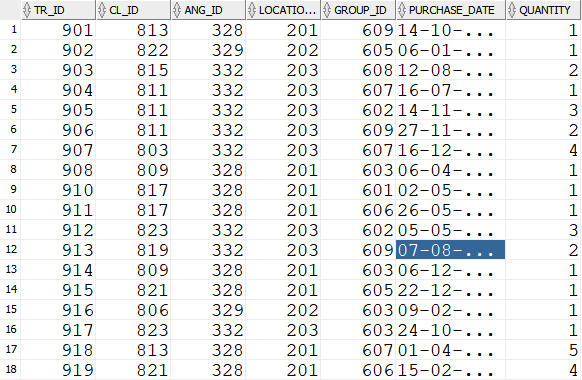
(tr\_id, cl\_id, ang\_id, location\_id, group\_id, purchase\_date, quantity)

values (903, 815, 332, 203, 608, to\_date('2021-08-12','YYYY-MM-DD'), 2);

insert into tranzactii

(tr\_id, cl\_id, ang\_id, location\_id, group\_id, purchase\_date, quantity)

values (904, 811, 332, 203, 607, to\_date('2017-07-16','YYYY-MM-DD'), 1);

****

1. **Adoptii**

create table adoptii (

ad\_id number(4),

cl\_id number(4),

dog\_id number(4),

dog\_name varchar(20),

ang\_id number(4),

location\_id number(4),

adoption\_date date default sysdate,

constraint pk\_adoption primary key (ad\_id)

);

insert into adoptii

(ad\_id, cl\_id, dog\_id, dog\_name, ang\_id, location\_id, adoption\_date)

values (1001, 813, 137, 'Darwin', 328, 201, to\_date('2019-07-30','YYYY-MM-DD'));

insert into adoptii

(ad\_id, cl\_id, dog\_id, dog\_name, ang\_id, location\_id, adoption\_date)

values (1002, 811, 131, 'Goofy', 332, 203, to\_date('2021-12-16','YYYY-MM-DD'));

insert into adoptii

(ad\_id, cl\_id, dog\_id, dog\_name, ang\_id, location\_id, adoption\_date)

values (1003, 817, 129, 'Dodo', 328, 201, to\_date('2018-05-26','YYYY-MM-DD'));

insert into adoptii

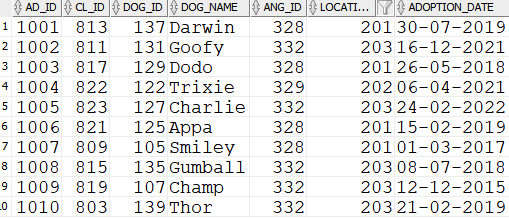
(ad\_id, cl\_id, dog\_id, dog\_name, ang\_id, location\_id, adoption\_date)

values (1004, 822, 122, 'Trixie', 329, 202, to\_date('2021-04-06','YYYY-MM-DD'));

insert into adoptii

(ad\_id, cl\_id, dog\_id, dog\_name, ang\_id, location\_id, adoption\_date)

values (1005, 823, 127, 'Charlie', 332, 203, to\_date('2022-02-24','YYYY-MM-DD'));

****

**CONSTRANGERI:**

alter table caini

add constraint fk\_caini\_loc\_id foreign key (location\_id) references locatii(location\_id);

alter table angajati

add constraint fk\_ang\_loc\_id foreign key (location\_id) references locatii (location\_id);

alter table locatii

add constraint fk\_loc\_man\_id foreign key (manager\_id) references angajati (ang\_id);

alter table angajati

add constraint fk\_ang\_job\_id foreign key (job\_id) references jobs (job\_id);

alter table angajati

add constraint fk\_ang\_dep\_id foreign key (dep\_id) references departamente (dep\_id);

alter table departamente

add constraint fk\_dep\_man\_id foreign key (manager\_id) references angajati (ang\_id);

alter table produse

add constraint fk\_prods\_prod\_id foreign key (prod\_id) references produs (prod\_id);

alter table produs

add constraint fk\_prod\_group\_id foreign key (group\_id) references produse (group\_id);

alter table produs

add constraint fk\_prod\_location\_id foreign key (location\_id) references locatii (location\_id);

alter table clienti

add constraint fk\_cl\_location\_id foreign key (location\_id) references locatii (location\_id);

alter table tranzactii

add constraint fk\_tranz\_cl\_id foreign key (cl\_id) references clienti (cl\_id);

alter table tranzactii

add constraint fk\_tranz\_ang\_id foreign key (ang\_id) references angajati (ang\_id);

alter table tranzactii

add constraint fk\_tranz\_location\_id foreign key (location\_id) references locatii (location\_id);

alter table tranzactii

add constraint fk\_tranz\_group\_id foreign key (group\_id) references produse (group\_id);

alter table adoptii

add constraint fk\_ad\_cl\_id foreign key (cl\_id) references clienti (cl\_id);

alter table adoptii

add constraint fk\_ad\_dog\_id foreign key (dog\_id) references caini (dog\_id);

alter table adoptii

add constraint fk\_ad\_ang\_id foreign key (ang\_id) references angajati (ang\_id);

alter table adoptii

add constraint fk\_ad\_location\_id foreign key (location\_id) references locatii (location\_id);

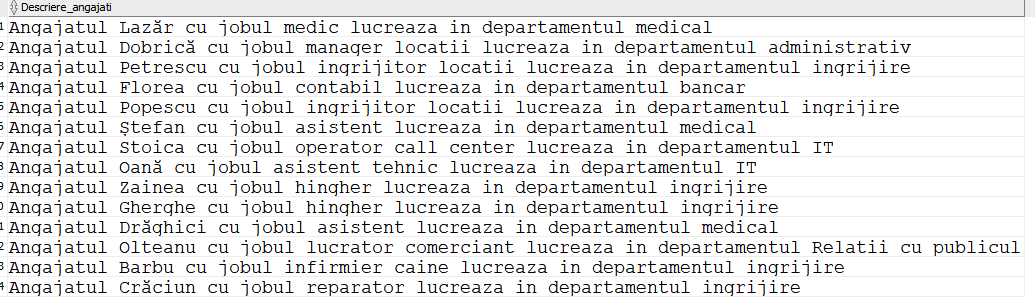
1. **Cereri complexe**
2. Sa se afiseze numele jobului, numele departamentului si numele de familie al angajatilor care lucreaza in orasul Galati. Sa se afiseze raspsurile intr-o singura propozitie

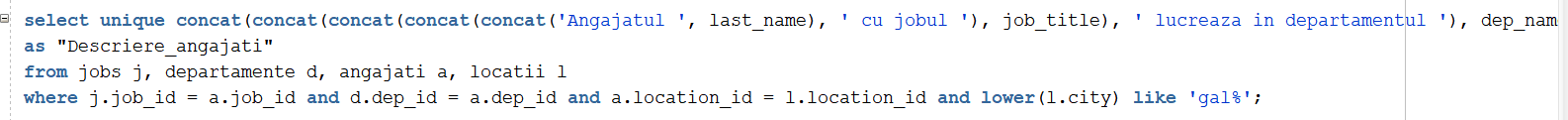
select unique concat(concat(concat(concat(concat('Angajatul ', last\_name), ' cu jobul '), job\_title), ' lucreaza in departamentul '), dep\_name)

as "Descriere\_angajati"

from jobs j, departamente d, angajati a, locatii l

where j.job\_id = a.job\_id and d.dep\_id = a.dep\_id and a.location\_id = l.location\_id and lower(l.city) like 'gal%';





1. Sa se afiseze cainii cu varsta peste varsta medie a tutror cainilor.

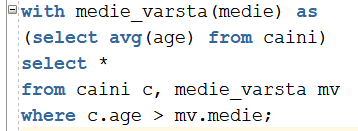
with medie\_varsta(medie) as

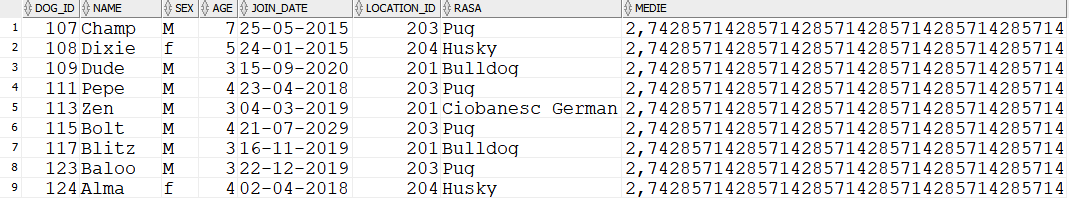
(select avg(age) from caini)

select \*

from caini c, medie\_varsta mv

where c.age > mv.medie;





1. Sa se afiseze in ordine cronologica, id-ul, numele, data la care a fost luat in grija si cate luni au trecut de la data respective pana la ultima zi din luna in care a fost luat in grija cel mai longeviv caine din locatiile unde se afla jucarii spre vanzare.

select dog\_id, name, join\_date,

months\_between(

join\_date,

last\_day((select min(join\_date) from

caini c, locatii l, produs p, produse po

where c.location\_id = l.location\_id and

l.location\_id = p.location\_id and

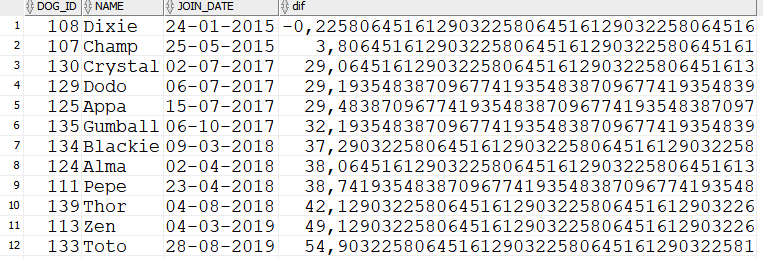
po.group\_id = p.group\_id

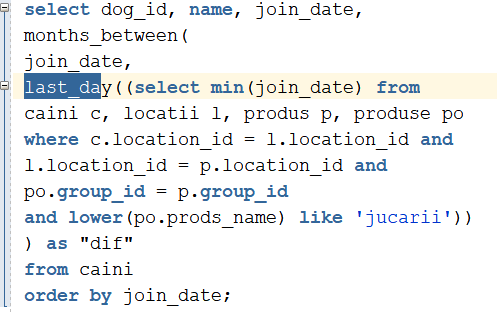
and lower(po.prods\_name) like 'jucarii'))

) as "dif"

from caini

order by join\_date;





1. Sa se afiseze id-ul si locatia produselor mai ieftine decat lotiune pentru ochi din Bucuresti.

De asemenea sa se afiseze “DA” daca culoarea produselor este roise si “NU” in caz contrar.

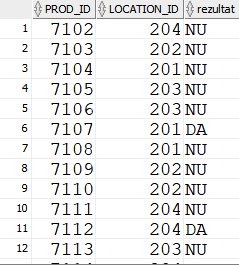
select prod\_id, p.location\_id, decode(p.color, 'rosu', 'DA', 'NU') as "rezultat"

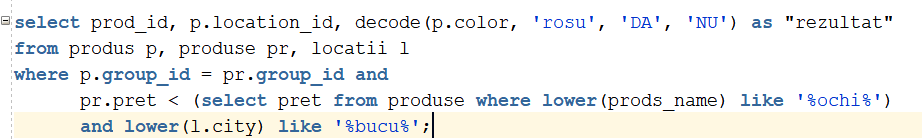
from produs p, produse pr, locatii l

where p.group\_id = pr.group\_id and

pr.pret < (select pret from produse where lower(prods\_name) like '%ochi%')

and lower(l.city) like '%bucu%';





1. Pentru fiecare locatie, sa se afiseze cate produse nu au culoare. Sa se spuna si daca numarul total de produse depaseste 130.

select p.location\_id, city, count(\*),

case

when (nvl(city, null)) is null then 'Peste 130 de produse'

else 'Sub 130 de produse'

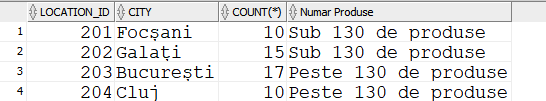
end as "Numar Produse"

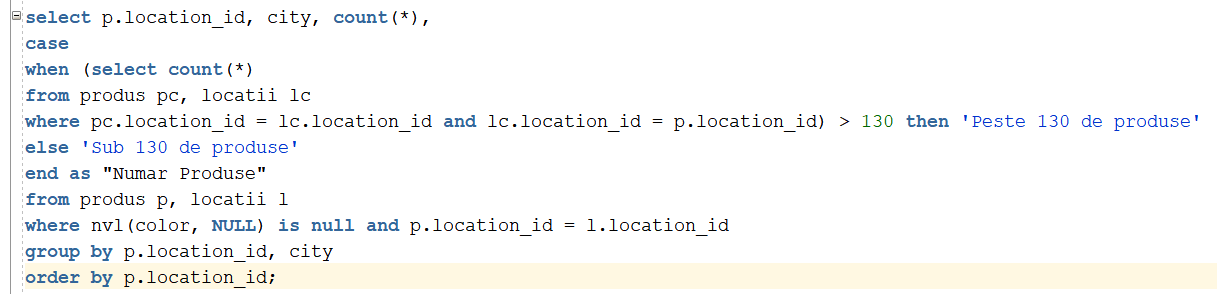
from produs p, locatii l

where nvl(color, NULL) is null and p.location\_id = l.location\_id

group by p.location\_id, city

order by p.location\_id;



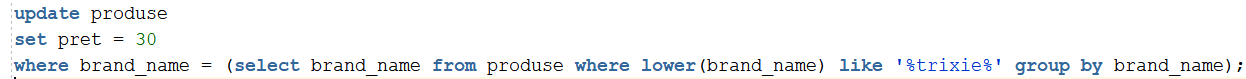


1. **Actualizare si suprimare**
2. Pretul tuturor produselor de la firma Trixie devine 30.

update produse

set pret = 30

where brand\_name = (select brand\_name from produse where lower(brand\_name) like '%trixie%' group by brand\_name);



1. Produsul vandut de cele mai multe ori primeste o reducere de 10 la suta.

update produse

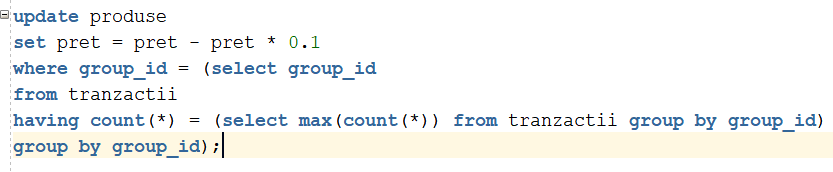
set pret = pret - pret \* 0.1

where group\_id = (select group\_id

from tranzactii

having count(\*) = (select max(count(\*)) from tranzactii group by group\_id)

group by group\_id);



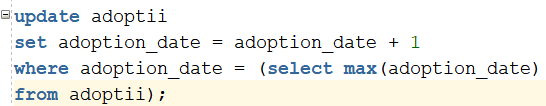
1. S-a produs o greseala si ultima adoptie a fost inregistrata cu o zi mai devreme decat trebuia. Sa se corecteze greseala.

update adoptii

set adoption\_date = adoption\_date + 1

where adoption\_date = (select max(adoption\_date)

from adoptii);



4. Cainii cu varsta mai mare decat media varstelor tuturor cainilor sunt stersi.

delete from caini

where age >= (select avg(age) from caini);



5, Se sterge prima tranzactie din anul 2016.

delete from tranzactii

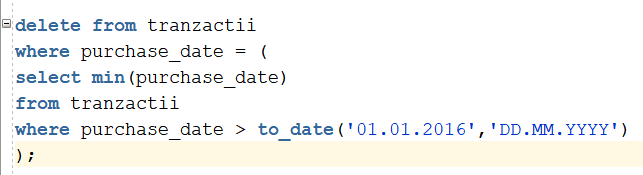
where purchase\_date = (

select min(purchase\_date)

from tranzactii

where purchase\_date > to\_date('01.01.2016','DD.MM.YYYY')

);

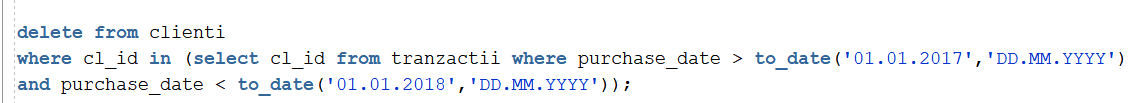


1. Se sterg clientii care au efectuat tranzactii in anul 2017.

delete from clienti

where cl\_id in (select cl\_id from tranzactii where purchase\_date > to\_date('01.01.2017','DD.MM.YYYY')

and purchase\_date < to\_date('01.01.2018','DD.MM.YYYY'));



**9, Realizarea normalizării până la forma normala 3 (FN1-FN3)**

***FN1***

O relație se află în FN1 dacă fiecărui atribut care o compune îi corespunde o valoare

indivizabilă.

Forma normală 1 este și cea care impune și faptul că fiecare înregistrare să fie definită astfel

încât să fie identificată unic prin intermediul unei chei primare.

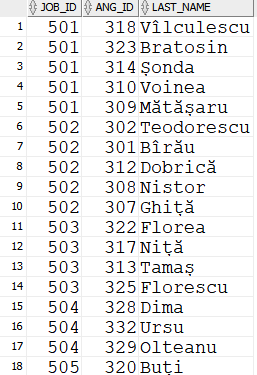
În cadrul bazei mele de date pe care am documentat-o aici, voi utiliza un exemplu pentru a

demonstra că aceasta e în forma normală 1.

Tabela ANGAJATI se leaga de tabela JOBS prin job\_id, astfel stim ce job are fiecare angajat. Dar mai multi angajati pot avea acelasi job dupa cum putem vedea mai jos:

|  |  |  |
| --- | --- | --- |
| Job\_id | Ang\_id | Last\_name |
| 501 | 318  323  314  310 | Vîlculescu  Bratosin  Șonda  Voinea |
| 502 | 301  312  308  307 | Bîrău  Dobrică  Nistor  Ghiță |
| 503 | 322  317  313  325 | Florea  Niță  Tamaș  Florescu |
| 504 | 328  332  329 | Dima  Ursu  Olteanu |

Relația de mai sus nu se află în FN1, pentru că atributelor ang\_id si last\_name nu le corespunde o valoare indivizabilă.



Se poate observa că după normalizare, fiecărui atribut care compune relația îi corespunde o valoare indivizibilă. Deci, pentru a asigura unicitatea unei înregistrări se va folosi cheia primară. În exemplul din stanga, prin introducerea unei coloane de tip întreg se asigură unicitatea fiecărei înregistrări.

***FN2***

O relație se află în FN2 daca și numai dacă această relație este deja în FN1 și fiecare atribut

care nu este cheie primară este dependent de întreaga cheie primară, astfel

**FN2 = FN1 + eliminarea dependentelor partiale.**

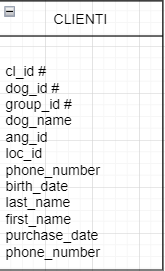
FN2 interzice existența dependențelor funcționale parțiale în cadrul relației.

Dacă unul sau mai mute elemente sunt dependente funcțional numai de o parte a cheii

primare, atunci ele trebuie să fie separate în tabele diferite. Dacă o tabelă are o cheie primară

formată doar dintr-un atribut, atunci ea e automat în FN2.

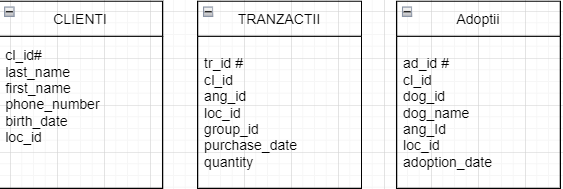
Vom folosi ca exemplu tabela Client, care ar fi aratat in mod initial asa:



In aceasta tabela, cheia primara este formata din cl\_id (id-ul unic al clientului), dog\_id (id-ul cainelui adoptat) si group\_id (tipul produsului cumparat). Avem multe dependente partiale, precum dog\_name care depinde doar de dog\_id, sau last\_name si first\_name care depind doar de cl\_id.

Alta problema apare cand un client nu a cumparat inca nimic, astfel group\_id ar trebui sa fie null, sau daca clientul respectiv cumpara mai intai un produs, iar la o saptamana dupa cumpara alt produs, vechea tranzactie va fi pierduta, de asemenea purchase\_date este un atribut definit prost deroace nu se stie daca se refera la cumpararea unui produs sau a unui caine.

Astfel, pentru a rezolva aceste probleme, am impartit tabela Clienti de mai sus in 3 tabele: Tranzactii, Clienti si Adoptii care arata asa:



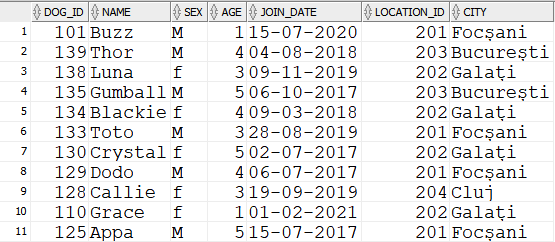
Astfel, se elimina toate dependentele partiale, cat si restul problemelor, relatia fiind acum in FN2.

***FN3***

O relatie este in a treia forma normala daca si numai daca este in FN2 si fiecare atribut care nu este cheie, depinde direct de cheia primara si numai de ea.

**FN3 = FN2 + eliminarea dependentelor tranzitive**

Pentru exemplificare, ne vom uita la tabela CAINI care la inceput ar fi aratat asa:



Dupa cum se poate vedea, dog\_id (cheia primara) determina toate celelalte atribute, mai putin CITY, care este determinat de LOCATION\_ID si astfel apar date duplicate in tabel.

Pentru aducerea in FN3, am creeat o tabela noua numita LOCATII, care contine date despre fiecare locatie in parte, iar in tabela CAINI am lasat doar atributul LOCATION\_ID, indepartand atributul CITY.

