Proiect Baze de Date

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

Modelul de date va gestiona informații legate de organizarea și funcționarea unui lanț de magazine ce vinde sisteme de calculatoare și periferice. Un client poate cumpăra dintr-un magazin unul sau mai multe produse, produsele fiind organizate pe categorii(sisteme desktop, monitoare, casti). Clientul poate fi o persoană fizică sau o persoană juridică.

Într-un oraș pot fi unul sau mai multe magazine, orașele la rândul lor fiind organizate pe județe. Angajații sunt împarțiți pe diverse departamente(management, vânzări, curățenie, it), iar pentru fiecare angajat se cunoaște job-ul. Un angajat poate lucra fie într-un magazin, fie într-un sediu.

1. Prezentarea constrângerilor(restricții, reguli) impuse asupra modelului:

* Clientul este fie o persoană juridică, fie una fizică
* Data facturii nu poate să fie mai veche decat data înfințării magazinului
* Stocul unui produs dintr-un magazin este mereu mai mare sau egal decat 0
* În aceeași locație nu pot exista mai multe magazine sau mai multe sedii, dar poate exista atât un magazin cât și un sediu
* Un angajat fie lucrează într-un magazin, fie lucrează într-un sediu
* Unei facturi trebuie mereu asociate unul sau mai multe produse
* Pentru fiecare persoana fizică și juridică se va cunoaște maxim un numar de telefon și o adresa de email
* Pot fi înregistrați mai mulți clienți la aceesi locație

1. Descrierea entitaților, incluzând precizarea cheii primare

Pentru modelul de date referitor la prezentărilde de modă, structurile CLIENT, FACTURA, CATEGORIE, PRODUS, MAGAZIN, LOCATIE, JUDET, SEDIU, ANGAJAT, DEPARTAMENT, JOB reprezintă entitați.

CLIENT = persoană fizica sau persoana juridică care a realizat cel puțin o achiziție în cadrul unui magazin. Despre un client sunt cunoscute datele de contat, cât și adresa. Cheia primară a entitații este client\_id

FACTURA = evidența a achizițiilor unui client pentru fiecare magazin. Se cunoaște și data când s-a realizat o factura. Cheia primara este factura\_id

MAGAZIN = locatie de unde pot fi cumpărate diverse produse, în limita stocului disponibil la momentul actual. Într-un magazin pot lucra unul sau mai mulți angajațti. Cheia primara este magazin\_id

PRODUS = obiect comercilizat în cadrul unuia sau mai multe magazine, poate fi achiziționat de unul sau mai mulți clienți. Cheia primara este produs\_id

CATEGORIE = entitate ce are rolul de a grupa produsele asemănătoare. Cheia primară este categorie\_id

LOCATIE = entitate ce reține informatii în legătură cu adresa magazinelor, sediilor, dar și ale clienților. Cheia primara este locație\_id

JUDEȚ = entitate ce reține informații în legătură despre județ. Cheia primară este județ\_id

SEDIU = locație unde lucreaza unul sau mai mulți angajați, care nu au funcții ce țin de magazine. Cheia primară este sediu\_id

ANGAJAT = persoană ce lucrează în cadrul lanțului de magazine, poate la rândul ei conduce unul sau mai mulți angajați și poate fi condusă de un alt angajat. Cheia primară este angajat\_id

DEPARTAMENT = entitate ce grupeaza angajati cu funcții similare. Fiecare departament este condus de un angajat. Cheia primară este department\_id

JOB = functie în cadrul lanțului de magzine ce poate fi indeplinită de unul sau mai mulți angajați. Cheia primara este job\_id

1. Descrierea relațiilor, incluzând precizarea cadrinalității acestora

Vom prezenta relațiile modelului de date, dând o descrierere completă fiecăreia.

CLIENT\_are\_FACTURA= relație care leagă entitățile CLIENT și FACTURA, reflectând legătura dintre acestea (pentru ca o persoană fizică sau juridică să fie considerată client trebuie sa aibă cel puțin o factură). Ea are cardinalitatea minimă 1:1 ( un client trebuie sa aiba cel puțin o factură) și cardinalitatea maximă 1:n (un client poate avea mai multe facturi, dar o factură poate aparținea unui singur client)

FACTURA\_apartine\_MAGAZIN= relație care leagă entitățile FACTURA și MAGAZIN, reflectând legătura dintre acestea( ce facturi au fost facute în cadrul unui magazin). Relația are cardinalitatea minimă 1:0 (o factură trebuie sa aparțină unui magazin, dar în cadrul unui magazin se poate să nu fi fost făcută nicio factură).

FACTURA\_conține\_PRODUS= relație care leagă entitățile FACTURA și PRODUS, reflectând legătura dintre ele ( ce produse au fost cumpărate în cadrul unei facturi). Relația are cardinalitatea minimă 1:0( o factura trebuie sa conțină cel puțin un produs, dar un produs poate să nu fie inclus in nicio factură) și cardinalitatea maxima m:n( o factura poate conține mai multe produse și un produs poate fi inclus în mai multe facturi).

PRODUS\_apartine\_CATEGORIE = relație care leagă entitățile CATEGORIE și PRODUS,reflectând legătura dintre ele (ce produse se încadrează într-o anumită categorie), relația are cardinilalitatea minimă 1:1 (un produs aparține unei categorii și o categorie conține cel puțin un produs) și maximă 1:n(un produs aparține unei categorii și o categorie conține mai multe produse).

MAGAZIN\_are\_PRODUSE = relație care leagă entitățile MAGAZIN și PRODUS, reflectând legătura dintre ele (ce produse are un magazin). Relația are cardinalitatea minimă 0:0(un magazin poate sa nu aibă produse și un produs poate sa nu fie în niciun magazin) și maxim m:n(un magazin poate sa aibă mai multe produse și un produs poate sa fie intr-unul sau mai multe magazine).

MAGAZIN\_se\_afla\_LOCATIA = relație care leagă entitățile MAGAZIN și LOCAȚIE, reflectând legătura dintre ele(unde este localizat un MAGAZIN), relația are cardinalitatea minimă 1:0 (un magazin se afla intr-o anumită locație, dar la o anumită locație poate să nu existe un magazin) și cardinalitatea maximă 1:1 (un magazine se află intr-o anumită locație, iar la o anumită locație se afla un magazin). O astfel de relație este definită și pentru SEDIU și LOCATIE, dar și între CLIENT și LOCAȚIE, permițând localizarea sediilor și clienților.

LOCATIE\_aparține\_JUDEȚ = relație care leagă entitățile LOCAȚIE și JUDET, reflectând legătura dintre ele(de ce județ aparține o anumită locație). Relația are cardinalitatea minimă 1:0( o locație aparține unui singur județ, dar intr-un județ nu se află nicio locație) și maximă M:1( o locație aparține unui singur județ, iar într-un județ se afla mai multe locații).

ANGAJAT\_lucrează\_MAGAZIN = relațtie care leagă entitățile ANGAJAT și SEDIU, reflectând legătura dintre ele(în ce magazin lucreaza un angajat). Relația are cardinalitatea minimă 0:0(un angajat poate să nu lucreze in niciun magazin, iar intr-un magazin poate sa nu lucreze nimeni) și cardinalitatea maximă m:1, (un angajat poate lucra într-un singur magazin și într-un magazin pot lucra mai mulți angajați). O astfel de relație este definită si pentru ANGAJAT și SEDIU, pentru a se putea determina dacă un angajat lucrează într-un sediu, sau într-un MAGAZIN.

ANGAJAT\_are\_JOB = relație care leagă entitățile ANGAJAT și JOB, reflectând legătura dintre ele, (ce job are un anumit angajat). Relația are cardinalitatea minimă 1:0(un angajat are un singur job, dar un job poate sa nu fie îndeplinit de niciun angajat) și cardinalitatea maximă 1:M(un angajat are un singur job, iar un job poate fi îndeplinit de mai mulți angajați).

ANGAJAT\_conduce\_ANGAJAT = relație reflexivă ce leagă tabela ANGAJAT de tabela ANGAJAT, reflectând legătura dintre ele( de cine este condus un anumit angajat). Relația are cardinalitatea minimă 0:0(angajatul nu este condus de nimeni , iar angajatul poate să nu conducă pe nimeni) și maximă 1:M( angajatul conduce mai mulți angajați, iar angajatul poate fi condus de un alt angajat).

ANGAJAT\_lucreaza\_DEPARTAMENT = relațtie ce leagă entitățile ANGAJAT și DEPARTAMENT, reflectând legătura dintre ele(în ce departament lucrează un anumit angajat). Relația are cardinalitatea minimă 1:0(angajatul lucreză într-un singur departament, iar într-un departament poate să nu lucreze nimeni) și maximă 1:m( un angajat lucrează într-un singur department, iar într-un department pot lucra mai mulți angajați).

ANGAJAT\_conduce\_DEPARTAMENT = relație ce leagă entitățile ANGAJAT și DEPARTAMENT, reflectând legătura dintre ele(ce angajat conduce un anumit department). Relația are cardinalitatea minimă 0:0 și maximă 1:1.

1. Descrierea atributelor, incluzând tipul lor

Entitatea independentă CLIENT are ca atribute:

client\_id= variabilă de tip întreg, de lungime maximă 6, care reprezintă codul unui client

email = variabilă de tip caracter, de lungime maximă 40, care reprezintă email-ul unui client

nr\_telefon = variabilă de tip caracter, de lungime maximă 20, care reprezintă numarul de t elefon al unui client

Subentitatea PF are ca atribute:

nume=variabilă de tip caracter, de lungime maximă 20, care reprezintă numele unei persoane fizice

prenume= variabilă de de tip caracter, de lungime maximă 20, care reprezintă prenumele unei persoane fizice

data\_nastere=variabilă de tip dată, reprezentâd data nasterii

Subentitatea PJ are ca atribute:

nume\_companie=variabilă de tip caracter reprezentând numele unei companii

Entitatea independentă MAGAZIN are ca atribute:

magazin\_id=variabilă de tip intreg, de lungime maximă 6, reprezentând codul unui magazin

dată\_contstruire=variabilă de tip dată calendaristică, care reprezintă data construirii unui magazin

dimensiune=variabilă de tip intreg, de lunime maximă 10, care reprezintă dimensiunea unui magazin, in metri pătrați

Entitatea independentă LOCATIE are ca atribute:

strada=variabilă de tip caracter, de lungime maximă 20, reprezentând numele unei străzi

numar=variabilă de tip întreg, de lungime maximă 6, reprezentând numărul unei adrese

oras=variabilă de tip caracter, de lungime maximă 20, reprezentând numele unui oraș

Entitatea independent JUDET are ca atribute:

nume=variabilă de tip caracter, de lungime maximă 20, reprezentând numele unui județ

cod\_judet=variabilă de tip caracter, de lungime maximă 2, reprezentând codul unui județ

Entitatea independentă SEDIU are ca atribute:

sediu\_id=variabilă de tip întreg, de lungime maximă 6, reprezentând id-ul unui sediu

dimensiune=variabilă de tip întreg de lungime maximă 10, care reprezintă dimensiunea unui sediu, în metri pătrați

capacitate=variabilă de tip întreg, de lungime maximă 6, reprezentând numărul maxim de angajați care pot lucra într-un sediu

Entitatea independentă ANGAJAT are atributele:angajat\_id, nume, prenume, data\_angjare, nr\_telefon, email, salariu

Entitatea CATEGORIE are atributele: id\_categorie, nume\_categorie

Entitatea PRODUS are atributele: produs\_id, pret, denumire

Entitatea DEPARTAMENT are atributele: nume\_departament, departament\_id

Entitatea FUNCTIE are atributele:functie\_id, denumire

1. Realizarea diagramei entitate relație

Diagram, engineering drawing

Description automatically generated

1. Realizarea diagramei conceptuale

Diagram

Description automatically generated

1. Enumerarea schemelor relaționale

CLIENT(client\_id#, id\_pf, id\_pj, locatie\_id, email, nr\_telefon)

PF(client\_id#, nume, prenume, data\_nastere)

PJ(client\_id#, nume\_companie)

FACTURA(factura\_id#, client\_id, magazin\_id, data\_achizitie)

MAGAZIN(magazin\_id#, locatie\_id, dimensiune, data\_construire)

LOCATIE(locatie\_id#, cod\_judet, strada, numar, oras)

JUDET(cod\_judet#, nume)

SEDIU(sediu\_id#, locatie\_id#, dimensiune, data\_construire, capacitate)

ANGAJAT(angajat\_id#, job\_id, manager\_id, departament\_id, locatie\_id,nume, prenume, data\_angajare, nr\_telefon, email, salariu)

FUNCTIE(functie\_id#, denumire)

DEPARTMANET(departament\_id#, manager\_id, nume\_departament)

PRODUS(produs\_id#, categorie\_id, pret, denumire)

STOC(produs\_id#, magazin\_id#, cantitate)

ATASAT(produs\_id#, factura\_id#, cantitate)

CATEGORIE(categorie\_id#, nume\_categorie)

1. Normalizare:

1.Exemplu non-FN1:

Presupunem ca am fi avut atributul PRODUSE pentru fiecare factură.Se observă ca atributul PRODUSE este compus, așa că vom introduce componentele acestuia în relație, iar pentru fiecare relație obținută cheia primară va fi formată din din cheia primară a facturii (atribut repetitiv) și codul produsului.

2.Exemplu non-FN2

Presupunem că exista o relație de forma LOCATIE(oraș#, cod\_judet#, stradă, nume\_județ). Se observă că există dependența cod\_județ->nume\_județ. Pentru a aduce relația în FN2 aplicăm regula Casey-Delobel și vom înlocui LOCAȚIE prin doua proiecții LOCAȚIE(oraș#, județ#, stradă) și JUDEȚ(cod\_județ#, nume\_județ).

3.Exemplu non-FN3:

Presupunem că există o relație de forma ANGAJAT(cod\_angajat#, nume, prenume, departament, nume\_departament). Se observă ca nume\_departament depinde tranzitiv de cod\_angajat, deoarece exisă dependențele cod\_angajat->department și departament ->nume\_departament. Se va descompune relația în ANGAJAT(cod\_angajat#, nume,prenume, departament) și DEPARTAMENT(cod\_departament#, nume\_departament).

1. Crearea secventă

CREATE SEQUENCE secventa

start with 1

increment by 1;



create table JUDET(

cod\_judet varchar(2) constraint pk\_judete primary key,

nume\_judet varchar(20) constraint null\_judete\_nume not null);

create table LOCATIE(

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

cod\_judet varchar(2) constraint fk\_loc\_jud references judet(cod\_judet),

strada varchar(40),

oras varchar(20) constraint null\_loc\_oras not null,

numar number(6));

create table MAGAZIN(

magazin\_id number(6) constraint pk\_magazin primary key,

locatie\_id number(6) constraint fk\_mag\_loc references locatie(locatie\_id),

dimensiune number(6) constraint null\_mag\_dim not null,

data\_construire date constraint null\_mag\_data not null

);

alter table magazin add constraint unq\_loc unique(locatie\_id);

create table SEDIU(

sediu\_id number(6) constraint pk\_sediu primary key,

locatie\_id number(6) constraint fk\_sed\_loc references locatie(locatie\_id),

dimensiune number(6) constraint null\_sed\_dim not null,

data\_construire date constraint null\_sed\_data not null,

capacitate number(6));

create table FUNCTIE(

functie\_id number(6) constraint pk\_job primary key,

denumire varchar(20) constraint null\_job\_den not null);

create table DEPARTAMENT(

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

manager\_id number(6),

nume\_departament varchar(20));

alter table departament

add constraint fk\_dep\_ang foreign key(manager\_id) references angajat(angajat\_id);

create table ANGAJAT(

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

job\_id number(6) constraint fk\_ang\_func references functie(functie\_id),

manager\_id number(6) constraint fk\_ang\_ang references angajat(angajat\_id),

departament\_id number(6) constraint fk\_ang\_dep references departament(departament\_id),

locatie\_id number(6) constraint fk\_ang\_loc references locatie(locatie\_id),

nume varchar(20) constraint null\_ang\_nume not null,

prenume varchar(20) constraint null\_ang\_prenume not null,

data\_angajare date constraint null\_data\_ang not null,

nr\_telefon varchar(20),

email varchar(30),

salary number(6) constraint null\_ang\_sal not null);

alter table angajat

add MAGAZIN\_ID number(6);

alter table angajat

add SEDIU\_ID number(6);

alter table angajat

add constraint cck\_sed\_mag check((magazin\_id is null and sediu\_id is not null) or (sediu\_id is null and magazin\_id is not null));

commit;

create table CATEGORIE(

categorie\_id number(6) constraint pk\_categorie primary key,

nume\_categorie varchar(20) constraint null\_cat\_num not null);

create table PRODUS(

produs\_id number(6) constraint pk\_produs primary key,

categorie\_Id number(6) constraint fk\_prod\_cat references categorie(categorie\_id),

pret number(6) constraint null\_prod\_pret not null,

denumire varchar(20) constraint null\_prod\_den not null);

create table STOC(

produs\_id number(6) constraint fk\_stoc\_prod references produs(produs\_id),

magazin\_id number(6) constraint fk\_prod\_mag references magazin(magazin\_id),

cantitate number(6) constraint null\_stoc\_cant not null,

constraint pk\_stoc primary key(produs\_id, magazin\_id));

alter table stoc add constraint cck\_stoc\_poz check(cantitate>=0);

create table PF(

pf\_id number(6) constraint pk\_pf primary key,

nume varchar(20) constraint null\_pf\_nume not null,

prenume varchar(20) constraint null\_pf\_pren not null,

data\_nastere date constraint null\_pf\_data not null);

create table PJ(

pj\_id number(6) constraint pk\_pj primary key,

denumire varchar(20) constraint null\_pj\_den not null);

create table CLIENT(

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

id\_pf number(6) constraint fk\_pf references pf(pf\_id),

id\_pj number(6) constraint fk\_pj references pj(pj\_id),

locatie number(6) constraint fk\_client\_loc references locatie(locatie\_id),

email varchar(20),

nr\_telefon varchar(20),

constraint cck\_tip\_client check((id\_pf is null and id\_pj is not null) or (id\_pf is not null and id\_pj is null)));

alter table client add constraint unq\_cl\_pf unique(id\_pj);

alter table client add constraint unq\_cl\_pj unique(id\_pf);

create table FACTURA(

factura\_id number(6) constraint pk\_factura primary key,

client\_id number(6) constraint fk\_fact\_client references client(client\_id),

magazin\_id number(6) constraint fk\_fact\_mag references magazin(magazin\_id),

data\_achizitie date constraint null\_fact\_data not null);

create table ATASAT(

factura\_id number(6) constraint fk\_at\_factura references factura(factura\_id),

produs\_id number(6) constraint fk\_at\_produs references produs(produs\_id),

cantitate number(6) constraint null\_at\_cant not null,

constraint pk\_atasat primary key(factura\_id, produs\_id));

insert into judet

values('B','Bucuresti');

insert into judet

values('IF', 'Ilfov');

insert into judet

values ('CT', 'Constanta');

insert into judet

values ('PH', 'Prahova');

insert into judet

values ('CJ', 'Cluj');

commit;

insert into locatie

values(secventa.nextval, 'B', 'Sos Victoriei','Bucuresti',123);

insert into locatie

values(secventa.nextval, 'B', 'Telescopului','Bucuresti',145);

insert into locatie

values(secventa.nextval, 'CJ', 'Aviatorilui','Cluj-Napoca',13);

insert into locatie

values(secventa.nextval, 'CT', 'Soarelui','Constanta',45);

insert into locatie

values(secventa.nextval, 'CT', 'Soarelui','Constanta',47);

insert into locatie

values(secventa.nextval, 'PH', 'Nordului','Ploiesti',132);

insert into locatie

values(secventa.nextval, 'CT', 'Aviatorilor','Constanta',59);

insert into locatie

values(secventa.nextval, 'CT', 'Aviatorilor','Constanta',89);

insert into locatie

values(secventa.nextval, 'B', 'Malului','Bucuresti',145);

insert into locatie

values(secventa.nextval, 'CJ', 'Aviatorilor','Cluj-Napoca',13);

insert into locatie

values(secventa.nextval, 'CT', 'Abecedarului','Cluj-Napoca',60);

insert into locatie

values(secventa.nextval, 'CT', 'Abecedarului','Cluj-Napoca',90);

commit;

update locatie

set cod\_judet='CJ'

where locatie\_id=9;

insert into magazin

values(secv\_mag.nextval, 2, 234, '01-jan-2002');

insert into magazin

values(secv\_mag.nextval, 3, 150, '08-jun-2005');

insert into magazin

values(secv\_mag.nextval, 1, 160, '09-jun-2010');

insert into magazin

values(secv\_mag.nextval, 5, 180, '13-aug-2006');

insert into magazin

values(secv\_mag.nextval, 4, 130, '26-oct-2004');

insert into sediu

values(100, 2, 432, '01-jan-2003',150);

insert into sediu

values(101, 3, 150, '08-jun-2002',50);

insert into sediu

values(102, 1, 160, '09-jun-2010',40);

insert into sediu

values(103, 5, 180, '13-aug-2006', 51);

insert into sediu

values(104, 9, 120, '26-oct-2004', 60);

delete from magazin

where 1>0;

rollback;

insert into functie

values(1, 'CEO');

insert into functie

values(2, 'Paznic');

insert into functie

values(3, 'Casier');

insert into functie

values(4, 'Vanzator');

insert into functie

values(5, 'Om-de-servici');

insert into functie

values(6, 'Expert-markerting');

insert into functie

values(7, 'Expert-HR');

insert into functie

values(8, 'Programator');

insert into functie

values(9, 'Manager-magazin');

commit;

insert into departament

values(100, null, 'Management');

insert into departament

values(101, null, 'Paza');

insert into departament

values(102, null, 'Mentenanta');

insert into departmanet

values(103, null, 'Marketing');

insert into departament

values(104, null, 'HR');

insert into departament

values(105, null, 'Vanzari');

commit;

delete from angajat;

insert into angajat

values(100, 1, null, 100, 3, 'Teodorescu', 'George', '02-oct-2002', '0733333333', 'tgt@yahoo.com', 10000, null,101);

insert into angajat

values(102, 9, 100, 100, 3, 'Ionescu', 'Alex', '02-feb-2003', '0733333444', 'ialex@yahoo.com', 4000, 100, null);

insert into angajat

values(103, 3, 102, 105, 4, 'Popescu', 'Andrei', '03-feb-2004', '0754303030', 'pandrei@yahoo.com', 2000, 100, null);

insert into angajat

values(104, 5, 102, 105, 7, 'Marinescu', 'Marius', '07-jun-2004', '0754023132', 'mmarius@yahoo.com', 2000, 100, null);

rollback;

insert into angajat

values(105, 9, 100, 100, 9, 'Titulescu', 'Alex', '02-feb-2003', '0733333444', 'talex@yahoo.com', 3000, 101, null);

insert into angajat

values(106, 3, 105, 105, 21, 'Ionescu', 'LUCA', '02-feb-2006', '0733333444', 'iluca@yahoo.com', 2000, 101, null);

insert into angajat

values(107, 5, 105, 105, 22, 'Alexandrescu', 'Mihai', '05-feb-2006', '0712234423','amihai@gmail.com', 2000, 101, null);

insert into angajat

values(108, 7, 100, 105, 2, 'Popescu', 'Mihai', '05-feb-2006', '0712234423','pmihai@gmail.com', 5000, null, 102);

commit;

update departament

set manager\_id=108

where departament\_id in(104);

commit;

insert into categorie

values(1, 'Sisteme');

insert into categorie

values(2, 'Casti');

insert into categorie

values(3, 'Monitoare');

insert into categorie

values(4, 'Tastaturi');

insert into categorie

values(5, 'Mousepads');

insert into categorie

values(6, 'Boxe');

commit;

insert into produs

values(100, 1, 9000, 'Sistem-asus');

insert into produs

values(101, 1, 7000, 'Sistem-dell');

insert into produs

values(102, 3, 3000, 'Monitor-asus');

insert into produs

values(103, 3, 1500, 'Monitor-samsung');

insert into produs

values(104, 4, 350, 'Tastatura-razer');

insert into produs

values(105, 4, 700, 'Tastatura-rog');

insert into produs

values(106,2, 900, 'Casti-razer');

commit;

insert into stoc

values(100, 101, 3);

insert into stoc

values(100, 103, 5);

insert into stoc

values(100, 104, 0);

insert into stoc

values(101, 102, 4);

insert into stoc

values(103, 104, 3);

insert into stoc

values(105, 104, 3);

insert into stoc

values(102, 103, 4);

insert into stoc

values(106, 101, 0);

insert into stoc

values(103, 102, 6);

insert into stoc

values(103, 101, 3);

commit;

insert into pf

values(1, 'Ionescu', 'Matei', '01-jan-1990');

insert into pf

values(2, 'Constantinescu' , 'Mihai', '10-oct-2001');

insert into pf

values(3, 'Stefanescu', 'Alex', '12-jan-2003');

insert into pf

values(4, 'Popa', 'Robert', '25-jul-2005');

insert into pf

values(5, 'Popescu', 'Iulian', '30-may-2006');

commit;

insert into pj

values(1, 'Bitdefender');

insert into pj

values(2, 'Xerox\_SRL');

insert into pj

values(3, 'Reparatii\_SRL');

insert into pj

values(4, 'LibrarieOnline');

insert into pj

values(5, 'MATEI\_SRL');

rollback;

insert into client

values(100, 1, null, 3, 'abc@gmail.com', '0777777777');

insert into client

values(101, null, 2, 4, 'fgb@yahoo.com', '0777777777');

insert into client

values(102, null, 3, 5, 'oip@yahoo.com', '0777777897');

insert into client

values(103, null, 4, 6, 'ads@yahoo.com', '0777777777');

insert into client

values(104, null, 5, 6, 'asa@yahoo.com', '0777777777');

insert into client

values(105, null, 1, 6, 'asa@yahoo.com', '0777777777');

insert into client

values(106, 2, null, 6, 'asa@yahoo.com', '0777777777');

insert into client

values(107, 3, null, 22, 'asa@yahoo.com', '0777777777');

insert into client

values(108, 4, null, 21, 'asda@gmail.com', '0777778888');

commit;

insert into factura

values(100, 100, 100, '02-feb-2002');

insert into factura

values(101, 102, 100, '04-feb-2002');

insert into factura

values(102, 104, 100, '04-feb-2002');

insert into factura

values(103, 104, 102, '04-feb-2010');

insert into factura

values(104, 100, 102, '04-feb-2010');

insert into factura

values(105, 105, 102, '04-feb-2010');

insert into factura

values(106, 105, 103, '09-sep-2006');

insert into factura

values(107, 107, 103, '09-sep-2006');

insert into factura

values(108, 108, 103, '10-sep-2006');

insert into factura

values(109, 108, 104, '10-nov-2004');

insert into factura

values(110, 108, 104, '10-nov-2004');

commit;

insert into atasat

values(100, 100, 2);

insert into atasat

values(100, 101, 1);

insert into atasat

values(101, 103, 4);

insert into atasat

values(101, 104, 4);

insert into atasat

values(102, 101, 2);

insert into atasat

values(103, 104, 3);

insert into atasat

values(104, 106, 2);

insert into atasat

values(104, 102, 1);

insert into atasat

values(105, 101, 2);

insert into atasat

values(106, 104, 3);

insert into atasat

values(107, 102, 1);

insert into atasat

values(108, 101, 3);

insert into atasat

values(109, 100, 5);

insert into atasat

values(109, 101, 5);

insert into atasat

values(105, 100, 6);

insert into atasat

Table

Description automatically generated with medium confidencevalues(110, 100, 6);

Table

Description automatically generated with low confidenceGraphical user interface, application

Description automatically generatedGraphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Table

Description automatically generated

Table

Description automatically generated with medium confidence

A picture containing table

Description automatically generated

Table

Description automatically generated

Graphical user interface

Description automatically generated with medium confidence

Table

Description automatically generated

Table

Description automatically generated

1. 5 Cereri SQL:

Graphical user interface, text, application, email

Description automatically generated1.Să se afișeze clienții care au cumpărat mai mult de 5 produse din categoria sisteme ordonați dupa numarul de sisteme cumpărate, respective câte au cumpărat.

* operația join pe 4 tabele
* grupări de date, funcții de grup, filtrare la nivel de grupuri
* o funcție pe șiruri de caractere
* ordonări

select f.client\_id, sum(NR)

from factura f,

(select f.factura\_id, sum(a.cantitate) NR

from factura f, atasat a,

(select p.produs\_id PID

from produs p, categorie c

where p.categorie\_id=c.categorie\_id

and lower(nume\_categorie) like '%sisteme%') b

where f.factura\_id=a.factura\_id and a.produs\_id=PID

group by f.factura\_id

) numar

where f.factura\_id=numar.factura\_id

group by f.client\_id

having sum(NR)>=5

order by 2;

2. Să se afișeze angajații care au subșirul ‘ul’ în nume și care lucreaza într-un departament în care există 2 angajați care stau la aceeași adresă.

Graphical user interface, text, application

Description automatically generated

* subcereri sincronizate pe 3 tabele
* filtrare pe linii
* o functie pe șiruri de caractere

select angajat\_id

from angajat a

where exists( select 1

from angajat b

where b.departament\_id=a.departament\_id

and exists(select 1

from angajat c

where c.locatie\_id=b.locatie\_id and b.angajat\_id!=angajat\_id and b.departament\_id=departament\_Id))

and instr(lower(nume),'ul')!=0;

Graphical user interface, application, Word

Description automatically generated3. Să se afișeze produsele care se mai găsesc în stoc într-un magazin construit in primele 7 luni are anului 2005 și care sunt localizate in județul cu numele Cluj.

* subcereri nesincronizate în care intervin cel puțin 3 tabele
* o funcție pe date calendaristice

select produs\_id

from stoc

where magazin\_id in (select magazin\_id

from magazin

where data\_construire>(to\_date('01-jan-2005')) and months\_between(data\_construire,to\_date('01-jan-2005'))<=7

and locatie\_id in(select locatie\_id

from locatie l, judet j

where l.cod\_judet=j.cod\_judet

and lower(trim(j.nume\_judet)) = 'cluj'))

and cantitate>0;

4. Să se afișeze suma salariilor angajaților pentru fiecare departament.

Text

Description automatically generated

* funcțiile nvl și decode
* with

with nr\_ang as(

select sum(salary) SUMA, nume\_departament NUME

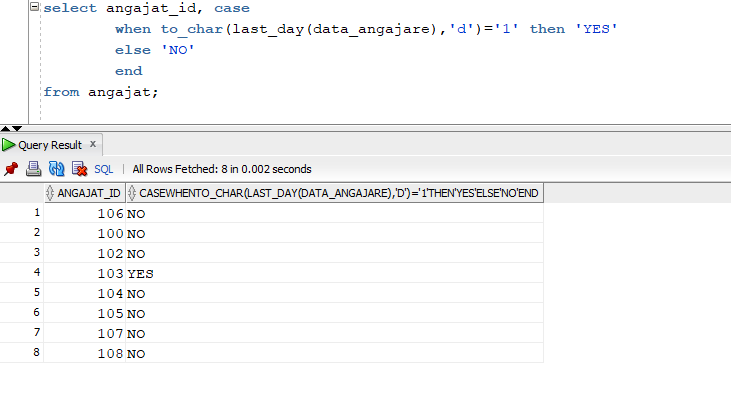
from departament d, angajat a

where a.departament\_id(+)=d.departament\_id

group by nume\_departament)

select decode(nvl(suma,0), 0, 0, suma), nume

from nr\_ang;

5. Să se afișeze pentru fiecare angajat dacă a fost angajat intr-o lună, în care ultima zi era luni.

* case
* o funcție calendaristică

select angajat\_id, case

when to\_char(last\_day(data\_angajare),'d')='1' then 'YES'

else 'NO'

end

from angajat;

1. Implementarea a trei operații de actualizare și suprimare a datelor
2. Sa se modifice manager-ul angajaților care lucrează în același departament cu angajatul citit de la tastatură într-o valoare citită de la tastatură.

Graphical user interface, text, application, email

Description automatically generated

update angajat

set manager\_id=&mng\_cod

where departament\_id=(select departament\_id from angajat where angajat\_id=&ang\_cod);

2.Să se șteargă din tabela departament, departamentele în care nu lucrează nimeni

Graphical user interface, text, application, email

Description automatically generated

delete from departament

where departament\_id not in (select distinct departament\_id from angajat);

3.Să se modifice numele străzii locatiilor cu numele județului dat de la tastatură, cu un șir citit de la tastatură.

Graphical user interface, text, application, email

Description automatically generated

update locatie

set strada='&sir'

where cod\_judet=(select cod\_judet from judet where trim(nume\_judet)='&cod\_judet');

16. Să se afișeze suma salariilor angajaților care lucreaza în magazine pentru fiecare județ și 0 unde nu există magazine sau angajați.

Graphical user interface, application, Word

Description automatically generated

select nvl(sum(salary),0), j.cod\_judet

from angajat a, magazin m, locatie l, judet j

where a.magazin\_id(+)=m.magazin\_id and m.locatie\_id(+)=l.locatie\_id and l.cod\_judet(+)=j.cod\_judet

group by j.cod\_judet;

Să se afișeze produsele care se află în stoc în toate magazinele din judetul cu codul B

Graphical user interface, text, application

Description automatically generated

with numar as(select count(magazin\_id) NR from magazin m, locatie l where l.cod\_judet='B' and m.locatie\_id=l.locatie\_Id)

select produs\_id

from stoc

where magazin\_id in (select magazin\_id NR from magazin m, locatie l where l.cod\_judet='B' and m.locatie\_id=l.locatie\_Id)

group by produs\_id

having count(magazin\_id) = (select NR from numar);

Să se afle clienții care au cumpărat din fiecare magazin înființat dupa 2006

Graphical user interface, text, application, email

Description automatically generated

select client\_id

from (select distinct client\_id from factura) c

where not exists(

select magazin\_id

from magazin

where data\_construire>to\_date('01-01-2006','dd-mm-yyyy')

minus

select magazin\_id

from factura f

where c.client\_id=f.client\_id

);

17

Diagram, engineering drawing

Description automatically generated

S1=PROJECT(NATURAL JOIN(LOCATIE, SELECT(JUDET, cod=’B’)),cod\_judet, cod\_locatie, strada)

S2=SELECT(NATURAL JOIN(MAGAZIN, ANGAJAT), salariu>4000)

R=NATURALJOIN(S1, S2)

Diagram, engineering drawing

Description automatically generated

R=NATURAL JOIN(NATURAL JOIN(PROJECT(LOCATIE, cod\_locatei, strada, cod\_judet),

PROJECT(JUDET, cod\_judet)), NATURAL JOIN(SELECT(ANGAJAT, salariu>4000), MAGAZIN))