

# Managementul companiei Electrohub

Bănilean Alexandru-Ioan

Grupa 152

Baze de date – Anul 1

**Descrierea modelului real – Ex. 1**

Acest proiect își propune să prezinte implementarea unei baze de date pentru managementul unei companii ce vinde produse electronice, intitulată ElectroHub. În această secțiune se va prezenta modelul de date, utilitatea acestuia și regulile de funcționare.

Baza de date conține informații despre produsele comercializate, clienții companiei, utilizatorii site-ului asociat, diverse informații interne precum date despre angajați, depozite, colete sau ierarhia din cadrul companiei.

Scopul creării acestei baze de date este de a pune la dispoziție o imagine de ansamblu asupra business-ului și de a permite o gestiune rapidă și facilă a afacerii, pentru a oferi clienților o experiență cât mai plăcută în interacțiunea lor cu ElectroHub.

Compania are o structură ierarhică bine definită, cu existența unor departamente, supervizate de un manager, fiecare având sediul la o anumită locație. Fiecare locație se află într-o regiune dintr-o anumită țară. În cadrul unui departament există mai mulți angajați, ce ocupă o anumită poziție în companie, cu un titlu aferent.

ElectroHub comercializează produse atât în magazine fizice, cât și pe site-ul companiei, unde fiecare client își poate crea unul sau mai mulți utilizatori, astfel adresându-se unei game extinse de clienți.

Comenzile sunt procesate în funcție de cantitatea produselor ce intră în componența acestora, sunt împachetate în colete și distribuite către depozite. În cazul în care un depozit este plin, acesta nu va mai putea depozita colete, pentru acest lucru fiind stocată suprafața și capacitatea acestora.

După ce un client cumpără un produs, acesta poate scrie un review în care să își descrie experiența avută, ajutând astfel și comunitatea prin informațiile oferite.

Pentru a dispune și de anumite date statistice, vor fi reținute și accesările utilizatorilor pe site, cu scopul de a prezenta produsele populare cu o prioritate mai mare.

**Restricții de funcționare – Ex. 2**

Modelul de date prezintă anumite constrângeri:

- O țară trebuie să aibă una sau mai multe regiuni, iar o regiune aparține unei singure țări.
- O regiune trebuie să aibă în componență unul sau mai multe orașe, iar un oraș aparține unei singure regiuni.
- Un oraș trebuie să aibă în componență una sau mai multe locații, iar o locație aparține unui singur oraș.
- La o anumită locație reținută trebuie să existe cel puțin un departament, un depozit sau un magazin, iar aceste trei tipuri trebuie să se afle la o singură locație.
- Un departament are cel puțin un angajat și poate avea un manager.
- Un angajat trebuie să aparțină unui departament, indiferent de statutul acestuia.
- Fiecare angajat are asociat un singur job.
- Un angajat poate lucra la un magazin, la un depozit sau la un departament.
- Un magazin, împreună cu un client, poate determina una sau mai multe comenzi.
- O comandă trebuie să fie asociată unui magazin și unui client.
- De asemenea, un utilizator poate face una sau mai multe comenzi, iar o comandă de acest tip trebuie să fie asociată unui utilizator.
- În momentul în care se face o comandă, trebuie să specifice o adresă de livrare.
- În cadrul unei comenzi trebuie să apară unul sau mai multe produse, într-o anumită cantitate.
- Un produs cu cantitatea asociată poate apărea unic într-o comandă.
- Un depozit poate avea unul sau mai multe colete, iar un colet trebuie să se afle într-un singur depozit.
- Un client poate avea unul sau mai mulți utilizatori, însă orice utilizator trebuie să fie asociat cu un singur client.
- Un utilizator poate avea una sau mai multe accesări, iar o accesare trebuie să aparțină unui utilizator.
- Un client poate lăsa unul sau mai multe review-uri la un anumit produs.
- Fiecare review are un rating ce variază pe o scară de la 1 la 10.

**Descrierea entităților – Ex. 3**

1. Tari
  - Reprezintă un tabel cu țările în care compania își desfășoară activitatea.
  - Atribute: id\_tara, nume\_tara, nr\_locuitori, suprafata
  - Cheia primară: id\_tara
2. Regiuni
  - Reprezintă regiunile dintr-o țară în care compania își desfășoară activitatea.
  - Atribute: id\_regiune, id\_tara, nume\_regiune, suprafata, resedinta
  - Cheia primară: id\_regiune
3. Orase
  - Reprezintă orașele dintr-o regiune în care compania își desfășoară activitatea.
  - Atribute: id\_oras, id\_regiune, nume\_oras
  - Cheia primară: id\_oras
4. Angajati
  - Reprezintă angajații companiei.
  - Atribute: id\_angajat, id\_departament, nume, prenume, email, nr\_telefon, data\_angajare, salariu, id\_job, id\_manager, id\_depozit, id\_magazin
  - Cheia primară: id\_angajat
5. Departamente
  - Reprezintă departamentele companiei.
  - Atribute: id\_departament, id\_locatie, denumire, id\_manager
  - Cheia primară: id\_departament
6. Locatii
  - Reprezintă locațiile în care compania are un depozit, magazin sau departament.
  - Atribute: id\_locatie, id\_regiune, adresa, nr\_cladiri
  - Cheia primară: id\_locatie
7. Depozite
  - Reprezintă depozitele companiei.
  - Atribute: id\_depozit, id\_locatie, capacitate, suprafata, este\_plin
  - Cheia primară: id\_depozit
8. Colete
  - Reprezintă coletele ce se află în depozite.
  - Atribute: id\_colet, id\_depozit, data\_preluare, data\_livrare, greutate, volum
  - Cheia primară: id\_colet

#### 9. Joburi

- Reprezintă joburile angajaților.
- Atribute: id\_job, nume\_job, salariu\_minim, salariu\_maxim
- Cheia primară: id\_job

#### 10. Magazine

- Reprezintă magazinele companiei.
- Atribute: id\_magazin, id\_locatie, nr\_anagajati
- Cheia primară: id\_magazin

#### 11. Produse

- Reprezintă produsele comercializate.
- Atribute: id\_produc, ambalaj, pret, dimensiune, greutate, descriere
- Cheia primară: id\_produc

#### 12. Comenzi

- Reprezintă comenzile date de clienți.
- Atribute: id\_comanda, adresa\_livrare, id\_utilizator
- Cheia primară: id\_comanda

#### 13. Reviews

- Reprezintă review-urile clienților la anumite produse.
- Atribute: id\_review, descriere, rating
- Cheia primară: id\_review

#### 14. Clienți

- Reprezintă clienții companiei.
- Atribute: id\_client, nume, prenume, email, nr\_telefon, adresa
- Cheia primară: id\_client

#### 15. Utilizatori

- Reprezintă utilizatorii site-ului.
- Atribute: id\_utilizator, id\_client, nume\_utilizator, parola, preferinte, email, data\_creare
- Cheia primară: id\_utilizator

#### 16. Accesări

- Reprezintă accesările utilizatorilor pe site.
- Atribute: id\_accesare, data\_accesare, id\_utilizator
- Cheia primară: id\_accesare

Tabele asociative

## 17. Cantitati

- Reprezintă cantitatea unor produse ce apar într-o anumită comandă.
- Atribute: id\_comanda, id\_produș, cantitate
- Cheia primară: id\_comanda, id\_produș

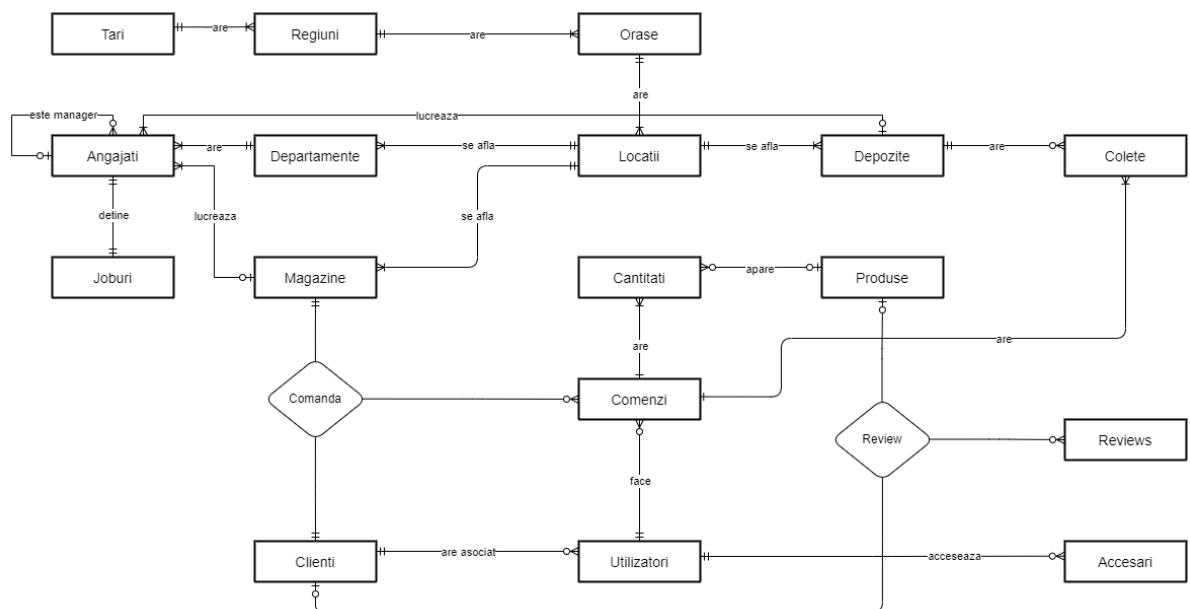
## 18. Comanda\_Magazin

- Reprezintă asocierea dintre o comandă făcută într-un magazin, magazin și client.
- Atribute: id\_magazin, id\_client, id\_comanda
- Cheia primară: id\_magazin, id\_client, id\_comanda

## 19. Review\_Produș

- Reprezintă asocierea dintre un review, un produs și client.
- Atribute: id\_produș, id\_client, id\_review
- Cheia primară: id\_produș, id\_client, id\_review

## Descrierea relațiilor – Ex. 4



Descrierea atributelor – Ex. 5

Atribut	Descriere	Exemplu	Tip de date	Constrângeri
<b>Tari</b>				
id_tara	Identificator unic țară	25	Number(8, 0)	Primary key
nume_tara	Denumire	România	Varchar2(30)	Not null
nr_locuitori	Număr locuitori	19290000	Number(15, 0)	-
suprafata	Suprafața țării	238397	Number(15, 0)	-
<b>Regiuni</b>				
id_regiune	Identificator unic regiune	4	Number(8, 0)	Primary key
id_tara	Identificatorul țării	25	Number(8, 0)	Not null, Foreign key
nume_regiune	Denumire	Muntenia	Varchar2(30)	Not null
resedinta	Oraș reședință	București	Varchar2(30)	-
<b>Orașe</b>				
id_oras	Identificator unic oraș	10	Number(8, 0)	Primary key
id_regiune	-	53	Number(8, 0)	Not null, Foreign key
nume_oras	Denumire	Barcelona	Varchar2(30)	Not null
<b>Angajati</b>				
id_angajat	Identificator unic angajat	44	Number(8, 0)	Primary key
id_departament	Identificator departament	30	Number(8, 0)	Foreign key
nume	-	Popescu	Varchar2(30)	Not null
prenume	-	Ionel	Varchar2(30)	Not null
email	-	ionel.pop@gmail.com	Varchar2(50)	Not null
nr_telefon	-	0742123456	Varchar2(15)	-
data_angajare	-	2008-11-11 13:23:44	Date	Not null
salariu	-	5000	Number(8, 0)	Default: 0, Check(>=salariu_minim), Check(<=salariu_maxim)
id_job	Identificator job	23	Number(8, 0)	Not null, Foreign key



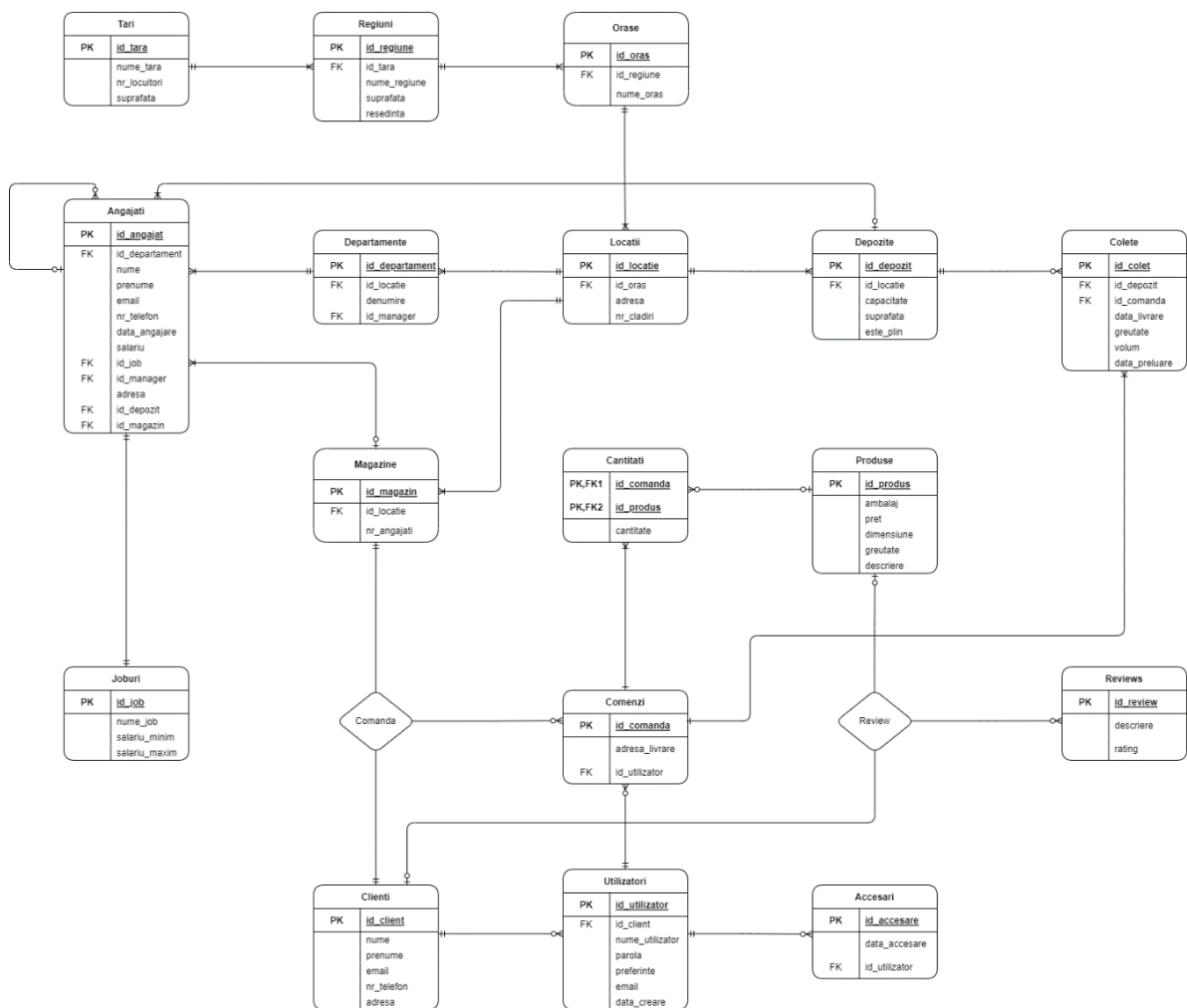
id_manager	Identificator unic manager	19	Number(8, 0)	Foreign key
id_depozit	-	54	Number(8, 0)	Foreign key
id_magazin	-	22	Number(8, 0)	Foreign key
<b>Departamente</b>				
id_departament	Identificator unic departament	30	Number(8, 0)	Primary key
id_locatie	Identificator locație	24	Number(8, 0)	Not null, Foreign key
denumire	-	Resurse umane	Varchar2(30)	Not null
id_manager	Identificator manager	47	Number(8, 0)	Foreign key
<b>Locatii</b>				
id_locatie	Identificator unic locație	25	Number(8, 0)	Primary key
id_oras	Identificator oras	30	Number(8, 0)	Not null, Foreign key
adresa	-	Str. Academiei, nr. 33	Varchar2(50)	Not null
nr_cladiri	Număr de cladiri	12	Number(8, 0)	-
<b>Depozite</b>				
id_depozit	Identificator unic depozit	24	Number(8, 0)	Primary key
id_locatie	Identificator locație	35	Number(8, 0)	Not null, Foreign key
capacitate	Capacitatea depozitului în metri cubi	10000	Number(8, 0)	Not null
suprafata	Suprafața	1500	Number(8, 0)	Not null
este_plin	Verifică starea depozitului	Da	Varchar2(5)	-
<b>Colete</b>				
id_colet	Identificator unic colet	500	Number(8, 0)	Primary key
id_comanda	Identificator comanda	267	Number(8, 0)	Not null, Foreign key
id_depozit	Identificator depozit	24	Number(8, 0)	Not null, Foreign key
data_preluare	-	2007-10-11 13:23:44	Date	Not null
data_livrare	-	2007-10-13 13:23:44	Date	-

greutate	Greutate colet în kg	20	Number(4, 0)	-
volum	Volum colet în metri cubi	1	Number(4, 0)	-
<b>Joburi</b>				
id_job	Identificator unic job	25	Number(8, 0)	Primary key
nume_job	-	Contabil	Varchar2(30)	Not null
salariu_minim	Salariul minim în lei	3000	Number(8, 0)	-
salariu_maxim	Salariul maxim în lei	5000	Number(8, 0)	-
<b>Magazine</b>				
id_magazin	Identificator unic magazin	13	Number(8, 0)	Primary key
id_locatie	Identificator locație	26	Number(8, 0)	Not null, Foreign key
nr_angajati	Număr angajați	20	Number(8, 0)	-
<b>Produse</b>				
id_produc	Identificator unic produs	523	Number(8, 0)	Primary key
ambalaj	Descriere ambalaj	Metal	Varchar2(50)	-
pret	Preț în lei	250	Number(8, 0)	Not null
dimensiune	Dimensiunile produsului	220x400x20	Varchar2(50)	-
greutate	Greutate în g	300	Number(8, 0)	-
descriere	-	Mașină de spălat	Varchar2(50)	-
<b>Comenzi</b>				
id_comanda	Identificator unic comandă	2663	Number(8, 0)	Primary key
adresa_livrare	-	Str. Lalelelor, nr. 22	Varchar2(50)	Not null
id_utilizator	Identificator utilizator	266	Number(8, 0)	Foreign key
<b>Reviews</b>				
id_review	Identificator unic review	2623	Number(8, 0)	Primary key
descriere	-	Un produs calitativ	Varchar2(50)	-
rating	-	10	Number(2, 0)	Not null, check(> 0), check(<11)

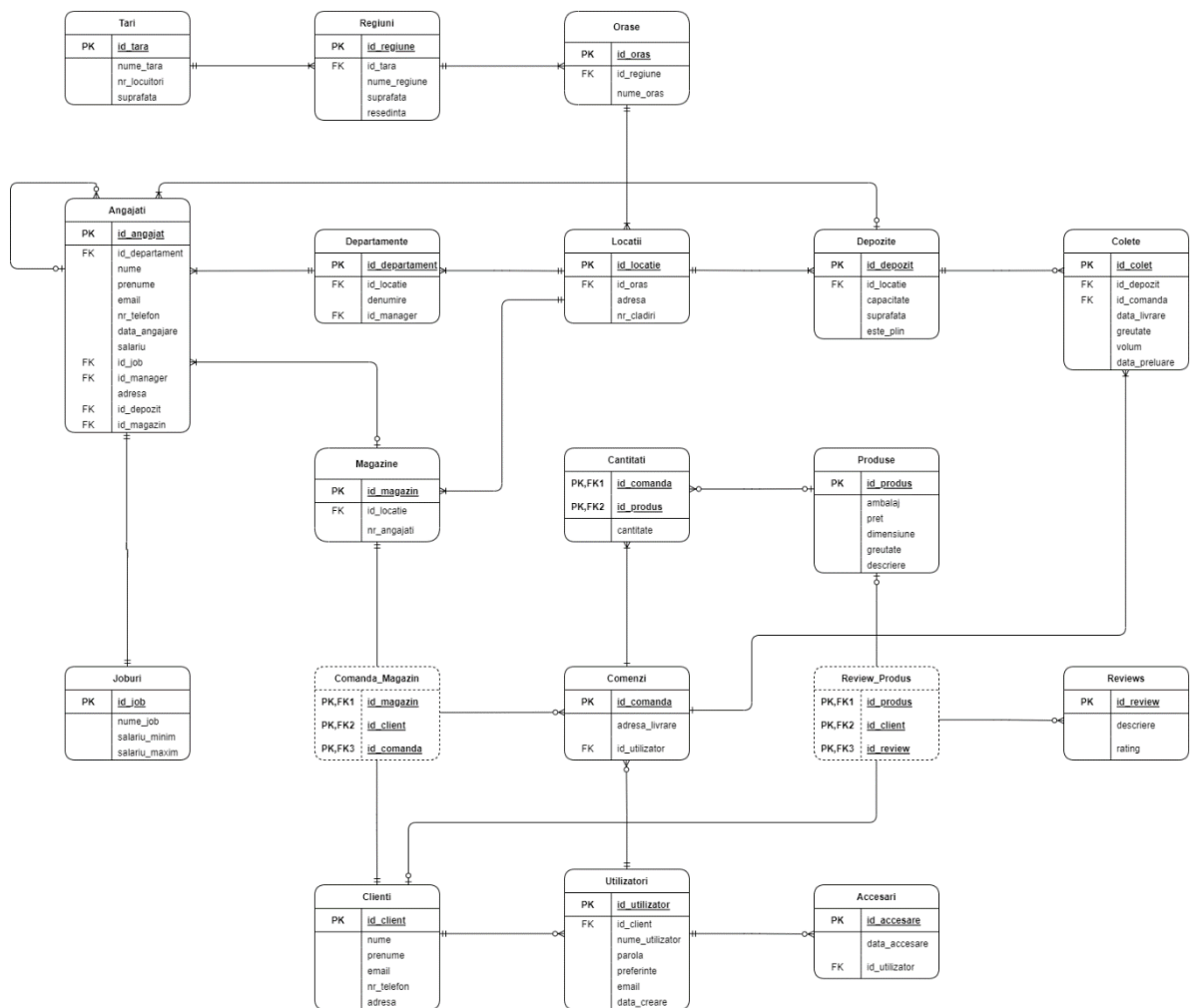
Clienti				
id_client	Identificator unic client	152	Number(8, 0)	Primary key
nume	-	Georgescu	Varchar2(30)	Not null
prenume	-	Gigel	Varchar2(30)	Not null
email	-	gigel.gg@gmail.com	Varchar2(50)	-
nr_telefon	-	0741235496	Varchar2(15)	-
adresa	-	Str. Crinului, nr.24	Varchar2(50)	-
Utilizatori				
id_utilizator	Identificator unic utilizator	6454	Number(8, 0)	Primary key
id_client	Identificator client	45	Number(8, 0)	Not null, Foreign key
nume_utilizator	-	Ionel22	Varchar2(30)	Not null
parola	-	1234	Varchar2(30)	Not null
preferinte	-	Dark theme	Varchar2(50)	-
email	-	ionel124@gmail.com	Varchar2(50)	Not null
data_creare	-	2012-10-11 15:23:44	Date	Not null
Accesari				
id_accesare	Identificator unic accesare	34353	Number(8, 0)	Primary key
data_accesare	-	2007-10-11 13:23:44	Date	Not null
id_utilizator	Identificator utilizator	4334	Number(8, 0)	Not null, Foreign key
Tabele asociative				
Cantitati				
id_comanda	Identificator comandă	46	Number(8, 0)	Primary key, Foreign key
id_produs	Identificator produs	565	Number(8, 0)	Primary key, Foreign key
cantitate	Cantitate produs	5462	Number(4, 0)	Default: 1
Comanda_Magazin				
id_magazin	Identificator magazin	25	Number(8, 0)	Primary key, Foreign key

id_client	Identificator client	54	Number(8, 0)	Primary key, Foreign key
id_comanda	Identificator comandă	255	Number(8, 0)	Primary key, Foreign key
<b>Review_Produs</b>				
id_produs	Identificator produs	456	Number(8, 0)	Primary key, Foreign key
id_client	Identificator client	55	Number(8, 0)	Primary key, Foreign key
id_review	Identificator review	544	Number(8, 0)	Primary key, Foreign key

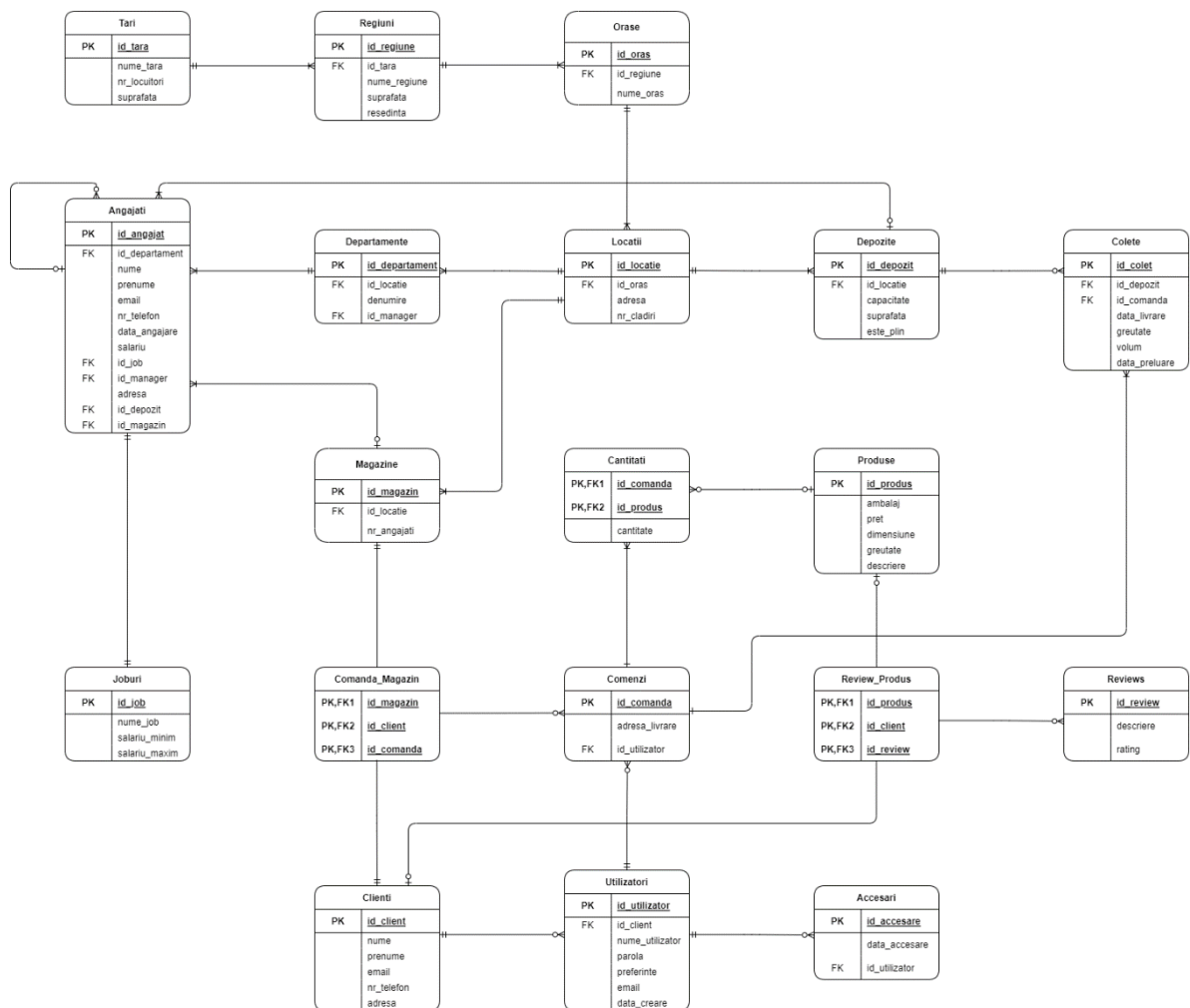
## Diagrama E/R – Ex. 6



## Diagrama conceptuală – Ex. 7



## Design Logic



## Scheme relaționale – Ex. 8

- TARI(#id\_tara, nume\_tara, nr\_locuitori, suprafata)
- REGIUNI(#id\_regiune, id\_tara, nume\_regiune, suprafata, resedinta)
- ORASE(#id\_oras, id\_regiune, nume\_oras)
- ANGAJATI(#id\_angajat, id\_departament, nume, prenume, email, nr\_telefon, data\_angajare, salariu, id\_job, id\_manager, id\_depozit, id\_magazin)
- DEPARTAMENTE(#id\_departament, id\_locatie, denumire, id\_manager)
- LOCATII(#id\_locatie, id\_regiune, adresa, nr\_cladiri)
- DEPOZITE(#id\_depozit, id\_locatie, capacitate, suprafata, este\_plin)
- COLETE(#id\_colet, id\_depozit, data\_preluare, data\_livrare, greutate, volum)
- JOBURI(#id\_job, nume\_job, salariu\_minim, salariu\_maxim)
- MAGAZINE(#id\_magazin, id\_locatie, nr\_anagajati)
- PRODUSE(#id\_produș, amabalaj, pret, dimensiune, greutate, descriere)
- COMENZI(#id\_comanda, adresa\_livrare, id\_utilizator)
- REVIEWS(#id\_review, descriere, rating)
- CLIENTI(#id\_client, nume, prenume, email, nr\_telefon, adresa)
- UTILIZATORI(#id\_utilizator, id\_client, nume\_utilizator, parola, preferinte, email, data\_creare)
- ACCESARI(#id\_accesare, data\_accesare, id\_utilizator)
- CANTITATI(#id\_comanda, #id\_produș, cantitate)
- COMANDA\_MAGAZIN(#id\_magazin, #id\_client, #id\_comanda)
- REVIEW\_PRODUS(#id\_produș, #id\_client, #id\_review)



### Normalizare și denormalizare – Ex. 9 / 18

Baza de date din acest proiect este normalizată, fiind adusă până la BCNF.

#### ▪ **Forma normală 1(FN1)**

O relație se află în FN1 dacă pe toate atributele sale există doar valori atomice(indivizibile).

Exemplu non-FN1:

Fie tabelul următor, care nu este în FN1.

#id_client	nume	email
1	Popescu	a12.popescu@gmail.com
2	Georgescu	georgescu.mihai@gmail.com mihaigeorgescu@yahoo.com

Se observă faptul că instanța cu numele Georgescu are 2 adrese de email.

Trecem la atribute atomice, pentru a aduce în FN1:

#id_client	nume	email
1	Popescu	a12.popescu@gmail.com
2	Georgescu	georgescu.mihai@gmail.com
3	Georgescu	mihaigeorgescu@yahoo.com

#### ▪ **Forma normală 2(FN2)**

O relație se află în FN2 dacă este în FN1 și toate atributele care nu sunt chei primare depind de întreaga cheie primară.

Exemplu non-FN2:

Fie tabelul următor, care nu este în FN2:

LOCATII(#id\_locatie, #id\_oras, nume\_oras, adresa)

Avem dependențele:

#id\_locatie => adresa

#id\_oras => nume\_oras

Descompunere fără pierdere de informație, pentru a aduce în FN2:

ORASE(#id\_oras, nume\_oras)

LOCATII(#id\_locatie, id\_oras, adresa) (în acest caz id\_oras este cheia străină)

- **Forma normală 3(FN3)**

O relație este în FN3 dacă este în FN2 și fiecare atribut care nu este cheie depinde direct de cheia primară.

Exemplu non-FN3:

Fie tabelul următor, care nu respectă FN3:

COMENZI(#id\_comanda, adresa\_livrare, id\_utilizator, nume\_utilizator)

Avem dependențele:

#id\_comanda => adresa\_livrare, id\_utilizator

id\_utilizator => nume\_utilizator

Descompunere fără pierdere de informație, pentru a aduce în FN3:

COMENZI(#id\_comanda, adresa\_livrare, id\_utilizator)

UTILIZATORI(id\_utilizator, nume\_utilizator)

- **Forma normală Boyce-Codd(BCNF)**

O relație este în BCNF dacă este în FN3 și orice cheie este în același timp și supercheie.

Exemplu non-BCNF:

Fie tabelul următor:

CANTITATI(#id\_comanda, #id\_produș, cantitate)

Avem dependențele, considerând faptul că o cantitate apare unic într-o comandă și determină identificatorul unic al produsului:

(#id\_comanda, #id\_produș) => cantitate

cantitate => #id\_produș

Descompunere fără pierdere de informație, pentru a aduce în BCNF, având în vedere dependența cantitate => #id\_produș:

CANTITATI\_1(#id\_comanda, cantitate)

CANTITATI\_2(id\_produș, cantitate)

De menționat este faptul că ipoteza de la care am plecat nu se regăsește în acest model de date, unde o cantitate nu determină identificatorul unic al produsului.

- **Forma normală 4(FN4)**

O relație este în FN4 dacă este în BCNF și orice dependență netrivială multivalorică are în partea stângă o supercheie. Aducerea în FN4 presupune eliminarea redundanțelor datorate dependențelor multiple.

Fie tabelul următor:

DEPOZITE(#id\_depozit, id\_locatie, denumire)

Presupun faptul că un depozit poate avea mai multe locații (fiind divizat) și mai multe denumiri.

Am avea dependențele:

#id\_depozit => id\_locatie

#id\_depozit => denumire

Pentru a aduce în FN4, vom descompune prin proiecție:

DEPOZITE\_1(#id\_depozit, id\_locatie)

DEPOZITE\_2(#id\_depozit, denumire)

#### ▪ **Forma normală 5(FN5)**

O relație este în FN5 dacă este în FN4 și orice join-dependență este determinată de o cheie candidat. Aducerea în FN5 presupune deci eliminarea join-dependențelor.

Fie tabelul:

DEPOZITE(id\_depozit, id\_locatie, denumire\_depozit).

Presupunem că există dependențe multiple între:

(id\_depozit, id\_locatie)

(id\_depozit, denumire\_depozit)

(id\_locatie, denumire\_depozit)

Pentru a aduce în FN5, descompunem tabelul inițial:

DEPOZITE\_LOCATII(id\_depozit, id\_locatie)

DEPOZITE\_DENUMIRI(id\_depozit, denumire\_depozit)

LOCATII\_DENUMIRI(id\_locatie, denumire\_depozit)

#### ▪ **Denormalizare**

Denormalizarea este procesul invers procesului de normalizare. Prin denormalizare se adăugă date redundante într-o bază de date relațională ce a fost normalizată pentru a crește performanțele interogărilor, în special pe cele de citire. Astfel, se reduce numărul de tabele și nu mai este necesară colectarea datelor ce se pot afla în alte tabele, în alte fișiere sau chiar în alte baze de date.

Luând în considerare baza de date din acest proiect, în cazul în care nu există un număr semnificativ de locații, pentru a crește rapiditatea interogărilor, numele orașului din entitatea ORASE ar putea fi reținut în cadrul adresei din entitatea LOCATII, iar prima entitate menționată nu ar mai fi necesară.

## Creare și inserare în SQL – Ex. 10

```

----- TARI -----
CREATE TABLE TARI
(
    id_tara NUMBER(8, 0) CONSTRAINT PKEY_TARI PRIMARY KEY,
    nume_tara VARCHAR2(30) CONSTRAINT nume_tara NOT NULL,
    nr_locuitori NUMBER(15, 0),
    suprafata NUMBER(15, 0)
);

INSERT INTO TARI VALUES (1, 'România', 19290000, 238397);
INSERT INTO TARI VALUES (2, 'Germania', 83222442, 357588);
INSERT INTO TARI VALUES (3, 'Franța', 67813000, 543940);
INSERT INTO TARI VALUES (4, 'Italia', 58983122, 302073);
INSERT INTO TARI VALUES (5, 'Spania', 47326687, 505990);

SELECT * FROM TARI;

```













































SQL   All Rows Fetched: 5 in 0.013 seconds				
ID_TARA	NUME_TARA	NR_LOCUITORI	SUPRAFATA	
1	1 România	19290000	238397	
2	2 Germania	83222442	357588	
3	3 Franța	67813000	543940	
4	4 Italia	58983122	302073	
5	5 Spania	47326687	505990	

----- Regiuni -----

```
CREATE TABLE REGIUNI
(
    id_regiune NUMBER(8, 0) CONSTRAINT PKEY_REGIUNI PRIMARY KEY,
    id_tara NUMBER(8, 0) CONSTRAINT id_tara_nn NOT NULL,
    nume_regiune VARCHAR2(30) CONSTRAINT nume_regiune NOT NULL,
    resedinta VARCHAR2(30),
    CONSTRAINT FK_REGIUNI_TARI FOREIGN KEY (id_tara) REFERENCES TARI(id_tara)
);
```

```
INSERT INTO REGIUNI VALUES (1, 1, 'Muntenia', 'București');
INSERT INTO REGIUNI VALUES (2, 1, 'Moldova', 'Iași');
INSERT INTO REGIUNI VALUES (3, 1, 'Transilvania', 'Cluj-Napoca');
INSERT INTO REGIUNI VALUES (4, 1, 'Dobrogea', 'Constanța');
INSERT INTO REGIUNI VALUES (5, 1, 'Maramureș', 'Baia Mare');
```

```
SELECT * FROM REGIUNI;
```

                                            All Rows Fetched: 5 in 0.004 seconds

ID_REGIUNE	ID_TARA	NUME_REGIUNE	RESEDINTA
1	1	1Muntenia	București
2	2	1Moldova	Iași
3	3	1Transilvania	Cluj-Napoca
4	4	1Dobrogea	Constanța
5	5	1Maramureș	Baia Mare

----- Orase -----

```
CREATE TABLE ORASE
```

```
(
  id_oras NUMBER(8, 0) CONSTRAINT PKEY_ORASE PRIMARY KEY,
  id_regiune NUMBER(8, 0) CONSTRAINT id_regiune_nn NOT NULL,
  nume_oras VARCHAR2(30) CONSTRAINT nume_oras NOT NULL,
  CONSTRAINT FK_ORASE_REGIUNI FOREIGN KEY (id_regiune) REFERENCES REGIUNI(id_regiune)
);
```

```
INSERT INTO ORASE VALUES (1, 1, 'București');
INSERT INTO ORASE VALUES (2, 2, 'Iași');
INSERT INTO ORASE VALUES (3, 3, 'Cluj-Napoca');
INSERT INTO ORASE VALUES (4, 4, 'Constanța');
INSERT INTO ORASE VALUES (5, 5, 'Baia Mare');
```

```
SELECT * FROM ORASE;
```

ID_ORAS	ID_REGIUNE	NUME_ORAS
1	1	1 București
2	2	2 Iași
3	3	3 Cluj-Napoca
4	4	4 Constanța
5	5	5 Baia Mare

----- Locatii -----

```
CREATE TABLE LOCATII
(
    id_locatie NUMBER(8, 0) CONSTRAINT PKEY_LOCATII PRIMARY KEY,
    id_oras NUMBER(8, 0) CONSTRAINT id_oras_nn NOT NULL,
    adresa VARCHAR2(50) CONSTRAINT adresa_locatie NOT NULL,
    nr_cladiri NUMBER(8, 0),
    CONSTRAINT FK_LOCATII_ORASE FOREIGN KEY (id_oras) REFERENCES ORASE(id_oras)
);
```

```
INSERT INTO LOCATII VALUES (1, 1, 'Str. Academiei, nr. 33', 1);
INSERT INTO LOCATII VALUES (2, 1, 'Str. Lalelelor, nr. 2', 2);
INSERT INTO LOCATII VALUES (3, 1, 'Str. Mihai Eminescu, nr. 1', 1);
INSERT INTO LOCATII VALUES (4, 3, 'Str. Crizantemelor, nr. 25', 1);
INSERT INTO LOCATII VALUES (5, 3, 'Str. Liviu Rebreanu, nr. 17', 2);
```

```
SELECT * FROM LOCATII;
```

ID_LOCATIE	ID_ORAS	ADRESA	NR_CLADIRI
1	1	Str. Academiei, nr. 33	1
2	2	Str. Lalelelor, nr. 2	2
3	3	Str. Mihai Eminescu, nr. 1	1
4	4	Str. Crizantemelor, nr. 25	1
5	5	Str. Liviu Rebreanu, nr. 17	2

----- Departamente -----

```
CREATE TABLE DEPARTAMENTE
```

```
(
  id_departament NUMBER(8, 0) CONSTRAINT PKEY_DEPARTAMENTE PRIMARY KEY,
  id_locatie NUMBER(8, 0) CONSTRAINT id_locatie_nn NOT NULL,
  denumire VARCHAR2(30) CONSTRAINT denumire_departament NOT NULL,
  id_manager NUMBER(8, 0),
  CONSTRAINT FK_DEPARTAMENTE_LOCATII FOREIGN KEY (id_locatie) REFERENCES LOCATII(id_locatie)
);
```

```
INSERT INTO DEPARTAMENTE VALUES (1, 1, 'Resurse umane', NULL);
```

```
INSERT INTO DEPARTAMENTE VALUES (2, 1, 'IT', NULL);
```

```
INSERT INTO DEPARTAMENTE VALUES (3, 1, 'Vânzări', NULL);
```

```
INSERT INTO DEPARTAMENTE VALUES (4, 3, 'Resurse umane', NULL);
```

```
INSERT INTO DEPARTAMENTE VALUES (5, 3, 'IT', NULL);
```

```
SELECT * FROM DEPARTAMENTE;
```

SQL | All Rows Fetched: 5 in 0.007 seconds

ID_DEPARTAMENT	ID_LOCATIE	DENUMIRE	ID_MANAGER
1	1	1 Resurse umane	(null)
2	2	1 IT	(null)
3	3	1 Vânzări	(null)
4	4	3 Resurse umane	(null)
5	5	3 IT	(null)



----- Depozite -----

```
CREATE TABLE DEPOZITE
(
    id_depozit NUMBER(8, 0) CONSTRAINT PKEY_DEPOZITE PRIMARY KEY,
    id_locatie NUMBER(8, 0) CONSTRAINT depozite_id_locatie_nn NOT NULL,
    capacitate NUMBER(8, 0) CONSTRAINT depozite_capacitate NOT NULL,
    suprafata NUMBER(8, 0) CONSTRAINT depozite_suprafata NOT NULL,
    este_plin VARCHAR2(5),
    CONSTRAINT FK_DEPOZITE_LOCATII FOREIGN KEY (id_locatie) REFERENCES LOCATII(id_locatie)
);

INSERT INTO DEPOZITE VALUES (1, 1, 5000, 2000, 'Nu');
INSERT INTO DEPOZITE VALUES (2, 3, 10000, 4000, 'Nu');
INSERT INTO DEPOZITE VALUES (3, 5, 4000, 1800, 'Nu');
INSERT INTO DEPOZITE VALUES (4, 2, 2000, 1000, 'Nu');
INSERT INTO DEPOZITE VALUES (5, 4, 1000, 200, 'Da');

SELECT * FROM DEPOZITE;
```





ID_DEPOZIT	ID_LOCATIE	CAPACITATE	SUPRAFATA	ESTE_PLIN
1	1	5000	2000	Nu
2	3	10000	4000	Nu
3	5	4000	1800	Nu
4	2	2000	1000	Nu
5	4	1000	200	Da

----- Magazine -----

```
CREATE TABLE MAGAZINE
(
    id_magazin NUMBER(8, 0) CONSTRAINT PKEY_MAGAZINE PRIMARY KEY,
    id_locatie NUMBER(8, 0) CONSTRAINT magazine_id_locatie_nn NOT NULL,
    nr_angajati NUMBER(8, 0),
    CONSTRAINT FK_MAGAZINE_LOCATII FOREIGN KEY (id_locatie) REFERENCES LOCATII(id_locatie)
);
```

```
INSERT INTO MAGAZINE VALUES (1, 1, 24);
INSERT INTO MAGAZINE VALUES (2, 3, 60);
INSERT INTO MAGAZINE VALUES (3, 5, 22);
INSERT INTO MAGAZINE VALUES (4, 2, 12);
INSERT INTO MAGAZINE VALUES (5, 4, 5);
```

```
SELECT * FROM MAGAZINE;
```

    SQL | All Rows Fetched: 5 in 0.008 seconds

	ID_MAGAZIN	ID_LOCATIE	NR_ANGAJATI
1	1	1	24
2	2	3	60
3	3	5	22
4	4	2	12
5	5	4	5

----- Joburi -----

```
CREATE TABLE JOBURI
```

```
(
    id_job NUMBER(8, 0) CONSTRAINT PKEY_JOBURI PRIMARY KEY,
    nume_job VARCHAR2(30) CONSTRAINT nume_job NOT NULL,
    salariu_minim NUMBER(8, 0),
    salariu_maxim NUMBER(8, 0)
);
```

```
INSERT INTO JOBURI VALUES (1, 'Programator', 3000, 12000);
INSERT INTO JOBURI VALUES (2, 'Consultant vânzări', 2000, 6000);
INSERT INTO JOBURI VALUES (3, 'Contabil', 3000, 8000);
INSERT INTO JOBURI VALUES (4, 'Manager vânzări', 4000, 8000);
INSERT INTO JOBURI VALUES (5, 'Manager resurse umane', 4000, 8000);
```

```
SELECT * FROM JOBURI;
```

SQL   All Rows Fetched: 5 in 0.006 seconds				
ID_JOB	NUME_JOB	SALARIU_MINIM	SALARIU_MAXIM	
1	1 Programator	3000	12000	
2	2 Consultant vânzări	2000	6000	
3	3 Contabil	3000	8000	
4	4 Manager vânzări	4000	8000	
5	5 Manager resurse umane	4000	8000	

```

----- Angajati -----

CREATE TABLE ANGAJATI
(
    id_angajat NUMBER(8, 0) CONSTRAINT PKEY_ANGJATI PRIMARY KEY,
    id_departament NUMBER(8, 0),
    nume VARCHAR2(30) CONSTRAINT angajati_nume NOT NULL,
    prenume VARCHAR2(30) CONSTRAINT angajati_prenume NOT NULL,
    email VARCHAR2(50) CONSTRAINT angajati_email NOT NULL,
    nr_telefon VARCHAR2(15),
    data_angajare DATE CONSTRAINT angajati_data_angajare NOT NULL,
    salariu NUMBER(8, 0) DEFAULT 0,
    id_job NUMBER(8, 0) CONSTRAINT anajati_id_job_nn NOT NULL,
    id_manager NUMBER(8, 0),
    id_depozit NUMBER(8, 0),
    id_magazin NUMBER(8, 0),
    CONSTRAINT FK_ANGAJATI_DEPARTAMENTE FOREIGN KEY (id_departament) REFERENCES DEPARTAMENTE(id_departament),
    CONSTRAINT FK_ANGAJATI_JOBURI FOREIGN KEY (id_job) REFERENCES JOBURI(id_job),
    CONSTRAINT FK_ANGAJATI_MANAGERI FOREIGN KEY (id_manager) REFERENCES ANGAJATI(id_angajat),
    CONSTRAINT FK_ANGAJATI_DEPOZITE FOREIGN KEY (id_depozit) REFERENCES DEPOZITE(id_depozit),
    CONSTRAINT FK_ANGAJATI_MAGAZINE FOREIGN KEY (id_magazin) REFERENCES MAGAZINE(id_magazin)
);

INSERT INTO ANGAJATI VALUES (1, 1, 'Popescu', 'Ion', 'popion@gmail.com', '0722467569', TO_DATE('2007-10-11', 'YYYY-MM-DD'), 4000, 1, NULL, 2, NULL);
INSERT INTO ANGAJATI VALUES (2, 2, 'Georgescu', 'Marcel', 'marecelgg@gmail.com', '0725467868', TO_DATE('2010-05-06', 'YYYY-MM-DD'), 3000, 2, NULL, NULL, 4);
INSERT INTO ANGAJATI VALUES (3, 3, 'Ionescu', 'Doru', 'dionescu@gmail.com', '0725367368', TO_DATE('2012-07-06', 'YYYY-MM-DD'), 2700, 3, NULL, NULL, 1);
INSERT INTO ANGAJATI VALUES (4, 4, 'Popa', 'Mircea', 'mpopa@gmail.com', '0723267368', TO_DATE('2010-08-12', 'YYYY-MM-DD'), 5000, 4, NULL, 3, NULL);
INSERT INTO ANGAJATI VALUES (5, 4, 'Titulescu', 'Bogdan', 'bogdantt@gmail.com', '0735464864', TO_DATE('2012-08-25', 'YYYY-MM-DD'), 4500, 5, NULL, NULL, 5);

ALTER TABLE DEPARTAMENTE
ADD CONSTRAINT FK_DEPARTAMENTE_ANGAJATI FOREIGN KEY (id_manager) REFERENCES ANGAJATI(id_angajat);

SELECT * FROM ANGAJATI;

```

SQL All Rows Fetched: 5 in 0.01 seconds

	ID_ANGAJAT	ID_DEPARTAMENT	NUME	PRENUME	EMAIL	NR_TELEFON	DATA_ANGAJARE	SALARIU	ID_JOB	ID_MANAGER	ID_DEPOZIT	ID_MAGAZIN
1	1	1	Popescu	Ion	popion@gmail.com	0722467569	11-OCT-07	4000	1	(null)	2	(null)
2	2	2	Georgescu	Marcel	marecelgg@gmail.com	0725467868	06-MAY-10	3000	2	(null)	(null)	4
3	3	3	Ionescu	Doru	dionescu@gmail.com	0725367368	06-JUL-12	2700	3	(null)	(null)	1
4	4	4	Popa	Mircea	mpopa@gmail.com	0723267368	12-AUG-10	5000	4	(null)	3	(null)
5	5	4	Titulescu	Bogdan	bogdantt@gmail.com	0735464864	25-AUG-12	4500	5	(null)	(null)	5



----- Clienti -----

```
CREATE TABLE CLIENTI
```

```
(
  id_client NUMBER(8, 0) CONSTRAINT PKEY_CLIENTI PRIMARY KEY,
  nume VARCHAR2(30) CONSTRAINT clienti_nume NOT NULL,
  prenume VARCHAR2(30) CONSTRAINT clienti_prenume NOT NULL,
  email VARCHAR2(50),
  nr_telefon VARCHAR2(15),
  adresa VARCHAR2(50)
);
```

```
INSERT INTO CLIENTI VALUES (1, 'Cristea', 'Mihai', 'mcristea@gmail.com', '0729486537', NULL);
INSERT INTO CLIENTI VALUES (2, 'Cristescu', 'Ștefan', 'stefcristescu@gmail.com', '0726484531', NULL);
INSERT INTO CLIENTI VALUES (3, 'Mircescu', 'Robert', 'rmircescu@gmail.com', '0725436532', NULL);
INSERT INTO CLIENTI VALUES (4, 'Rotaru', 'David', 'davidrt@gmail.com', '0729426232', NULL);
INSERT INTO CLIENTI VALUES (5, 'Nae', 'Cristian', 'cristinae@gmail.com', '0749446547', NULL);
```

```
SELECT * FROM CLIENTI;
```

  SQL | All Rows Fetched: 5 in 0.005 seconds

ID_CLIENT	NUME	PRENUME	EMAIL	NR_TELEFON	ADRESA
1	1 Cristea	Mihai	mcristea@gmail.com	0729486537	(null)
2	2 Cristescu	Ștefan	stefcristescu@gmail.com	0726484531	(null)
3	3 Mircescu	Robert	rmircescu@gmail.com	0725436532	(null)
4	4 Rotaru	David	davidrt@gmail.com	0729426232	(null)
5	5 Nae	Cristian	cristinae@gmail.com	0749446547	(null)

```

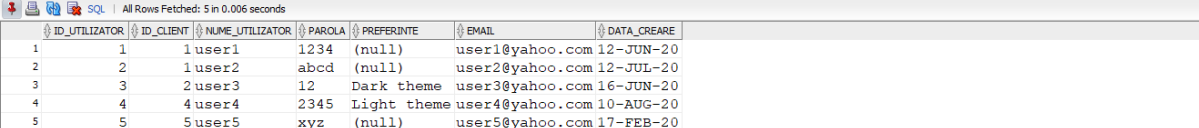
----- Utilizatori -----

CREATE TABLE UTILIZATORI
(
    id_utilizator NUMBER(8, 0) CONSTRAINT PKEY_UTILIZATORI PRIMARY KEY,
    id_client NUMBER(8, 0) CONSTRAINT utilizatori_id_client NOT NULL,
    nume_utilizator VARCHAR2(30) CONSTRAINT utilizatori_nume NOT NULL,
    parola VARCHAR2(30) CONSTRAINT utilizatori_parola NOT NULL,
    preferinte VARCHAR2(50),
    email VARCHAR2(50) CONSTRAINT utilizatori_email NOT NULL,
    data_creat DATE CONSTRAINT utilizatori_data_creat NOT NULL,
    CONSTRAINT FK_UTILIZATORI_CLIENTI FOREIGN KEY (id_client) REFERENCES CLIENTI(id_client)
);

INSERT INTO UTILIZATORI VALUES (1, 1, 'user1', '1234', NULL, 'user1@yahoo.com', TO_DATE('2020-06-12', 'YYYY-MM-DD'));
INSERT INTO UTILIZATORI VALUES (2, 1, 'user2', 'abcd', NULL, 'user2@yahoo.com', TO_DATE('2020-07-12', 'YYYY-MM-DD'));
INSERT INTO UTILIZATORI VALUES (3, 2, 'user3', '12', 'Dark theme', 'user3@yahoo.com', TO_DATE('2020-06-16', 'YYYY-MM-DD'));
INSERT INTO UTILIZATORI VALUES (4, 4, 'user4', '2345', 'Light theme', 'user4@yahoo.com', TO_DATE('2020-08-10', 'YYYY-MM-DD'));
INSERT INTO UTILIZATORI VALUES (5, 5, 'user5', 'xyz', NULL, 'user5@yahoo.com', TO_DATE('2020-02-17', 'YYYY-MM-DD'));

SELECT * FROM UTILIZATORI;

```



ID_UTILIZATOR	ID_CLIENT	NUME_UTILIZATOR	PAROLA	PREFERINTE	EMAIL	DATA_CREARE
1	1	user1	1234	(null)	user1@yahoo.com	12-JUN-20
2	1	user2	abcd	(null)	user2@yahoo.com	12-JUL-20
3	2	user3	12	Dark theme	user3@yahoo.com	16-JUN-20
4	4	user4	2345	Light theme	user4@yahoo.com	10-AUG-20
5	5	user5	xyz	(null)	user5@yahoo.com	17-FEB-20

----- Accesari -----

```
CREATE TABLE ACCESARI
(
    id_accesare NUMBER(8, 0) CONSTRAINT PKEY_ACCESARI PRIMARY KEY,
    data_accesare DATE CONSTRAINT accesari_data_accesare NOT NULL,
    id_utilizator NUMBER(8, 0) CONSTRAINT accesari_id_utilizator NOT NULL,
    CONSTRAINT FK_ACCESARI_UTILIZATORI FOREIGN KEY (id_utilizator) REFERENCES UTILIZATORI(id_utilizator)
);

INSERT INTO ACCESARI VALUES (1, TO_DATE('2020-04-15', 'YYYY-MM-DD'), 1);
INSERT INTO ACCESARI VALUES (2, TO_DATE('2021-05-15', 'YYYY-MM-DD'), 2);
INSERT INTO ACCESARI VALUES (3, TO_DATE('2020-06-15', 'YYYY-MM-DD'), 2);
INSERT INTO ACCESARI VALUES (4, TO_DATE('2020-08-15', 'YYYY-MM-DD'), 3);
INSERT INTO ACCESARI VALUES (5, TO_DATE('2021-09-15', 'YYYY-MM-DD'), 3);

SELECT * FROM ACCESARI;
```

ID_ACCESARE	DATA_ACCESARE	ID_UTILIZATOR
1	15-APR-20	1
2	21-MAY-21	2
3	15-JUN-20	2
4	15-AUG-20	3
5	15-SEP-21	3

----- Comenzi-----

```
CREATE TABLE COMENZI
(
    id_comanda NUMBER(8, 0) CONSTRAINT PKEY_COMENZI PRIMARY KEY,
    adresa_livrare VARCHAR2(50) CONSTRAINT comenzi_adresa NOT NULL,
    id_utilizator NUMBER(8, 0),
    CONSTRAINT FK_COMENZI_UTILIZATORI FOREIGN KEY (id_utilizator) REFERENCES UTILIZATORI(id_utilizator)
);
```

```
INSERT INTO COMENZI VALUES (1, 'Str. Crinului, nr. 22', NULL);
INSERT INTO COMENZI VALUES (2, 'Str. Crinului, nr. 22', NULL);
INSERT INTO COMENZI VALUES (3, 'Str. Crinului, nr. 22', NULL);
INSERT INTO COMENZI VALUES (4, 'Str. Crinului, nr. 22', NULL);
INSERT INTO COMENZI VALUES (5, 'Str. Crinului, nr. 22', NULL);
INSERT INTO COMENZI VALUES (6, 'Str. Speranței, nr. 34', NULL);
INSERT INTO COMENZI VALUES (7, 'Str. Speranței, nr. 34', NULL);
INSERT INTO COMENZI VALUES (8, 'Str. Cireșului, nr. 12', NULL);
INSERT INTO COMENZI VALUES (9, 'Str. Pinului, nr. 7', NULL);
INSERT INTO COMENZI VALUES (10, 'Str. Revoluției, nr. 3', NULL);
```

```
SELECT * FROM COMENZI;
```

SQL | All Rows Fetched: 10 in 0.01 seconds

ID_COMANDA	ADRESA_LIVRARE	ID_UTILIZATOR
1	1Str. Crinului, nr. 22	(null)
2	2Str. Crinului, nr. 22	(null)
3	3Str. Crinului, nr. 22	(null)
4	4Str. Crinului, nr. 22	(null)
5	5Str. Crinului, nr. 22	(null)
6	6Str. Speranței, nr. 34	(null)
7	7Str. Speranței, nr. 34	(null)
8	8Str. Cireșului, nr. 12	(null)
9	9Str. Pinului, nr. 7	(null)
10	10Str. Revoluției, nr. 3	(null)



```

----- Colete -----

CREATE TABLE COLETE
(
    id_colet NUMBER(8, 0) CONSTRAINT PKEY_COLETE PRIMARY KEY,
    id_depozit NUMBER(8, 0) CONSTRAINT colete_id_depozit NOT NULL,
    id_comanda NUMBER(8, 0) CONSTRAINT colete_id_comanda NOT NULL,
    data_preluare DATE CONSTRAINT colete_data_preluare NOT NULL,
    data_livrare DATE,
    greutate NUMBER(4, 0),
    volum NUMBER(4, 0),
    CONSTRAINT FK_COLETE_DEPOZITE FOREIGN KEY (id_depozit) REFERENCES DEPOZITE(id_depozit),
    CONSTRAINT FK_COLETE_COMENZI FOREIGN KEY (id_comanda) REFERENCES COMENZI(id_comanda)
);

-- Secvență

CREATE SEQUENCE SECV_COLETE
INCREMENT BY 1
START WITH 1
NOCYCLE
ORDER;

INSERT INTO COLETE VALUES (SECV_COLETE.nextval, 1, 2, TO_DATE('2020-06-05', 'YYYY-MM-DD'), NULL, 2, 1);
INSERT INTO COLETE VALUES (SECV_COLETE.nextval, 2, 5, TO_DATE('2020-03-05', 'YYYY-MM-DD'), NULL, 10, 5);
INSERT INTO COLETE VALUES (SECV_COLETE.nextval, 1, 4, TO_DATE('2020-06-03', 'YYYY-MM-DD'), NULL, 2, 1);
INSERT INTO COLETE VALUES (SECV_COLETE.nextval, 3, 10, TO_DATE('2021-06-17', 'YYYY-MM-DD'), NULL, 4, 3);
INSERT INTO COLETE VALUES (SECV_COLETE.nextval, 4, 3, TO_DATE('2019-04-02', 'YYYY-MM-DD'), NULL, 5, 1);

SELECT * FROM COLETE;

```

SQL | All Rows Fetched: 5 in 0.007 seconds

ID_COLET	ID_DEPOZIT	ID_COMANDA	DATA_PRELUARE	DATA_LIVRARE	GREUTATE	VOLUM
1	1	1	2 05-JUN-20	(null)	2	1
2	2	2	5 05-MAR-20	(null)	10	5
3	3	1	4 03-JUN-20	(null)	2	1
4	4	3	10 17-JUN-21	(null)	4	3
5	5	4	3 02-APR-19	(null)	5	1

```

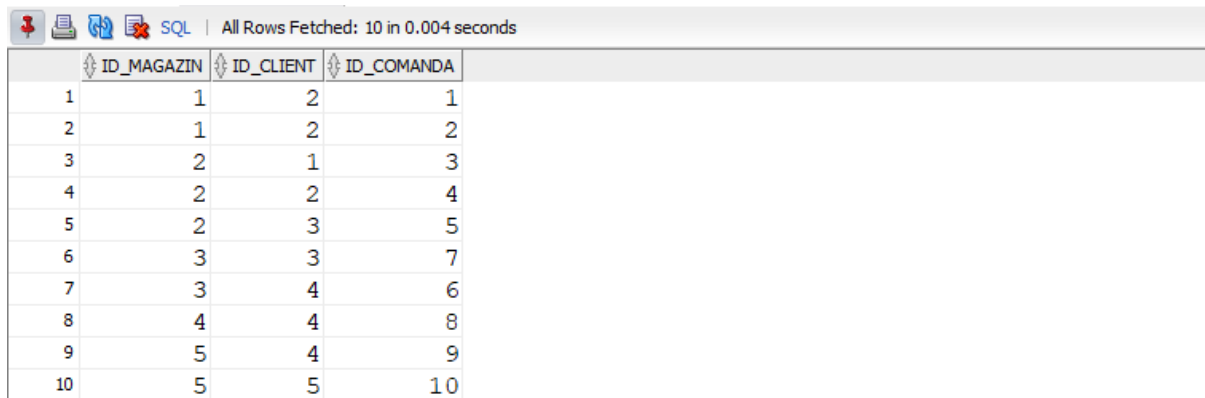
----- Comanda_Magazin -----

CREATE TABLE COMANDA_MAGAZIN
(
    id_magazin NUMBER(8, 0),
    id_client NUMBER(8, 0),
    id_comanda NUMBER(8, 0),
    CONSTRAINT PK_COMANDA_MAGAZIN PRIMARY KEY (id_magazin, id_client, id_comanda),
    CONSTRAINT FK_COMANDA_MAGAZIN_MAGAZINE FOREIGN KEY (id_magazin) REFERENCES MAGAZINE(id_magazin),
    CONSTRAINT FK_COMANDA_MAGAZIN_CLIENTI FOREIGN KEY (id_client) REFERENCES CLIENTI(id_client),
    CONSTRAINT FK_COMANDA_MAGAZIN_COMENZI FOREIGN KEY (id_comanda) REFERENCES COMENZI(id_comanda)
);

INSERT INTO COMANDA_MAGAZIN VALUES (1, 2, 1);
INSERT INTO COMANDA_MAGAZIN VALUES (1, 2, 2);
INSERT INTO COMANDA_MAGAZIN VALUES (2, 1, 3);
INSERT INTO COMANDA_MAGAZIN VALUES (2, 2, 4);
INSERT INTO COMANDA_MAGAZIN VALUES (2, 3, 5);
INSERT INTO COMANDA_MAGAZIN VALUES (3, 4, 6);
INSERT INTO COMANDA_MAGAZIN VALUES (3, 3, 7);
INSERT INTO COMANDA_MAGAZIN VALUES (4, 4, 8);
INSERT INTO COMANDA_MAGAZIN VALUES (5, 4, 9);
INSERT INTO COMANDA_MAGAZIN VALUES (5, 5, 10);

SELECT * FROM COMANDA_MAGAZIN;

```



	ID_MAGAZIN	ID_CLIENT	ID_COMANDA	
1	1	2	1	
2	1	2	2	
3	2	1	3	
4	2	2	4	
5	2	3	5	
6	3	3	7	
7	3	4	6	
8	4	4	8	
9	5	4	9	
10	5	5	10	

----- Produse -----

```
CREATE TABLE PRODUSE
```

```
(
  id_produs NUMBER(8, 0) CONSTRAINT PK_PRODUSE PRIMARY KEY,
  ambalaj VARCHAR2(50),
  pret NUMBER(8, 0) CONSTRAINT produse_pret NOT NULL,
  dimensiune VARCHAR2(50),
  greutate NUMBER(8, 0),
  descriere VARCHAR2(50)
);
```

```
CREATE SEQUENCE SECV_PRODUSE
```

```
INCREMENT BY 1
```

```
START WITH 1
```

```
NOCYCLE
```

```
ORDER;
```

```
INSERT INTO PRODUSE VALUES (SECV_PRODUSE.nextval, 'plastic', 75, '18 cm x 12 cm x 2 cm', 50, 'Husa telefon');
INSERT INTO PRODUSE VALUES (SECV_PRODUSE.nextval, 'carton', 450, '21 cm x 5 cm x 1 cm', 250, 'Smartwatch');
INSERT INTO PRODUSE VALUES (SECV_PRODUSE.nextval, 'plastic', 1800, '18 cm x 12 cm x 2 cm', 450, 'Smartphone');
INSERT INTO PRODUSE VALUES (SECV_PRODUSE.nextval, 'silicon', 5000, '30 cm x 20 cm x 5 cm', 1800, 'Ultrabook');
INSERT INTO PRODUSE VALUES (SECV_PRODUSE.nextval, 'plastic', 25, '200 cm x 500 mm x 500 mm', 30, 'Cablul USB-C');
```

```
SELECT * FROM PRODUSE;
```

ID_PRODUS	AMBALAJ	PRET	DIMENSIUNE	GREUTATE	DESCRIERE
1	1plastic	75	18 cm x 12 cm x 2 cm	50	Husa telefon
2	2carton	450	21 cm x 5 cm x 1 cm	250	Smartwatch
3	3plastic	1800	18 cm x 12 cm x 2 cm	450	Smartphone
4	4silicon	5000	30 cm x 20 cm x 5 cm	1800	Ultrabook
5	5plastic	25	200 cm x 500 mm x 500 mm	30	Cablul USB-C

----- Cantitati -----

```
CREATE TABLE CANTITATI
(
    id_comanda NUMBER(8, 0),
    id_produș NUMBER(8, 0),
    cantitate NUMBER(4, 0) DEFAULT 1,
    CONSTRAINT PK_CANTITATI PRIMARY KEY (id_comanda, id_produș),
    CONSTRAINT FK_CANTITATI_COMENZI FOREIGN KEY (id_comanda) REFERENCES COMENZI(id_comanda),
    CONSTRAINT FK_CANTITATI_PRODUSE FOREIGN KEY (id_produș) REFERENCES PRODUSE(id_produș)
);
```

```
INSERT INTO CANTITATI VALUES (1, 1, 2);
INSERT INTO CANTITATI VALUES (1, 2, 3);
INSERT INTO CANTITATI VALUES (1, 4, 5);
INSERT INTO CANTITATI VALUES (2, 1, 1);
INSERT INTO CANTITATI VALUES (2, 3, 1);
INSERT INTO CANTITATI VALUES (3, 3, 3);
INSERT INTO CANTITATI VALUES (4, 5, 2);
INSERT INTO CANTITATI VALUES (5, 1, 4);
INSERT INTO CANTITATI VALUES (4, 4, 3);
INSERT INTO CANTITATI VALUES (5, 2, 1);
```

```
SELECT * FROM CANTITATI;
```

SQL | All Rows Fetched: 10 in 0.013 seconds

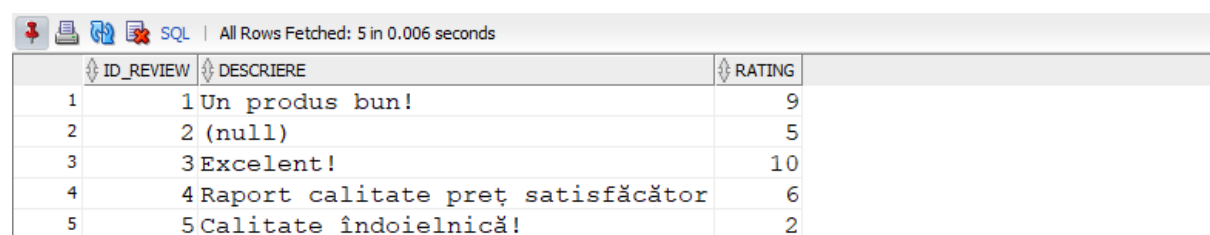
ID_COMANDA	ID_PRODUS	CANTITATE
1	1	2
2	1	3
3	1	5
4	2	1
5	2	1
6	3	3
7	4	2
8	5	4
9	4	3
10	5	1

----- Reviews -----

```
CREATE TABLE REVIEWS
(
    id_review NUMBER(8, 0) CONSTRAINT PK_REVIEWS PRIMARY KEY,
    descriere VARCHAR2(50),
    rating NUMBER(2, 0) CONSTRAINT review_rating_nn NOT NULL,
    CONSTRAINT reviews_rating CHECK (rating > 0 AND rating < 11)
);

INSERT INTO REVIEWS VALUES (1, 'Un produs bun!', 9);
INSERT INTO REVIEWS VALUES (2, NULL, 5);
INSERT INTO REVIEWS VALUES (3, 'Excelent!', 10);
INSERT INTO REVIEWS VALUES (4, 'Raport calitate preț satisfăcător', 6);
INSERT INTO REVIEWS VALUES (5, 'Calitate îndoielnică!', 2);

SELECT * FROM REVIEWS;
```






ID_REVIEW	DESCRIERE	RATING
1	Un produs bun!	9
2	(null)	5
3	Excelent!	10
4	Raport calitate preț satisfăcător	6
5	Calitate îndoielnică!	2

----- Review\_Produs -----

```
CREATE TABLE REVIEW_PRODUS
(
    id_produs NUMBER(8, 0),
    id_client NUMBER(8, 0),
    id_review NUMBER(8, 0),
    CONSTRAINT PK_REVIEW_PRODUS PRIMARY KEY (id_produs, id_client, id_review),
    CONSTRAINT FK_REVIEW_PRODUS_PRODUSE FOREIGN KEY (id_produs) REFERENCES PRODUSE(id_produs),
    CONSTRAINT FK_REVIEW_PRODUS_CLIENTI FOREIGN KEY (id_client) REFERENCES CLIENTI(id_client),
    CONSTRAINT FK_REVIEW_PRODUS_REVIEWS FOREIGN KEY (id_review) REFERENCES REVIEWS(id_review)
);
```

```
INSERT INTO REVIEW_PRODUS VALUES (1, 2, 3);
INSERT INTO REVIEW_PRODUS VALUES (1, 1, 4);
INSERT INTO REVIEW_PRODUS VALUES (2, 4, 1);
INSERT INTO REVIEW_PRODUS VALUES (2, 5, 3);
INSERT INTO REVIEW_PRODUS VALUES (3, 1, 2);
INSERT INTO REVIEW_PRODUS VALUES (3, 3, 1);
INSERT INTO REVIEW_PRODUS VALUES (4, 4, 2);
INSERT INTO REVIEW_PRODUS VALUES (4, 5, 1);
INSERT INTO REVIEW_PRODUS VALUES (5, 2, 1);
INSERT INTO REVIEW_PRODUS VALUES (5, 3, 3);
```

```
SELECT * FROM REVIEW_PRODUS;
```

   | All Rows Fetched: 10 in 0.007 seconds

	ID_PRODUS	ID_CLIENT	ID_REVIEW
1	1	1	4
2	1	2	3
3	2	4	1
4	2	5	3
5	3	1	2
6	3	3	1
7	4	4	2
8	4	5	1
9	5	2	1
10	5	3	3

## Cereri complexe – Ex. 11

----- Exercitii -----

```
-- 1. Formulați în limbaj natural și implementați 5 cereri SQL complexe ce vor utiliza, în
--    ansamblul lor, următoarele elemente:
--    • operație join pe cel puțin 4 tabele
--    • filtrare la nivel de linii
--    • subcereri sincronizate în care intervin cel puțin 3 tabele
--    • subcereri nesincronizate în care intervin cel puțin 3 tabele
--    • grupări de date, funcții grup, filtrare la nivel de grupuri
--    • ordonări
--    • utilizarea a cel puțin 2 funcții pe șiruri de caractere, 2 funcții pe date calendaristice, a
--    funcțiilor NVL și DECODE, a cel puțin unei expresii CASE
--    • utilizarea a cel puțin 1 bloc de cerere (clauza WITH)
```

```
-- 1. Să se afișeze toate comenzile primite de magazinul la care lucrează angajatul Ionescu Doru.
```

```
SELECT cm.id_client, c.id_comanda, c.adresa_livrare
FROM comenzi c
JOIN comanda_magazin cm ON c.id_comanda = cm.id_comanda
JOIN magazine m ON cm.id_magazin = m.id_magazin
WHERE m.id_magazin =
(
    SELECT id_magazin
    FROM angajati a
    WHERE LOWER(a.num) LIKE '%ionescu%' AND TRIM(BOTH ' ' FROM UPPER(a.prenume)) = 'DORU'
)
ORDER BY c.id_comanda;
```

All Rows Fetched: 2 in 0.006 seconds			
ID_CLIENT	ID_COMANDA	ADRESA_LIVRARE	
1	2	1 Str. Crinului, nr. 22	
2	2	2 Str. Crinului, nr. 22	

```
-- 2. Să se afișeze pentru fiecare angajat ce a fost angajat în urmă cu minim 12 ani salariul, data angajării, numele și id-ul managerului, dacă există.

SELECT a.numa || ' ' || a.prenume AS "Nume angajat", TO_CHAR(a.data_angajare, 'DD-MM-YY') AS "Data angajare",
       DECODE(a.id_manager, NULL, 'Nu are manager', 'Are manager') AS "Manager", a.salariu AS "Salariu"
FROM angajati a
JOIN joburi j ON a.id_job = j.id_job
WHERE EXTRACT(YEAR FROM SYSDATE) - EXTRACT(YEAR FROM a.data_angajare) >= 12;
```

SQL All Rows Fetched: 3 in 0.01 seconds

	Nume angajat	Data angajare	Manager	Salariu
1	Popescu Ion	11-10-07	Nu are manager	4000
2	Georgescu Marcel	06-05-10	Nu are manager	2400
3	Popa Mircea	12-08-10	Nu are manager	5000

```
-- 3. Să se afișeze pentru fiecare client media suprafeței depozitelor la care au ajuns coletele din comenzile sale, unde această medie trebuie să fie mai mare de 2000.
```

```
WITH clienti_orase AS
(
  SELECT c.id_client, c.numa, c.prenume, d.capacitate, d.suprafata
  FROM clienti c
  JOIN comanda_magazin cm ON cm.id_client = c.id_client
  JOIN comenzi cz ON cz.id_comanda = cm.id_comanda
  JOIN colete cl ON cl.id_comanda = cz.id_comanda
  JOIN depozite d ON d.id_depozit = cl.id_depozit
)
SELECT id_client, numa, prenume, AVG(suprafata) AS "MEDIE"
FROM clienti_orase
GROUP BY id_client, numa, prenume
HAVING AVG(suprafata) >= 2000
ORDER BY AVG(suprafata) DESC;
```

SQL All Rows Fetched: 2 in 0.016 seconds

	ID_CLIENT	NUMA	PRENUME	MEDIE
1	3	Mircescu Robert	4000	
2	2	Cristescu Ștefan	2000	



-- 4. Să se afișeze pentru fiecare angajat care lucrează la un departament din București dacă acesta are manager.

```
SELECT id_angajat, nume || ' ' || prenume AS "Nume", CASE(NVL(id_manager, -1)) WHEN -1 THEN 'Nu are manager' ELSE 'Are manager' END AS "Manager"
FROM angajati
WHERE id_angajat IN (
    SELECT id_angajat
    FROM angajati a
    JOIN departamente d ON a.id_departament = d.id_departament
    JOIN locatii l ON l.id_locatie = d.id_locatie
    JOIN orase o ON o.id_oras = l.id_oras
    WHERE LOWER(nume_oras) = 'bucurești'
);
```

ID_ANGAJAT	Nume	Manager
1	1 Popescu Ion	Nu are manager
2	2 Georgescu Marcel	Nu are manager
3	3 Ionescu Doru	Nu are manager
4	4 Popa Mircea	Nu are manager
5	5 Titulescu Bogdan	Nu are manager

-- 5. Afișați joburile a căror valoare a salariului minim estimată este respectată.

```
SELECT id_job, nume_job
FROM joburi
WHERE salariu_minim < ALL (
    SELECT salariu
    FROM angajati a
    JOIN magazine m ON a.id_magazin = m.id_magazin
    JOIN locatii l ON l.id_locatie = m.id_locatie
    WHERE a.id_job = id_job
);
```

ID_JOB	NUME_JOB
1	2 Consultant vânzări

## Operații cu subcereri – Ex. 12

```
--2. Implementarea a 3 operații de actualizare sau suprimare a datelor utilizând subcereri.

-- Să se scadă salariul cu un procent de 20% angajaților ce lucrează la magazinele cu număr minim de comenzi.

UPDATE angajati
SET salariu = salariu * 0.8
WHERE id_magazin IN (
    SELECT cm.id_magazin
    FROM comanda_magazin cm
    GROUP BY cm.id_magazin
    HAVING COUNT(cm.id_comanda) = (
        SELECT MIN(COUNT(id_comanda))
        FROM comanda_magazin
        GROUP BY id_magazin
    )
);

-- Să se dubleze prețul produselor ce au o medie a rating-ului mai mare de 7 și au o greutate mai mică de 1 kg.

UPDATE produse
SET pret = 2 * pret
WHERE id_produc IN (
    SELECT p.id_produc
    FROM produse p
    JOIN review_produc rp ON rp.id_produc = p.id_produc
    JOIN reviews r ON rp.id_review = r.id_review
    WHERE p.greutate < 1000
    GROUP BY p.id_produc
    HAVING AVG(r.rating) > 7
);

-- Să se șteargă utilizatorii ce nu au avut nicio accesare.

DELETE FROM utilizatori
WHERE id_utilizator NOT IN (
    SELECT DISTINCT id_utilizator
    FROM accesari
);

COMMIT
```

 Task completed in 0.041 seconds

```
1 row updated.

3 rows updated.

2 rows deleted.
```

## Creare secvență – Ex. 13

```
-- Secvență  
  
CREATE SEQUENCE SECV_COLETE  
INCREMENT BY 1  
START WITH 1  
NOCYCLE  
ORDER;
```

## Creare vizualizare compusă – Ex. 14

```
--3. Crearea unei vizualizări compuse. Dați un exemplu de operație LMD permisă pe vizualizarea respectivă și un exemplu de operație LMD nepermisă.

-- Vizualizare care conține date despre clienții ce au făcut minim 2 comenzi fizice (la un magazin, nu online).

CREATE OR REPLACE VIEW vizualizare_clienti_produce
AS
SELECT id_client, nume || ' ' || prenume AS "Nume client", nr_telefon, id_comanda, adresa_livrare
FROM clienti c
JOIN comanda_magazin cm USING (id_client)
JOIN comenzi cz USING (id_comanda)
WHERE id_client IN (
    SELECT id_client
    FROM comanda_magazin
    GROUP BY id_client
    HAVING COUNT(*) > 1
);

UPDATE vizualizare_clienti_produce SET "Nume client" = 'Cristian' WHERE id_client = 1; -- nepermisa, update pe coloana virtuala, nu e updatable

DELETE FROM vizualizare_clienti_produce WHERE id_client = 3; -- permisa, sterge doua linii din comanda_magazin, nu si din comenzi deoarece ON DELETE CASCADE nu e activat

COMMIT
```

Task completed in 0.259 seconds

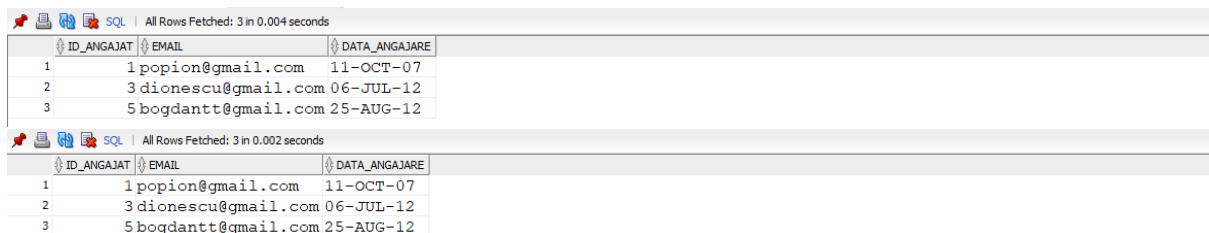
View VIZUALIZARE\_CLIENTI\_PRODUSE created.

```
Error starting at line : 108 in command -
UPDATE vizualizare_clienti_produce SET "Nume client" = 'Cristian' WHERE id_client = 1
Error at Command Line : 108 Column : 40
Error report -
SQL Error: ORA-01733: virtual column not allowed here
01733. 00000 - "virtual column not allowed here"
*Cause:
*Action:

2 rows deleted.
```

## Creare index optimizare – Ex. 15

```
--4. Crearea unui index care să optimizeze o cerere de tip căutare cu 2 criterii. Specificați cererea.  
  
CREATE INDEX idx_email_angajat ON angajati(email);  
  
-- Să se afișeze email-urile angajaților ce conțin litera 'n' și au fost angajați înainte de 15 martie 2015.  
  
SELECT id_angajat, email, data_angajare  
FROM angajati  
WHERE LOWER(email) LIKE '%n%' AND data_angajare < TO_DATE('2015-03-15', 'YYYY-MM-DD');  
  
COMMIT
```



ID_ANGAJAT	EMAIL	DATA_ANGAJARE
1	1popion@gmail.com	11-OCT-07
2	3dionescu@gmail.com	06-JUL-12
3	5bogdanttt@gmail.com	25-AUG-12

ID_ANGAJAT	EMAIL	DATA_ANGAJARE
1	1popion@gmail.com	11-OCT-07
2	3dionescu@gmail.com	06-JUL-12
3	5bogdanttt@gmail.com	25-AUG-12

## Cereri speciale – Ex. 16

```
--5. Formulați în limbaj natural și implementați în SQL: o cerere ce utilizează operația outerjoin pe minimum 4 tabele și două cereri ce utilizează operația division.

-- Să se afișeze pentru fiecare client orașele în care se află magazinele la care a făcut minim o comandă.

SELECT c.id_client, c.numa || ' ' || c.prenume AS "Nume client", o.numa_oras
FROM clienti c
LEFT OUTER JOIN comanda_magazin cm ON c.id_client = cm.id_client
LEFT OUTER JOIN magazine m ON cm.id_magazin = m.id_magazin
LEFT OUTER JOIN locatii l ON m.id_locatie = l.id_locatie
LEFT OUTER JOIN orase o ON l.id_oras = o.id_oras;
```

All Rows Fetched: 10 in 0.005 seconds

ID_CLIENT	Nume client	NUME_ORAS
1	2Cristescu Ștefan	București
2	2Cristescu Ștefan	București
3	4Rotaru David	București
4	1Cristea Mihai	București
5	2Cristescu Ștefan	București
6	3Mircescu Robert	București
7	4Rotaru David	Cluj-Napoca
8	5Nae Cristian	Cluj-Napoca
9	3Mircescu Robert	Cluj-Napoca
10	4Rotaru David	Cluj-Napoca

```
-- Să se afișeze toți angajații al căror departament se află la o locație cu cel mult o clădire.
```

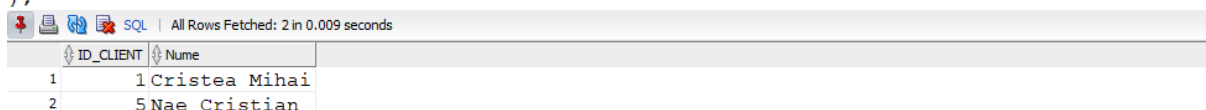
```
SELECT a.id_angajat, a.numa || ' ' || a.prenume AS "Nume"
FROM angajati a
WHERE NOT EXISTS
(
    SELECT 1
    FROM locatii l
    WHERE l.nr_cladiri > 2 AND NOT EXISTS
    (
        SELECT 'c'
        FROM departamente d
        WHERE d.id_locatie = l.id_locatie AND d.id_departament = a.id_departament
    )
);
```

All Rows Fetched: 5 in 0.004 seconds

ID_ANGAJAT	Nume
1	1Popescu Ion
2	2Georgescu Marcel
3	3Ionescu Doru
4	4Popa Mircea
5	5Titulescu Bogdan

-- Să se afișeze toți clienții care au dat cel mult o comandă.

```
SELECT c.id_client, c.nume || ' ' || c.prenume AS "Nume"
FROM clienti c
WHERE NOT EXISTS
(
    SELECT 'x'
    FROM comenzi cz
    WHERE
        (SELECT COUNT(*)
         FROM comanda_magazin cm1
         WHERE cm1.id_client = c.id_client
        ) > 1
    AND
    NOT EXISTS
    (
        SELECT 'y'
        FROM comanda_magazin cm
        WHERE cm.id_client = c.id_client AND cm.id_comanda = cz.id_comanda
    )
);
```



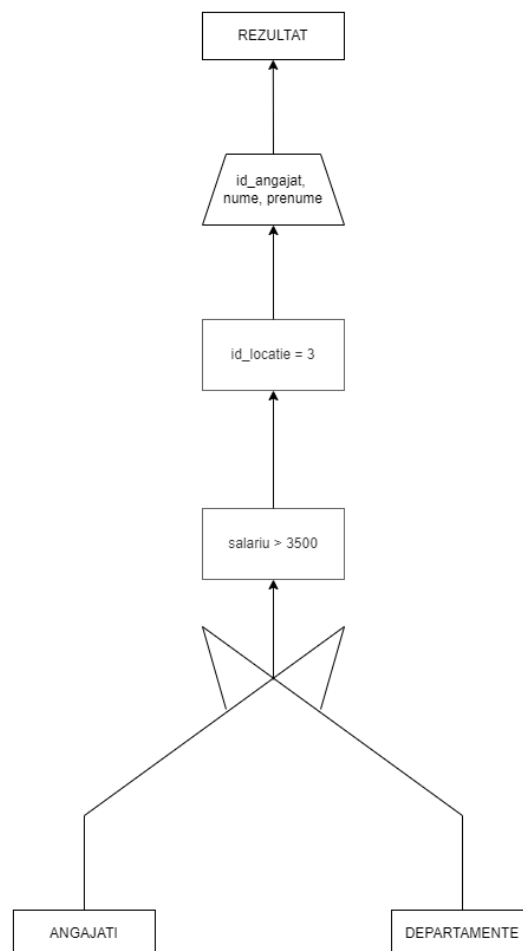
ID_CLIENT	Nume
1	1Cristea Mihai
2	5Nae Cristian

## Optimizare cu algebră relațională – Ex. 17

```
-- Să se afișeze numele angajaților ce lucrează la departamentul ce se află la locația cu id-ul 3 și au salariul strict mai mare de 3500 RON.

-- varianta inițială
SELECT a.id_angajat, a.nume, a.prenume
FROM angajati a
JOIN departamente d ON d.id_departament = a.id_departament
WHERE a.salariu > 3500 AND d.id_locatie = 3;

-- Expresia algebrică:
-- R1 = JOIN(angajati, departamente)
-- R2 = SELECT(R1, salariu > 3500)
-- R3 = SELECT(R2, id_locatie = 3)
-- RESULTAT = R4 = PROJECT(R3, id_angajat, nume, prenume)
```





```

-- varianta optimizată

SELECT t1.id_angajat, t1.nume, t1.prenume
FROM
(
  SELECT id_angajat, nume, prenume, id_departament
  FROM angajati
  WHERE salariu > 3500
) t1
JOIN
(
  SELECT id_departament
  FROM departamente
  WHERE id_locatie = 3
) t2 ON (t1.id_departament = t2.id_departament);

-- Expresia algebrică:
-- R1 = SELECT(angajati, salariu > 3500)
-- R2 = PROJECT(R1, id_angajat, nume, prenume, id_departament)
-- R3 = SELECT(departamente, id_locatie = 3)
-- R4 = PROJECT(R3, id_departament)
-- REZULTAT = R5 = JOIN(R2, R4)

```

ID_ANGAJAT	NUME	PRENUME
1	5 Titulescu	Bogdan
2	4 Popa	Mircea

