

Proiect Baze de Date

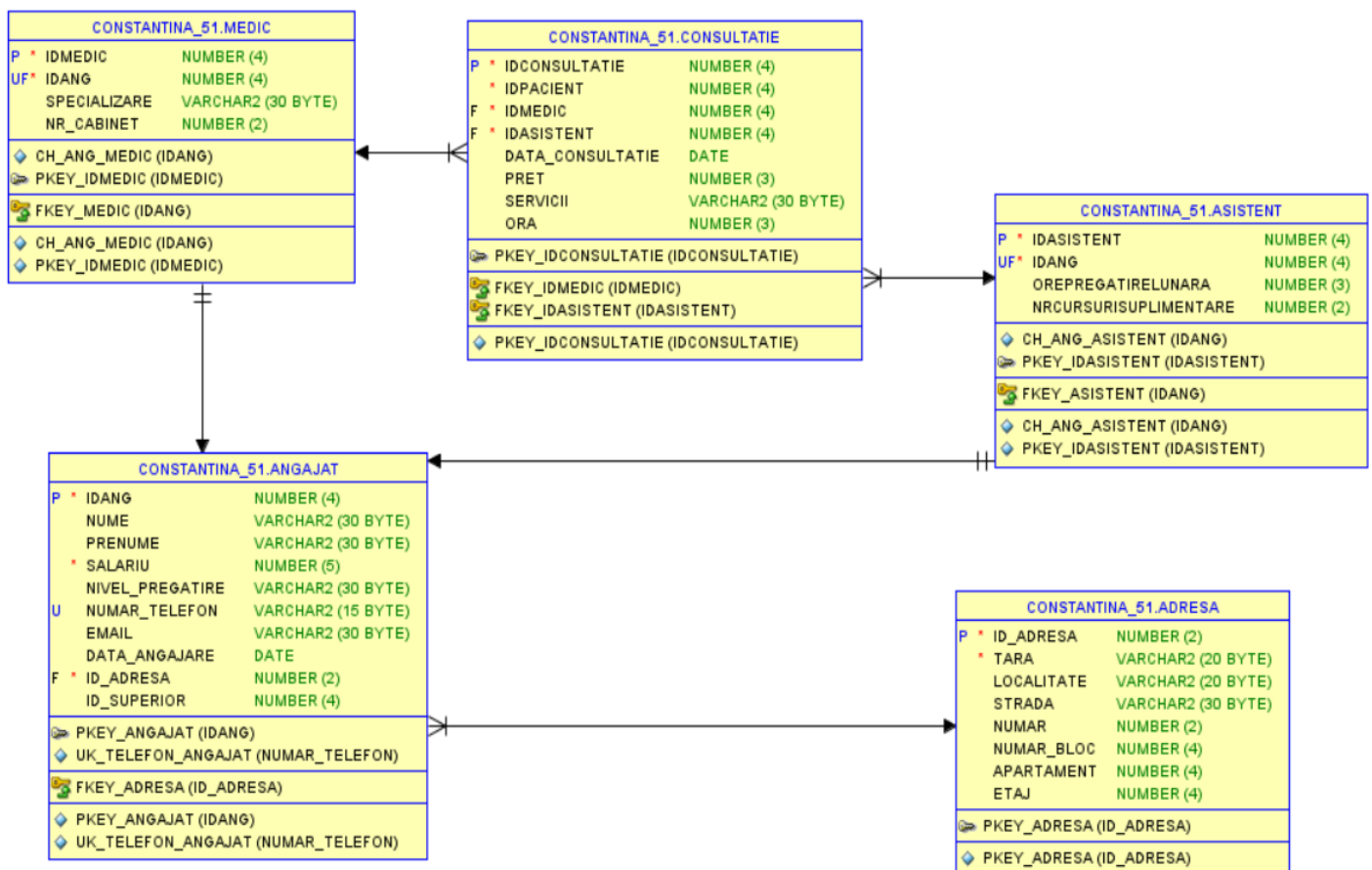
~Constantin Andreea Maria 1051, seria C~

Obiectivul Proiectului

Obiectivul proiectului este reprezentat de realizarea unei baze de date, ce are ca scop administrarea unui centru medical, respectiv a consultațiilor fiecărui pacient în parte. Aceasta permite vizualizarea medicului ce susține consultația, cât și a asistentului acestuia, și bineînțeles, toate datele aferente fiecăruia dintre ei. Aceasta cuprinde 6 tabele, respectiv: Pacient, Consultatie, Angajat, Medic, Asistent și Adresă

Modelul Conceptual al Bazei de Date

Pentru o examinare mai în detaliu a schemei bazei de date, [CLICK AICI](#).



LEGĂTURILE DINTRE TABELE

- **Relația Pacient - Consultație**

Între tabela Pacient și tabela Consultație este o relație ONE-TO-MANY, deoarece un pacient se poate programa la mai multe consultații în cadrul centrului medical. Relația este obligatorie, întrucât Pacienții sunt înregistrați numai în urma programării unei consultații.

- **Relația Medic - Consultație**

Între tabela Medic și tabela Consultație este o relație ONE-TO-MANY, deoarece un medic poate să asiste la mai multe consultații, iar fiecare consultație necesită un singur medic. Relația este obligatorie, întrucât nu există o consultație care să se susțină în absența unui medic, iar medicul, atât timp cât este angajat în cadrul centrului medical, este obligat să susțină consultații.

- **Relația Asistent - Consultație**

Între tabela Asistent și tabela Consultație este o relație ONE-TO-MANY, deoarece un asistent poate fi asignat mai multor consultații, iar o consultație necesită un singur asistent. Relația este obligatorie, întrucât nu există o consultație care să se susțină în absența unui asistent, iar asistentul, atât timp cât este angajat în cadrul centrului medical, este obligat să asiste la consultații.

- **Relația Adresa - Pacient**

Între tabela Adresa și tabela Pacient este o relație ONE-TO-MANY, deoarece considerăm adresa pacientului înregistrată în C.I. În schimb, este luată posibilitatea ca mai mulți pacienți să aibă înregistrați aceeași adresă, întrucât împart același domiciliu. Adresa este obligatorie, pentru că înregistrarea unui pacient este valabilă doar dacă conține și adresa acestuia.

- **Relația Adresa - Angajat**

Între tabela Adresa și tabela Angajat este o relație ONE-TO-MANY, deoarece considerăm adresa angajatului înregistrată în C.I. În schimb, este luată posibilitatea ca mai mulți angajați să aibă înregistrați aceeași adresă, întrucât împart același domiciliu. Adresa este obligatorie, pentru că înregistrarea unui pacient este valabilă doar dacă conține și adresa acestuia.

- **Relația Angajat-Medic și Relația Angajat-Asistent**

Între tabela Angajat și tabela Medic este o relație ONE-TO-ONE. Este utilizată tabela Angajat pentru a stoca datele comune între tabela Medic și tabela Asistent. Astfel, un angajat poate fi doar medic sau doar asistent, și bineînțeles, este obligatoriu una dintre cele două, întrucât am considerat că baza de date va gestiona strict personalul medical al centrului medical.

Definirea schemei bazei de date – tabele, restricții de integritate (CREATE, ALTER, DROP)

Notă: la crearea tabelor, unele câmpuri nu au toate constrângerile necesare pentru a întruni schema bazei de date, acestea fiind adăugate ulterior.

1. Crearea tabelii ADRESĂ

Atributele tabelii și tipul de date al fiecăruia în parte:

- id_adresa - de tip number, primary key
- tara - de tip varchar2
- localitate - de tip varchar2
- strada - de tip varchar2
- numar - de tip number
-

Comenzile SQL:

```
create table ADRESA (id_adresa number(2)
constraint PKey_adresa primary key,
tara varchar2(20) not null,
localitate varchar2(20), strada varchar2(15),
numar number(2));
```

2. Crearea tabelii ANGAJAT

Atributele tabelii și tipul de date al fiecăruia în parte:

- idAng - de tip number, primary key
- nume - de tip varchar2
- prenume - de tip varchar2
- salariu - de tip number
- nivel_pregatire - de tip varchar2
- numar_telefon - de tip varchar2
- email - de tip varchar2
- data_angajare - de tip date
- id_superior
- id_adresa - de tip number, cu foreign key-ul FKey_adresă, care face legătura cu tabela ADRESA

Comenzile SQL:

```
create table ANGAJAT
(idAng number(4) constraint PKey_angajat primary key,
nume varchar2(30), prenume varchar2(30),
salariu number(5), nivel_pregatire varchar2(30),
numar_telefon varchar2(15), email varchar2(30),
data_angajare date,
id_adresa number(2),
id_superior number(4),
constraint FKey_adresa foreign key(id_adresa) references adresa(id_adresa));
```

3. Crearea tabelii MEDIC

Atributele tabelii și tipul de date al fiecăruia în parte:

- idMedic - de tip number, primary key
- idAng - de tip number, cu foreign key-ul FKey_medic, face legătura cu tabela ANGAJAT
- specializare - de tip varchar2
- nr_cabinet - de tip number

Comezile SQL:

```
create table MEDIC
(idMedic number(2) constraint PKey_idMedic primary key,
idAng number(2),
specializare varchar2(30),
nr_cabinet number(2),
constraint FKey_medic foreign key(idAng) references angajat(idAng)
);
```

4. Crearea tabelii ASISTENT

Atributele tabelii și tipul de date al fiecăruia în parte:

- idAsistent - de tip number, primary key
- idAng - de tip number, cu foreign key-ul FKey_asistent, face legătura cu tabela ANGAJAT
- orePregatireLunara - de tip number
- nrCursuriSuplimentare - de tip number

Comenzile SQL:

```
create table ASISTENT
(idAsistent number(2) constraint PKey_idAsistent primary key,
idAng number(2),
orePregatireLunara number(3),
nrCursuriSuplimentare number(2),
constraint FKey_asistent foreign key(idAng) references angajat(idAng)
);
```

5. Crearea tabelii PACIENT

Atributele tabelii și tipul de date al fiecăruia în parte:

- idPacient - de tip number, primary key
- nume - de tip varchar2
- prenume - de tip varchar2
- varsta - de tip number
- email - de tip varchar2
- telefon - de tip varchar2
- id_adresă - de tip number, cu foreign key-ul FKey_idAdresaPacient, care face legătura cu tabela ADRESA

Comenzile SQL:

```
create table PACIENT
(idPacient number(4) constraint PKey_idPacient primary key,
nume varchar2(30),
prenume varchar2(30),
varsta number(3),
email varchar2(30),
telefon varchar2(30),
id_adresa number(2),
data_nasterii date default sysdate,
constraint FKey_idAdresaPacient foreign key(id_adresa)
references adresa(id_adresa)
);
```

6. Crearea tabelii CONSULTAȚIE

Atributele tabelii și tipul de date al fiecăruia în parte:

- idConsultație - de tip number, primary key
- idPacient - de tip number, foreign key-ul FKey_idPacient, pentru a face legătura cu tabela PACIENT
- idMedic - de tip number, cu foreign key-ul FKey_idMedic, pentru a face legătura cu tabela MEDIC
- idAsistent - de tip number, cu foreign key-ul FKey_idAsistent, pentru a face legătura cu tabela ASISTENT
- data_consultatie - de tip date, default-ul fiind data curentă (sysdate)
- ora - de tip number
- pret - de tip number
- servicii - de tip varchar

Comenzile SQL:

```
create table CONSULTATIE
(idConsultatie number(4)
constraint PKey_idConsultatie primary key,
idPacient number(4),
idMedic number(2), idAsistent number(2),
data_consultatie date default sysdate,
pret number(3), servicii varchar2(30),
constraint FKey_idPacient foreign key(idPacient)
references PACIENT(idPacient),
constraint FKey_idMedic foreign key(idMedic)
references MEDIC(idMedic),
constraint FKey_idAsistent foreign key(idAsistent)
references ASISTENT(idAsistent)
);
```

7. Să se adauge restricția ca email-ul să conțină "@gmail.com" sau "@yahoo.com" la final

Comenzile SQL:

```
alter table pacient add constraint CK_EMAIL check
(email like '%@gmail.com' or email like '%yahoo.com');
```

8. Să se adauge constrângerea ca fiecare număr de telefon este unic, atât pentru pacienți, cât și pentru angajați

Comenzile SQL:

```
alter table pacient add constraint UK_TELEFON_Pacient unique(telefon);
alter table angajat add constraint UK_TELEFON_Angajat unique(numar_telefon);
```

9. Să se adauge constrângerea ca salariul angajatului să nu fie null

Comenzile SQL:

```
alter table angajat modify salariu not null;
```

10. Să se adauge constrângerea ca tipul de servicii aferent unei consultații să se încadreze în următoarea listă: { consultație, analize, operație }.

Comenzile SQL:

```
alter table CONSULTATIE add constraint CH_SERVICII check
( SERVICII IN ('analize','operatie', 'consultatie'));
```

11. Să se adauge constrângerea ca nivelul de pregătire al unui angajat să fie în următoarea listă: { licență, masterat, doctorat }.

Comenzile SQL:

```
alter table ANGAJAT add constraint CH_NivelStudii  
check ( NIVEL_PREGATIRE IN ('licenta', 'master', 'doctorat'));
```

12. Să se adauge constrângerea ca specializarea medicului să se regăsească în următoarea listă: { oncolog, pediatru, ortoped, medic generalist, urolog, dermatolog, neurolog, nefrolog, gastroenterolog, radiolog, cardiolog, chirurg, reumatolog}.

Comenzile SQL:

```
alter table MEDIC add constraint CH_Specializare check (  
SPECIALIZARE IN ('oncolog', 'pediatru', 'ortoped',  
'medic generalist', 'urolog', 'dermatolog', 'neurolog',  
'nefrolog', 'gastroenterolog', 'radiolog', 'cardiolog',  
'chirurg', 'reumatolog'));
```

13. Să se dezactiveze și să se activeze la loc constrângerea ca specializarea medicului să se regăsească în listă.

Comenzile SQL:

```
alter table MEDIC disable constraint CH_Specializare;  
alter table MEDIC enable constraint CH_Specializare;
```

14. Să se elimine și să se adauge la loc email-ul pentru fiecare pacient.

Comenzile SQL:

```
alter table PACIENT drop constraint CK_EMAIL;  
alter table PACIENT drop column EMAIL;  
alter table PACIENT add EMAIL VARCHAR2(30);  
alter table PACIENT add constraint CK_EMAIL  
check (EMAIL like '%@gmail.com' or EMAIL like '%yahoo.com');
```

15. Să se adauge următoarele coloane: bloc(de tip number), etaj(de tip number) și apartament(de tip number), tabele ADRESĂ.

Comenzile SQL:


```
alter table ADRESA add numar_bloc number(4);  
alter table ADRESA add apartament number(4);  
alter table ADRESA add etaj number(4);
```

16. Adăugarea coloanei oră în tabela CONSULTAȚIE

Comenzile SQL:

```
alter table consultatie add ora number(3);
```

17. Să se adauge constrângerea de interval orar pentru programarea consultațiilor în programul de muncă (de la ora 8 până la ora 21).

Comenzile SQL:

```
alter table consultatie add constraint CK_ORA check(ora >= 8 and ora <= 21);
```

18. Să se modifice tipul coloanei email, în varchar(50).

Comenzile SQL:

```
alter table pacient  
modify (email VARCHAR2(50));
```

19. Să se șteargă și să se recupereze tabela PACIENT.

Comenzile SQL:

```
drop table PACIENT cascade constraints;  
flashback table PACIENT to before drop;
```

20. Să se adauge constrângerea ca id-ul angajatului să fie unic pentru tabela medic și cea a angajatului. Astfel, se crează relația one-to-one.

Comenzile SQL:

```
alter table MEDIC add constraint CH_ANG_MEDIC unique(idAng);  
alter table ASISTENT add constraint CH_ANG_ASISTENT unique(idAng);
```

21. Să se adauge constrângerile not null, pentru a crea legături obligatorii între tabele, conform celor din schema conceptuală.

Comenzile SQL:

```
alter table MEDIC modify (idAng not null);  
alter table ASISTENT modify (idAng not null);  
alter table ANGAJAT modify (id_adresa not null);  
alter table PACIENT modify (id_adresa not null);  
alter table CONSULTATIE modify (idPacient not null);  
alter table CONSULTATIE modify (idMedic not null);  
alter table CONSULTATIE modify (idAsistent not null);
```

Exemple cu operațiile de actualizare a datelor

(INSERT, UPDATE, DELETE).

22. Popularea tabelii ADRESA

Pentru adăugarea înregistrărilor în tabela ADRESA folosim următorul șablon:

INSERT INTO adresa VALUES(id, 'Țara', 'Oraș', 'Strada', nr_stradă);

Exemplu de înregistrări adăugate în tabelă:

```
insert into adresa values(1, 'Romania', 'Bucuresti', 'Strada Sperantei', 3);
insert into adresa values(2, 'Romania', 'Bucuresti', 'Aleea Gornesti', 1);
insert into adresa values(3, 'Romania', 'Ilfov', 'Strada Leordeni', 15);
insert into adresa values(4, 'Romania', 'Bucuresti', 'Strada Lacramioarei', 4);
insert into adresa values(5, 'Romania', 'Bucuresti', 'Aleea Ghiocailor', 13);
insert into adresa values(6, 'Romania', 'Bucuresti', 'Strada Gheorghe Posu', 43);
insert into adresa values(7, 'Romania', 'Bucuresti', 'Bulevardul Roman', 33);
insert into adresa values(8, 'Romania', 'Bucuresti', 'Strada Tudor Arghezi', 23);
```

23. Popularea tabelii ANGAJAT

Pentru adăugarea înregistrărilor în tabela ANGAJAT folosim următorul șablon:

*INSERT INTO angajat VALUES(id, 'Nume', 'Prenume', salariu, 'nivel_pregatire', 'numar_telefon',
'email', TO_DATE('01/01/2023', 'dd/mm/yyyy'), id_adresă, id_superor);*

Exemplu de înregistrări adăugate în tabelă:

```
insert into angajat values(1, 'Constantin', 'Andreea', 5000, 'Masterat', '0727590226',  
'costareebsbwa@gmail.com', to_date('28/01/2003', 'dd/mm/yyyy'), 1, 56);
insert into angajat values(2, 'Carasel', 'Valeriu', 3000, 'Licenta', '0727390246',  
'ce45e4tea@gmail.com', to_date('24/01/2021', 'dd/mm/yyyy'), 35, 1);
insert into angajat values(3, 'Caravan', 'Elena', 2000, 'Licenta', '0727520246',  
'cost4e5ee545wa@gmail.com', to_date('23/04/2021', 'dd/mm/yyyy'), 2, 10);
insert into angajat values(4, 'Chiriac', 'Madalina', 8000, 'Masterat', '0747590326',  
'cose45e45a@gmail.com', to_date('26/12/2004', 'dd/mm/yyyy'), 2, 55);
insert into angajat values(5, 'Tudor', 'Oana', 9000, 'Masterat', '0727390426',  
'c4e5e45bwa@gmail.com', to_date('20/02/2013', 'dd/mm/yyyy'), 3, 54);
```

24. Popularea tabelii MEDIC

Pentru adăugarea înregistrărilor în tabela MEDIC folosim următorul șablon:

INSERT INTO medic VALUES(id_medic, id_angajat, 'specializare', nr_cabinet);

Exemplu de înregistrări adăugate în tabelă:

```
insert into medic values(1, 56, 'pediatru', 30);
insert into medic values(3, 54, 'oncolog', 28);
insert into medic values(4, 53, 'ortoped', 27);
insert into medic values(6, 51, 'medic generalist', 25);
insert into medic values(7, 50, 'urolog', 24);
insert into medic values(8, 49, 'neurolog', 23);]
insert into medic values(11, 46, 'dermatolog', 20);
insert into medic values(12, 45, 'nefrolog', 19);
insert into medic values(14, 43, 'gastroenterolog', 17);
insert into medic values(17, 40, 'oftalmolog', 14);
insert into medic values(19, 38, 'radiolog', 12);
insert into medic values(21, 36, 'cardiolog', 10);]
insert into medic values(23, 34, 'chirurg', 8);
insert into medic values(24, 33, 'anestezist', 7);
insert into medic values(25, 32, 'reumatolog', 6);
insert into medic values(26, 31, 'chirurg', 5);
```

25. Popularea tabelii ASISTENT

Pentru adăugarea înregistrărilor în tabela ASISTENT folosim următorul șablon:

INSERT INTO medic VALUES(idAsistent, idAngajat, orePregatire, nrCursuriLunare);

Exemplu de înregistrări adăugate în tabelă:

```
insert into asistent values(1, 29, 65, 4);
insert into asistent values(2, 28, 75, 4);
insert into asistent values(3, 27, 45, 3);
insert into asistent values(4, 26, 25, 3);
insert into asistent values(5, 25, 43, 2);
```

26. Popularea tabelii CONSULTAȚIE

Pentru adăugarea înregistrărilor în tabela CONSULTATIE folosim următorul șablon:

*INSERT INTO consultatie VALUES(idConsultatie, idMedic, idAsistent,
TO_DATE('01/01/2023', 'dd/mm/yyyy'), ora, pret, 'servicii');*

Exemplu de înregistrări adăugate în tabelă:

```

insert into consultatie values(1, 45, 1, 1, to_date('12/02/2023','dd/mm/yyyy'), 14, 564, 'consultatie');
insert into consultatie values(2, 44, 27, 29, to_date('12/02/2023','dd/mm/yyyy'), 15, 900, 'operatie');
insert into consultatie values(3, 43, 26, 28, to_date('12/02/2023','dd/mm/yyyy'), 12, 900, 'operatie');
insert into consultatie values(4, 42, 25, 27, to_date('13/02/2023','dd/mm/yyyy'), 8, 300, 'analize');
insert into consultatie values(5, 41, 24, 26, to_date('13/02/2023','dd/mm/yyyy'), 9, 300, 'analize');
insert into consultatie values(6, 40, 23, 25, to_date('13/02/2023','dd/mm/yyyy'), 16, 564, 'consultatie');
insert into consultatie values(7, 39, 22, 24, to_date('14/02/2023','dd/mm/yyyy'), 17, 300, 'analize');
insert into consultatie values(8, 38, 21, 23, to_date('14/02/2023','dd/mm/yyyy'), 18, 564, 'consultatie');
insert into consultatie values(9, 37, 20, 22, to_date('14/02/2023','dd/mm/yyyy'), 10, 300, 'analize');
insert into consultatie values(10, 36, 19, 21, to_date('15/02/2023','dd/mm/yyyy'), 9, 564, 'consultatie');
insert into consultatie values(11, 35, 18, 20, to_date('15/02/2023','dd/mm/yyyy'), 11, 300, 'analize');

```

27. Popularea tabelii PACIENT

Pentru adăugarea înregistrărilor în tabela PACIENT folosim următorul șablon:

```

INSERT INTO pacient VALUES(idPacient, 'Nume',
                             'Prenume', varsta, 'email', 'telefon', idAdresa);

```

Exemplu de înregistrări adăugate în tabelă:

```

insert into pacient values(1, 'Constantin', 'Andreea', 20, 'costareebswa@gmail.com', '0727590226',
                           35, to_date('28/01/2003','dd/mm/yyyy'));
insert into pacient values(2, 'Carasel', 'Valeriu', 20, 'variu13@gmail.com', '0727550436',
                           34, to_date('08/08/2002','dd/mm/yyyy'));
insert into pacient values(3, 'Cojocaru', 'Flavius', 21, 'flaviuscojocaru@gmail.com', '0727698265',
                           33, to_date('24/01/2002','dd/mm/yyyy'));
insert into pacient values(4, 'Caravan', 'Elena', 20, 'hellen@gmail.com', '0727390118',
                           32, to_date('12/04/2003','dd/mm/yyyy'));
insert into pacient values(5, 'Nicolae', 'Diana', 20, 'diananico@gmail.com', '0727777090',
                           31, to_date('17/09/2003','dd/mm/yyyy'));
insert into pacient values(6, 'Visan', 'Rares', 20, 'raresvisan@gmail.com', '0727540281',
                           30, to_date('02/11/2003','dd/mm/yyyy'));
insert into pacient values(7, 'Tudor', 'Gabriel', 20, 'tudorgabriel@gmail.com', '0737888226',
                           29, to_date('23/10/2003','dd/mm/yyyy'));

```

28. Să se selecteze etajul astfel:

- adresele cu id-ul în intervalul [1-10] -> etajul 1
- adresele cu id-ul în intervalul [10-20] -> etajul 2
- adresele cu id-ul în intervalul (20-30] -> etajul 3
- adresele cu id-ul în intervalul (30-40] -> etajul 4
- adresele cu id-ul în intervalul (40-50] -> etajul 5
- adresele cu id-ul în intervalul (50-60] -> etajul 6
- adresele cu id-ul în intervalul (60-70] -> etajul 7
- adresele cu id-ul în intervalul (70-80] -> etajul 8
- adresele cu id-ul în intervalul (80-90] -> etajul 9
- adresele cu id-ul în intervalul [1-10] -> etajul 10

Comenzile SQL:

```
update adresa set etaj = 1 where id_adresa>=1 and id_adresa<=10;  
update adresa set etaj = 2 where id_adresa>10 and id_adresa<=20;  
update adresa set etaj = 3 where id_adresa>20 and id_adresa<=30;  
update adresa set etaj = 4 where id_adresa>30 and id_adresa<=40;  
update adresa set etaj = 5 where id_adresa>40 and id_adresa<=50;  
update adresa set etaj = 6 where id_adresa>50 and id_adresa<=60;  
update adresa set etaj = 7 where id_adresa>60 and id_adresa<=70;  
update adresa set etaj = 8 where id_adresa>70 and id_adresa<=80;  
update adresa set etaj = 9 where id_adresa>80 and id_adresa<=90;  
update adresa set etaj = 10 where id_adresa>90 and id_adresa<=100;
```

29. Să se șteargă adresele care nu se întâlnesc la niciun pacient sau la niciun angajat

Comenzile SQL:

```
delete from ADRESA  
where id_adresa not in(select id_adresa from PACIENT)  
or id_adresa not in(select id_adresa from ANGAJAT);
```

Interogarea tabelor

30. Să se afișeze toate datele aferente pacientului cu id-ul 3.

Comenzile SQL:

```
select * from pacient  
where idPacient = 3;
```

Rezultatul:

IDPACIENT	IDCONSULTATIE	NUME	PRENUME	VARSTA	TELEFON	ID_ADRESA	DATA_NASTERII	EMAIL
1	3	43 Cojocaru	Flavius	21	0727698265	33	24-JAN-02	flaviuscojocaru@gmail.com

31. Să se afișeze numele, prenumele și salariul angajaților cu salariul între 3000 și 4000 RON.

Comenzile SQL:

```
select nume, prenume, salariu from angajat  
where salariu between 3000 and 4000;
```

Rezultatul:

	NUME	PRENUME	SALARIU
1	Carasel	Valeriu	3000
2	Constantin	Daniel	4000
3	Diaconescu	Maria	4000
4	Elen	Paula	3000
5	Telespan	Claudia	4000
6	Dobrincu	Sebastian	3000
7	Georgian	Alexandru	4000
8	Margineanu	Matei	4000
9	Molnar	Alexandra	4000
10	Stanica	Alexandru	3000
11	Nistor	Cosmin	3000
12	Hingan	Iulia	3000
13	Ivan	Maria	4000
14	Badea	Mircea	4000
15	Foriu	Tristian	3000
16	Verni	Stoi	3000

32. Să se afișeze specializarea medicilor cu data angajării înainte de 2008.

Comenzile SQL:

```
select a.num, a.prenume, m.specializare
from medic m, angajat a
where a.idAng = m.idAng and
extract (year from a.data_angajare)<2008;
```

Rezultatul:

	NUME	PRENUME	SPECIALIZARE
1	Foriu	Tristian	pediatru
2	Badea	Mircea	oncolog
3	Mazilu	Mircea	dermatolog
4	Gont	Vlad	oftalmolog
5	Ivan	Maria	radiolog
6	Hingan	Iulia	radiolog
7	Stangaciu	Loredana	cardiolog
8	Stanciu	Andrei	cardiolog
9	Lazar	Tiberiu	chirurg
10	Stanica	Alexandru	chirurg

33. Să se afișeze toate informațiile legate de toate consultațiile mai puțin cele cu tip de servicii "analize", ordonate descrescător, în funcție de prețul acestora.

Comenzile SQL:

```
select * from consultatie
where servicii<>'analize'
order by pret desc;
```

Rezultatul:

	IDCONSULTATIE	IDMEDIC	IDASISTENT	DATA_CONSULTATIE	ORA	PRET	SERVICII
1	2	27	29	12-FEB-23	15	900	operatie
2	3	26	28	12-FEB-23	12	900	operatie
3	27	2	4	20-FEB-23	15	900	operatie
4	26	3	5	20-FEB-23	14	900	operatie
5	12	17	19	15-FEB-23	16	564	consultatie
6	13	16	18	16-FEB-23	17	564	consultatie
7	15	14	16	16-FEB-23	15	564	consultatie
8	16	13	15	17-FEB-23	14	564	consultatie
9	18	11	13	17-FEB-23	8	564	consultatie
10	19	10	12	18-FEB-23	11	564	consultatie
11	20	9	11	18-FEB-23	9	564	consultatie
12	22	7	9	19-FEB-23	17	564	consultatie
13	44	17	15	26-FEB-23	13	564	consultatie
14	24	5	7	19-FEB-23	15	564	consultatie
15	29	2	2	21-FEB-23	8	564	consultatie

34. Să se afișeze numele, prenumele și strada pe care acesta locuiește, împreună cu tipul serviciului solicitat în cadrul consultației recent programate.

Comenzile SQL:


```
select nume || ' ' || prenume || ' locuieste pe strada ' || strada
|| ', nr.' || numar ||
' si este programat in cadrul centrului medical pentru ' || servicii
from pacient p, adresa a, consultatie c
where p.idConsultatie = c.idConsultatie and p.id_adresa = a.id_adresa;
```

Rezultatul:

NUME ' ' PRENUME LOCUIESTEPESTRADA STRADA ',NR.' NUMAR SIESTEPROGRAMATINCADRULCENTRULUIMEDICALPENTRU SERVICII
Baciu Vasile locuieste pe strada Strada Sperantei, nr.3 si este programat in cadrul centrului medical pentru analize
Vasilescu Gigel locuieste pe strada Aleea Gornesti, nr.1 si este programat in cadrul centrului medical pentru consultatie
Danila Daniel locuieste pe strada Aleea Gornesti, nr.1 si este programat in cadrul centrului medical pentru consultatie
Ionescu Calin locuieste pe strada Strada Leordeni, nr.15 si este programat in cadrul centrului medical pentru consultatie
Faurescu Gabriel locuieste pe strada Strada Leordeni, nr.15 si este programat in cadrul centrului medical pentru analize
Ionescu Andrei locuieste pe strada Strada Lacramioarei, nr.4 si este programat in cadrul centrului medical pentru analize
Hristache Alexandra locuieste pe strada Strada Lacramioarei, nr.4 si este programat in cadrul centrului medical pentru c...
Borta Rares locuieste pe strada Aleea Ghiocilor, nr.13 si este programat in cadrul centrului medical pentru consultatie

35. Să se calculeze diferit discountul aferent fiecărei consultații pentru pacienții clinicii astfel:

- dacă pacientul a fost programat la o singura consultatie atunci discountul este de 10% ;
- dacă pacientul a fost programat la două consultații atunci discountul este de 15%;
- dacă pacientul a fost programat la trei consultații atunci discountul este de 20%.

Din acestea să se elimine înregistrările încheiate de clienții care încep cu litera C.

Comenzile SQL:

```
select p.numa || ' ' || p.prenume,
(case when count(c.idPacient)=1 then 0.1
when count(c.idPacient)=2 THEN 0.15
when count(c.idPacient)>=3 THEN 0.2
else 0 end) discount from consultatie c, pacient p
where c.idPacient=p.idPacient group by p.prenume, p.numa
minus
select p.numa || ' ' || p.prenume,
(case when count(c.idPacient)=1 then 0.1
when count(c.idPacient)=2 THEN 0.15
when count(c.idPacient)>=3 THEN 0.2
else 0 end) discountfrom consultatie c, pacient p
where c.idPacient=p.idPacient and nume like 'C%' group by p.prenume, p.numa;
```

Rezultatul:

	P.NUME ' ' P.PRENUME	DISCOUNT
1	Anton Luana	0.1
2	Antonescu Gabriela	0.1
3	Baciu Vasile	0.1
4	Bagarea Erika	0.1
5	Baltac Alexandru	0.15
6	Boboia Tiberiu	0.15
7	Boja Victor	0.1
8	Bojan Paul	0.1
9	Borta Rares	0.1
10	Danila Daniel	0.1
11	Dumitrache Mihaela	0.2
12	Dumitrescu Daniela	0.2
13	Gont Cezin	0.15
14	Hristache Alexandra	0.1
15	Ionescu Andrei	0.1
16	Ionescu Calin	0.1
17	Ivan Vlad	0.15
18	Lazar Maria	0.15
19	Luca Robert	0.1

36. Să se calculeze distinct bonusul de salariu pentru asistenții centrului medical, astfel:

- Dacă este înregistrat la mai mult de 2 consultații bonusul salarial va fi de 300 RON;
- Dacă este înregistrat la mai mult de 4 consultații bonusul salarial va fi de 500 RON;
- Dacă este înregistrat la mai mult de 6 consultații bonusul salarial va fi de 1000 RON.

Comenzile SQL:

```
select ang.num || ' ' || ang.prenume,
(case
when count(c.idAsistent)>=6 then 1000
when count(c.idAsistent)>=4 then 500
when count(c.idAsistent)>=2 then 300
else 0 end
) bonus_salarial
FROM asistent a, consultatie c, angajat ang
WHERE a.idAsistent = c.idAsistent
and a.idAng=ang.idAng
GROUP BY ang.num, ang.prenume;
```

Rezultatul:

	ANG.NUME ' ' ANG.PRENUME	BONUS_SALARIAL
1	Constantin Andreea	0
2	Carasel Valeriu	0
3	Constantin Daniel	0
4	Safaru Cosmin	0
5	Popescu Simona	0
6	Elen Paula	300
7	Dobrincu Sebastian	300
8	Munteanu Cristian	1000
9	Deaconu Alina	0
10	Grigore Diana	0
11	Telespan Laura	300
12	Molnar Alexandra	500
13	Sandulescu Daria	300
14	Margineanu Matei	1000
15	Sfetcu Grigoras	300
16	Scafaru Antonia	300
17	Telespan Claudia	300
18	Georgian Alexandru	500
19	Chiriac Madalina	0

37. Să se calculeze salariul complet al asistenților centrului medical, astfel încât fiecare primește la salariul normal un bonus. În funcție de numărul de consultații ținute, bonusul se calculează astfel:

- Dacă este înregistrat la mai mult de 2 consultații bonusul salarial va fi de 300 RON;
- Dacă este înregistrat la mai mult de 4 consultații bonusul salarial va fi de 500 RON;
- Dacă este înregistrat la mai mult de 6 consultații bonusul salarial va fi de 1000 RON.
- Dacă nu se încadrează în regulile de mai sus, bonusul va fi 0.

Comenzile SQL:

```
select ang.num|| ' ' ||ang.prenume, count(c.idAsistent) numar_consultatii, (1000 + ang.salariu) as salariu_complet
from angajat ang, consultatie c, asistent a where ang.idAng = a.idAng and c.idAsistent = a.idAsistent
group by ang.num, ang.prenume, ang.salariu having count(c.idAsistent)>=6
UNION
select ang.num|| ' ' ||ang.prenume, count(c.idAsistent) numar_consultatii, (500+ ang.salariu) as salariu_complet
from angajat ang, consultatie c, asistent a where ang.idAng = a.idAng and c.idAsistent = a.idAsistent
group by ang.num, ang.prenume, ang.salariu having count(c.idAsistent) in (4, 5)
UNION
select ang.num|| ' ' ||ang.prenume, count(c.idAsistent) numar_consultatii, (300 + ang.salariu) as salariu_complet
from angajat ang, consultatie c, asistent a where ang.idAng = a.idAng
and c.idAsistent = a.idAsistent group by ang.num, ang.prenume, ang.salariu having count(c.idAsistent) in (2,3)
UNION
select ang.num|| ' ' ||ang.prenume, count(c.idAsistent) numar_consultatii, ang.salariu as salariu_complet
from angajat ang, consultatie c, asistent a where ang.idAng = a.idAng and c.idAsistent = a.idAsistent
group by ang.num, ang.prenume, ang.salariu having count(c.idAsistent) <2;
```

Rezultatul:

ANG.NUME ' ' ANG.PRENUME	NUMAR_CONSULTATII	SALARIU_COMPLET
1 Carasel Valeriu	1	3000
2 Caravan Elena	1	2000
3 Chiriac Madalina	1	8000
4 Constantin Andreea	1	5000
5 Constantin Daniel	1	4000
6 Deaconu Alina	1	5000
7 Diaconescu Maria	1	4000
8 Dobrinu Sebastian	2	3300
9 Dumitrescu Vlad	2	8300
10 Elen Paula	2	3300
11 Georgian Alexandru	5	4500
12 Grigore Diana	1	6000
13 Margineanu Matei	7	5000
14 Molnar Alexandra	4	4500
15 Munteanu Cristian	7	7000
16 Pandone Claudiu	11	6000
17 Popescu Simona	1	5000
18 Safaru Cosmin	1	5000
19 Sandulescu Daria	2	6300
20 Scafaru Antonia	2	6300
21 Semanti Bogdan	11	10000
22 Sfetcu Grigoras	2	5300
23 Stanescu Adela	1	5000

38. Să se afișeze asistenții ce vor primi primă pe sărbători. Condițiile pentru obținerea primei, sunt următoarele:

- nrOrePregatire > 45
- nrCursuriSuplimentare > 5

Comenzile SQL:

```
select ang.numa || ' ' || ang.prenume,
count(c.idAsistent) as nrConsultatii,
a.orePregatireLunara, a.nrCursuriSuplimentare
from asistent a, angajat ang, consultatie c
where orePregatireLunara > 45
and ang.idAng = a.idAng and a.idAsistent = c.idAsistent
group by ang.numa, ang.prenume, a.orePregatireLunara, a.nrCursuriSuplimentare
INTERSECT
select ang.numa || ' ' || ang.prenume,
count(c.idAsistent) as nrConsultatii,
a.orePregatireLunara, a.nrCursuriSuplimentare
from asistent a, angajat ang, consultatie c
where nrCursuriSuplimentare > 5
and ang.idAng = a.idAng and a.idAsistent = c.idAsistent
group by ang.numa, ang.prenume, a.orePregatireLunara, a.nrCursuriSuplimentare;
```

Rezultatul:

ANG.NUME ' 'ANG.PRENUME	NRCONSULTATII	OREPREGATIRELUNARA	NRCURSURI SUPPLEMENTARE
1 Carasel Valeriu	1	60	14
2 Caravan Elena	1	61	14
3 Chiriac Madalina	1	62	13
4 Constantin Andreea	1	87	14
5 Constantin Daniel	1	64	12
6 Grigore Diana	1	95	10
7 Popescu Simona	1	55	9
8 Safaru Cosmin	1	85	9
9 Stoenescu Ana	1	75	10
10 Symeonidis Oana	2	95	6
11 Telespan Laura	2	85	6
12 Tudor Oana	1	63	13

39. Se scrie un proiect de lege prin care se dorește creșterea salariilor angajaților cu studii superioare astfel:

- 0,2 * salariu pentru absolvenții de licenta
- 0,4 * salariu pentru absolvenții de masterat
- 0,6 * salariu pentru absolvenții de doctorat

Să se afișeze care ar fi salariile angajaților în urma adoptării acestei legi.

Comenzile SQL:

```
SELECT nume, prenume, nivel_pregatire,
DECODE(UPPER(nivel_pregatire), 'LICENTA' ,
0.2, 'MASTERAT', 0.4, 'DOCTORAT' , 0.6, 0)
as marireSalariu,
(salariu * DECODE(UPPER(nivel_pregatire),
'LICENTA' , 0.2, 'MASTERAT', 0.4, 'DOCTORAT',
0.6, 0) + salariu) as salariu_marit FROM angajat;
```

Rezultatul:

NUME	PRENUME	NIVEL_PREGATIRE	MARIRESALARIU	SALARIU_MARIT
1 Constantin	Andreea	Masterat	0.4	7000
2 Carasel	Valeriu	Licenta	0.2	3600
3 Caravan	Elena	Licenta	0.2	2400
4 Chiriac	Madalina	Masterat	0.4	11200
5 Tudor	Oana	Masterat	0.4	12600
6 Constantin	Daniel	Licenta	0.2	4800
7 Diaconescu	Maria	Masterat	0.4	5600
8 Deaconu	Alina	Masterat	0.4	7000
9 Stanescu	Adela	Masterat	0.4	7000
10 Stoenescu	Ana	Masterat	0.4	8400
11 Grigore	Diana	Masterat	0.4	8400
12 Safaru	Cosmin	Masterat	0.4	7000
13 Popescu	Simona	Masterat	0.4	7000
14 Dumitrescu	Vlad	Masterat	0.4	11200
15 Sfetcu	Grigoras	Masterat	0.4	7000
16 Scafaru	Antonia	Masterat	0.4	8400
17 Elen	Paula	Licenta	0.2	3600
18 Symeonidis	Oana	Licenta	0.2	2400
19 Telespan	Laura	Licenta	0.2	2400

40. Să se afișeze numele, prenumele, id-ul și id-ul superiorului, pentru asistenții inferiori medicului cu id-ul de angajat 56.

Comenzile SQL:

```
SELECT idAng, nume, prenume, id_superior, LEVEL FROM angajat
CONNECT BY PRIOR idAng= id_superior
START WITH idAng = 56;
```

Rezultatul:

IDANG	NUME	PRENUME	ID_SUPERIOR	LEVEL
1	56 Verni	Stoi	(null)	1
2	1 Constantin	Andreea	56	2
3	2 Carasel	Valeriu	1	3

41. Sa se afișeze numele asistenților inferiori medicului cu id-ul de angajat 56, sub forma de organigrama.

Comenzile SQL:

```
SELECT LEVEL, LPAD(' ', LEVEL)|| nume FROM angajat
CONNECT BY PRIOR idAng = id_superior
START WITH idAng= 56;
```

Rezultatul:

LEVEL	LPAD(' ',LEVEL) NUME
1	1 Verni
2	2 Constantin
3	3 Carasel

42. Să se afișeze toți subordonații angajatului cu prenumele 'Stoi'.

Comenzile SQL:

```
SELECT idAng, prenume, salariu, nume,
id_superior, LEVEL FROM angajat
CONNECT BY PRIOR idAng = id_superior
START WITH prenume= 'Stoi'
ORDER BY LEVEL;
```

Rezultatul:

IDANG	PRENUME	SALARIU	NUME	ID_SUPERIOR	LEVEL
1	56 Stoi	3000	Verni	(null)	1
2	1 Andreea	5000	Constantin	56	2
3	2 Valeriu	3000	Carasel	1	3

43. Să se afișeze toți subordonații cu prenumele 'Stoi' fără cei cu salariu mai mare de 4000.

Comenzile SQL:

```
SELECT idAng, nume, id_superior,
salariu, LEVEL FROM angajat
WHERE salariu<4000
CONNECT BY PRIOR idAng = id_superior
START WITH prenume = 'stoi'
ORDER BY LEVEL;
```

Rezultatul:

IDANG	NUME	ID_SUPERIOR	SALARIU	LEVEL
1	56 Verni	(null)	3000	1
2	2 Carasel	1	3000	3

44. Să se afișeze adresele pacienților, chiar acestea nu sunt asociate strict pacienților, ci și angajaților.

Comenzile SQL:

```
SELECT a.* FROM pacient p, adresa a WHERE a.id_adresa = p.id_adresa (+);
```

Rezultatul:

ID_ADRESA	TARA	LOCALITATE	STRADA	NUMAR	NUMAR_BLOC	APARTAMENT	ETAJ
1	1 Romania	Bucuresti	Strada Sperantei	3	51	14	1
2	2 Romania	Bucuresti	Aleea Gornesti	1	56	9	1
3	2 Romania	Bucuresti	Aleea Gornesti	1	56	9	1
4	3 Romania	Ilfov	Strada Leordeni	15	34	5	1
5	3 Romania	Ilfov	Strada Leordeni	15	34	5	1
6	4 Romania	Bucuresti	Strada Lacramioarei	4	6	7	1
7	4 Romania	Bucuresti	Strada Lacramioarei	4	6	7	1
8	5 Romania	Bucuresti	Aleea Ghiocailor	13	77	66	1
9	5 Romania	Bucuresti	Aleea Ghiocailor	13	77	66	1
10	6 Romania	Bucuresti	Strada Gheorghe Posu	43	85	3	1
11	6 Romania	Bucuresti	Strada Gheorghe Posu	43	85	3	1
12	7 Romania	Bucuresti	Bulevardul Roman	33	56	206	1
13	7 Romania	Bucuresti	Bulevardul Roman	33	56	206	1
14	8 Romania	Bucuresti	Strada Tudor Arghezi	23	78	5	1
15	8 Romania	Bucuresti	Strada Tudor Arghezi	23	78	5	1
16	9 Romania	Bucuresti	Aleea Bucuriei	33	30	204	1
17	9 Romania	Bucuresti	Aleea Bucuriei	33	30	204	1

45. Să se afișeze superiorii direcți ai angajaților ce au superiori.

Comenzile SQL:

```
SELECT a.num||'|' are ca superior direct pe: '|s.num  
FROM angajat a, angajat s  
WHERE a.id_superior=s.idAng;
```

Rezultatul:

A.NUME ARECASUPERIORDIRECTPE' S.NUME
1 Carasel are ca superior direct pe: Constantin
2 Caravan are ca superior direct pe: Stoenescu
3 Pandone are ca superior direct pe: Dobrincu
4 Molnar are ca superior direct pe: Stanica
5 Munteanu are ca superior direct pe: Lazar
6 Margineanu are ca superior direct pe: Nistor
7 Semanti are ca superior direct pe: Vrincut
8 Visan are ca superior direct pe: Antonescu
9 Georgian are ca superior direct pe: Stangaciu
10 Dobrincu are ca superior direct pe: Hingan
11 Sandulescu are ca superior direct pe: Ivan
12 Telespan are ca superior direct pe: Gont
13 Telespan are ca superior direct pe: Baltac
14 Symeonidis are ca superior direct pe: Crudu
15 Elen are ca superior direct pe: Horatiu
16 Scafaru are ca superior direct pe: Chetran
17 Sfetcu are ca superior direct pe: Ciausescu

46. Să se afișeze toți pacienții care au aceeași dată de naștere ca pacientul cu id-ul 25.

Comenzile SQL:

```
SELECT * FROM pacient WHERE data_nasterii IN  
(SELECT data_nasterii FROM pacient WHERE idPacient = 25);
```

Rezultatul:

IDPACIENT	NUME	PRENUME	VARSTA	TELEFON	ID_ADRESA	DATA_NASTERII	EMAIL
1	25	Ciausescu	Mircea	21 0724590526	11	07-APR-02	mirceaciasca@gmail.com
2	46	Stanescu	Cristiana	21 0798322343	12	07-APR-02	cristianastanescu@yahoo.com

47. Să se afișeze datele referitoare la angajații ce sunt înregistrați în cadrul centrului medical de la începutul anului 2012.

Comenzile SQL:

```
select * from ANGAJAT where data_angajare  
between to_date( '01/01/2012', 'DD/MM/YYYY') and sysdate;
```


Rezultatul:

IDANG	NUME	PRENUME	SALARIU	NIVEL_PREGATIRE	NUMAR_TELEFON	EMAIL	DATA_ANGAJARE	ID_ADRESA	ID_SUPERIOR
1	3 Caravan	Elena	2000	Licenta	0727520246	cost4e5ee545wa@gmail.com	23-APR-21	1	10
2	5 Tudor	Oana	9000	Masterat	0727390426	c4e5e45bwa@gmail.com	20-FEB-13	3	54
3	6 Constantin	Daniel	4000	Licenta	0327540226	ertw4tsbwa@gmail.com	18-NOV-13	4	53
4	8 Deaconu	Alina	5000	Masterat	0727490426	ct4ee4ta@gmail.com	18-DEC-15	6	51
5	9 Stanescu	Adela	5000	Masterat	0727555226	tetetwa@gmail.com	28-AUG-19	7	50
6	12 Safaru	Cosmin	5000	Masterat	0727550526	bwadrgdr@gmail.com	16-MAR-19	11	47
7	13 Popescu	Simona	5000	Masterat	0707595226	crd564ge5w5a@gmail.com	21-APR-13	12	46
8	14 Dumitrescu	Vlad	8000	Masterat	0720590126	54erdggdra@gmail.com	18-JUL-18	13	45
9	15 Sfetcu	Grigoras	5000	Masterat	0727191216	drdrgsbwa@gmail.com	09-NOV-17	14	44
10	16 Scafaru	Antonia	6000	Masterat	0727190345	stadgdreebsbwa@gmail.com	11-FEB-22	15	43
11	17 Elen	Paula	3000	Licenta	0725590123	cdgdra@gmail.com	21-JAN-16	16	42
12	18 Symeonidis	Oana	2000	Licenta	0727577726	drgrbsfthbwa@gmail.com	17-NOV-19	17	41
13	20 Telespan	Claudia	4000	Masterat	0777895226	fdhftsba@gmail.com	12-JAN-13	19	39
14	21 Sandulescu	Daria	6000	Masterat	0727590626	cgfhebsbwa@gmail.com	15-APR-19	20	38
15	24 Visan	Rares	7000	Masterat	0727590676	costafghfghfreebsbwa@gmail.com	12-MAR-19	23	35
16	25 Semanti	Bogdan	9000	Masterat	0727590126	costareebbswa@gmail.com	19-JUN-22	24	34
17	26 Pandone	Claudiu	5000	Masterat	0727590006	costareebghfbsbwa@gmail.com	19-JUL-20	25	22

48. Să se afișeze toate datele referitoare la medicii înregistrați într-un an diferit de 2013, dar care au salariul mai mic decât oricare dintre salariile medicilor înregistrați în anul 2013, ordonați descrescător, în funcție de salariu.

Comenzile SQL:

```
SELECT * FROM angajat a, medic m
WHERE a.salariu < ANY
(SELECT a.salariu FROM angajat a, medic m
WHERE a.idAng = m.idAng and
extract(year from a.data_angajare)=2013)
AND a.idAng = m.idAng and
extract(year from a.data_angajare)<>2013
ORDER BY salariu DESC;
```

Rezultatul:

IDANG	NUME	PRENUME	SALARIU	NIVEL_PREGATIRE	NUMAR_TELEFON	EMAIL	DATA_ANGAJARE	ID_ADRESA	ID_SUPERIOR	IDMEDIC	IDANG_1	SPECIALIZARE	NR_CABINET
1	53 Cirstian	Diana	7000	Masterat	0727575226	5yergd5ysbwa@gmail.com	08-AUG-08	20	(null)	4	53	ortoped	27
2	49 Popoiu	Raluca	7000	Masterat	0721510121	rdgdrwa@gmail.com	09-APR-21	18	(null)	8	49	neurolog	23
3	42 Horatiu	Valentin	7000	Masterat	0727588226	fdhdfbsbwa@gmail.com	21-MAY-21	19	(null)	15	42	gastroenterolog	16
4	46 Mazilu	Mircea	7000	Masterat	0727591226	codfwa@gmail.com	08-MAR-00	5	(null)	11	46	dermatolog	20
5	33 Stanciu	Andrei	6000	Masterat	0727590999	costareebbsbwhdfa@gmail.com	05-JUL-03	32	(null)	24	33	cardiolog	7
6	47 Gologan	Alexandra	5000	Masterat	0727111111	fdhcfdsbwa@gmail.com	10-NOV-12	2	(null)	10	47	dermatolog	21
7	39 Gont	Vlad	5000	Masterat	0727595636	costebsbwa@gmail.com	23-AUG-03	16	(null)	18	39	oftalmolog	13
8	40 Baltac	Cristian	5000	Masterat	0727770226	cosfhdsbwa@gmail.com	10-JAN-22	17	(null)	17	40	oftalmolog	14
9	41 Crudu	Cristian	5000	Masterat	0727575426	efhbsbwa@gmail.com	27-MAR-21	18	(null)	16	41	oftalmolog	15
10	54 Badea	Mircea	4000	Masterat	0727590799	tfhftfbsbwa@gmail.com	05-MAY-05	30	(null)	3	54	oncolog	28
11	38 Ivan	Maria	4000	Masterat	0727556326	costafdhfbswa@gmail.com	28-APR-03	15	(null)	19	38	radiolog	12
12	37 Hingan	Iulia	3000	Masterat	0727590333	cofdhfebsbwa@gmail.com	25-JAN-07	14	(null)	20	37	radiolog	11
13	30 Stanica	Alexandru	3000	Masterat	0729092949	cosdfhdhtareebbswa@gmail.com	18-DEC-05	29	(null)	27	30	chirurg	4
14	55 Foriu	Tristian	3000	Licenta	0727587826	tfhftfbswa@gmail.com	03-MAR-03	16	(null)	2	55	pediatru	29
15	34 Vrintuc	Paul	2000	Licenta	0727590000	costareebbsbhfdfwa@gmail.com	09-JUN-22	33	(null)	23	34	chirurg	8

49. Să se afișeze numele, id-ul medicului, prenumele și specializarea medicilor cu specializarea de cardiolog.

Comenzile SQL:

```
select m.idMedic, a.numa, a.prenume, m.specializare
from ANGAJAT a, MEDIC m
where upper(m.specializare) = 'CARDIOLOG'
and m.idAng = a.idAng;
```

Rezultatul:

	IDMEDIC	NUME	PRENUME	SPECIALIZARE
1	24	Stanciu	Andrei	cardiolog
2	22	Antonescu	Gabriela	cardiolog
3	21	Stangaciu	Loredana	cardiolog

50. Să se afișeze id-ul, numele și prenumele angajaților, dar și id-ul superiorului. În cazul în care angajatul nu are un superior, atunci este afișat -1 în locul id-ului.

Comenzile SQL:

```
select idAng, numa || ' ' || prenume as nume_angajat,
nvl(id_superior, -1) id_superior
from angajat;
```

Rezultatul:

	IDANG	NUME_ANGAJAT	ID_SUPERIOR
19	21	Sandulescu Daria	38
20	22	Dobrincu Sebastian	37
21	23	Georgian Alexandru	36
22	24	Visan Rares	35
23	25	Semanti Bogdan	34
24	26	Pandone Claudiu	22
25	27	Margineanu Matei	32
26	28	Munteanu Cristian	31
27	29	Molnar Alexandra	30
28	30	Stanica Alexandru	-1
29	31	Lazar Tiberiu	-1
30	32	Nistor Cosmin	-1
31	33	Stanciu Andrei	-1
32	34	Vrincut Paul	-1
33	35	Antonescu Gabriela	-1
34	36	Stangaciu Loredana	-1

Tabele Virtuale

51. Să se realizeze o tabelă virtuală cu toți medicii cu master.

Comenzile SQL:

```
CREATE OR REPLACE VIEW v_angajati_master  
AS SELECT * FROM angajat  
WHERE upper(nivel_pregatire)='MASTERAT';
```

Rezultatul:

```
View V_ANGAJATI_MASTER created.
```

52. Să se afișeze toate datele stocate în tabela virtuală.

Comenzile SQL:

```
SELECT * FROM v_angajati_master;
```

Rezultatul:

IDANG	NUME	PRENUME	SALARIU	NIVEL_PREGATIRE	NUMAR_TEFON	EMAIL	DATA_ANGAJARE	ID_ADRESA	ID_SUPERIOR	
1	4	Chiriac	Madalina	8000	Masterat	0747590326	cose45e45a@gmail.com	26-DEC-04	2	55
2	5	Tudor	Oana	9000	Masterat	0727390426	c4e5e45bwa@gmail.com	20-FEB-13	3	54
3	7	Diaconescu	Maria	4000	Masterat	0724590426	te44te4tbwa@gmail.com	30-JAN-07	5	52
4	8	Deaconu	Alina	5000	Masterat	0727490426	ct4ee4ta@gmail.com	18-DEC-15	6	51
5	9	Stanescu	Adela	5000	Masterat	0727555226	tetetwa@gmail.com	28-AUG-19	7	50
6	10	Stoenescu	Ana	6000	Masterat	0727596264	certerta@gmail.com	02-OCT-03	8	49
7	11	Grigore	Diana	6000	Masterat	0724590256	rtags@gmail.com	01-JAN-03	9	48
8	12	Safaru	Cosmin	5000	Masterat	0727550526	bwadrgdr@gmail.com	16-MAR-19	11	47
9	13	Popescu	Simona	5000	Masterat	0707595226	crd564ge5w5a@gmail.com	21-APR-13	12	46
10	14	Dumitrescu	Vlad	8000	Masterat	0720590126	54erdggdra@gmail.com	18-JUL-18	13	45
11	15	Sfetcu	Grigoras	5000	Masterat	0727191216	drdrgsbwa@gmail.com	09-NOV-17	14	44
12	16	Scafaru	Antonia	6000	Masterat	0727190345	stadgdreebsbwa@gmail.com	11-FEB-22	15	43
13	20	Telespan	Claudia	4000	Masterat	0777895226	fdhftsbwa@gmail.com	12-JAN-13	19	39
14	21	Sandulescu	Daria	6000	Masterat	0727590626	cgfheebbswa@gmail.com	15-APR-19	20	38
15	23	Georgian	Alexandru	4000	Masterat	0724590226	costareebwa@gmail.com	09-MAR-04	22	36
16	24	Visan	Rares	7000	Masterat	0727590676	costafghfghfreebsbwa@gmail.com	12-MAR-19	23	35
17	25	Semanti	Bogdan	9000	Masterat	0727590126	costareebbswa@gmail.com	19-JUN-22	24	34
18	26	Pandone	Claudiu	5000	Masterat	0727590006	costareebghfbsbwa@gmail.com	19-JUL-20	25	22
19	27	Margineanu	Matei	4000	Masterat	07270000000	costareebbsbfghfowa@gmail.com	08-MAY-15	26	32

53. Să se seteze id-ul superiorilor ca fiind -1 în cazul în care acesta este null.

Comenzile SQL:

```
UPDATE v_angajati_master
SET id_superior = decode(id_superior, null, -1, id_superior);
```

Rezultatul:

IDANG	NUME	PRENUME	SALARIU	NIVEL_PREGATIRE	NUMAR_TELEFON	EMAIL	DATA_ANGAJARE	ID_ADRESA	ID_SUPERIOR
16	24 Visan	Rares	7000	Masterat	0727590676	costafghfghfreebsbwa@gmail.com	12-MAR-19	23	35
17	25 Semanti	Bogdan	9000	Masterat	0727590126	costareebbsbwa@gmail.com	19-JUN-22	24	34
18	26 Pandone	Claudiu	5000	Masterat	0727590006	costareebgfghfsbwa@gmail.com	19-JUL-20	25	22
19	27 Margineanu	Matei	4000	Masterat	07270000000	costareebbsfghfgwa@gmail.com	08-MAY-15	26	32
20	28 Munteanu	Cristian	6000	Masterat	0757550255	costaredhdfbsbwa@gmail.com	02-FEB-17	27	31
21	29 Molnar	Alexandra	4000	Masterat	0727901314	fdhdcostareebbsbwa@gmail.com	12-MAY-16	28	30
22	30 Stanica	Alexandru	3000	Masterat	0729092949	cosdfhdhtareebbsbwa@gmail.com	18-DEC-05	29	-1
23	31 Lazar	Tiberiu	8000	Masterat	0727533326	costafhdhreebsbwa@gmail.com	27-NOV-00	30	-1
24	32 Nistor	Cosmin	3000	Masterat	0727590666	costareebbsbwa@gmail.com	21-MAR-13	31	-1
25	33 Stanciu	Andrei	6000	Masterat	0727590999	costareebbsbwhdfa@gmail.com	05-JUL-03	32	-1
26	35 Antonescu	Gabriela	9000	Masterat	0727980226	costareebbsbwa@gmail.com	22-FEB-22	34	-1
27	36 Stangaciu	Loredana	9000	Masterat	0736390226	costareebbsbwa@gmail.com	07-JUL-07	35	-1
28	37 Hingan	Iulia	3000	Masterat	0727590333	cofdhfebsbwa@gmail.com	25-JAN-07	14	-1
29	38 Ivan	Maria	4000	Masterat	0727556326	costafdhfbwa@gmail.com	28-APR-03	15	-1
30	39 Gont	Vlad	5000	Masterat	0727595636	costebsbwa@gmail.com	23-AUG-03	16	-1
31	40 Baltac	Cristian	5000	Masterat	0727770226	cosfhdtbsbwa@gmail.com	10-JAN-22	17	-1

54. Adăugarea condiției ca anul în care angajatul s-a alăturat centrului medical să fie 2019.

Comenzile SQL:

```
select * from v_angajati_master where extract(year from data_angajare) = 2019;
```

Rezultatul:

IDANG	NUME	PRENUME	SALARIU	NIVEL_PREGATIRE	NUMAR_TELEFON	EMAIL	DATA_ANGAJARE	ID_ADRESA	ID_SUPERIOR
1	9 Stanescu	Adela	5000	Masterat	0727555226	tetetwa@gmail.com	28-AUG-19	7	50
2	12 Safaru	Cosmin	5000	Masterat	0727550526	bwadrgdr@gmail.com	16-MAR-19	11	47
3	21 Sandulescu	Daria	6000	Masterat	0727590626	cgfhfebsbwa@gmail.com	15-APR-19	20	38
4	24 Visan	Rares	7000	Masterat	0727590676	costafghfghfreebsbwa@gmail.com	12-MAR-19	23	35

55. Să se creeze o tabelă virtuală care să se actualizeze cu ajutorul declanșatorilor, pentru mărirea salariilor medicilor, în funcție de nivelul de pregătire:

- doctorat -> 2000 RON
- masterat -> 1000 RON
- licență -> 500 RON

Comenzile SQL:

```
CREATE OR REPLACE TRIGGER t_date_utilite_medici
INSTEAD OF UPDATE ON v_date_utilite_medici
BEGIN
UPDATE v_date_utilite_medici
SET salariu = salariu +
(case
when nivel_pregatire = 'doctorat' then 2000
when nivel_pregatire = 'masterat' then 1000
when nivel_pregatire = 'licenta' then 500
else 0 end);
end;
```

Rezultatul:

```
Trigger T_DATE_UTILE_MEDICI compiled
```

56. Aplicați opțiunea de read only pentru tabela virtuală v_angajati_master.

Comenzile SQL:

```
CREATE OR REPLACE VIEW v_angajati_master  
AS SELECT * FROM angajat  
WHERE upper(nivel_pregatire)='MASTERAT'  
WITH READ ONLY;
```

Rezultatul:

```
View V_ANGAJATI_MASTER created.
```

57. Să se șteargă tabela virtuală v_angajati_master.

Comenzile SQL:

```
drop view v_angajati_master;
```

Rezultatul:

```
View V_ANGAJATI_MASTER dropped.
```

De menționat!!! Ținând cont că, unele dintre rezultate sunt trunchiate, pentru a nu aglomera prea tare documentul, **AICI** este disponibil **ÎNTREGUL SCRIPT**, încărcat pe drive.