CARUTASU ADRIANA STEFANIA *GRUPA 243*

ORGANIZAREA ACTIVITATILOR DINTR-UN SPITAL

Proiect-baza de date pentru organizarea activitatilor dintr-un spital

Utilitatea bazei de date

Am implementat o baza de date pentru gestiunea activitatilor dintr-un spital.
Pentru aceasta am creat entitati pentru pacienti, doctori (si o tabela pentru pastrarea adresei complete a acestora), laboratoare de analize, camere, programul operatiilor, fise de internare si rezultatele analizelor medicale.
Un pacient poate avea mai multi doctori, iar un doctor poate avea mai multi pacienti.
Fiecarui pacient i se adauga cate o fisa de internare la fiecare vizita in spital.
Un doctor poate efectua mai multe operatii si la o operatie pot asista mai multi doctor.
Un pacient poate avea mai multe operatii si in timpul unei operatii este operat un singur pacient.
La fiecare internare unui pacient i se face o fisa noua de internare (datele fiind pastrate in tabela PATIENT_ADMISSION) si sta intr-o singura camera, dar intr-o camera pot fi internati mai multi pacienti.
Fiecarui pacient i se pot face mai multe teste, iar un test apartine unui singur pacient.
Intr-un laborator sunt facute mai multe teste.
Adresele pacientilor si medicilor sunt pastrate intr-o tabela separata.

Independent de aceste entitati am creat tabela AUDIT_ORCLDB pentru a pastra informatii despre actualizarile facute asupra celorlalte tabele.

Diagrame:

Diagrama Entitate-Relatie

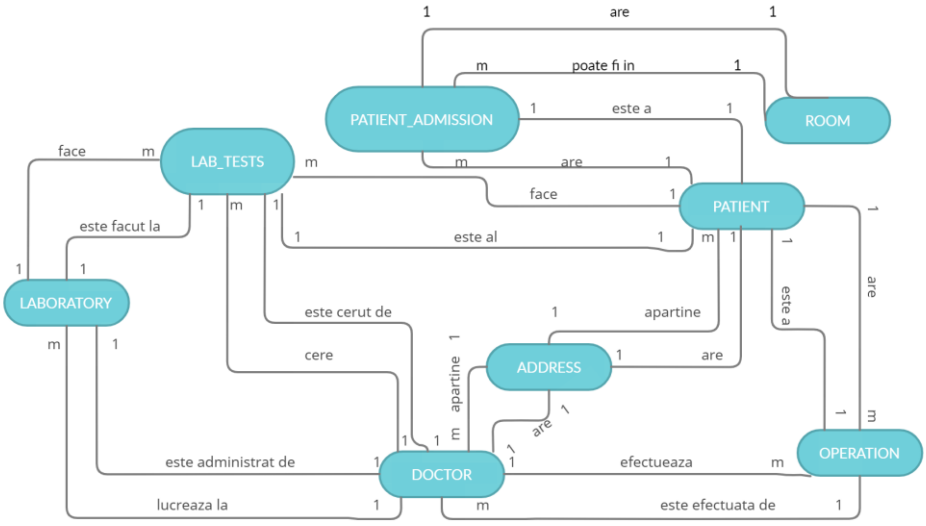
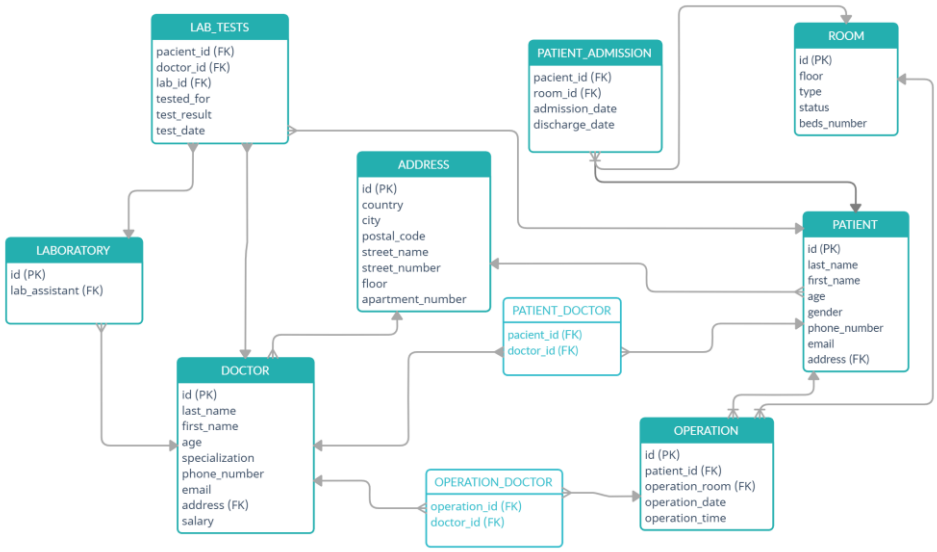


Diagrama Conceptuala



Implementarea in Oracle a diagramei relationale create

```
--creare PACIENT (Ulterior am modificat numele in PATIENT)
CREATE TABLE pacient (
id number(5) primary key,
last_name varchar2(25) not null,
first_name varchar2(25) not null,
age number(3) not null,
gender varchar2(15) not null,
phone_number varchar2(10) not null,
email varchar2(20),
address varchar2(40));

alter table pacient
modify email varchar2(30);

alter table pacient
add room_id number(4);
alter table pacient
add constraint room_fk FOREIGN KEY (room_id) REFERENCES room(id) on delete cascade;

alter table pacient
add room_id number(4);

alter table pacient
add constraint room_fk FOREIGN KEY (room_id) REFERENCES room(id) on delete cascade;

alter table pacient
drop column address;

alter table pacient
add address number(5);
alter table pacient
add constraint fk_address foreign key (address) references address(id) on delete set null;


alter table patient
drop column room_id;

--creare DOCTOR
create table doctor(
id number(5) primary key,
```

```
last_name varchar2(25),
first_name varchar2(25),
age number(3),
specialization varchar2(25),
phone_number varchar2(10),
email varchar2(20),
address varchar2(40));



alter table doctor
drop column address;

alter table doctor
add address number(5);

alter table doctor
add constraint fk_address_doc foreign key (address) references address(id) on delete set null;

--creare TABEL ASOCIATIV PACIENT_DOCTOR (Ulterior am modificat numele in PATIENT_DOCTOR)
create table pacient_doctor (
pacient_id number(5),
doctor_id number(5));

alter table pacient_doctor
add constraint pd_fk FOREIGN KEY (pacient_id) REFERENCES pacient(id) on delete cascade;

alter table pacient_doctor
add constraint pd_fk_doctor FOREIGN KEY (doctor_id) REFERENCES doctor(id) on delete cascade;

alter table pacient_doctor
add constraint pk_pd primary key (pacient_id, doctor_id);


--creare LABORATORY
create table laboratory (
id number(5) primary key,
lab_assistant varchar2(40) not null);

alter table laboratory
```

```
modify lab_assistant number(5);
alter table laboratory
add constraint lab_assistant_fk FOREIGN KEY (lab_assistant) references doctor(id) on delete
cascade;


--create ROOM
create table room (
id NUMBER(4) primary key,
floor number(2) not null,
type varchar2(20),
status varchar2(20) not null);

ALTER TABLE room
ADD beds_number number(2);
update room
set beds_number = 3
where id = 124;


--creare LAB_TESTS
create table lab_tests (
pacient_id number(5),
doctor_id number(5),
lab_id number(5),
tested_for varchar2(30),
test_result varchar2(30),
test_date date,
FOREIGN key (pacient_id) references pacient(id),
FOREIGN key (doctor_id) references doctor(id),
FOREIGN key (lab_id) references laboratory(id));

alter table lab_tests
add constraint pk_lab_tests primary key (pacient_id, doctor_id, lab_id);

alter table lab_tests
drop constraint pk_lab_tests;

alter table lab_tests
add constraint pk_lab_tests primary key (pacient_id, doctor_id, tested_for, test_date);


--creare OPERATIONS (Ulterior am schimbat numele in OPERATION)
```

```
create table operations (
id number(5) primary key,
patient_id number(5),
operation_room NUMBER(4),
operation_date date,
operation_time varchar2(10),
foreign key (patient_id) references patient(id));

--creare OPERATION_DOCTOR
create table operation_doctor (
doctor_id number(5),
operation_id number(5),
foreign key (doctor_id) references doctor(id),
foreign key (operation_id) references operation(id));

alter table operation_doctor
add constraint od_pk primary key (operation_id, doctor_id);

--creare PACIENT_ADMISSION (Ulterior am schimbat numele in PATIENT_ADMISSION)
create table pacient_admission (
pacient_id number(5),
room_id number(4),
admission_date date,
discharge_date date);

alter table pacient_admission
add constraint fk_p_add foreign key (pacient_id) references pacient(id) on delete cascade;

alter table pacient_admission
add constraint fk_room_add foreign key (room_id) references room(id) on delete cascade;

alter table pacient_admission
add constraint pk_admission primary key (pacient_id, room_id, admission_date);


--creare ADDRESS
create table address (
id number(5) primary key,
country varchar2(25),
city varchar2(25),
postal_code varchar2(10),
street_name varchar2(25),
street_number number(4),
```

```
floor number(2),
apartment_number number(3));
```

Popularea cu date a tabelelor create anterior

```
--inserare in ROOM
insert into room
values (123, 1, 'Ward', 'Full');
insert into room
values (124, 1, 'Ward', 'Empty');
insert into room
values (134, 3, 'Operation Room', 'Empty');
insert into room
values (135, 3, 'Operation Room', 'Full');
insert into room
values (101, 0, 'ER', '-');
alter table room
add beds_number number(2);
insert into room
values (125, 1, 'Ward', 'Available', 3);
```

	ID	FLOOR	TYPE	STATUS	BEDS_NUMBER
1	125	1	Ward	Available	3
2	123	1	Ward	Full	2
3	124	1	Ward	Empty	3
4	134	3	Operation Room	Empty	(null)
5	135	3	Operation Room	Full	(null)
6	101	0	ER	-	(null)

```
--inserare in ADDRESS
insert into address
values(1, 'Romania', 'Bucharest', '012345', 'Teiul Doamnei', 7, 2, 5);
insert into address
values(2, 'Romania', 'Bucharest', '012346', 'Regina Elisabeta', 5, 1, 3);
insert into address
values(3, 'Romania', 'Bucharest', '013345', 'Grigore Ionescu', 43, 8, 36);
insert into address
values(4, 'Romania', 'Bucharest', '012445', 'Otesani', 3, 0, 1);
insert into address
values(5, 'Romania', 'Bucharest', '112345', 'Dristorului', 10, 6, 20);
insert into address
values(6, 'Romania', 'Bucharest', '222345', 'Ion Alexe', 23, 4, 13);
insert into address
values(7, 'Romania', 'Bucharest', '012355', 'Grigore Ionescu', 7, 2, 5);
insert into address
values(8, 'Romania', 'Bucharest', '012355', 'Grigore Ionescu', 73, 5, 20)
insert into address
```

values(9, 'Romania', 'Bucharest', '222345', 'Ion Alexe', 41, 4, 13);
insert into address
values(10, 'Romania', 'Bucharest', '012345', 'Teiul Doamnei', 44, 7, 21);

	ID	COUNTRY	CITY	POSTAL_CODE	STREET_NAME	STREET_NUMBER	FLOOR	APARTMENT_NUMBER
1	10	Romania	Bucharest	012345	Teiul Doamnei	44	7	21
2	1	Romania	Bucharest	012345	Teiul Doamnei	7	2	5
3	2	Romania	Bucharest	012346	Regina Elisabeta	5	1	3
4	3	Romania	Bucharest	013345	Grigore Ionescu	43	8	36
5	4	Romania	Bucharest	012445	Otesani	3	0	1
6	8	Romania	Bucharest	012355	Grigore Ionescu	73	5	20
7	9	Romania	Bucharest	222345	Ion Alexe	41	4	13
8	5	Romania	Bucharest	112345	Dristorului	10	6	20
9	6	Romania	Bucharest	222345	Ion Alexe	23	4	13
10	7	Romania	Bucharest	012355	Grigore Ionescu	7	2	5

--inserare PACIENT (Ulterior am schimbat numele in PATIENT)
insert into pacient
values (100, 'King', 'Steven', 34, 'Male', '0723', 'stevenking@gmail.com', 123, 1);
insert into pacient
values (101, 'Ernst', 'Bruce', 22, 'Male', '0724', 'bruceernst@gmail.com', 123, 2);
insert into pacient
values (102, 'Popp', 'Louise', 55, 'Female', '0725', 'popplouise@gmail.com', 124, 3);
insert into pacient
values (103, 'King', 'Nancy', 30, 'Female', '0726', 'nancyking@gmail.com', 124, 1);
insert into pacient
values (104, 'Sigal', 'Tobias', 18, 'Male', '0727', 'tobiassigal@gmail.com', 124, 6);
insert into pacient
values (105, 'Khan', 'Elena', 21, 'Female', '0787', 'elenakhan@gmail.com', 125, 10);
(Coloana ce continea ID-ul camerei in care sta pacientul a fost stearsa ulterior)

	ID	LAST_NAME	FIRST_NAME	AGE	GENDER	PHONE_NUMBER	EMAIL	ADDRESS
1	105	Khan	Elena	21	Female	0787	elenakhan@gmail.com	10
2	100	King	Steven	34	Male	0723	stevenking@gmail.com	1
3	101	Ernst	Bruce	22	Male	0724	bruceernst@gmail.com	2
4	102	Popp	Louise	55	Female	0725	popplouise@gmail.com	3
5	103	King	Nancy	30	Female	0726	nancyking@gmail.com	1
6	104	Sigal	Tobias	18	Male	0727	tobiassigal@gmail.com	6

--inserare DOCTOR
insert into doctor
values (200, 'Lorentz', 'Andrew', 40, 'Cardiology', '0727', 'andrewlorentz@gmail.com', 5);
insert into doctor
values (201, 'Ernst', 'Anna', 55, 'Pediatrics', '0728', 'anaernst@gmail.com', 2);
insert into doctor
values (202, 'Lorentz', 'Jhon', 28, 'Orthopaedics', '0729', 'jhonlorentz@gmail.com', 5);
insert into doctor
values (203, 'Sigal', 'Ana', 65, 'Radiology', '0733', 'anasigal@gmail.com', 6);

```
insert into doctor
values (204, 'Hall', 'Peter',  40, 'Dermatology', '0734', 'peterhall@gmail.com', 7);
insert into doctor
values (205, 'Popp', 'Andrew',  56, 'Lab assistant', '0747', 'andrewpopp@gmail.com', 3);
insert into doctor
values (206, 'Grant', 'Douglas',  25, 'Lab assistant', '0749', 'douglasgrant@gmail.com', 8);
insert into doctor
values (207, 'Grant', 'Tina',  26, 'Lab assistant', '0750', 'tinagrant@gmail.com', 8);
insert into doctor
values (208, 'Higgins', 'Shelly',  40, 'Anesthetist', '0799', 'shellyhiggins@gmail.com', 9)
 (Ulterior a fost adaugata coloana salary)
update doctor
set salary = 5000
where lower(specialization) != lower('Lab assistant') and lower(specialization) !=
lower('Anesthetist');
update doctor
set salary = 4500
where lower(specialization) = 'lab assistant';
update doctor
set salary = 4200
where lower(specialization) = 'anesthetist';
```

	ID	LAST_NAME	FIRST_NAME	AGE	SPECIALIZATION	PHONE_NUMBER	EMAIL	ADDRESS	SALARY
1	200	Lorentz	Andrew	40	Cardiology	0727	andrewlorentz@gmail.com	5	5000
2	201	Ernst	Anna	55	Pediatrics	0728	anaernst@gmail.com	2	5000
3	202	Lorentz	Jhon	28	Orthopaedics	0729	jhonlorentz@gmail.com	5	5000
4	203	Sigal	Ana	65	Radiology	0733	anasigal@gmail.com	6	5000
5	204	Hall	Peter	40	Dermatology	0734	peterhall@gmail.com	7	5000
6	205	Popp	Andrew	56	Lab assistant	0747	andrewpopp@gmail.com	3	4500
7	206	Grant	Douglas	25	Lab assistant	0749	douglasgrant@gmail.com	8	4500
8	207	Grant	Tina	26	Lab assistant	0750	tinagrant@gmail.com	8	4500
9	208	Higgins	Shelly	40	Anesthetist	0799	shellyhiggins@gmail.com	9	4200

```
--inserare LABORATORY
insert into laboratory
values (1234, 205);
insert into laboratory
values (1235, 205);
insert into laboratory
values (1236, 206);
insert into laboratory
values (1237, 206);
insert into laboratory
values (1238, 207);
```

	ID	LAB_ASSISTANT
1	1234	205
2	1235	205
3	1236	206
4	1237	206
5	1238	207

--inserare LAB_TESTS
insert into lab_tests
values (100, 204, 1234, 'Iritation', 'Negative', to_date('10 May 2020'));
insert into lab_tests
values (101, 200, 1234, 'Palpitaions', 'Pozitive', to_date('19 July 2020'));
insert into lab_tests
values (101, 200, 1235, 'Palpitaions', 'Negative', '27 July 2020');
insert into lab_tests
values (101, 200, 1235, 'Palpitaions', 'Negative', '27 July 2020');
insert into lab_tests
values (104, 202, 1236, 'Broken arm', 'Pozitive', '27 July 2020');
insert into lab_tests
values (104, 202, 1236, 'Broken arm', 'Negative', '27 October 2020');

	PACIENT_ID	DOCTOR_ID	LAB_ID	TESTED_FOR	TEST_RESULT	TEST_DATE
1	100	204	1234	Iritation	Negative	10-MAY-20
2	101	200	1234	Palpitaions	Pozitive	19-JUL-20
3	101	200	1235	Palpitaions	Negative	27-JUL-20
4	104	202	1236	Broken arm	Pozitive	27-JUL-20
5	104	202	1236	Broken arm	Negative	27-OCT-20

--inserare OPERATIONS (ulterior am schimbat numele tabelei in OPERATION)
insert into operations
values (104, 200, 208, 134, '18 June 2020', '17;45');
insert into operations
values (104, 200, 208, 134, '20 June 2020', '18:00');
insert into operations
values (101, 200, 208, 135, '21 July 2020', '09:00');
insert into operations (pacient_id, doctor_id, operation_room, operation_date, operation_time)
values (103, 204, 135, '23 July 2020', '09:00');
insert into operations (pacient_id, doctor_id, operation_room, operation_date, operation_time)
values (102, 204, 135, '23 July 2020', '10:45');
insert into operations (pacient_id, doctor_id, operation_room, operation_date, operation_time)

values (105, 200, 135, '3-December-2020', '12:00');

	ID	PATIENT_ID	OPERATION_ROOM	OPERATION_DATE	OPERATION_TIME
1	1	104	134	18-JUN-20	17;45
2	2	104	134	20-JUN-20	18:00
3	3	101	135	21-JUL-20	09:00
4	4	103	135	23-JUL-20	09:00
5	5	102	135	23-JUL-20	10:45
6	6	105	135	03-DEC-20	12:00

--inserare PACIENT_ADMISSION (Ulterior am schimbat numele in PATIENT_ADMISSION)
insert into pacient_admission
values (104, 124, '17-June-2020', '25-June-2020');
insert into pacient_admission
values (101, 123, '20-July-2020', '25-July-2020');
insert into pacient_admission
values (103, 124, '23-July-2020', '24-July-2020');
insert into pacient_admission (pacient_id, room_id, admission_date)
values (102, 124, '08-December-2020');
insert into pacient_admission (pacient_id, room_id, admission_date)
values (104, 124, '08-December-2020');
insert into pacient_admission (pacient_id, room_id, admission_date)
values (105, 124, '30-November-2020');

	PACIENT_ID	ROOM_ID	ADMISSION_DATE	DISCHARGE_DATE
1	105	124	30-NOV-20	(null)
2	104	124	17-JUN-20	25-JUN-20
3	101	123	20-JUL-20	25-JUL-20
4	103	124	23-JUL-20	24-JUL-20
5	102	124	08-DEC-20	(null)
6	104	124	08-DEC-20	(null)

--inserare PACIENT_DOCTOR (Ulterior am schimbat numele in PATIENT_DOCTOR
insert into pacient_doctor
values (100, 200);
insert into pacient_doctor
values (100, 204);
insert into pacient_doctor
values (101, 200);
insert into pacient_doctor
values (101, 202);
insert into pacient_doctor
values (101, 204);
insert into pacient_doctor
values (102, 200);
insert into pacient_doctor

values (102, 204);
insert into pacient_doctor
values (103, 202);
insert into pacient_doctor
values (103, 204);
insert into pacient_doctor
values (104, 200);
insert into pacient_doctor
values (104, 204);
insert into pacient_doctor
values (105, 204);
insert into pacient_doctor
values (105, 200);

	PACIENT_ID	DOCTOR_ID
1	100	200
2	100	204
3	101	200
4	101	202
5	101	204
6	102	200
7	102	204
8	103	202
9	103	204
10	104	200
11	104	204
12	105	200

--inserare OPERATION_DOCTOR
insert into operation_doctor values
(200, 1);
insert into operation_doctor values
(205, 1);
insert into operation_doctor values
(208, 1);

insert into operation_doctor values
(202, 2);
insert into operation_doctor values
(207, 2);
insert into operation_doctor values
(208, 2);

insert into operation_doctor values

(202, 3);
insert into operation_doctor values
(204, 3);
insert into operation_doctor values
(208, 3);

insert into operation_doctor values
(200, 4);
insert into operation_doctor values
(208, 4);

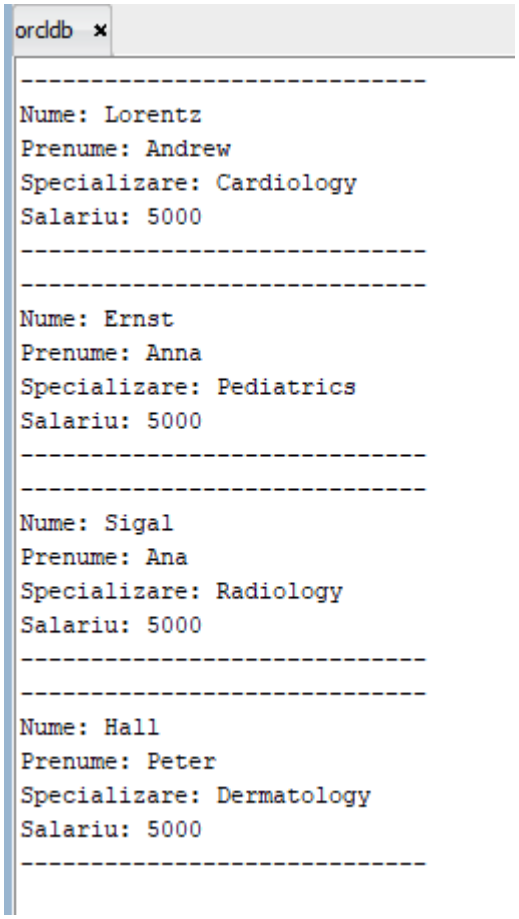
insert into operation_doctor values
(200, 5);
insert into operation_doctor values
(208, 5);

insert into operation_doctor values
(200, 6);
insert into operation_doctor values
(208, 6);

	DOCTOR_ID	OPERATION_ID
1	200	1
2	205	1
3	208	1
4	202	2
5	207	2
6	208	2
7	202	3
8	204	3
9	208	3
10	200	4
11	208	4
12	200	5
13	208	5
14	200	6
15	208	6

Utilizare de subprograme (punctele 6..9)

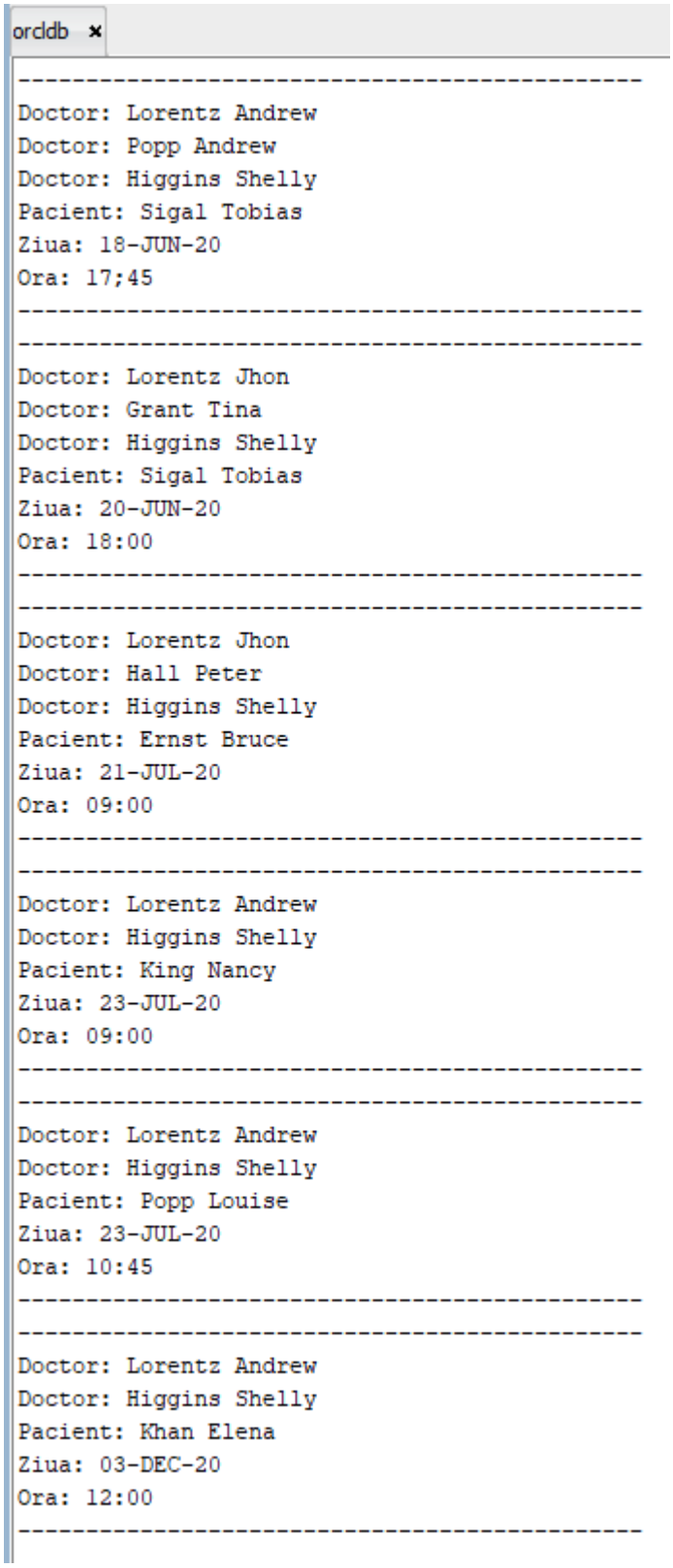
```
--6
--sa se afiseze numele, prenumele, specializarea si salariul pentru toti medicii
--peste 40 de ani
  create or replace procedure afisare_date_medic is
      type date_medic is record
      (nume doctor.last_name%type, prenume doctor.first_name%type, spec
doctor.specialization%type, salariu doctor.salary%type);
      type v_date_medic is table of date_medic;
      v_date v_date_medic;
   begin
      select last_name, first_name, specialization, salary
      bulk collect into v_date
      from doctor
      where lower(specialization) != 'lab assistant' and lower(specialization) != 'anesthetist'
and age > 80;
      if v_date.exists(1) then
         for i in v_date.first..v_date.last loop
            DBMS_OUTPUT.PUT_LINE ('------------------------------');
            DBMS_OUTPUT.PUT_LINE ('Nume: ' || v_date(i).nume);
            DBMS_OUTPUT.PUT_LINE ('Prenume: ' || v_date(i).prenume);
            DBMS_OUTPUT.PUT_LINE ('Specializare: ' || v_date(i).spec);
            DBMS_OUTPUT.PUT_LINE ('Salariu: ' || v_date(i).salariu);
            DBMS_OUTPUT.PUT_LINE ('------------------------------');
         end loop;
      else RAISE no_data_found;
      end if;
      EXCEPTION
         when no_data_found then
            DBMS_OUTPUT.PUT_LINE ('Nu exista astfel de medici');
          when others then
            DBMS_OUTPUT.PUT_LINE ('Alta eroare!');
   end afisare_date_medic;
   /

   EXECUTE afisare_date_medic;
```


orcldb
Nume: Lorentz
Prenume: Andrew
Specializare: Cardiology
Salariu: 5000
Nume: Ernst
Prenume: Anna
Specializare: Pediatrics
Salariu: 5000
Nume: Sigal
Prenume: Ana
Specializare: Radiology
Salariu: 5000
Nume: Hall
Prenume: Peter
Specializare: Dermatology
Salariu: 5000

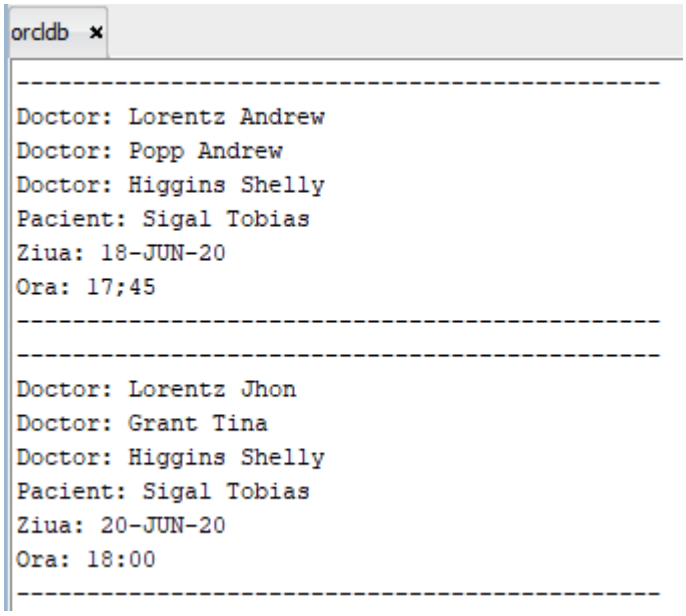
--7
--Sa se afiseze informatii din tabelul OPERATIONS in functie de optiunea
--data ca parametru procedurii
--pentru 1: toate datele din tabel
--pentru 2: toate operatiile unde nu a fost prezent niciun anestezist
--pentru 3: toate operatiile ce au avut loc in luna curenta

```
create or replace procedure afisare_operatii (optiune in number default 1)
is
   type op_type is ref cursor return operation%rowtype;
   v_op op_type;
   v_date operation%rowtype;
   type doc_data is record (nume_doctor doctor.last_name%type, prenume_doctor
doctor.first_name%type);
   type doc_type is table of doc_data;
   v_doc doc_type;
   nume_pacient patient.last_name%type;
   prenume_pacient patient.first_name%type;

begin
   if optiune = 1 then
      open v_op for select *
               from operation;
   elsif optiune = 2 then
      open v_op for select *
               from operation
               where operation_room = 134;
   elsif optiune = 3 then
      open v_op for select *
               from operation
               where to_char(operation_date, 'mm') = to_char(sysdate, 'mm') and
to_char(operation_date, 'yyyy') = to_char(sysdate, 'yyyy');
   else
      dbms_output.put_line('Nu exista optiunea introdusa');
   end if;
      loop
         fetch v_op into v_date;
         exit when v_op%notfound;
         select last_name, first_name
         bulk collect into v_doc
```

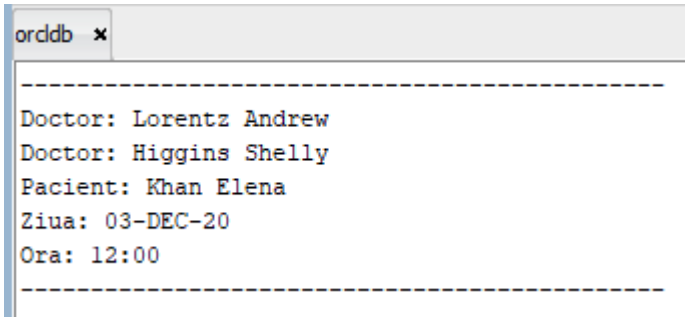
```
        from doctor d join operation_doctor od on (d.id = od.doctor_id)
        where od.operation_id = v_date.id;
        select last_name, first_name
        into nume_pacient, prenume_pacient
        from patient
        where id = v_date.patient_id;
        DBMS_OUTPUT.PUT_LINE('---------------------------------------------');
        for i in v_doc.first..v_doc.last loop
          DBMS_OUTPUT.PUT_LINE('Doctor: '|| v_doc(i).nume_doctor || ' ' ||
v_doc(i).prenume_doctor);
        end loop;
        DBMS_OUTPUT.PUT_LINE('Pacient: '||nume_pacient || ' ' || prenume_pacient);
        DBMS_OUTPUT.PUT_LINE('Ziua: '|| v_date.operation_date);
        DBMS_OUTPUT.PUT_LINE('Ora: ' || v_date.operation_time);
        DBMS_OUTPUT.PUT_LINE('---------------------------------------------');
      end loop;
      close v_op;
   EXCEPTION
      when no_data_found then
         DBMS_OUTPUT.PUT_LINE('Nu exista informatiile cerute');
      when others then
         DBMS_OUTPUT.PUT_LINE('S-a produs o eroare!');
end afisare_operatii;
/

execute afisare_operatii(1);
```



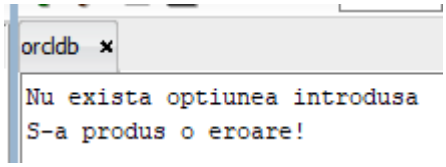
execute afisare_operatii(2);



execute afisare_operatii(3);



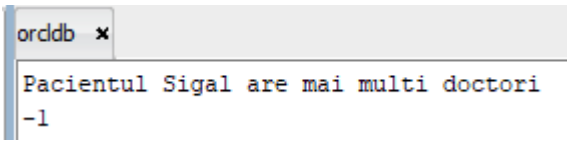
execute afisare_operatii(4);



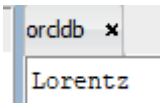
--8

--sa se afiseze numele medicului unui pacient al carui nume este dat

```
create or replace function medic_pacient (nume_pacient in patient.last_name%type default
'Sigal')
return doctor.last_name%type
is
medic doctor.last_name%type;
begin
   select d.last_name
   into medic
   from doctor d join patient_doctor pd on (d.id = pd.doctor_id)
              join patient p on (pd.pacient_id = p.id)
   where lower(p.last_name) = lower(nume_pacient);
   return medic;
   EXCEPTION
      when no_data_found then
         DBMS_OUTPUT.PUT_LINE('Nu exista un pacient cu acest nume!');
         return -1;
      when too_many_rows then
         DBMS_OUTPUT.PUT_LINE('Pacientul ' || nume_pacient || ' are mai multi doctori');
         return -1;
      when others then
         DBMS_OUTPUT.PUT_LINE('Alta eroare!');
         return -1;
end medic_pacient;
/
begin
   DBMS_OUTPUT.PUT_LINE(medic_pacient);
end;
/
```



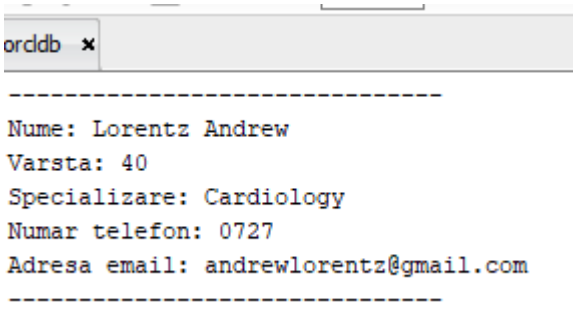

```
begin
   DBMS_OUTPUT.PUT_LINE(medic_pacient('Khan'));
end;
/
```



--9
--se se afiseze informatii despre toti doctorii care au operat pacienti care
--au fost operati in luna curenta si au fost internati luna precedenta.

```
create or replace procedure proc_afisare_doctori
is
cursor c_date_medic is
   select d.last_name, d.first_name, d.age, specialization, d.phone_number, d.email
   from doctor d join operation_doctor od on (d.id = od.doctor_id)
                     join operation o on (od.operation_id = o.id)
                     join patient p on (o.patient_id = p.id)
                     join patient_admission pa on (p.id = pa.pacient_id)
   where (to_char(o.operation_date, 'mm') = to_char(sysdate, 'mm') and
to_number(to_char(pa.admission_date, 'mm')) = to_number(to_char(sysdate, 'mm')) - 1)
      or (to_char(o.operation_date, 'mm') = 'January' and to_char(pa.admission_date, 'mm')
= 'December' and
      to_number(to_char(o.operation_date, 'yyyy')) =
to_number(to_char(pa.admission_date, 'yyyy')) + 1);

type date_doctor is record (nume doctor.last_name%type, prenume
doctor.first_name%type, varsta doctor.age%type, spec doctor.specialization%type,
nr_telefon doctor.phone_number%type, email doctor.email%type);
type tablou_date_doctori is table of date_doctor;
tablou_doctori tablou_date_doctori;
begin
   open c_date_medic;
   fetch c_date_medic bulk collect into tablou_doctori;
   close c_date_medic;
   if tablou_doctori.exists(1) then
      for i in tablou_doctori.first..tablou_doctori.last loop
         DBMS_OUTPUT.PUT_LINE ('------------------------------');
         DBMS_OUTPUT.PUT_LINE ('Nume: ' || tablou_doctori(i).nume || ' ' ||
tablou_doctori(i).prenume);
         DBMS_OUTPUT.PUT_LINE ('Varsta: ' || tablou_doctori(i).varsta);
         DBMS_OUTPUT.PUT_LINE ('Specializare: ' || tablou_doctori(i).spec);
         DBMS_OUTPUT.PUT_LINE ('Numar telefon: ' || tablou_doctori(i).nr_telefon);
         DBMS_OUTPUT.PUT_LINE ('Adresa email: ' || tablou_doctori(i).email);
         DBMS_OUTPUT.PUT_LINE ('------------------------------');
      end loop;
   else raise no_data_found;
   end if;
   EXCEPTION
```

```
    when no_data_found then
      DBMS_OUTPUT.PUT_LINE ('Nu exista medici cu pacienti operati luna curenta si
internati luna precedenta');
    when others then
      DBMS_OUTPUT.PUT_LINE ('Alta eroare!');
end proc_afisare_doctori;
/


EXECUTE proc_afisare_doctori;
```



Utilizare TRIGGERS

```
--10
--triggerul se va declansa daca se incearca adaugarea de operatii noi de Craciun sau in
weekend

create or replace trigger restrictionare_operatii
   before insert or update on operation
begin
      if (to_char(sysdate, 'dd') = 25) and (to_char(sysdate, 'mm') = 12)
         then raise_application_error(-20001, 'Nu se pot face programari pentru operatii de
Craciun');
      elsif (to_char(sysdate,'D') = 1) OR (to_char(sysdate,'D') = 7)
         then raise_application_error(-20002, 'Nu se pot face programari pentru operatii in
weekend');
      end if;
end;
```

Am folosit data de 29 pentru a putea exemplifica utilizarea trigger-ului.

```
insert into operation
values (7, 105, 134, '29-DEC-2020', '12:00');
```

```
Error starting at line : 21 in command -
insert into operation
values (7, 105, 134, '29-DEC-2020', '12:00')
Error report -
ORA-20001: Nu se pot face programari pentru operatii de Craciun
ORA-06512: at "C##STEFI.RESTRICTIONARE_OPERATII", line 5
ORA-04088: error during execution of trigger 'C##STEFI.RESTRICTIONARE_OPERATII'
```

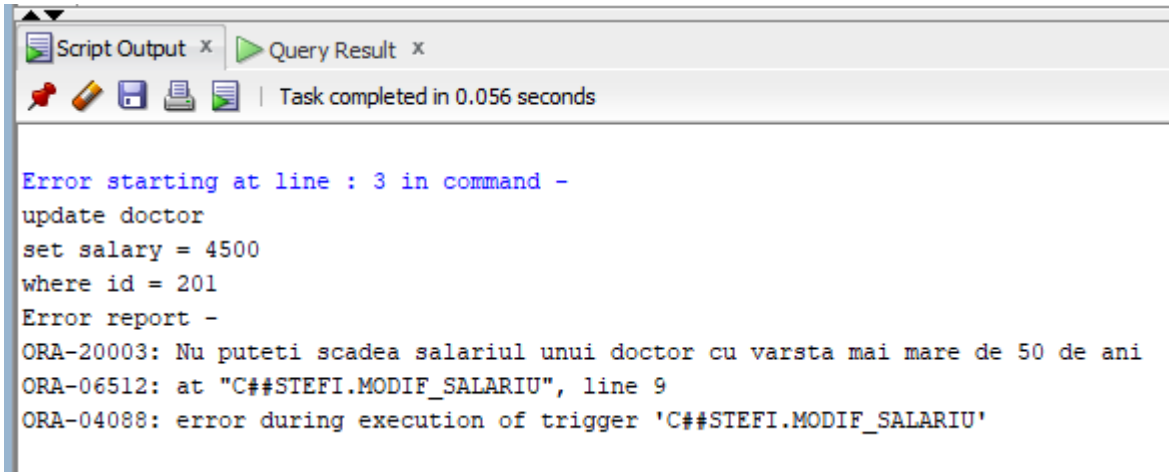
--11

--trigger-ul se va declansa cand se va incerca scaderea salariului unui medic cu varsta mai mare de 50 de ani

```
create or replace trigger modif_salariu
before update of salary on doctor
FOR EACH ROW
declare
exceptie EXCEPTION;
begin
   if :OLD.salary > :NEW.salary and :NEW.age >= 50 then
      raise exceptie;
   end if;
  EXCEPTION
WHEN exceptie THEN
RAISE_APPLICATION_ERROR (-20003, 'Nu puteti scadea salariul unui doctor cu varsta mai mare de 50 de ani');
end;
/
select specialization, age, salary
from doctor
where id = 201;
```

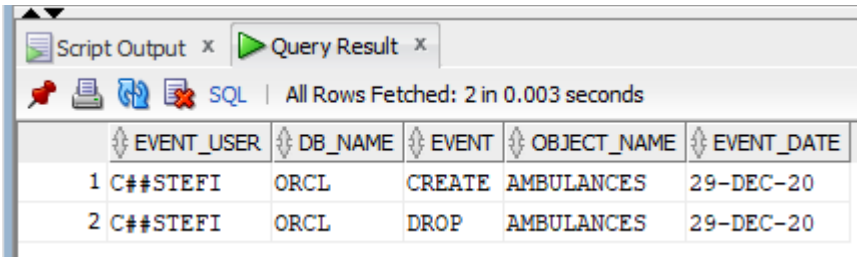
	SPECIALIZATION	AGE	SALARY
1	Pediatrics	55	5000

```
update doctor
set salary = 4500
where id = 201;
```



--12
--Trigger-ul se va declansa de fiecare data cand se executa o comanda LDD asupra bazei de date
--Modificarile aparute vor fi inserate in tabela audit_orcldb

```
create table audit_orcldb
   (event_user varchar2(30),
   db_name varchar2(30),
   event varchar2(20),
   object_name varchar2(30),
   event_date date);

create or replace trigger insert_event
   after create or alter or drop on schema
begin
   insert into audit_orcldb
   values (sys.login_user, sys.database_name, sys.sysevent, sys.dictionary_obj_name,
sysdate);
end;
/
create table ambulances (
plate_numer varchar2(20),
functionality varchar2(20));
drop table ambulances;
```



	EVENT_USER	DB_NAME	EVENT	OBJECT_NAME	EVENT_DATE
1	C##STEFI	ORCL	CREATE	AMBULANCES	29-DEC-20
2	C##STEFI	ORCL	DROP	AMBULANCES	29-DEC-20

Utilizare pachete

```
--13
create or replace package pachet_spital as
   procedure afisare_date_medic;
   procedure afisare_operatii (optiune in number default 1);
   function medic_pacient (nume_pacient in patient.last_name%type default 'Sigal')
      return doctor.last_name%type;
   procedure proc_afisare_doctori;
end pachet_spital;
```

```
Package PACHET_SPITAL compiled
```

```
create or replace package body pachet_spital as
   procedure afisare_date_medic is
      type date_medic is record
      (nume doctor.last_name%type, prenume doctor.first_name%type, spec
doctor.specialization%type, salariu doctor.salary%type);
      type v_date_medic is table of date_medic;
      v_date v_date_medic;
   begin
      select last_name, first_name, specialization, salary
      bulk collect into v_date
      from doctor
      where lower(specialization) != 'lab assistant' and lower(specialization) != 'anesthetist'
and age > 80;
      if v_date.exists(1) then
         for i in v_date.first..v_date.last loop
            DBMS_OUTPUT.PUT_LINE ('-----------------------------');
            DBMS_OUTPUT.PUT_LINE ('Nume: ' || v_date(i).nume);
            DBMS_OUTPUT.PUT_LINE ('Prenume: ' || v_date(i).prenume);
            DBMS_OUTPUT.PUT_LINE ('Specializare: ' || v_date(i).spec);
            DBMS_OUTPUT.PUT_LINE ('Salariu: ' || v_date(i).salariu);
            DBMS_OUTPUT.PUT_LINE ('-----------------------------');
         end loop;
      else RAISE no_data_found;
      end if;
      EXCEPTION
         when no_data_found then
            DBMS_OUTPUT.PUT_LINE ('Nu exista astfel de medici');
          when others then
            DBMS_OUTPUT.PUT_LINE ('Alta eroare!');
   end afisare_date_medic;
```

```
   procedure afisare_operatii (optiune in number default 1)
is
   type op_type is ref cursor return operation%rowtype;
   v_op op_type;
   v_date operation%rowtype;
   type doc_data is record (nume_doctor doctor.last_name%type, prenume_doctor
doctor.first_name%type);
   type doc_type is table of doc_data;
   v_doc doc_type;
   nume_pacient patient.last_name%type;
   prenume_pacient patient.first_name%type;
begin
   if optiune = 1 then
      open v_op for select *
               from operation;
   elsif optiune = 2 then
      open v_op for select *
               from operation
               where operation_room = 134;
   elsif optiune = 3 then
      open v_op for select *
               from operation
               where to_char(operation_date, 'mm') = to_char(sysdate, 'mm') and
to_char(operation_date, 'yyyy') = to_char(sysdate, 'yyyy');
   else
      dbms_output.put_line('Nu exista optiunea introdusa');
   end if;
      loop
         fetch v_op into v_date;
         exit when v_op%notfound;
         select last_name, first_name
         bulk collect into v_doc
         from doctor d join operation_doctor od on (d.id = od.doctor_id)
         where od.operation_id = v_date.id;
         select last_name, first_name
         into nume_pacient, prenume_pacient
         from patient
         where id = v_date.patient_id;
         DBMS_OUTPUT.PUT_LINE('-----------------------------------------------');
         for i in v_doc.first..v_doc.last loop
            DBMS_OUTPUT.PUT_LINE('Doctor: '|| v_doc(i).nume_doctor || ' ' ||
v_doc(i).prenume_doctor);
```

```
        end loop;
        DBMS_OUTPUT.PUT_LINE('Pacient: '||nume_pacient || ' ' || prenume_pacient);
        DBMS_OUTPUT.PUT_LINE('Ziua: '|| v_date.operation_date);
        DBMS_OUTPUT.PUT_LINE('Ora: ' || v_date.operation_time);
        DBMS_OUTPUT.PUT_LINE('-----------------------------------------------');
      end loop;
      close v_op;
   EXCEPTION
      when no_data_found then
         DBMS_OUTPUT.PUT_LINE('Nu exista informatiile cerute');
      when others then
         DBMS_OUTPUT.PUT_LINE('S-a produs o eroare!');
end afisare_operatii;

   function medic_pacient (nume_pacient in patient.last_name%type default 'Sigal')
return doctor.last_name%type
is
medic doctor.last_name%type;
begin
   select d.last_name
   into medic
   from doctor d join patient_doctor pd on (d.id = pd.doctor_id)
            join patient p on (pd.pacient_id = p.id)
   where lower(p.last_name) = lower(nume_pacient);
   return medic;
   EXCEPTION
      when no_data_found then
        DBMS_OUTPUT.PUT_LINE('Nu exista un pacient cu acest nume!');
        return -1;
      when too_many_rows then
        DBMS_OUTPUT.PUT_LINE('Pacientul ' || nume_pacient || ' are mai multi doctori');
        return -1;
      when others then
        DBMS_OUTPUT.PUT_LINE('Alta eroare!');
        return -1;
end medic_pacient;

   procedure proc_afisare_doctori
is
cursor c_date_medic is
   select d.last_name, d.first_name, d.age, specialization, d.phone_number, d.email
   from doctor d join patient_doctor pd on (d.id = pd.doctor_id)
            join patient p on (pd.pacient_id = p.id)
```

```
            join operation o on (p.id = o.patient_id)
            join patient_admission pa on (p.id = pa.pacient_id)
   where (to_char(o.operation_date, 'mm') = to_char(sysdate, 'mm') and
to_number(to_char(pa.admission_date, 'mm')) = to_number(to_char(sysdate, 'mm')) - 1)
      or (to_char(o.operation_date, 'mm') = 'January' and to_char(pa.admission_date, 'mm')
= 'December' and
      to_number(to_char(o.operation_date, 'yyyy')) =
to_number(to_char(pa.admission_date, 'yyyy')) + 1);
type date_doctor is record (nume doctor.last_name%type, prenume
doctor.first_name%type, varsta doctor.age%type, spec doctor.specialization%type,
nr_telefon doctor.phone_number%type, email doctor.email%type);
type tablou_date_doctori is table of date_doctor;
tablou_doctori tablou_date_doctori;
begin
   open c_date_medic;
   fetch c_date_medic bulk collect into tablou_doctori;
   close c_date_medic;
   if tablou_doctori.exists(1) then
      for i in tablou_doctori.first..tablou_doctori.last loop
         DBMS_OUTPUT.PUT_LINE ('------------------------------');
         DBMS_OUTPUT.PUT_LINE ('Nume: ' || tablou_doctori(i).nume || ' ' ||
tablou_doctori(i).prenume);
         DBMS_OUTPUT.PUT_LINE ('Varsta: ' || tablou_doctori(i).varsta);
         DBMS_OUTPUT.PUT_LINE ('Specializare: ' || tablou_doctori(i).spec);
         DBMS_OUTPUT.PUT_LINE ('Numar telefon: ' || tablou_doctori(i).nr_telefon);
         DBMS_OUTPUT.PUT_LINE ('Adresa email: ' || tablou_doctori(i).email);
         DBMS_OUTPUT.PUT_LINE ('------------------------------');
      end loop;
   else raise no_data_found;
   end if;
   EXCEPTION
      when no_data_found then
         DBMS_OUTPUT.PUT_LINE ('Nu exista medici cu pacienti operati luna curenta si
internati luna precedenta');
      when others then
         DBMS_OUTPUT.PUT_LINE ('Alta eroare!');
end proc_afisare_doctori;
end pachet_spital;
/
```

```
Package Body PACHET_SPITAL compiled
```

```
--Apel subprograme pachet
begin
```

```
    dbms_output.put_line('-------------------------');
    dbms_output.put_line('--AFISARE_DATE_MEDIC--');
    pachet_spital.afisare_date_medic;
    dbms_output.put_line('-------------------------');
    dbms_output.put_line('--AFISARE_OPERATII--');
    pachet_spital.afisare_operatii(1);
    dbms_output.put_line('-------------------------');
    pachet_spital.afisare_operatii(2);
    dbms_output.put_line('-------------------------');
    pachet_spital.afisare_operatii(3);
    dbms_output.put_line('-------------------------');
    pachet_spital.afisare_operatii(4);
    dbms_output.put_line('-------------------------');
    dbms_output.put_line('--MEDIC_PACIENT--');
    dbms_output.put_line(pachet_spital.medic_pacient);
    dbms_output.put_line('-------------------------');
    dbms_output.put_line('--PROC_AFISARE_DOCTORI--');
    pachet_spital.proc_afisare_doctori;
end;
/
```

orcldb

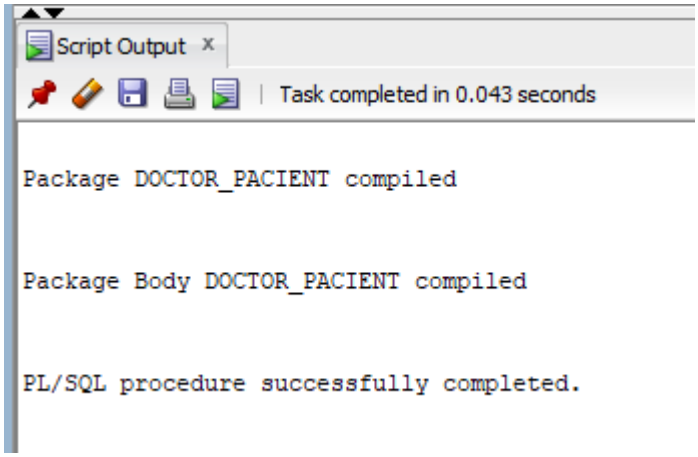
```
-------------------------
--AFISARE_DATE_MEDIC--
Nu exista astfel de medici
-------------------------
--AFISARE_OPERATII--
-----------------------------------------------
Doctor: Lorentz Andrew
Doctor: Popp Andrew
Doctor: Higgins Shelly
Pacient: Sigal Tobias
Ziua: 18-JUN-20
Ora: 17;45
-----------------------------------------------
-----------------------------------------------
Doctor: Lorentz Jhon
Doctor: Grant Tina
Doctor: Higgins Shelly
Pacient: Sigal Tobias
Ziua: 20-JUN-20
Ora: 18:00
-----------------------------------------------
-----------------------------------------------
Doctor: Lorentz Jhon
Doctor: Hall Peter
Doctor: Higgins Shelly
Pacient: Ernst Bruce
Ziua: 21-JUL-20
Ora: 09:00
-----------------------------------------------
-----------------------------------------------
Doctor: Lorentz Andrew
Doctor: Higgins Shelly
Pacient: King Nancy
Ziua: 23-JUL-20
Ora: 09:00
-----------------------------------------------
-----------------------------------------------
Doctor: Lorentz Andrew
Doctor: Higgins Shelly
Pacient: Popp Louise
Ziua: 23-JUL-20
Ora: 10:45
-----------------------------------------------
-----------------------------------------------
```

```
--------------------------------------------
Doctor: Lorentz Andrew
Doctor: Higgins Shelly
Pacient: Popp Louise
Ziua: 23-JUL-20
Ora: 10:45
--------------------------------------------
--------------------------------------------
Doctor: Lorentz Andrew
Doctor: Higgins Shelly
Pacient: Khan Elena
Ziua: 03-DEC-20
Ora: 12:00
--------------------------------------------
--------------------------
--------------------------------------------
Doctor: Lorentz Andrew
Doctor: Popp Andrew
Doctor: Higgins Shelly
Pacient: Sigal Tobias
Ziua: 18-JUN-20
Ora: 17;45
--------------------------------------------
--------------------------------------------
Doctor: Lorentz Jhon
Doctor: Grant Tina
Doctor: Higgins Shelly
Pacient: Sigal Tobias
Ziua: 20-JUN-20
Ora: 18:00
--------------------------------------------
--------------------------
--------------------------
Nu exista optiunea introdusa
S-a produs o eroare!
--------------------------
--MEDIC_PACIENT--
Pacientul Sigal are mai multi doctori
-1
--------------------------
--PROC_AFISARE_DOCTORI--
Nu exista medici cu pacienti operati luna curenta si internati luna precedenta
```

```
--14
--Sa se afiseze date despre doctori impreuna cu lista pacientilor lor

create or replace package doctor_pacient as
   --record pentru datele doctorului
   type doctor is record
      (cod C##STEFI.doctor.id%type, nume C##STEFI.doctor.last_name%type,
prenume C##STEFI.doctor.first_name%type, spec
C##STEFI.doctor.specialization%type,
      varsta C##STEFI.doctor.age%type, salariu C##STEFI.doctor.salary%type);
   --record pentru datele pacientului
   type pacient is record
      (cod patient.id%type, nume patient.last_name%type, prenume
patient.last_name%type, varsta patient.age%type, numar_telefon
patient.phone_number%type);
   --tip de date pentru datele pacientilor
   type v_pacienti is table of pacient;
   --tablou ce contine datele pacientilor
   vp v_pacienti;
   --tip de date pentru datele doctorului
   type v_doctor is table of doctor;
   --tablou ce contine datele doctorilor
   vd v_doctor;
   --tip de date pentru pastrarea fiecarui doctor in parte cu pacientii lui
   type doc_pacienti is record
      (doc doctor, pacienti v_pacienti);
   type v_doc_pacienti is table of doc_pacienti;
   colectie_afisabila v_doc_pacienti := v_doc_pacienti();
   procedure afisare_medici_pacienti;
end doctor_pacient;
/
create or replace package body doctor_pacient as
   procedure selectare_date is
      begin
         select id, last_name, first_name, specialization, age, salary
         bulk collect into vd
         from doctor;
```

```
        for i in vd.first..vd.last loop
          colectie_afisabila.extend();
          colectie_afisabila(colectie_afisabila.last).doc := vd(i);
          select p.id, p.last_name, p.first_name, p.age, p.phone_number
          bulk collect into vp
          from patient p join patient_doctor pd on (p.id = pd.pacient_id)
          where pd.doctor_id = vd(i).cod;
          colectie_afisabila(colectie_afisabila.last).pacienti := vp;
        end loop;
      end selectare_date;
      procedure afisare_medici_pacienti is
        begin
          selectare_date;
          for i in colectie_afisabila.first..colectie_afisabila.last loop
            DBMS_OUTPUT.PUT_LINE ('-----------------------------------------');
            DBMS_OUTPUT.PUT_LINE ('Doctor: ' ||
colectie_afisabila(i).doc.nume || ' ' || colectie_afisabila(i).doc.prenume);
            DBMS_OUTPUT.PUT_LINE ('Specializare: ' ||
colectie_afisabila(i).doc.spec);
            DBMS_OUTPUT.PUT_LINE ('Varsta: ' ||
colectie_afisabila(i).doc.varsta);
            DBMS_OUTPUT.PUT_LINE ('Salariu: ' ||
colectie_afisabila(i).doc.salariu);
            DBMS_OUTPUT.PUT_LINE ('Lista pacienti: ');
            if colectie_afisabila(i).pacienti.count > 0 then
              for j in
colectie_afisabila(i).pacienti.first..colectie_afisabila(i).pacienti.last loop
              DBMS_OUTPUT.PUT_LINE ('Pacient:' ||
colectie_afisabila(i).pacienti(j).nume || ' ' ||
colectie_afisabila(i).pacienti(j).prenume);
              DBMS_OUTPUT.PUT_LINE ('Varsta: ' ||
colectie_afisabila(i).pacienti(j).varsta);
              DBMS_OUTPUT.PUT_LINE ('Numar telefon: ' ||
colectie_afisabila(i).pacienti(j).numar_telefon);
              DBMS_OUTPUT.PUT_LINE ('');
              end loop;
            else
```

```
                DBMS_OUTPUT.PUT_LINE ('Doctorul nu are in grija niciun
pacient');
              end if;
              DBMS_OUTPUT.PUT_LINE ('----------------------------------------');
            end loop;
          end afisare_medici_pacienti;
end doctor_pacient;
/
begin
   doctor_pacient.afisare_medici_pacienti;
end;
/
```



orcldb

```
-----------------------------------------
Doctor: Lorentz Andrew
Specializare: Cardiology
Varsta: 40
Salariu: 5000
Lista pacienti:
Pacient:Khan Elena
Varsta: 21
Numar telefon: 0787

Pacient:King Steven
Varsta: 34
Numar telefon: 0723

Pacient:Ernst Bruce
Varsta: 22
Numar telefon: 0724

Pacient:Popp Louise
Varsta: 55
Numar telefon: 0725

Pacient:Sigal Tobias
Varsta: 18
Numar telefon: 0727

-----------------------------------------


-----------------------------------------
Doctor: Ernst Anna
Specializare: Pediatrics
Varsta: 55
Salariu: 5000
Lista pacienti:
Doctorul nu are in grija niciun pacient
-----------------------------------------

-----------------------------------------
```

orcldb

```
-----------------------------------------
Doctor: Lorentz Jhon
Specializare: Orthopaedics
Varsta: 28
Salariu: 5000
Lista pacienti:
Pacient:Ernst Bruce
Varsta: 22
Numar telefon: 0724

Pacient:King Nancy
Varsta: 30
Numar telefon: 0726

-----------------------------------------


-----------------------------------------
Doctor: Sigal Ana
Specializare: Radiology
Varsta: 65
Salariu: 5000
Lista pacienti:
Doctorul nu are in grija niciun pacient
-----------------------------------------


-----------------------------------------
Doctor: Hall Peter
Specializare: Dermatology
Varsta: 40
Salariu: 5000
Lista pacienti:
Pacient:King Steven
Varsta: 34
Numar telefon: 0723

Pacient:Ernst Bruce
Varsta: 22
Numar telefon: 0724

Pacient:Popp Louise
Varsta: 55
Numar telefon: 0725

Pacient:King Nancy
Varsta: 30
Numar telefon: 0726
```

```
Pacient:Sigal Tobias
Varsta: 18
Numar telefon: 0727

-----------------------------------------


-----------------------------------------
Doctor: Popp Andrew
Specializare: Lab assistant
Varsta: 56
Salariu: 4500
Lista pacienti:
Doctorul nu are in grija niciun pacient
-----------------------------------------


-----------------------------------------
Doctor: Grant Douglas
Specializare: Lab assistant
Varsta: 25
Salariu: 4500
Lista pacienti:
Doctorul nu are in grija niciun pacient
-----------------------------------------


-----------------------------------------
Doctor: Grant Tina
Specializare: Lab assistant
Varsta: 26
Salariu: 4500
Lista pacienti:
Doctorul nu are in grija niciun pacient
-----------------------------------------


-----------------------------------------
Doctor: Higgins Shelly
Specializare: Anesthetist
Varsta: 40
Salariu: 4200
Lista pacienti:
Doctorul nu are in grija niciun pacient
-----------------------------------------
```