**Sistemul de Gestiune al Bazei de Date (SGBD)**

**BRAGĂ IONELA-CRISTIANA**

**Seria A, grupa 1033, Informatică-Economică**

**PROFESOR COORDONATOR:** Iuliana Botha

****

## CUPRINS

I.Descrierea bazei de date ..………………………………………………………………………………….3

II.Schema conceptuala………………………………………………………………………………………….5

III.Crearea tabelelor…………………….……………………………………………………………………….6

IV.Adaugarea inregistrarilor………………………………………………………………………………….8

V.Structuri alternative si repetitive si tratarea exceptiilor….……………………………….12

VI.Gestionarea cursorilor……………………………………………………………………………………16

VII.Functii. Proceduri. Pachete……………………………………………………………………………19

VIII.Declansatori…………………………………………………………………………………………………25

IX.Interfețe cu utilizatorul- APEX............................................................................27

GESTIUNEA UNUI SPITAL

1. **Descrierea bazei de date**

“St Andrew's Hospital”

O bază de date memorează informaţii despre secţiile, medicii şi pacienţii, unui spital din Marea Britanie. Internarea se realizează, atât pe baza unei recomandări făcute de medicul de familie, cât şi a unei confirmări din partea unui specialist existent în spital. La internare sunt înregistrate datele personale ale pacienţilor, iar o fişă separată, ţine evidenţa investigaţiilor făcute pe perioada internării cu tratamentul aplicat. Medicii sunt specialişti în diverse ramuri şi pot avea mai mulţi pacienţi sub supraveghere.

Baza de date trebuie să asigure gestionarea corespunzătoare a tuturor datelor referitoare la medicamentele şi produsele stocate in depozitele spitalului si inventarul acestuia, cât şi la listarea pacienţilor in funcţie de id-ul medicului, data internării, data externării, diagnostic sau tratament.

* Reguli structurale

Intrucat spitalul din Marea Britanie are mai multe **secţii**, unde la rândul lor, au mai mulţi **medici**, trebuie reţinute informaţii despre medici si sectii.

Totodată trebuie reţinute informaţii despre **pacienţii** spitalului, despre **fişele de medic** şi despre **tratamentele** care rezulta după cercetari amănunţite. Pe baza unui tratament recomandat de un medic, impreuna cu **medicamente** prescrise, va rezulta **cazul** pacientului respectiv.

Baza mea de date contine, 7 entitati, denumirea lor fiind cuvintele bold-uite in descrierea anterioara. Initial in structura, regaseam 2 relatii M:M, intre Pacienti - Medici si Tratamente-Medicamente, insa aceastea au disparut prin entitatile Fise\_medici si Cazuri, iar acum avem relatii 1:M. Pentru o intelegere mai exacta, am explicitat legaturile dintre entitati:

Fiecare secţie trebuie să aibă unul sau mai mulţi medici.

Fiecare medic trebuie să apartină unei singure secţii.

Fiecare medic trebuie să aibă unul sau mai mulţi pacienţi.

Fiecare pacient trebuie să apartină unuia sau mai multor medici.

Fiecare medic trebuie să aibă una sau mai multe fişe.

Fiecare fişă trebuie să apartină unui singur medic.

Fiecare fişă trebuie să realizeze unul sau mai multe tratamente.

Fiecare medicament trebuie să trateze unul sau mai multe cazuri.

Fiecare tratament duce la realizarea unui sau mai multor cazuri..

Prin modelarea entitatilor in PL/SQL Developer, acestea au devenit tabele si au urmatoarele atribute si restrictii.

* Secţii (deţine informaţii despre secţiile din cadrul spitalului din Marea Britanie): id\_secţie (cu restrictie de cheie primara PK si de NOT NULL), nume\_sectie (unique);
* Medici (deţine informaţii despre medicii din cadrul fiecărei secţii): id\_medic(PK si NOT NULL), nume, prenume, specializare, data\_naşterii, salariul, id\_sectie(cheie straina FK);
* Pacienti (deţine informaţii despre pacienţii spitalului) id\_pacient (PK si NOT NULL), nume, prenume, adresă;
* Fişe\_medici (detine informaţii despre colaborarea unui medic cu un pacient): id\_fişa (PK), data\_internării, data\_exterării, id\_medic(FK), id\_pacient(FK);
* Medicament (deţine informaţii despre medicamentele existente în spital): id\_medicament(PK), denumire, prospect;
* Tratamente (detine informatii despre tratamentul, pe care un pacient trebuie sa-l urmeze): id\_tratament(PK), doza, id\_fisa(FK),
* Cazuri (detine informaţii despre cazul final al pacientului): id\_caz (PK), observatie, id\_medicament (FK), id\_tratament (FK).

Observand explicitarea relatiilor de mai sus, observăm că nu există dependențe tranzitive. Deci relațiile sunt în forma normală 3, de unde rezultă ca baza de date este în FN3.

Toate aceste explicatii le puteti vizualiza mai bine in schema conceptuala de mai jos, dar si in print screen-urile realizate, atat asupra tabelelor, cat si a restrictiilor.

1. **Schema conceptuală**

SECTII

PACIENTI

MEDICI

are

**ID\_MEDIC NUMBER(5)PK**

NUME VARCHAR2(25)

PRENUME VARCHAR2(25)

SPECIALIZARE VARCHAR2(25)

DATA\_NASTERII DATE

SALARIUL NUMBER(5)

**ID\_SECTIE NUMBER (5)FK**

**ID\_SECTIE NUMBER(5)PK**

NUME VARCHAR2(25)

**ID\_PACIENT NUMBER(5)PK**

NUME\_PACIENT VARCHAR2(25)

PRENUME\_PACIENT VARCHAR2(25)

ADRESA VARCHAR2(25)

dfc

apartine

FISE\_MEDICI

**ID\_FISA NUMBER(5)PK**

DIAGNOSTIC VARCHAR2(25)

DATA\_INTERNARII DATE

DATA\_EXTERNARII DATE

**ID\_MEDIC NUMBER(5)FK**

**ID\_PACIENT NUMBER(5)FK**

apartine

poate avea

este realizat pe baza

TRATAMENTE

duce la realizarea

**ID\_TRATAMENT NUMBER(5)PK**

DOZA VARCHAR2(25)

**ID\_FISA** NUMBER(5)**FK**

CAZURI

**ID\_CAZ NUMBER(5)PK**

OBSERVATIE VARCHAR2(25)

**ID\_MEDICAMENT NUMBER(5)FK**

**ID\_TRATAMENT NUMBER(5)FK**

apartine

MEDICAMENTE

trateaza

**ID\_MEDICAMENT NUMBER(5)FK**

DENUMIRE VARCHAR2(25)

PROSPECT VARCHAR2(25)

1. **Crearea tabelelor**

create table SECTII

(

ID\_SECTIE NUMBER(5),

NUME\_SECTIE VARCHAR2(25)

);

alter table SECTII

add constraint ID\_SECTIE\_PK primary key (ID\_SECTIE);

alter table SECTII

add constraint ID\_SECTIE\_NN

check ("ID\_SECTIE" IS NOT NULL);

alter table SECTII

add constraint NUME\_SECTIE\_UN

unique ("NUME\_SECTIE" );

create table MEDICI

(

ID\_MEDIC NUMBER(5),

NUME VARCHAR2(25),

PRENUME VARCHAR2(25),

DATA\_NASTERII DATE,

SPECIALIZARE VARCHAR2(25),

SALARIUL NUMBER(5),

ID\_SECTIE NUMBER(5)

);

alter table MEDICI

add constraint ID\_MEDIC\_PK primary key (ID\_MEDIC);

alter table MEDICI

add constraint ID\_MEDIC\_NN

check ("ID\_MEDIC" IS NOT NULL);

alter table MEDICI

add constraint ID\_SECTIE\_FK foreign key (ID\_SECTIE)

references SECTII (ID\_SECTIE);

create table PACIENTI

(

ID\_PACIENT NUMBER(5),

NUME\_PACIENT VARCHAR2(25),

PRENUME\_PACIENT VARCHAR2(25),

ADRESA VARCHAR2(25)

);

alter table PACIENTI

add constraint ID\_PACIENT\_PK primary key (ID\_PACIENT);

alter table PACIENTI

add constraint ID\_PACIENT\_NN

check ("ID\_PACIENT" IS NOT NULL);

create table FISE\_MEDICI

(

ID\_FISA NUMBER(5),

DATA\_INTERNARII DATE,

DATA\_EXTERNARII DATE,

ID\_MEDIC NUMBER(5),

ID\_PACIENT NUMBER(5),

DIAGNOSTIC VARCHAR2(20)

);

alter table FISE\_MEDICI

add constraint ID\_FISA\_PK primary key (ID\_FISA);

alter table FISE\_MEDICI

add constraint ID\_MEDIC\_FK foreign key (ID\_MEDIC)

references MEDICI (ID\_MEDIC);

alter table FISE\_MEDICI

add constraint ID\_PACIENT\_FK foreign key (ID\_PACIENT)

references PACIENTI (ID\_PACIENT);

create table MEDICAMENTE

(

ID\_MEDICAMENT NUMBER(5),

DENUMIRE VARCHAR2(25),

PROSPECT VARCHAR2(25)

);

alter table MEDICAMENTE

add constraint ID\_MEDICAMENT\_PK primary key (ID\_MEDICAMENT);

create table TRATAMENTE

(

ID\_TRATAMENT NUMBER(5),

DOZA VARCHAR2(25),

ID\_FISA NUMBER(5),

);

alter table TRATAMENTE

add constraint ID\_TRATAMENT\_PK primary key (ID\_TRATAMENT);

alter table TRATAMENTE

add constraint ID\_FISA\_FK foreign key (ID\_FISA)

references FISE\_MEDICI (ID\_FISA);

create table CAZURI

(

ID\_CAZ NUMBER(5),

ID\_MEDICAMENT NUMBER(5),

ID\_TRATAMENT NUMBER(5),

OBSERVATIE VARCHAR2(25)

);

alter table CAZURI

add constraint ID\_CAZ\_PK primary key (ID\_CAZ);

alter table CAZURI

add constraint ID\_MEDICAMENT\_FK foreign key (ID\_MEDICAMENT)

references MEDICAMENTE (ID\_MEDICAMENT);

alter table CAZURI

add constraint ID\_TRATAMENT\_FK foreign key (ID\_TRATAMENT)

references TRATAMENTE (ID\_TRATAMENT);

1. **Adaugarea inregistrarilor**

INSERT INTO SECTII VALUES(1,'Pediatrie');

INSERT INTO SECTII VALUES(2,'Chirurgie');

INSERT INTO SECTII VALUES(3,'Dermatologie');

INSERT INTO SECTII VALUES(4,'Oftalmologie');

INSERT INTO SECTII VALUES(5,'Stomatologie');

INSERT INTO SECTII VALUES(6,'Ginecologie');

INSERT INTO SECTII VALUES(7,'Psihiatrie');

INSERT INTO SECTII VALUES(8,'Ortopedie');

INSERT INTO SECTII VALUES(9,'Boli infectioase');

INSERT INTO SECTII VALUES(10,'Sectie speciala');



INSERT INTO PACIENTI(ID\_PACIENT,NUME\_PACIENT,PRENUME\_PACIENT,ADRESA) VALUES (95, 'Halep','Adriana','33654,Suceava');

INSERT INTO PACIENTI(ID\_PACIENT,NUME\_PACIENT,PRENUME\_PACIENT,ADRESA) VALUES (96, 'Manea','Teodor','521468,Tulcea');

INSERT INTO PACIENTI(ID\_PACIENT,NUME\_PACIENT,PRENUME\_PACIENT,ADRESA) VALUES (97, 'Gutui','Florica','125664,Alexandria');

INSERT INTO PACIENTI(ID\_PACIENT,NUME\_PACIENT,PRENUME\_PACIENT,ADRESA) VALUES (98, 'Chitu','Andrei','665522,Valea Cascadelor');

INSERT INTO PACIENTI(ID\_PACIENT,NUME\_PACIENT,PRENUME\_PACIENT,ADRESA) VALUES (99, 'Zamfir','Ioana','954425,Focsani');

INSERT INTO PACIENTI(ID\_PACIENT,NUME\_PACIENT,PRENUME\_PACIENT,ADRESA) VALUES (100,'Tasbac','Maria','965235, Sos.Pipera');

INSERT INTO PACIENTI(ID\_PACIENT,NUME\_PACIENT,PRENUME\_PACIENT,ADRESA) VALUES (101,'Costea','Flavius','785126, Galati');

INSERT INTO PACIENTI(ID\_PACIENT,NUME\_PACIENT,PRENUME\_PACIENT,ADRESA) VALUES (102,'Tapu','Florina','365897, Rucar');

INSERT INTO PACIENTI(ID\_PACIENT,NUME\_PACIENT,PRENUME\_PACIENT,ADRESA) VALUES (103, 'Mitu','Madalina','111A,Iuliu Maniu');

INSERT INTO PACIENTI(ID\_PACIENT,NUME\_PACIENT,PRENUME\_PACIENT,ADRESA) VALUES(104,'Duta','Catalin','3022, Bd. Constructorilor');



INSERT INTO MEDICI VALUES (300,'Albulescu','Ion',to\_date('17-06-1987','DD-MM-YYYY'),'Medic Chirurg-Estetic',1500,2);

INSERT INTO MEDICI VALUES (301,'Nedelcu','Lucian',to\_date('23-04-1968','DD-MM-YYYY'),'Medic Pediatru',2000,1);

INSERT INTO MEDICI VALUES (302,'Ciobanu','Rares',to\_date('01-03-1970','DD-MM-YYYY'),'Medic Neurochirurg',3000,2);

INSERT INTO MEDICI VALUES (303,'Enache','Simona',to\_date('19-12-1982','DD-MM-YYYY'),'Medic Oftalmolog ',2500,4);

INSERT INTO MEDICI VALUES (304,'Boja','Violeta',to\_date('10-10-1971','DD-MM-YYYY'),'Medic Chirurg-cardiac',3100,2);

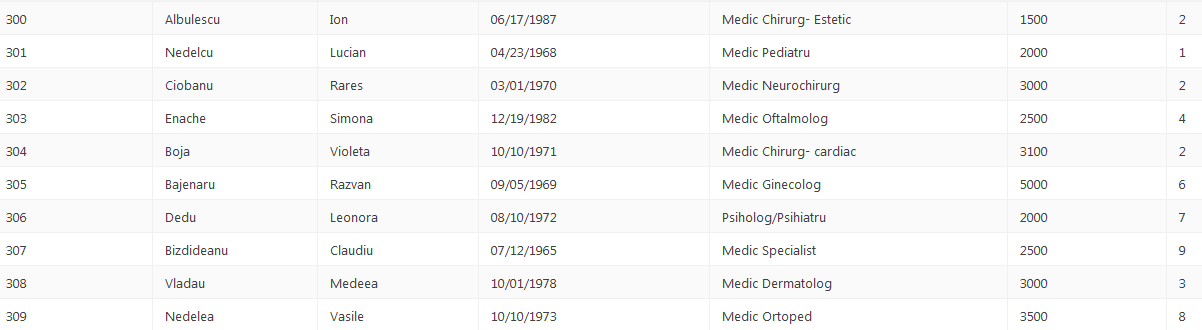
INSERT INTO MEDICI VALUES (305,'Bajenaru','Razvan',to\_date('05-09-1969','DD-MM-YYYY'),'Medic Ginecolog',5000,6);

INSERT INTO MEDICI VALUES (306,'Dedu','Leonora',to\_date('10-08-1972','DD-MM-YYYY'),'Psiholog/Psihiatru',2000,7);

INSERT INTO MEDICI VALUES (307,'Bizdideanu','Claudiu',to\_date('12-07-1965','DD-MM-YYYY'),'Medic Specialist',2500,9);

INSERT INTO MEDICI VALUES (308,'Vladau','Medeea',to\_date('01-10-1978','DD-MM-YYYY'),'Medic Dermatolog ',3000,3);

INSERT INTO MEDICI VALUES (309,'Nedelea','Vasile',to\_date('10-10-1973','DD-MM-YYYY'),'Medic Ortoped',3500,8);



INSERT INTO FISE\_MEDICI VALUES (365,to\_date('07-10-2015','DD-MM-YYYY'),to\_date('10-10-2015','DD-MM-YYYY'),300,101,'Obozitate faza II');

INSERT INTO FISE\_MEDICI VALUES (502,to\_date('23-07-2015','DD-MM-YYYY'),to\_date('10-08-2015','DD-MM-YYYY'),304,100,'Tahicardie');

INSERT INTO FISE\_MEDICI VALUES (715,to\_date('05-11-2015','DD-MM-YYYY'),to\_date('13-11-2015','DD-MM-YYYY'),302,104,'Parkinson');

INSERT INTO FISE\_MEDICI VALUES (847,to\_date('12-12-2015','DD-MM-YYYY'),to\_date('13-12-2015','DD-MM-YYYY'),303,103,'Miopie faza avansata');

INSERT INTO FISE\_MEDICI VALUES (906,to\_date('15-10-2015','DD-MM-YYYY'),to\_date('25-10-2015','DD-MM-YYYY'),302,102,'Tumoare cerebrala');

INSERT INTO FISE\_MEDICI VALUES (1005,to\_date('18-11-2015','DD-MM-YYYY'),to\_date('25-11-2015','DD-MM-YYYY'),301,99,'Pneumonie lobara');

INSERT INTO FISE\_MEDICI VALUES (1163,to\_date('08-08-2015','DD-MM-YYYY'),to\_date('12-08-2015','DD-MM-YYYY'),308,96,'Acnee Severa');

INSERT INTO FISE\_MEDICI VALUES (1255,to\_date('10-11-2015','DD-MM-YYYY'),to\_date('13-11-2015','DD-MM-YYYY'),309,98,'Osteogeneze');

INSERT INTO FISE\_MEDICI VALUES (1459,to\_date('15-08-2015','DD-MM-YYYY'),to\_date('15-09-2015','DD-MM-YYYY'),306,97,'Stres');

INSERT INTO FISE\_MEDICI VALUES (1500,to\_date('15-10-2015','DD-MM-YYYY'),to\_date('16-10-2015','DD-MM-YYYY'),303,95,'Inflamarea irisului');



INSERT INTO MEDICAMENTE VALUES (49,'Algocalmin','3/zi');

INSERT INTO MEDICAMENTE VALUES (48,'Supramax gel','dimineata si seara');

INSERT INTO MEDICAMENTE VALUES (47,'Extraveral','1capsula/18h');

INSERT INTO MEDICAMENTE VALUES (46,'Zinerit','3 aplicari/zi');

INSERT INTO MEDICAMENTE VALUES (50,'Dehydrobenzperidol','3/zi');

INSERT INTO MEDICAMENTE VALUES (51,'Ketonal','anti-inflamator,la durere');

INSERT INTO MEDICAMENTE VALUES (52,'Achetaminophen','2/zi,dupa mese');

INSERT INTO MEDICAMENTE VALUES (53,'Moldamin','1 injectabila/12h');

INSERT INTO MEDICAMENTE VALUES (54,'Ocunet','3 picaturi/zi');

INSERT INTO MEDICAMENTE VALUES (55,'Entamoeba','1 capsula/6h');



INSERT INTO TRATAMENTE VALUES (200,'2 cutii',365);

INSERT INTO TRATAMENTE VALUES (201,'1 cutie',502);

INSERT INTO TRATEMENTE VALUES (202,'3 tablete',715);

INSERT INTO TRATAMENTE VALUES (203,'1 tub',847);

INSERT INTO TRATAMENTE VALUES (204,'5 fiole',906);

INSERT INTO TRATAMENTE VALUES (205,'2 cutii',1005);

INSERT INTO TRATAMENTE VALUES (206,'1 cutie',1163);

INSERT INTO TRATEMENTE VALUES (207,'3 tablete',1255);

INSERT INTO TRATAMENTE VALUES (208,'1 tub',1459);

INSERT INTO TRATAMENTE VALUES (209,'5 fiole',1500);



INSERT INTO CAZURI VALUES (700,51,200,'la durere dupa operat');

INSERT INTO CAZURI VALUES (701,55,201,'Minim 2saptamani');

INSERT INTO CAZURI VALUES (702,50,202,'Pot exista efecte adverse');

INSERT INTO CAZURI VALUES (703,54,203,'Picaturi doar seara');

INSERT INTO CAZURI VALUES (704,53,204,'-');

INSERT INTO CAZURI VALUES (705,53,205,'Atentie la supradoza');

INSERT INTO CAZURI VALUES (706,46,206,'Poate crea iritatii!');

INSERT INTO CAZURI VALUES (707,48,207,'Aplicari usoare seara!');

INSERT INTO CAZURI VALUES (708,49,208,'-');

INSERT INTO CAZURI VALUES (709,54,209,'Picaturi doar seara');



1. **Structuri alernative si repetitive/Tratarea exceptiilor**
2. Afisati salariul mediu al medicilor,dar si salariul celor sub medie.

DECLARE

v\_sal medici.salariul%type;

v\_salmediu v\_sal%type;

i number(5):=300;

BEGIN

SELECT avg(salariul) into v\_salmediu from medici;

dbms\_output.put\_line('Salariul mediu: '||v\_salmediu);

loop

select salariul into v\_sal from medici where id\_medic=i;

dbms\_output.put\_line('Medicul cu id-ul '||i||' are salariul: '||v\_sal);

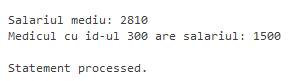
i:=i+1;

exit when v\_sal<v\_salmediu or i>310;

end loop;

end;

/



1. Afisati pentru fiecare medic, numele, prenumele, specializarea si salariul folosind structura while.

DECLARE

id medici.id\_medic%TYPE;

v\_nume medici.nume%TYPE;

v\_prenume medici.prenume%TYPE;

v\_min medici.id\_medic%TYPE;

v\_max medici.id\_medic%TYPE;

v\_salariul medici.salariul%TYPE;

v\_specializare medici.specializare%TYPE;

begin

select min(id\_medic),max(id\_medic) into v\_min,v\_max from medici;

while (v\_min <= v\_max)

loop

select nume, prenume, specializare, salariul into v\_nume,v\_prenume,v\_specializare, v\_salariul

from medici

where id\_medic=v\_min;

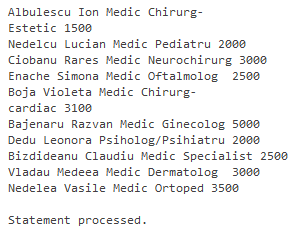
DBMS\_OUTPUT.PUT\_LINE(v\_nume||' '||v\_prenume||' '||v\_specializare||' '||v\_salariul);

v\_min:=v\_min+1;

END loop;

END;

/



1. Sa se afiseze salariul minim si sa se genereze o exceptie daca acesta este mai mic de 2000.

DECLARE

salariul\_minim\_exception EXCEPTION;

v\_salariul medici.salariul%type;

BEGIN

SELECT min(salariul) into v\_salariul

from medici;

if v\_salariul<2000 then

RAISE salariul\_minim\_exception;

else

DBMS\_OUTPUT.PUT\_LINE('SALARIUL MINIM ESTE '|| v\_salariul);

end if;

EXCEPTION

when salariul\_minim\_exception then

DBMS\_OUTPUT.PUT\_LINE('SALARIUL MINIM ESTE MAI MIC DE

2000');

END;

/

C:\Users\Cristiana\Desktop\Untitled.png

1. Pentru medicii din sectia 2/sectia 10, sa se afiseze numele. Sa se trateze exceptiile NO\_DATA\_FOUND si TOO\_MANY\_ROWS.

DECLARE

v\_nume VARCHAR2(20);

BEGIN

SELECT nume INTO v\_nume

FROM medici

WHERE id\_sectie=10;

dbms\_output.put\_line(v\_nume);

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

dbms\_output.put\_line('Nu exista medic momentan care sa lucreze in aceasta sectie!');

WHEN TOO\_MANY\_ROWS THEN

DBMS\_OUTPUT.PUT\_LINE('Exista mai multi medici in aceasta sectie!');

END;

/

C:\Users\Cristiana\Desktop\Untitled.png🡪sectia 10

C:\Users\Cristiana\Desktop\Untitled.png🡪sectia 2

1. Afisati id, numele si varsta tuturor medicilor si de asemenea, realizati o clasificare in functie de varsta acestora. Daca numarul este mai mic de 35, atunci medicul este unul TANAR, intre 35-50 ani ‘Medic cu varsta medie’ iar peste 50 ‘Medic aproape de pensionare’.

DECLARE

v\_nume varchar2(25);

v\_prenume varchar2(25);

v\_nr number(5);

v\_nume1 varchar2(50);

BEGIN

for i in 300..309 loop

select nume,prenume, round((sysdate-data\_nasterii)/365) into v\_nume,v\_prenume,v\_nr

from medici

where id\_medic=i;

v\_nume1:= case

when v\_nr<35 then 'Medic tanar'

when v\_nr between 35 and 50 then 'Medic cu varsta medie'

else 'Medic aproape de pensionare'

end;

dbms\_output.put\_line('Medicul cu codul '||i||' are numele '||v\_nume||' '||v\_prenume||' si varsta-------'||v\_nr||' de ani. Status:-> '||v\_nume1);

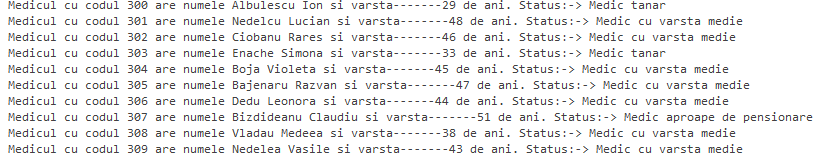
end loop;

EXCEPTION

when NO\_DATA\_FOUND then

dbms\_output.put\_line('Nu exista medici cu aceste coduri!');

end;



1. **Gestionarea cursorilor**
2. Schimbati denumirea medicamentului 547 in ‘Periculos’; tratati excepti in caz de acest medicament nu exista.

DECLARE

invalid\_medicament EXCEPTION;

BEGIN

UPDATE medicamente

SET denumire='PERICULOS'

WHERE id\_medicament=547;

IF SQL%NOTFOUND THEN

RAISE invalid\_medicament;

END IF;

EXCEPTION

WHEN invalid\_medicament THEN

DBMS\_OUTPUT.PUT\_LINE('Nu exista medicamentul cu acest ID');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('A aparut o eroare! Nu se poate actualiza denumirea medicamentului!');

END;

C:\Users\Cristiana\Desktop\Untitled.png

1. Stergeti pacientii care au stat mai mult de 4 luni in spital; tratati exceptia in caz de nu exista.

DECLARE

invalid\_fisa EXCEPTION;

BEGIN

DELETE

FROM fise\_medici

WHERE data\_externarii-data\_internarii>120;

IF SQL%NOTFOUND THEN

RAISE invalid\_fisa;

END IF;

EXCEPTION

WHEN invalid\_fisa THEN

DBMS\_OUTPUT.PUT\_LINE('Nu exista pacienti, care sa stea mai mult de 4 luni in spital! ');

END;

C:\Users\Cristiana\Desktop\Untitled.png

1. Afisati numele, specializarea si data nasterii, pentru medicii ce lucreaza in sectia 2

DECLARE

cursor c is select nume, specializare, data\_nasterii

from medici

where id\_sectie=2;

v\_nume medici.nume%type;

v\_specializare medici.specializare%type;

v\_data date;

Begin

open c;

Loop

fetch c into v\_nume, v\_specializare, v\_data;

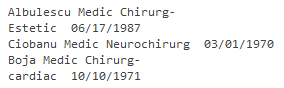
exit when c%notfound;

DBMS\_OUTPUT.PUT\_LINE(v\_nume||' '||v\_specializare||' '||v\_data);

end loop;

close c;

end;



1. Afisati pentru toti medicii, numele si varsta.

DECLARE

cursor c is select nume, prenume, round((sysdate-data\_nasterii)/365,0) varsta

from medici;

r c%rowtype;

Begin

open c;

Loop

fetch c into r;

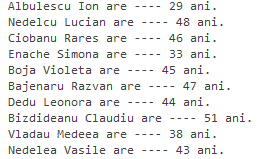
exit when c%notfound;

DBMS\_OUTPUT.PUT\_LINE(r.nume||' '||r.prenume||' are ---- '||r.varsta||' ani.');

end loop;

close c;

end;



1. Consultand fisele de medic, afisati primii 5 pacienti, care au fost internati in spital cele mai putine zile.

DECLARE

cursor c is select nume\_pacient,prenume\_pacient, data\_externarii-data\_internarii durata

from pacienti p,fise\_medici f

where p.id\_pacient=f.id\_pacient

order by data\_externarii-data\_internarii asc;

r c%rowtype;

BEGIN

open c;

Loop

fetch c into r;

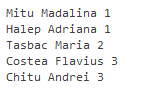
exit when c%rowcount=6;

DBMS\_OUTPUT.PUT\_LINE(r.nume\_pacient||' '||r.prenume\_pacient||' '|| r.durata);

end loop;

close c;

END;



1. **Functii. Proceduri. Pachete**
2. Functie ce contine un parametru de IN , testeaza daca un medic are salariul sau nu peste pragul de 2000 de lei.

CREATE OR REPLACE FUNCTION salariul\_medic (a IN NUMBER)

RETURN BOOLEAN

AS

v\_salariul number(5);

BEGIN

SELECT salariul into v\_salariul

from medici

where id\_medic=a;

IF v\_salariul > 2000 then

return true;

ELSE

return false;

end if;

EXCEPTION

WHEN no\_data\_found THEN

return NULL;

end;

C:\Users\Cristiana\Desktop\Untitled.png

BEGIN

IF (salariul\_medic(300) IS NULL) THEN

dbms\_output.put\_line('Acest medic nu exista!');

ELSIF (salariul\_medic(300)) THEN

dbms\_output.put\_line('Medicul are salariul peste 2000 de lei');

ELSE

dbms\_output.put\_line('Medicul nu are salariul peste 2000 de lei');

END IF;

END;

C:\Users\Cristiana\Desktop\Untitled.png

1. Functie cu un parametru de OUT, care intoarce cate sectii din spital sunt inca goale si nu exista niciun medic.

CREATE OR REPLACE FUNCTION sectie\_nula (a OUT NUMBER)

RETURN NUMBER

AS

BEGIN

select count(count(m.id\_medic)) into a

from medici m, sectii s

where m.id\_sectie(+)=s.id\_sectie

group by s.id\_sectie

having count(m.id\_medic)=0;

RETURN a;

EXCEPTION

WHEN no\_data\_found THEN

return 0;

end;

C:\Users\Cristiana\Desktop\Untitled.png

DECLARE

v\_nr number(3);

BEGIN

DBMS\_output.put\_line('In spital exista '||sectie\_nula(v\_nr)||' sectii, unde nu exista inca medici!');

end;

C:\Users\Cristiana\Desktop\Untitled.png

1. Functie care primeste 2 parametrii de IN, unul in care avem salariul fiecarui medic, iar cel de-al doilea un procent. S-a stabilit la nivel de spital ca in fiecare luna, 2% din salariul fiecarui medic sa se redirectioneze catre un fond al pacientilor fara venituri. Am afisat valoarea /luna ce se strange in acest fond.

CREATE OR REPLACE FUNCTION FONDURI (sal IN NUMBER, cotizatie IN NUMBER)

RETURN NUMBER

AS

BEGIN

RETURN (sal\*cotizatie);

END;

C:\Users\Cristiana\Desktop\Untitled.png

DECLARE

v\_sum number(3);

BEGIN

select sum(fonduri(salariul,0.02)) into v\_sum

from medici;

DBMS\_OUTPUT.PUT\_LINE('Cotizatia la nivel de luna pe care toti medicii au facut-o pentru pacientii fara venituri este de: '|| v\_sum ||' de lei!');

END;

C:\Users\Cristiana\Desktop\Untitled.png

1. Procedura fisa\_durata primeste printr-un parametru de IN id-ul fisei ce a fost realizata de un anumit medic pentru un pacient si unul de OUT care returneaza numarul de zile petrecute in spital.

CREATE OR REPLACE PROCEDURE FISA\_DURATA(a IN NUMBER, b OUT NUMBER)

AS

BEGIN

SELECT data\_externarii-data\_internarii into b

from fise\_medici

where id\_fisa=a;

dbms\_Output.put\_line('Pacientul a stat spitalizat '|| b ||' zile!');

EXCEPTION

when no\_data\_found then

dbms\_Output.put\_line('Nu exista aceasta fisa!');

end;

C:\Users\Cristiana\Desktop\Untitled.png

DECLARE

v\_nrZile number(3);

BEGIN FISA\_DURATA(1005,v\_nrZile);

end;

C:\Users\Cristiana\Desktop\Untitled.png

1. Procedura MEDICAMENT\_CAZ primeste 2 parametrii de OUT, unul prin care vom afisa denumirea medicamentului care a fost prescris de cele mai multe ori de catre medici, iar prin cel de-al doilea numarul de prescrieri.

CREATE OR REPLACE PROCEDURE MEDICAMENT\_CAZ(a OUT VARCHAR2, b OUT NUMBER)

AS

BEGIN

select denumire, count(m.id\_medicament) INTO a,b

from medicamente m,cazuri c

where m.id\_medicament=c.id\_medicament

group by denumire

having count(m.id\_medicament)=(select max(count(id\_medicament))

from cazuri

group by id\_medicament);

dbms\_Output.put\_line('Medicamentul '|| a ||' a fost prescris in '|| b ||' cazuri!');

EXCEPTION

WHEN TOO\_MANY\_ROWS THEN

dbms\_Output.put\_line('Mai multe medicamente au fost prescrise de numar maxim de ori!');

end;

C:\Users\Cristiana\Desktop\Untitled.png

DECLARE

v\_nume varchar2(20);

v\_nr number(2);

BEGIN

MEDICAMENT\_CAZ(v\_nume,v\_nr);

END;

C:\Users\Cristiana\Desktop\Untitled.png

1. Sa se creeze un packet in care sa existe o procedura ce afiseaza varsta unui medic si o functie care sa ne confirme/infirme daca el castiga peste 2000 de lei.

CREATE OR REPLACE PACKAGE pachet IS

PROCEDURE VARSTA\_MEDIC(a IN NUMBER);

FUNCTION salariul\_medic (a IN NUMBER)

RETURN BOOLEAN ;

END;

C:\Users\Cristiana\Desktop\Untitled.png

CREATE OR REPLACE PACKAGE BODY pachet IS

PROCEDURE VARSTA\_MEDIC(a IN NUMBER)

AS

v\_varsta number(30);

BEGIN

SELECT round((sysdate-data\_nasterii)/365,0) into v\_varsta

from medici

where id\_medic=a;

DBMS\_output.put\_line('Varsta medicului este '|| v\_varsta||' de ani!');

EXCEPTION

when no\_data\_found then

dbms\_Output.put\_line('NU s-au gasit date!');

end;

FUNCTION salariul\_medic (a IN NUMBER)

RETURN BOOLEAN

AS

v\_salariul number(5);

BEGIN

SELECT salariul into v\_salariul

from medici

where id\_medic=a;

IF v\_salariul > 2000 then

return true;

ELSE

return false;

end if;

EXCEPTION

WHEN no\_data\_found THEN

return NULL;

end;

END;

C:\Users\Cristiana\Desktop\Untitled.png

EXECUTE pachet.varsta\_medic(300);

C:\Users\Cristiana\Desktop\Untitled.png

1. Sa se creeze un pachet cu 2 proceduri de Adaugare, respectiv Stergere din sectiile spitalului, dar si 1 functie prin care sa se verifice daca exista o anumita sectie.

CREATE OR REPLACE PACKAGE operatii\_sectii AS

exceptie exception;

procedure adauga (v\_id NUMBER, v\_nume VARCHAR2);

procedure sterge(v\_id NUMBER);

function sectie(v\_id NUMBER)

return boolean;

END;

C:\Users\Cristiana\Desktop\Untitled.png

CREATE OR REPLACE PACKAGE BODY operatii\_sectii AS

procedure adauga (v\_id NUMBER, v\_nume VARCHAR2) AS

begin

if sectie(v\_id) then

raise exceptie;

else

insert into sectii values (v\_id,v\_nume);

end if;

EXCEPTION

when exceptie then

dbms\_output.put\_line('Sectie deja existenta!');

end;

procedure sterge(v\_id NUMBER) AS

begin

if sectie(v\_id) then

delete

from sectii

where id\_sectie=v\_id;

else

raise exceptie;

end if;

EXCEPTION

when exceptie then

dbms\_output.put\_line('Aceasta sectie nu exista!');

end;

function sectie (v\_id IN NUMBER)

RETURN boolean

AS

v\_nr number(5);

begin

select 1 into v\_nr

from sectii

where id\_sectie=v\_id;

return true;

EXCEPTION

when no\_data\_found then

return false;

end;

END;

C:\Users\Cristiana\Desktop\Untitled.png

EXECUTE operatii\_sectii.adauga(21,'Kinetoterapie’);

select \* from sectii;

EXECUTE operatii\_sectii.sterge(1);

select \* from sectii;

1. **Declansatori**
2. Se creează un declansator pentru a nu se permite depăşirea salariului de 5000 de lei pentru orice medic din spital

CREATE OR REPLACE TRIGGER medici\_restrictie

BEFORE INSERT or UPDATE on medici

FOR EACH ROW

DECLARE

v\_sal number(5);

BEGIN

select salariul into v\_sal

from medici;

IF :new.salariul>5000 then

RAISE\_APPLICATION\_ERROR (-20202, 'Nu se poate depasi salariul maxim in acest spital.');

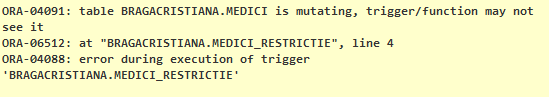
end if;

END;

**C:\Users\Cristiana\Desktop\Untitled.png**

update medici

set salariul =salariul\*2;

****

1. Se creează un declansator care asigură unicitatea codului produsului folosind valorile generate de o secvenţă

CREATE SEQUENCE pacienti\_secvet

START WITH 1

INCREMENT BY 1

MAXVALUE 1000

NOCYCLE;

C:\Users\Cristiana\Desktop\Untitled.png

CREATE OR REPLACE TRIGGER pacienti\_noi

BEFORE INSERT ON pacienti

FOR EACH ROW

BEGIN

SELECT pacienti\_secvet.nextval INTO :new.id\_pacient FROM dual;

END;

**C:\Users\Cristiana\Desktop\Untitled.png**

Insert into pacienti values(999,'Braga','Cristi','01125,Berceni');

Insert into pacienti values (999,'Mitu','Andreea','07545,Cluj');

Insert into pacienti values (999,'Dumitru','Sabina','04745,Iasi');

Insert into pacienti values(999,'Coman','Ionut','01125,Berceni');

select id\_pacient, nume\_pacient,prenume\_pacient

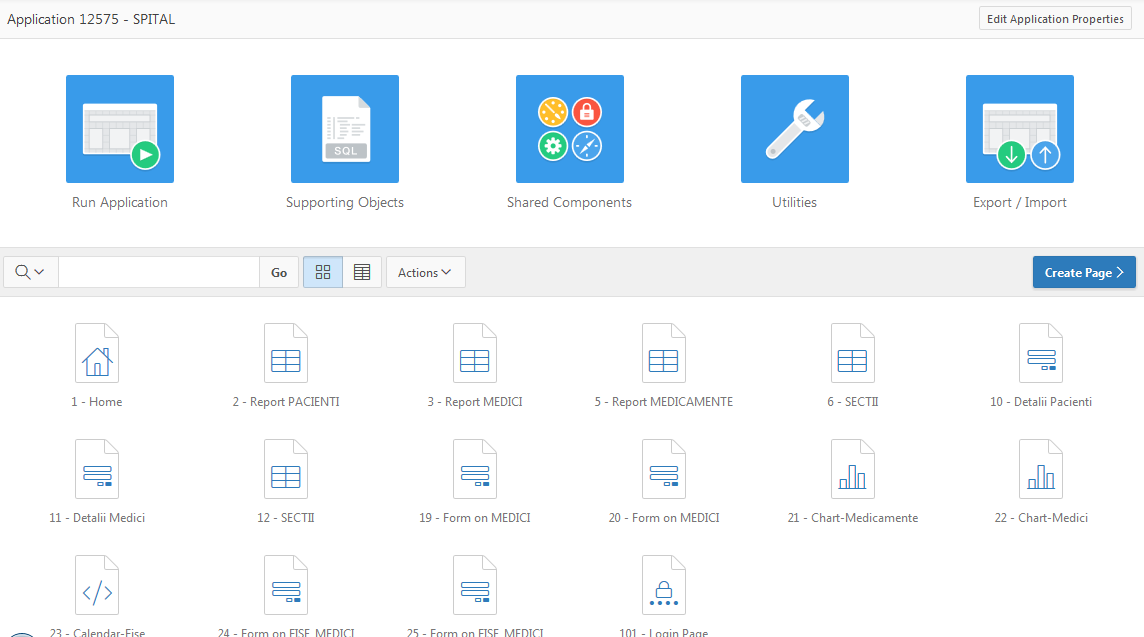
from pacienti

order by id\_pacient desc;



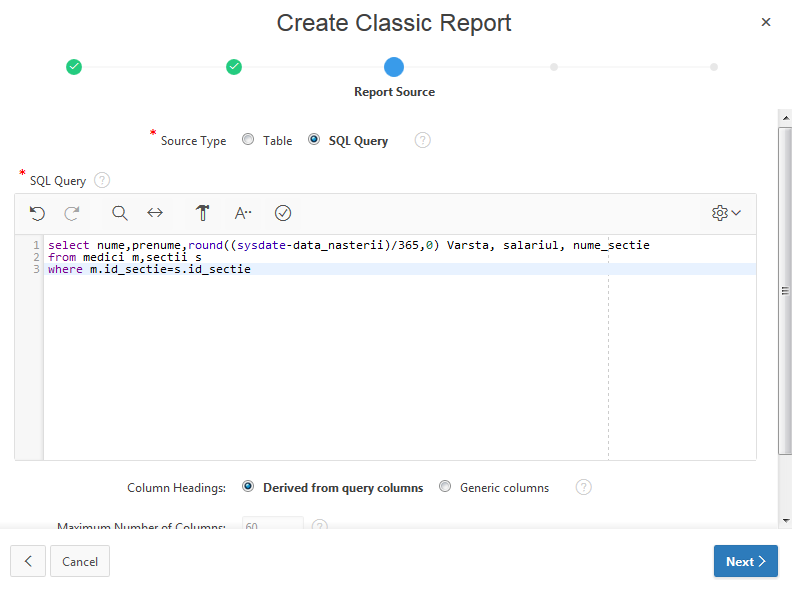
1. **APEX**

Am realizat o aplicatie Spital, unde regasim Raporarte, Formulare, Calendare dar si Chart-uri pe baze datelor existente.



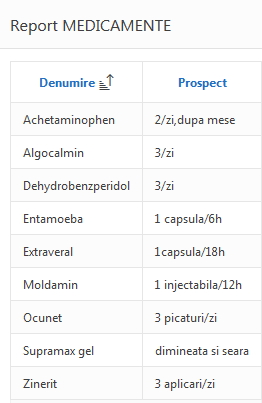
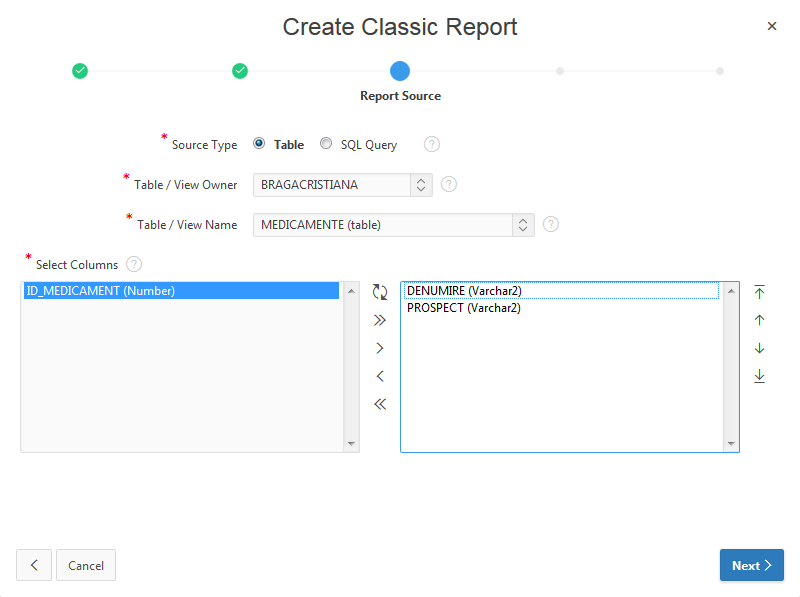
Le vom explicita pe fiecare in cele ce urmeaza.

* 3 Rapoarte Clasice cu medici existenti in spital, medicamentele disponibile si pacientii tratati:

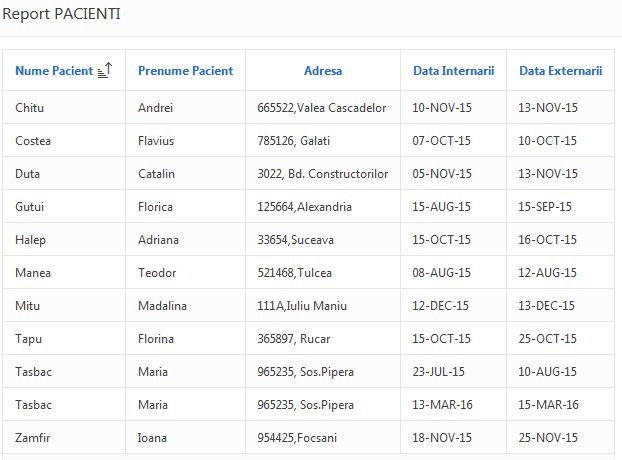
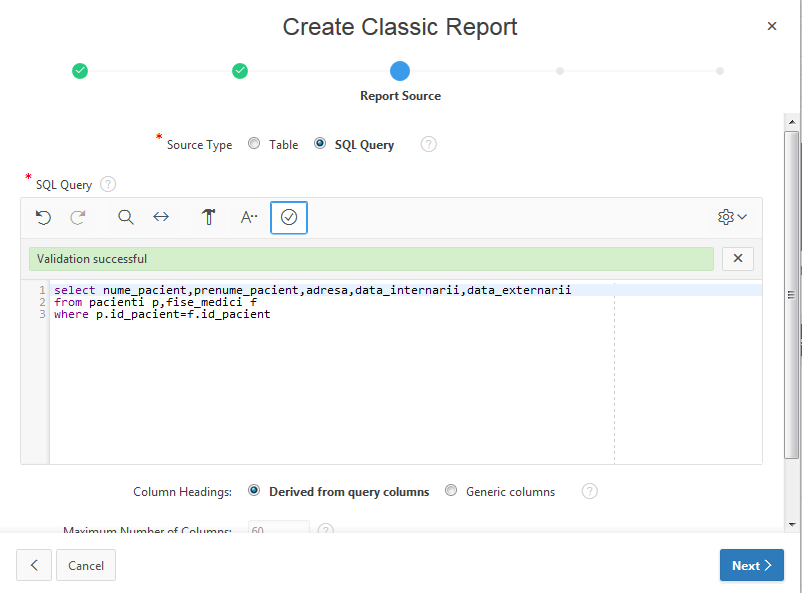


Pentru o usurinta in vizualizarea datelor, am afisat pentru fiecare medic varsta, dar si numele sectiei, unde acesta isi desfasoara activitatile.

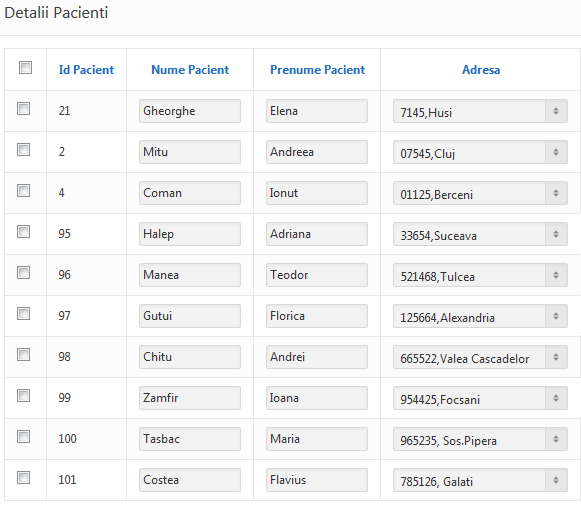
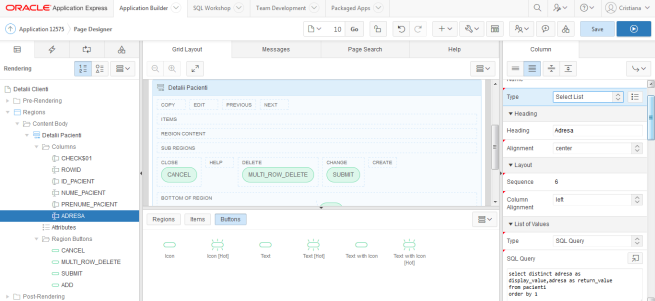
Medicamentele existente in stoc:



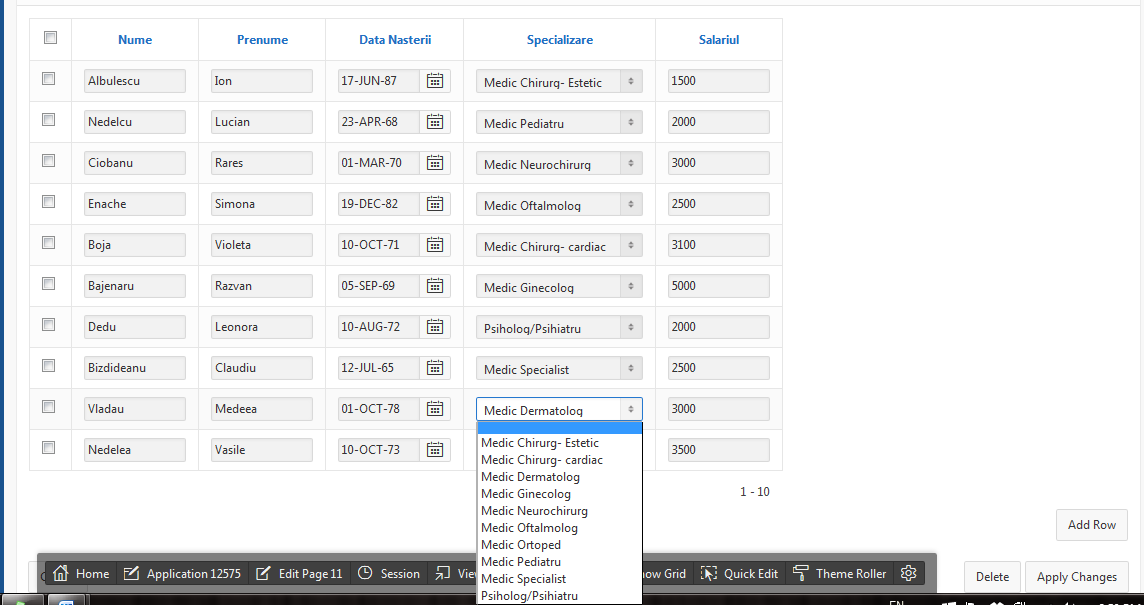
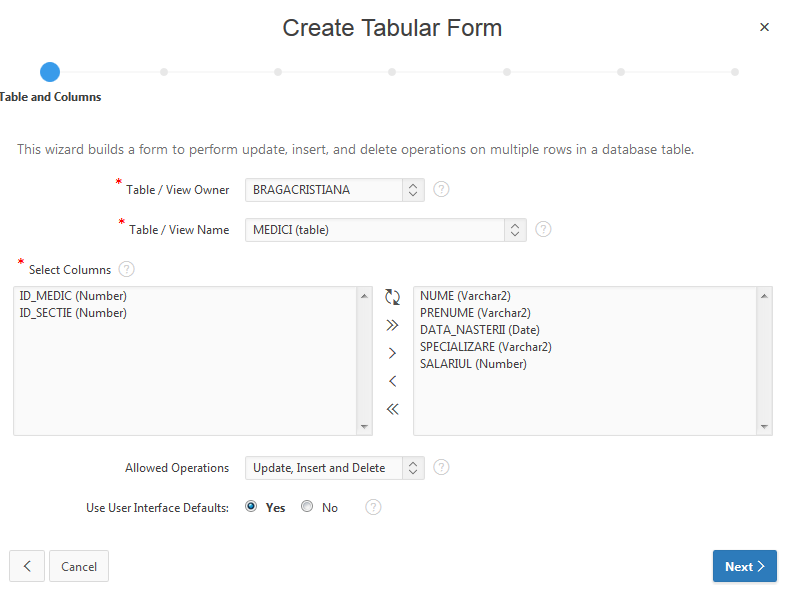
Numele, prenumele si adresa persoanelor tratate, insa si perioada de spitalizare.



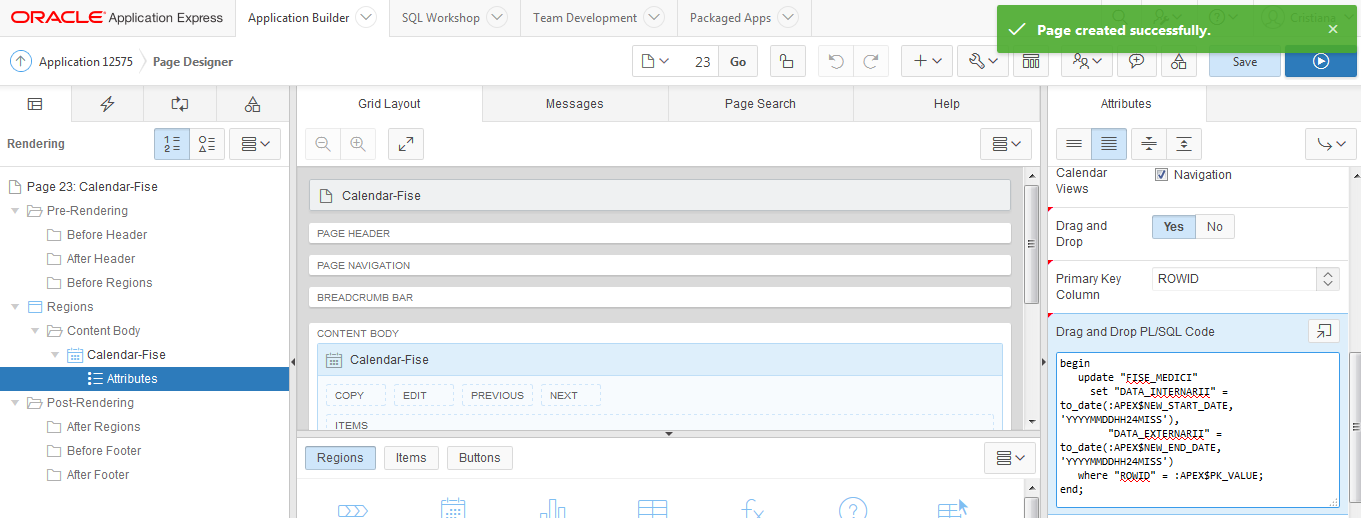
* 3 Formulare cu pacientii, medicii spitalului si fisele realizate. Observam un List of Value atat la adresa pacientilor, cat si la specializarea medicilor.

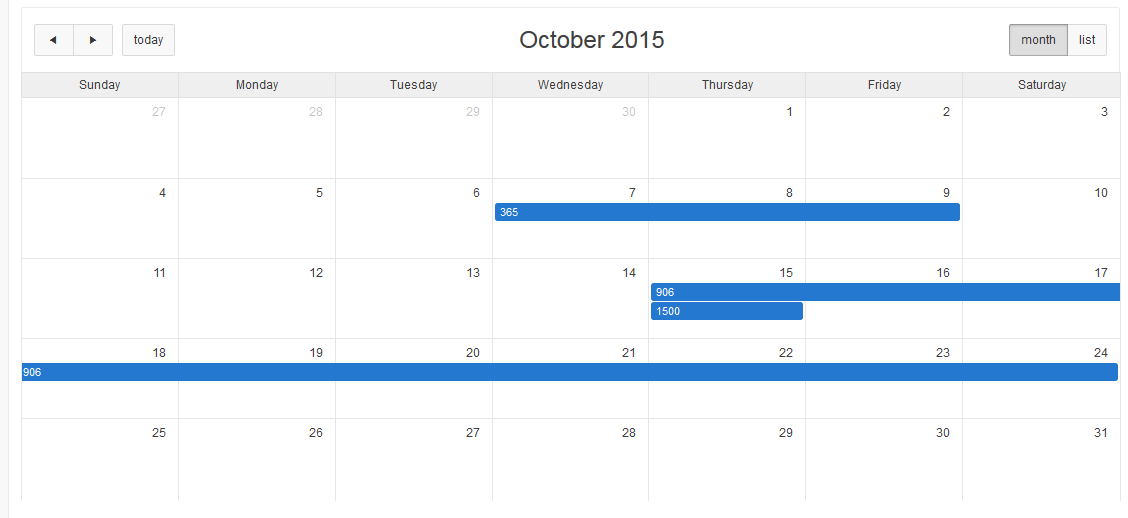
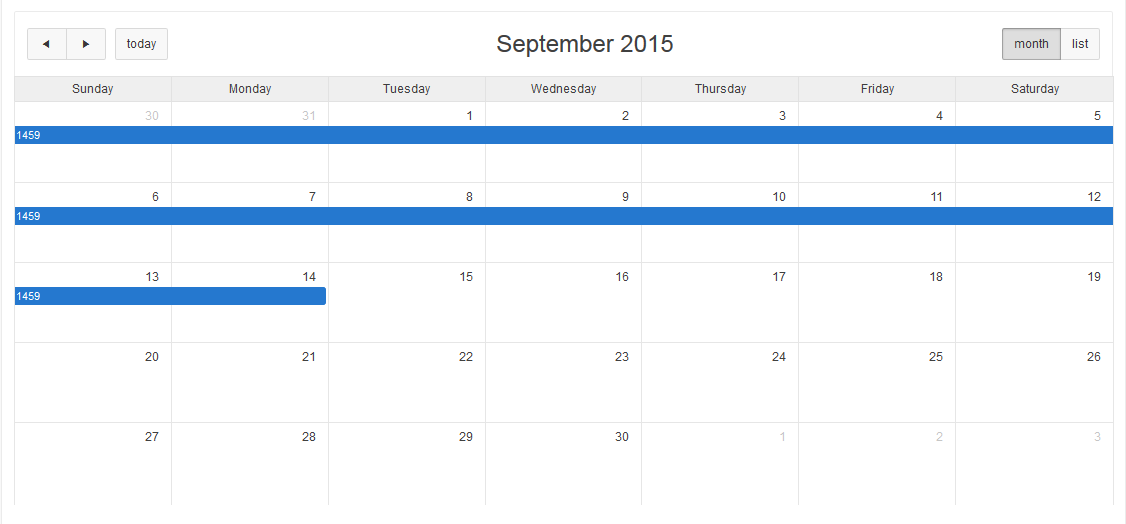


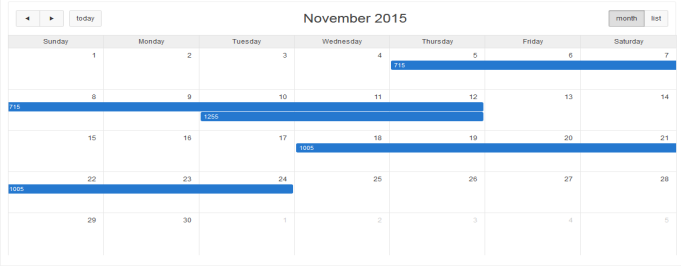
Observam buloanele din dreapta jos “Add Row” “Delete” “Apply changes”, unde putem adauga,sterge si modifica datele.



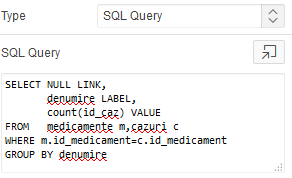
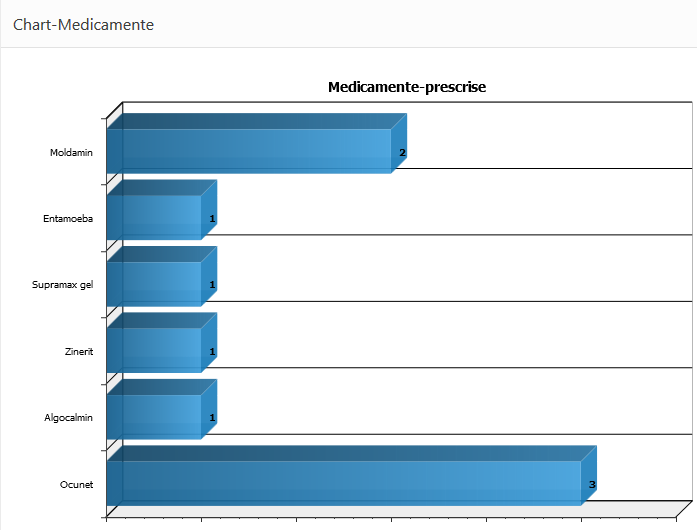
* Calendar cu fisele realizate de medici si durata spitalizarii.





 🡪Internarile din Septembrie, Octombrie si Noiembrie

* Chart-uri
* Grafic privind numarul de medicamente prescrise in tratarea cazurilor din spital.



* 3D Pie privind medicii si numarul de fise realizate.

