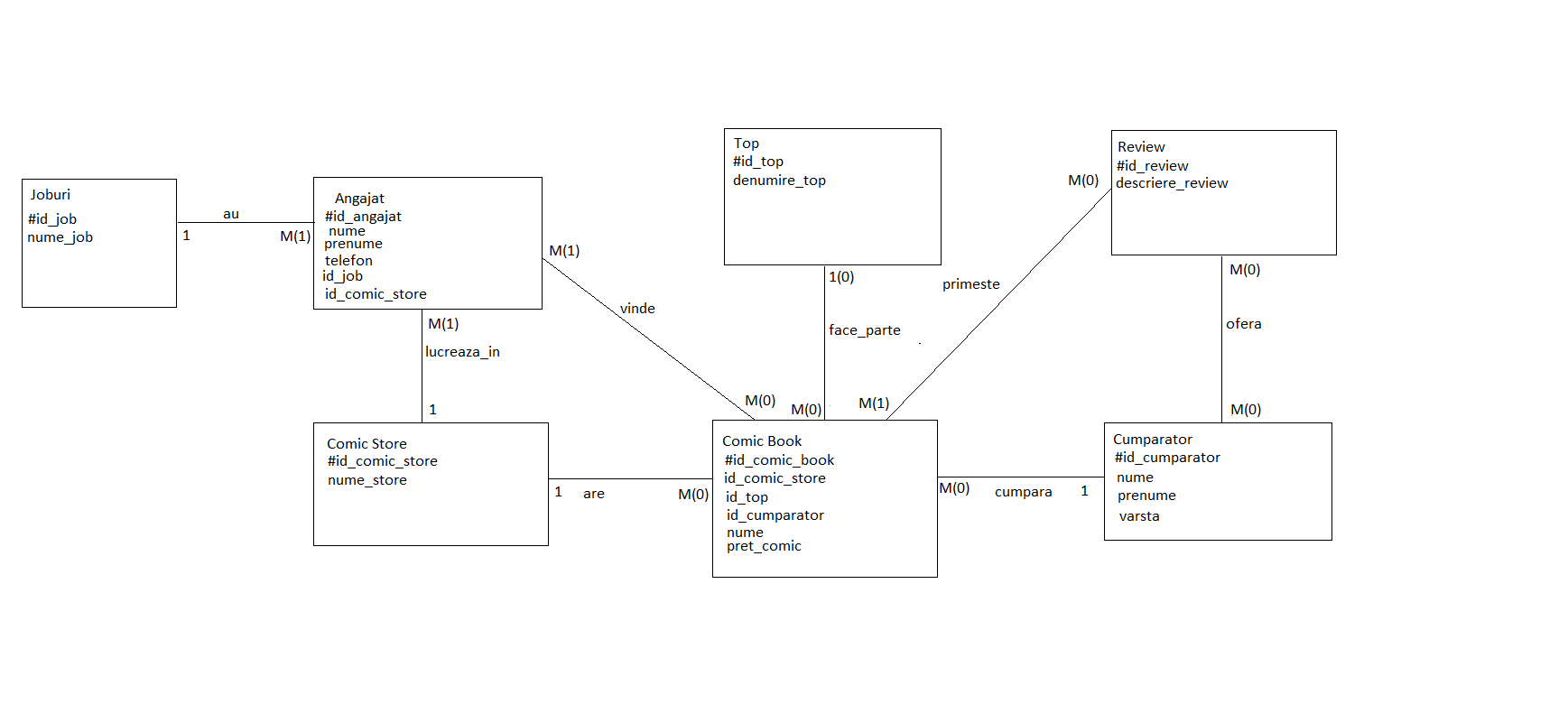
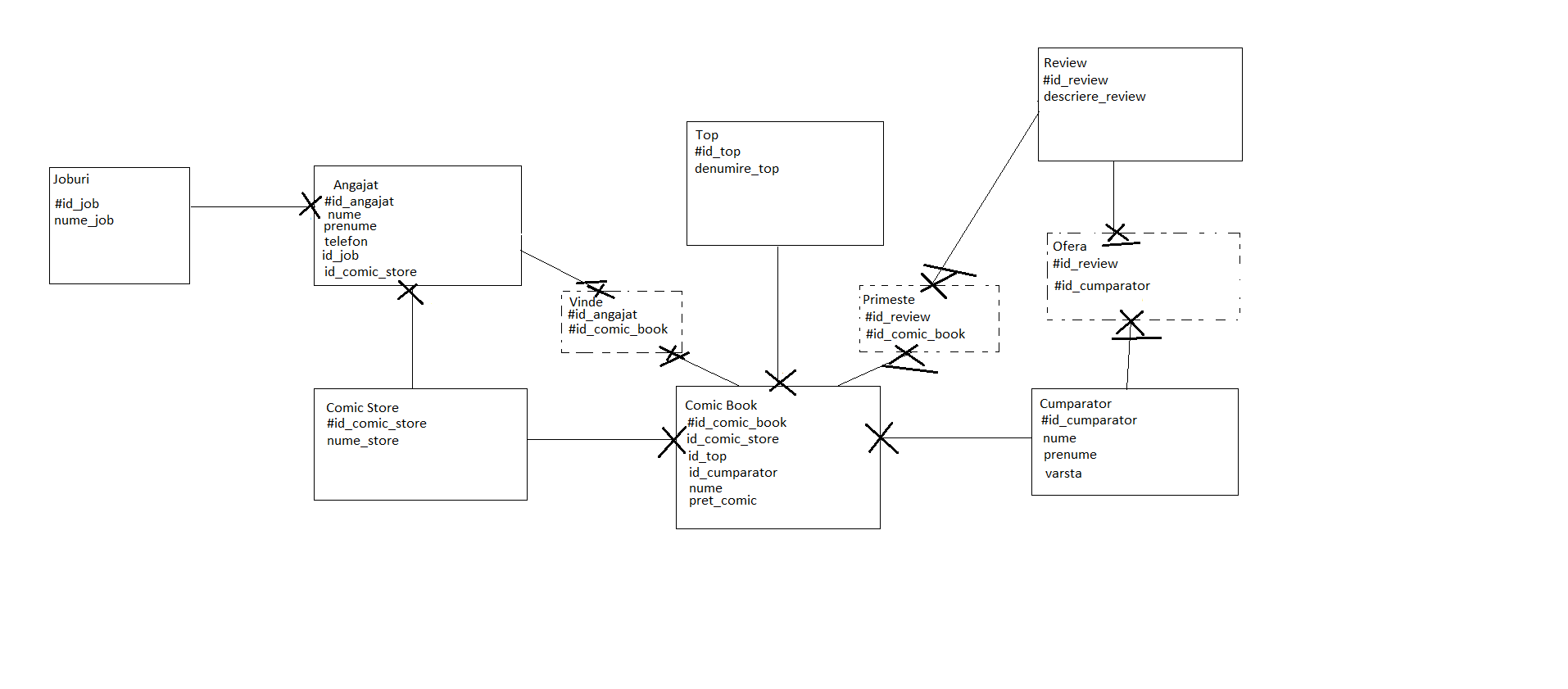
Proiect Sisteme de Gestiune a Bazelor de Date

Roșianu Robert – Grupa 233

1. Proiectul prezinta o baza de date folosita pentru administrarea unei afaceri ce se ocupa cu vanzarea benzilor desenate. Avem mai multe magazine unde se vand diverse reviste (vandute de anumiti angajati avand diverse joburi). Aceste reviste fac parte dintr-un top (nu este obligatoriu), au cate un comparator si primesc sau nu un review in functie de cat de bun este continutul revistei respective. Baza noastra de date ofera operatii ce ajuta la mentinerea informatiilor cat si la mentenanta acesteia.
2. Diagrama Entitate-Relatie:



1. Diagrama Conceptuala:



1. Tabele:

create table Joburi(

id\_job number(4) primary key,

nume\_job varchar2(10) not null

);

create table ComicStore(

id\_comic\_store number(4) primary key,

nume\_store varchar2(15) not null

);

create table Top(

id\_top number(4) primary key,

denumire\_top varchar2(20) not null

);

create table Cumparator(

id\_cumparator number(4) primary key,

nume varchar2(12) not null,

prenume varchar2(12) not null,

varsta number(4) not null

);

create table Review(

id\_review number(4) primary key,

descriere\_review varchar2(30) not null

);

create table ComicBook(

id\_comic\_book number(4) primary key,

id\_comic\_store number(4) references ComicStore(id\_comic\_store) on delete cascade,

id\_top number(4) references Top(id\_top) on delete cascade,

id\_cumparator number(4) references Cumparator(id\_cumparator) on delete cascade,

nume varchar2(20) not null,

pret number(4) not null

);

create table Angajat(

id\_angajat number(4) primary key,

id\_job number(4) references Joburi(id\_job) on delete cascade,

id\_comic\_store number(4) references ComicStore(id\_comic\_store) on delete cascade,

nume varchar2(10) not null,

prenume varchar2(10) not null,

telefon number(11) not null unique

);

create table Vinde(

id\_angajat number(4),

id\_comic\_book number(4),

primary key(id\_angajat, id\_comic\_book),

foreign key(id\_angajat) references Angajat(id\_angajat) on delete cascade,

foreign key(id\_comic\_book) references ComicBook(id\_comic\_book) on delete cascade

);

create table Primeste(

id\_review number(4),

id\_comic\_book number(4),

primary key(id\_review, id\_comic\_book),

foreign key(id\_review) references Review(id\_review) on delete cascade,

foreign key(id\_comic\_book) references ComicBook(id\_comic\_book) on delete cascade

);

create table Ofera(

id\_review number(4),

id\_cumparator number(4),

primary key(id\_review, id\_cumparator),

foreign key(id\_review) references Review(id\_review) on delete cascade,

foreign key(id\_cumparator) references Cumparator(id\_cumparator) on delete cascade

);

1. Inserare in tabele:

create sequence angajati

start with 1

nocycle;

create sequence jobs

start with 1

nocycle;

create sequence comics

start with 1

nocycle;

create sequence comic\_store

start with 1

nocycle;

create sequence tops

start with 1

nocycle;

create sequence cumparatori

start with 1

nocycle;

create sequence reviews

start with 1

nocycle;

insert into Joburi(id\_job, nume\_job)

values(jobs.nextval, 'Vanzator');

insert into Joburi(id\_job, nume\_job)

values(jobs.nextval, 'Manager');

insert into Joburi(id\_job, nume\_job)

values(jobs.nextval, 'Asistent');

insert into Joburi(id\_job, nume\_job)

values(jobs.nextval, 'Patron');

insert into Joburi(id\_job, nume\_job)

values(jobs.nextval, 'Curatator');

insert into Joburi(id\_job, nume\_job)

values(jobs.nextval, 'Investitor');

insert into Joburi(id\_job, nume\_job)

values(jobs.nextval, 'Asistent');

insert into ComicStore(id\_comic\_store, nume\_store)

values (comic\_store.nextval, 'Best Books');

insert into ComicStore(id\_comic\_store, nume\_store)

values (comic\_store.nextval, 'Hentai Heaven');

insert into ComicStore(id\_comic\_store, nume\_store)

values (comic\_store.nextval, 'Top Comics');

insert into ComicStore(id\_comic\_store, nume\_store)

values (comic\_store.nextval, 'Last Avengers');

insert into ComicStore(id\_comic\_store, nume\_store)

values (comic\_store.nextval, 'Downtown Comics');

insert into ComicStore(id\_comic\_store, nume\_store)

values (comic\_store.nextval, 'New Order Shop');

insert into ComicStore(id\_comic\_store, nume\_store)

values (comic\_store.nextval, 'Stranger Shop');

insert into ComicStore(id\_comic\_store, nume\_store)

values (comic\_store.nextval, 'Big Offer');

insert into ComicStore(id\_comic\_store, nume\_store)

values (comic\_store.nextval, 'New Nw');

insert into Angajat(id\_angajat, id\_job, id\_comic\_store, nume, prenume, telefon)

values (angajati.nextval,1,1, 'Ciocan', 'Madalin', 40763312934);

insert into Angajat(id\_angajat, id\_job, id\_comic\_store, nume, prenume, telefon)

values (angajati.nextval,2,1, 'Silviu', 'Lungu', 40763312314);

insert into Angajat(id\_angajat, id\_job, id\_comic\_store, nume, prenume, telefon)

values (angajati.nextval,3,1, 'Matea', 'Marian', 40762345674);

insert into Angajat(id\_angajat, id\_job, id\_comic\_store, nume, prenume, telefon)

values (angajati.nextval,4,2, 'Liviu', 'Sandu', 40743212934);

insert into Angajat(id\_angajat, id\_job, id\_comic\_store, nume, prenume, telefon)

values (angajati.nextval,5,2, 'Spataru', 'Sulean', 40743444934);

insert into Angajat(id\_angajat, id\_job, id\_comic\_store, nume, prenume, telefon)

values (angajati.nextval,1,2, 'Coltea', 'Spitalu', 40733312934);

insert into Angajat(id\_angajat, id\_job, id\_comic\_store, nume, prenume, telefon)

values (angajati.nextval,3,2, 'Cosmina', 'Ciortea', 40743299934);

insert into Angajat(id\_angajat, id\_job, id\_comic\_store, nume, prenume, telefon)

values (angajati.nextval,2,2, 'Sandu', 'Ciorba', 40712312734);

insert into Angajat(id\_angajat, id\_job, id\_comic\_store, nume, prenume, telefon)

values (angajati.nextval,5,1, 'Florin', 'Florin', 40743123454);

insert into Angajat(id\_angajat, id\_job, id\_comic\_store, nume, prenume, telefon)

values (angajati.nextval,4,1, 'Florin', 'Dinu', 40746754454);

insert into Angajat(id\_angajat, id\_job, id\_comic\_store, nume, prenume, telefon)

values (angajati.nextval,2,3, 'Martinel', 'Lungu', 40743555554);

insert into Angajat(id\_angajat, id\_job, id\_comic\_store, nume, prenume, telefon)

values (angajati.nextval,1,3, 'Guta', 'Nicolae', 40711123454);

insert into Angajat(id\_angajat, id\_job, id\_comic\_store, nume, prenume, telefon)

values (angajati.nextval,4,3, 'Trinu', 'Minu', 40743187874);

insert into Angajat(id\_angajat, id\_job, id\_comic\_store, nume, prenume, telefon)

values (angajati.nextval,5,4, 'Legion', 'Seven', 40741212654);

insert into Angajat(id\_angajat, id\_job, id\_comic\_store, nume, prenume, telefon)

values (angajati.nextval,2,4, 'Cultan', 'Oirr', 40755123552);

insert into Angajat(id\_angajat, id\_job, id\_comic\_store, nume, prenume, telefon)

values (angajati.nextval,1,4, 'Petrica', 'Simion', 40743666884);

insert into Angajat(id\_angajat, id\_job, id\_comic\_store, nume, prenume, telefon)

values (angajati.nextval,5,4, 'Minuc', 'Sad', 40724561834);

insert into Angajat(id\_angajat, id\_job, id\_comic\_store, nume, prenume, telefon)

values (angajati.nextval,1,5, 'Sidan', 'Pisan', 40215384912);

insert into Angajat(id\_angajat, id\_job, id\_comic\_store, nume, prenume, telefon)

values (angajati.nextval,2,5, 'Lionel', 'Messi', 40724444434);

insert into Angajat(id\_angajat, id\_job, id\_comic\_store, nume, prenume, telefon)

values (angajati.nextval,1,5, 'Cristiano', 'Ronaldo', 40724656224);

insert into Angajat(id\_angajat, id\_job, id\_comic\_store, nume, prenume, telefon)

values (angajati.nextval,5,5, 'Vecinu', 'Debloc', 40712361994);

insert into Angajat(id\_angajat, id\_job, id\_comic\_store, nume, prenume, telefon)

values (angajati.nextval,2,6, 'Ghioc', 'David', 40721122334);

insert into Angajat(id\_angajat, id\_job, id\_comic\_store, nume, prenume, telefon)

values (angajati.nextval,1,6, 'Flesa', 'Manar', 40754222811);

insert into Angajat(id\_angajat, id\_job, id\_comic\_store, nume, prenume, telefon)

values (angajati.nextval,3,6, 'Tarnaveanu', 'Sorin', 40724141794);

insert into Angajat(id\_angajat, id\_job, id\_comic\_store, nume, prenume, telefon)

values (angajati.nextval,4,6, 'Scovearca', 'Ramus', 40214563151);

insert into Angajat(id\_angajat, id\_job, id\_comic\_store, nume, prenume, telefon)

values (angajati.nextval,4,6, 'Sive', 'Relu', 40214999951);

insert into Angajat(id\_angajat, id\_job, id\_comic\_store, nume, prenume, telefon)

values (angajati.nextval,2,7, 'Svde', 'Rreu', 40223561711);

insert into Angajat(id\_angajat, id\_job, id\_comic\_store, nume, prenume, telefon)

values (angajati.nextval,1,1, 'SIle', 'Kamat', 40224441219);

insert into Angajat(id\_angajat, id\_job, id\_comic\_store, nume, prenume, telefon)

values (angajati.nextval,1,8, 'Ye', 'Kanye', 40234143297);

insert into Top(id\_top, denumire\_top)

values (tops.nextval,'Best Romance');

insert into Top(id\_top, denumire\_top)

values (tops.nextval,'Best Overall');

insert into Top(id\_top, denumire\_top)

values (tops.nextval,'Best Action');

insert into Top(id\_top, denumire\_top)

values (tops.nextval,'Most Recommended');

insert into Top(id\_top, denumire\_top)

values (tops.nextval,'Funniest');

insert into Review(id\_review, descriere\_review)

values (reviews.nextval, 'Foarte Faina');

insert into Review(id\_review, descriere\_review)

values (reviews.nextval, 'Buna');

insert into Review(id\_review, descriere\_review)

values (reviews.nextval, 'Mediocra');

insert into Review(id\_review, descriere\_review)

values (reviews.nextval, 'Slaba');

insert into Review(id\_review, descriere\_review)

values (reviews.nextval, 'Groaznica');

insert into Review(id\_review, descriere\_review)

values (reviews.nextval, 'Printre cele mai bune');

insert into Review(id\_review, descriere\_review)

values (reviews.nextval, 'Printre cele mai rele');

insert into Review(id\_review, descriere\_review)

values (reviews.nextval, 'Foarte Unica');

insert into Review(id\_review, descriere\_review)

values (reviews.nextval, 'Design Slab');

insert into Review(id\_review, descriere\_review)

values (reviews.nextval, 'Animatie buna');

insert into Cumparator(id\_cumparator, nume, prenume, varsta)

values (cumparatori.nextval, 'Ghinea', 'Silviu', 23);

insert into Cumparator(id\_cumparator, nume, prenume, varsta)

values (cumparatori.nextval, 'Florica', 'Nwo', 24);

insert into Cumparator(id\_cumparator, nume, prenume, varsta)

values (cumparatori.nextval, 'Tataee', 'Bug', 43);

insert into Cumparator(id\_cumparator, nume, prenume, varsta)

values (cumparatori.nextval, 'Maresalu', 'Antonio', 29);

insert into Cumparator(id\_cumparator, nume, prenume, varsta)

values (cumparatori.nextval, 'Pican', 'Antonio', 31);

insert into Cumparator(id\_cumparator, nume, prenume, varsta)

values (cumparatori.nextval, 'Goku', 'Saiyan', 42);

insert into ComicBook(id\_comic\_book, id\_comic\_store, id\_top, id\_cumparator, nume, pret)

values (comics.nextval, 1, 1, 2, 'Genshin Impact', 20);

insert into ComicBook(id\_comic\_book, id\_comic\_store, id\_top, id\_cumparator, nume, pret)

values (comics.nextval, 1, 3, 2, 'Big Adventure', 25);

insert into ComicBook(id\_comic\_book, id\_comic\_store, id\_top, id\_cumparator, nume, pret)

values (comics.nextval, 2, null, 3, 'Re:Zero', 18);

insert into ComicBook(id\_comic\_book, id\_comic\_store, id\_top, id\_cumparator, nume, pret)

values (comics.nextval, 2, null, 1, 'Horimiya', 22);

insert into ComicBook(id\_comic\_book, id\_comic\_store, id\_top, id\_cumparator, nume, pret)

values (comics.nextval, 3, 2, 3, 'Ian si Azteca', 29);

insert into ComicBook(id\_comic\_book, id\_comic\_store, id\_top, id\_cumparator, nume, pret)

values (comics.nextval, 5, null, 4, 'Dragon Ball Z', 26);

insert into ComicBook(id\_comic\_book, id\_comic\_store, id\_top, id\_cumparator, nume, pret)

values (comics.nextval, 2, 4, 5, 'Last Airbender', 21);

insert into ComicBook(id\_comic\_book, id\_comic\_store, id\_top, id\_cumparator, nume, pret)

values (comics.nextval, 5, 5, 6, 'Worst Date', 31);

insert into ComicBook(id\_comic\_book, id\_comic\_store, id\_top, id\_cumparator, nume, pret)

values (comics.nextval, 5, null, 4, 'Best Date', 31);

insert into ComicBook(id\_comic\_book, id\_comic\_store, id\_top, id\_cumparator, nume, pret)

values (comics.nextval, 4, null, 3, 'Worst Date', 37);

insert into ComicBook(id\_comic\_book, id\_comic\_store, id\_top, id\_cumparator, nume, pret)

values (comics.nextval, 7, null, 3, 'New Stuff', 33);

insert into ComicBook(id\_comic\_book, id\_comic\_store, id\_top, id\_cumparator, nume, pret)

values (comics.nextval, 8, null, null, 'AA problems', 42);

insert into Vinde(id\_angajat, id\_comic\_book)

values(1,3);

insert into Vinde(id\_angajat, id\_comic\_book)

values(2,4);

insert into Vinde(id\_angajat, id\_comic\_book)

values(1,5);

insert into Vinde(id\_angajat, id\_comic\_book)

values(1,1);

insert into Vinde(id\_angajat, id\_comic\_book)

values(2,6);

insert into Vinde(id\_angajat, id\_comic\_book)

values(3,3);

insert into Vinde(id\_angajat, id\_comic\_book)

values(3,6);

insert into Vinde(id\_angajat, id\_comic\_book)

values(4,1);

insert into Vinde(id\_angajat, id\_comic\_book)

values(5,8);

insert into Vinde(id\_angajat, id\_comic\_book)

values(22,5);

insert into Primeste(id\_review,id\_comic\_book)

values(1,1);

insert into Primeste(id\_review,id\_comic\_book)

values(2,2);

insert into Primeste(id\_review,id\_comic\_book)

values(3,4);

insert into Primeste(id\_review,id\_comic\_book)

values(4,6);

insert into Primeste(id\_review,id\_comic\_book)

values(5,8);

insert into Primeste(id\_review,id\_comic\_book)

values(6,9);

insert into Primeste(id\_review,id\_comic\_book)

values(7,10);

insert into Primeste(id\_review,id\_comic\_book)

values(8,3);

insert into Primeste(id\_review,id\_comic\_book)

values(9,5);

insert into Primeste(id\_review,id\_comic\_book)

values(10,7);

insert into Ofera(id\_review, id\_cumparator)

values(2,5);

insert into Ofera(id\_review, id\_cumparator)

values(1,4);

insert into Ofera(id\_review, id\_cumparator)

values(3,1);

insert into Ofera(id\_review, id\_cumparator)

values(4,2);

insert into Ofera(id\_review, id\_cumparator)

values(10,3);

insert into Ofera(id\_review, id\_cumparator)

values(9,6);

insert into Ofera(id\_review, id\_cumparator)

values(5,3);

insert into Ofera(id\_review, id\_cumparator)

values(6,3);

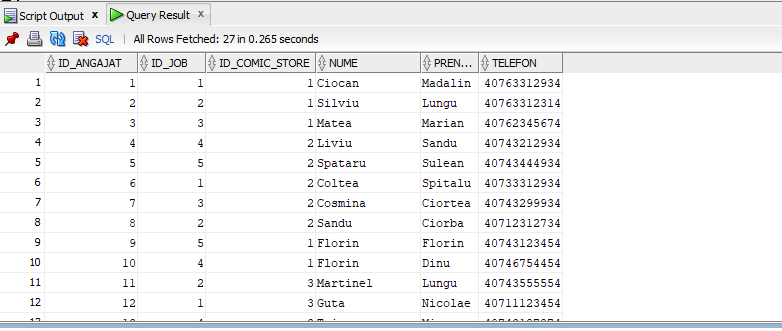
insert into Ofera(id\_review, id\_cumparator)

values(7,3);

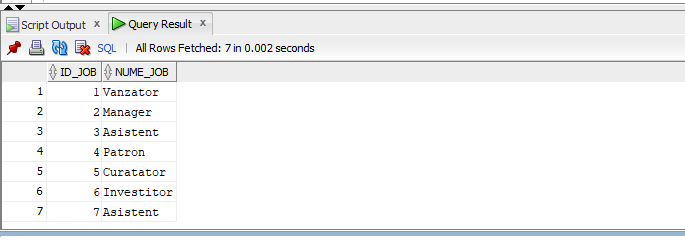
insert into Ofera(id\_review, id\_cumparator)

values(9,4);

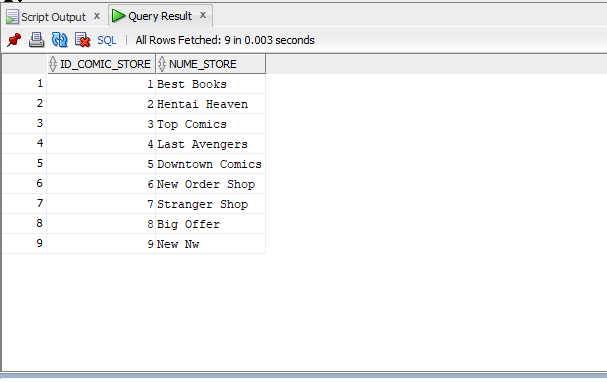
Tabelul Angajat:



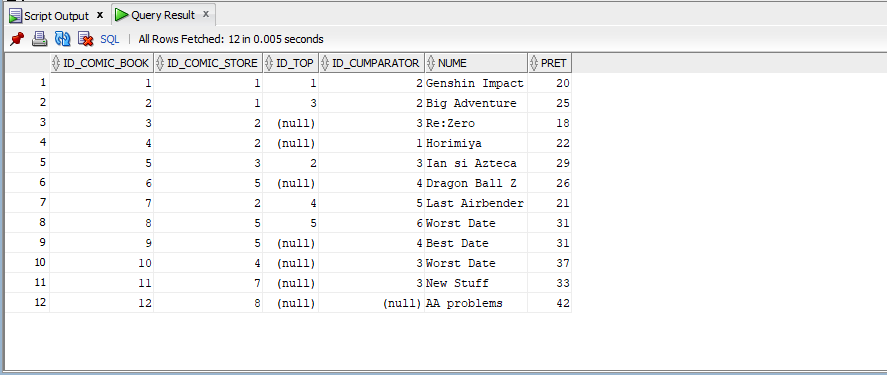
Tabelul Joburi:



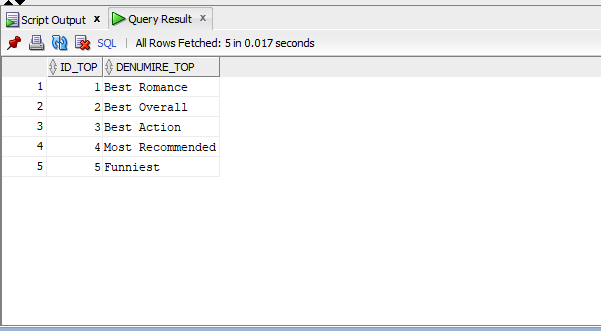
Tabelul ComicStore:



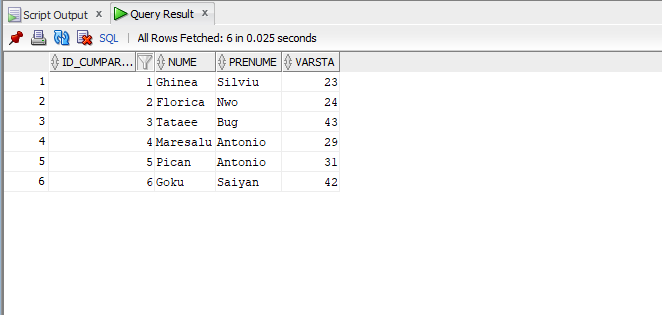
Tabelul ComicBook:



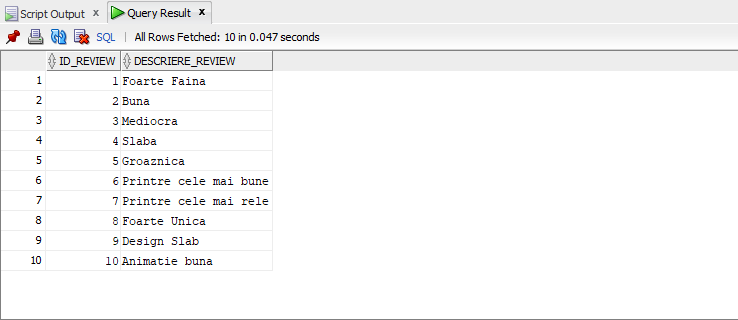
Tabelul Top:



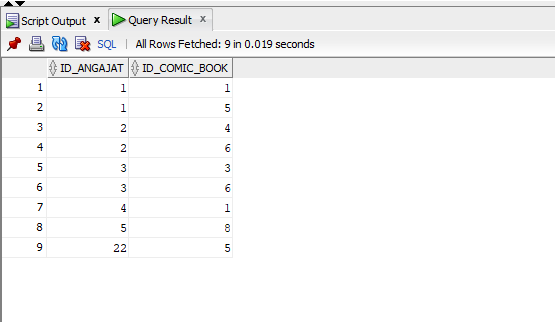
Tabelul Cumparator:



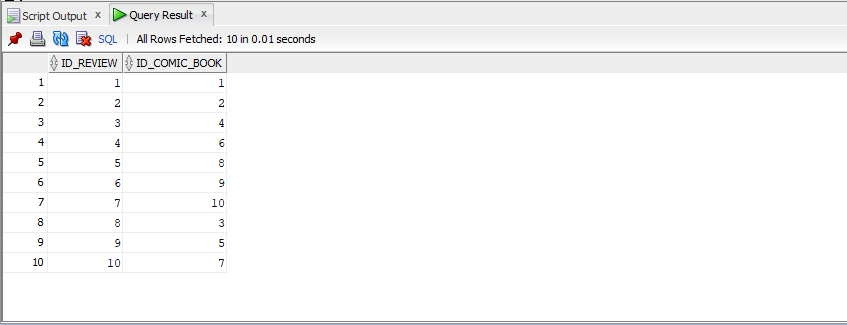
Tabelul Review:



Tabelul Vinde:



Tabelul Primeste:



Tabelul Ofera:



Exercitiul 6:

--6 Aflati toti angajatii din magazine si codul acestor magazine

create or replace procedure ang\_si\_magazine

as

type vector is varray(50) of ComicStore.id\_comic\_store%type;

type vector\_2 is varray(50) of Angajat.nume%type;

type table\_1 is table of vector\_2 index by pls\_integer;

cstore vector := vector();

cangajat table\_1;

chk number(1) := 0;

begin

select id\_comic\_store

bulk collect into cstore

from ComicStore;

if cstore.count < 3 then

RAISE\_APPLICATION\_ERROR(-20099, 'Prea putine magazine!');

end if;

for i in 1..cstore.count loop

select nume

bulk collect into cangajat(i)

from Angajat

join ComicStore on Angajat.id\_comic\_store = ComicStore.id\_comic\_store

where ComicStore.id\_comic\_store = cstore(i);

dbms\_output.put\_line('Id-ul magazinului:' || cstore(i));

for j in 1..cangajat(i).count loop

dbms\_output.put\_line(cangajat(i)(j));

end loop;

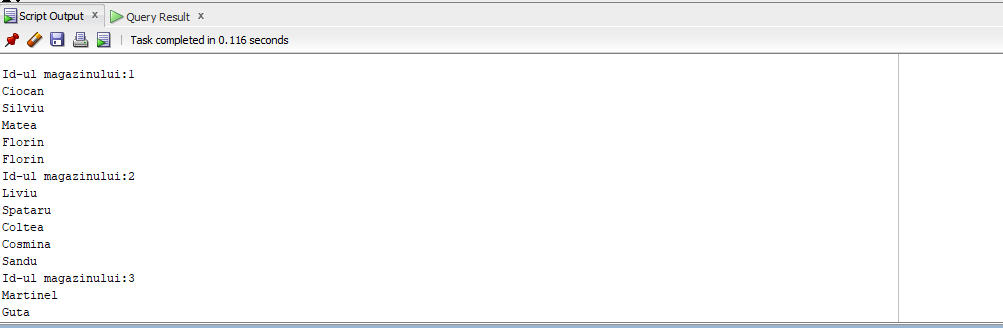
end loop;

end ang\_si\_magazine;

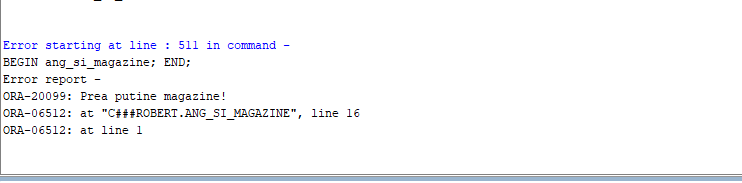
/

--Test:

execute ang\_si\_magazine;



Eroare atunci cand nu avem suficiente magazine:



Exercitiul 7:

--7.Afisati numele tuturor revistelor avute cat si revistele ce contin in nume un anumit cuvant --dat si care au pretul mai mare decat un pret dat.

create or replace procedure comics\_ex(p\_cod comicbook.pret%type, p\_cuvant in varchar2)

as

cursor cursor1 is

select \* from ComicBook;

cursor cursor2 (v\_pret comicbook.pret%type, v\_cuvant\_cheie varchar2) is

select \* from comicbook

where pret > v\_pret

and nume like '%' || v\_cuvant\_cheie || '%';

comic comicbook%rowtype;

begin

dbms\_output.put\_line('Toate revistele sunt: ');

open cursor1;

loop

fetch cursor1 into comic;

exit when cursor1%notfound;

dbms\_output.put\_line('Numele: ' || comic.nume);

end loop;

close cursor1;

dbms\_output.put\_line('Revistele cautate dupa nume si pret sunt: ');

open cursor2(p\_cod,p\_cuvant);

loop

fetch cursor2 into comic;

exit when cursor2%notfound;

dbms\_output.put\_line('Pretul este: ' || comic.pret || ' si numele este ' || comic.nume);

end loop;

close cursor2;

end comics\_ex;

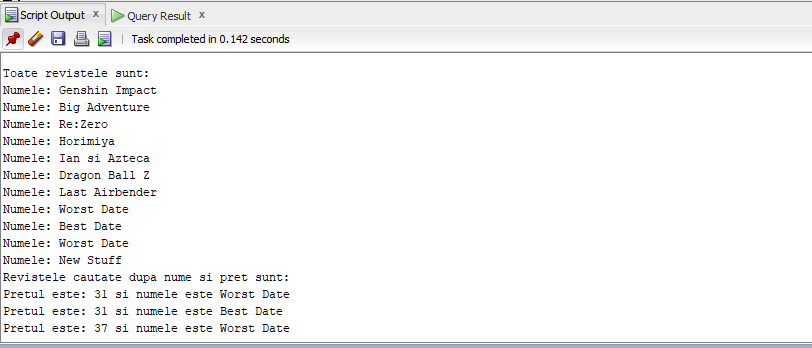
/

--Test:

begin

comics\_ex(29,'Date');

end;



Exercitiul 8:

--8 Pentru un id dat al unei reviste afisati pretul acestei reviste, magazinul in care se afla cat ---si angajatul asociat acesteia

create or replace function reviste(p\_id\_comic comicbook.id\_comic\_book%type, lungime number) return number is

v\_pret comicbook.pret%type;

v\_comic\_name comicbook.nume%type;

v\_angajat\_nume angajat.nume%type;

counter number;

pret\_prea\_mare exception;

nume\_scurt exception;

begin

select cb.pret, cs.nume\_store, a.nume

into v\_pret, v\_comic\_name, v\_angajat\_nume

from comicbook cb

join comicstore cs on cb.id\_comic\_store=cs.id\_comic\_store

join angajat a on cs.id\_comic\_store=a.id\_comic\_store

where cb.id\_comic\_book=p\_id\_comic;

if length(v\_angajat\_nume) < lungime then

raise nume\_scurt;

end if;

if v\_pret > 40 then

raise pret\_prea\_mare;

end if;

dbms\_output.put\_line('Numele magazinului este: ' || v\_comic\_name);

dbms\_output.put\_line('Numele angajatului asociat este: ' || v\_angajat\_nume);

dbms\_output.put\_line('Pretul pentru comic bookul cu id ul: ' || p\_id\_comic || ' este: ');

return v\_pret;

exception

when NO\_DATA\_FOUND then

RAISE\_APPLICATION\_ERROR(-20008, 'Nu exista comic-ul cu id ul dat');

when pret\_prea\_mare then

RAISE\_APPLICATION\_ERROR(-20007, 'Pretul dat este prea mare');

when nume\_scurt then

RAISE\_APPLICATION\_ERROR(-20009, 'Numele este prea scurt!');

when TOO\_MANY\_ROWS then

RAISE\_APPLICATION\_ERROR(-20006, 'Mai multe reviste!');

end reviste;

/

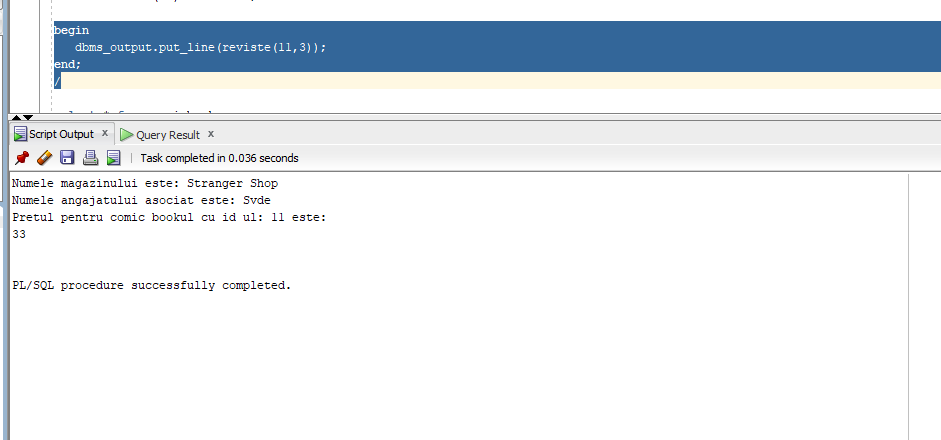
--Test:

begin

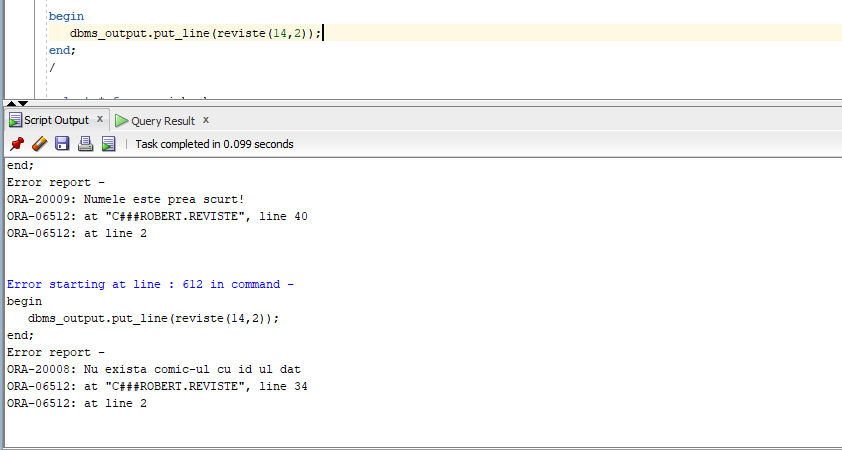
dbms\_output.put\_line(reviste(9,2));

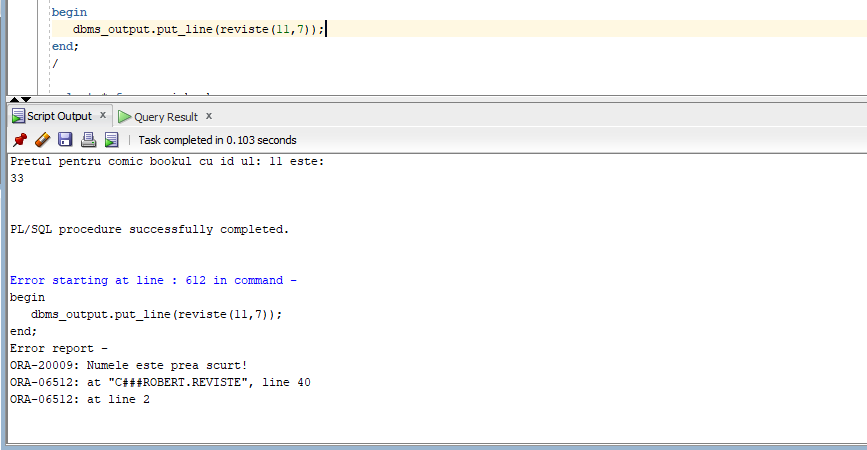
end;

/









Exercitiul 9:

--9 Aflati jobul avut de un angajat care lucreaza la un comic store care are o anumita revista ---ce se afla intr-un top cu titlul ce contine "Romance"

create or replace procedure job\_avut(id\_angajat\_dat angajat.id\_angajat%type)

is

denumire\_job joburi.nume\_job%type;

begin

select jb.nume\_job

into denumire\_job

from joburi jb

join angajat a on a.id\_job = jb.id\_job

join comicstore cs on a.id\_comic\_store = cs.id\_comic\_store

join comicbook cb on cs.id\_comic\_store = cb.id\_comic\_store

join top tp on cb.id\_top = tp.id\_top

where tp.denumire\_top like '%Romance%' and a.id\_angajat = id\_angajat\_dat;

dbms\_output.put\_line('Jobul avut de angajatul cu id dat este: ' || denumire\_job);

exception

when NO\_DATA\_FOUND then

RAISE\_APPLICATION\_ERROR(-20010, 'Nu s-au gasit joburi pe care lucreaza angajatul cu id-ul dat!');

end job\_avut;

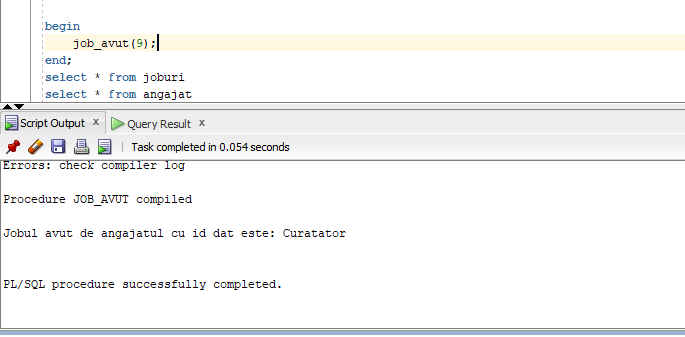
/

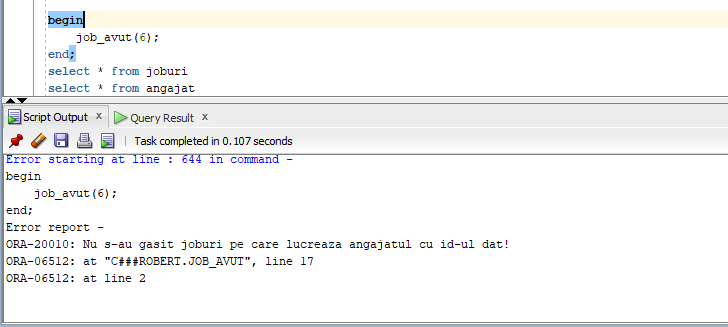
--Test:

begin

job\_avut(9);

end;





--10 Creati un trigger care nu ne permite modificarea tabelului angajat in ziua de luni si --duminica sau daca nu ne aflam in intervalul orar 8 -17

create or replace trigger actualizare\_tabel

before insert or update or delete on angajat

begin

if (to\_char(sysdate,'D') = 1) or (to\_char(sysdate,'D') = 2) or (to\_char(sysdate,'HH24') not between 8 and 17)

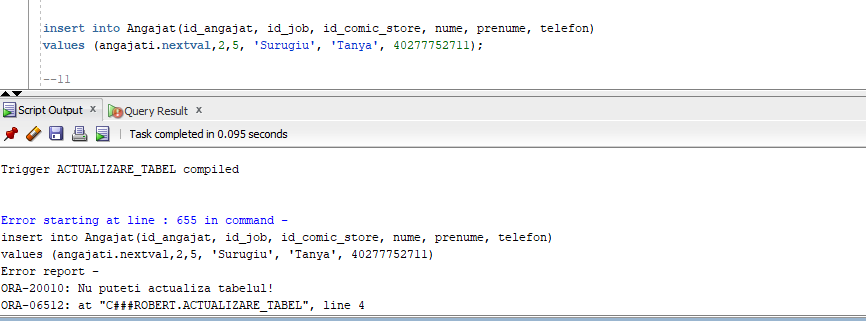
then

RAISE\_APPLICATION\_ERROR(-20010, 'Nu puteti actualiza tabelul!');

end if;

end;

/



--11 Creati un trigger care nu ne lasa sa modificam pretul revistelor daca diferenta dintre -------pretul vechi si cel nou este mai mica de 15

create or replace trigger actualizare\_pret

before update of pret on comicbook

for each row

declare

pret\_obligatoriu exception;

begin

if :old.pret - :new.pret < 15 then

raise pret\_obligatoriu;

end if;

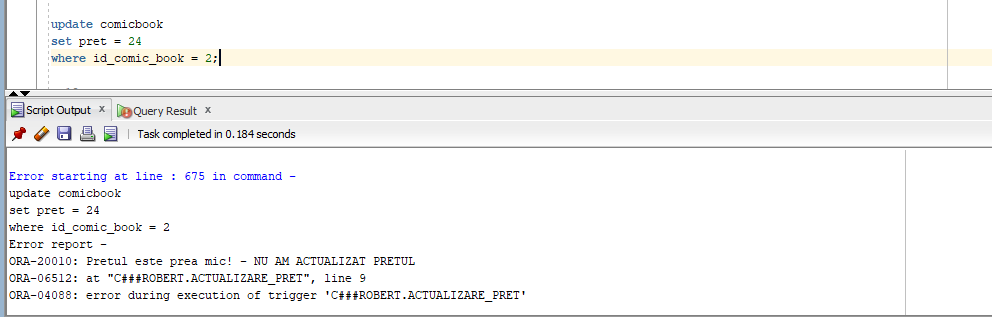
exception

when pret\_obligatoriu then

RAISE\_APPLICATION\_ERROR(-20010, 'Pretul este prea mic! - NU AM ACTUALIZAT PRETUL');

end;

/



--12 Creati un trigger care nu ne permite crearea unor tabele noi in baza de date vinerea si -----sambata sau daca numele acestuia este fie prea mic < 2, fie prea mare > 10

create or replace trigger actualizare\_database

before create on schema

declare

nume\_gresit exception;

data\_gresita exception;

begin

if (to\_char(sysdate,'D') between 6 and 7)

then

raise data\_gresita;

elsif length(ora\_dict\_obj\_name) < 2 or length(ora\_dict\_obj\_name) > 10

then

raise nume\_gresit;

end if;

exception

when nume\_gresit then

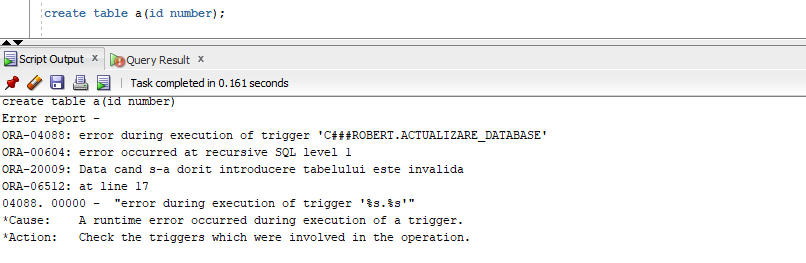
RAISE\_APPLICATION\_ERROR(-20010, 'Numele introdus este prea scurt sau prea lung - incorect!');

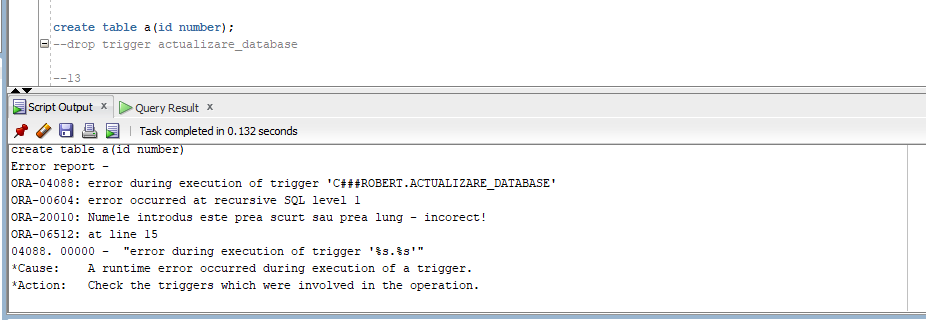
when data\_gresita then

RAISE\_APPLICATION\_ERROR(-20009, 'Data cand s-a dorit introducere tabelului este invalida');

end;

/





13. Pachetul cu obiectele:

create or replace package pachet\_13 as

procedure ang\_si\_magazine;

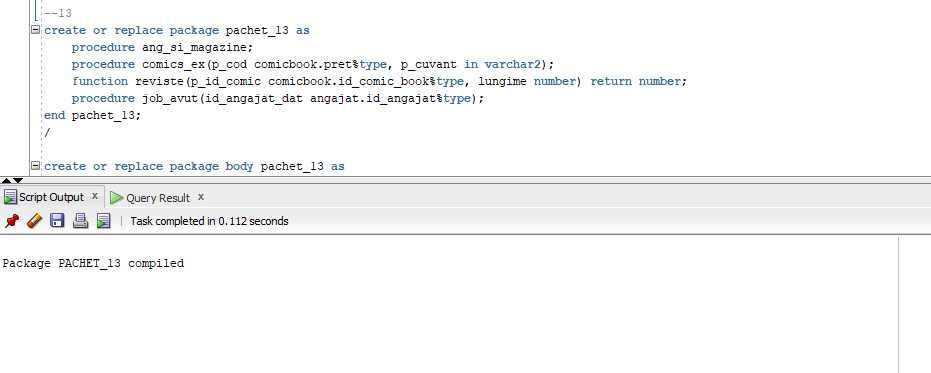
procedure comics\_ex(p\_cod comicbook.pret%type, p\_cuvant in varchar2);

function reviste(p\_id\_comic comicbook.id\_comic\_book%type, lungime number) return number;

procedure job\_avut(id\_angajat\_dat angajat.id\_angajat%type);

end pachet\_13;

/



create or replace package body pachet\_13 as

procedure ang\_si\_magazine

as

type vector is varray(50) of ComicStore.id\_comic\_store%type;

type vector\_2 is varray(50) of Angajat.nume%type;

type table\_1 is table of vector\_2 index by pls\_integer;

cstore vector := vector();

cangajat table\_1;

chk number(1) := 0;

begin

select id\_comic\_store

bulk collect into cstore

from ComicStore;

if cstore.count < 3 then

RAISE\_APPLICATION\_ERROR(-20099, 'Prea putine magazine!');

end if;

for i in 1..cstore.count loop

select nume

bulk collect into cangajat(i)

from Angajat

join ComicStore on Angajat.id\_comic\_store = ComicStore.id\_comic\_store

where ComicStore.id\_comic\_store = cstore(i);

dbms\_output.put\_line('Id-ul magazinului:' || cstore(i));

for j in 1..cangajat(i).count loop

dbms\_output.put\_line(cangajat(i)(j));

end loop;

end loop;

end ang\_si\_magazine;

procedure comics\_ex(p\_cod comicbook.pret%type, p\_cuvant in varchar2)

as

cursor cursor1 is

select \* from ComicBook;

cursor cursor2 (v\_pret comicbook.pret%type, v\_cuvant\_cheie varchar2) is

select \* from comicbook

where pret > v\_pret

and nume like '%' || v\_cuvant\_cheie || '%';

comic comicbook%rowtype;

begin

dbms\_output.put\_line('Toate revistele sunt: ');

open cursor1;

loop

fetch cursor1 into comic;

exit when cursor1%notfound;

dbms\_output.put\_line('Numele: ' || comic.nume);

end loop;

close cursor1;

dbms\_output.put\_line('Revistele cautate dupa nume si pret sunt: ');

open cursor2(p\_cod,p\_cuvant);

loop

fetch cursor2 into comic;

exit when cursor2%notfound;

dbms\_output.put\_line('Pretul este: ' || comic.pret || ' si numele este ' || comic.nume);

end loop;

close cursor2;

end comics\_ex;

function reviste(p\_id\_comic comicbook.id\_comic\_book%type, lungime number) return number is

v\_pret comicbook.pret%type;

v\_comic\_name comicbook.nume%type;

v\_angajat\_nume angajat.nume%type;

counter number;

pret\_prea\_mare exception;

nume\_scurt exception;

begin

select cb.pret, cs.nume\_store, a.nume

into v\_pret, v\_comic\_name, v\_angajat\_nume

from comicbook cb

join comicstore cs on cb.id\_comic\_store=cs.id\_comic\_store

join angajat a on cs.id\_comic\_store=a.id\_comic\_store

where cb.id\_comic\_book=p\_id\_comic;

if length(v\_angajat\_nume) < lungime then

raise nume\_scurt;

end if;

if v\_pret > 40 then

raise pret\_prea\_mare;

end if;

dbms\_output.put\_line('Numele magazinului este: ' || v\_comic\_name);

dbms\_output.put\_line('Numele angajatului asociat este: ' || v\_angajat\_nume);

dbms\_output.put\_line('Pretul pentru comic bookul cu id ul: ' || p\_id\_comic || ' este: ');

return v\_pret;

exception

when NO\_DATA\_FOUND then

RAISE\_APPLICATION\_ERROR(-20008, 'Nu exista comic-ul cu id ul dat');

when pret\_prea\_mare then

RAISE\_APPLICATION\_ERROR(-20007, 'Pretul dat este prea mare');

when nume\_scurt then

RAISE\_APPLICATION\_ERROR(-20009, 'Numele este prea scurt!');

when TOO\_MANY\_ROWS then

RAISE\_APPLICATION\_ERROR(-20006, 'Mai multe reviste!');

end reviste;

procedure job\_avut(id\_angajat\_dat angajat.id\_angajat%type)

is

denumire\_job joburi.nume\_job%type;

begin

select jb.nume\_job

into denumire\_job

from joburi jb

join angajat a on a.id\_job = jb.id\_job

join comicstore cs on a.id\_comic\_store = cs.id\_comic\_store

join comicbook cb on cs.id\_comic\_store = cb.id\_comic\_store

join top tp on cb.id\_top = tp.id\_top

where tp.denumire\_top like '%Romance%' and a.id\_angajat = id\_angajat\_dat;

dbms\_output.put\_line('Jobul avut de angajatul cu id dat este: ' || denumire\_job);

exception

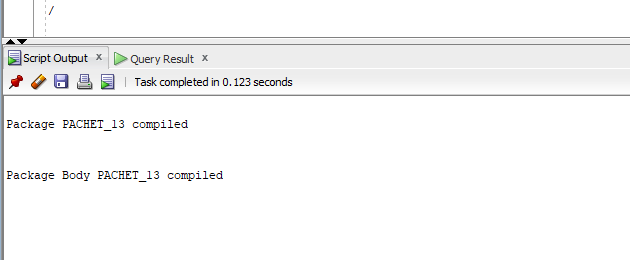
when NO\_DATA\_FOUND then

RAISE\_APPLICATION\_ERROR(-20010, 'Nu s-au gasit joburi pe care lucreaza angajatii cu id-ul dat!');

end job\_avut;

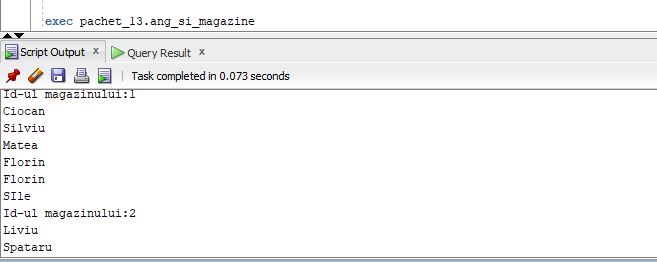
end pachet\_13;

/

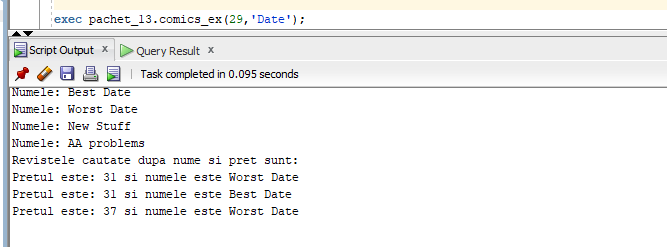


--Test pachet:

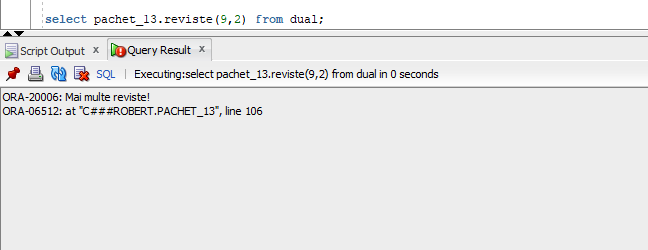
exec pachet\_13.ang\_si\_magazine



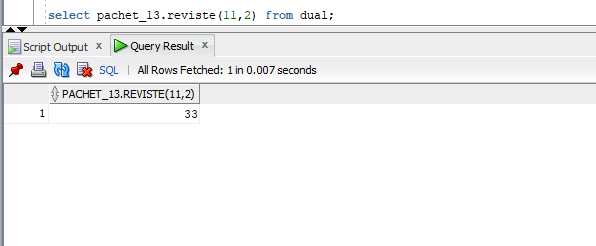
exec pachet\_13.comics\_ex(29,'Date');



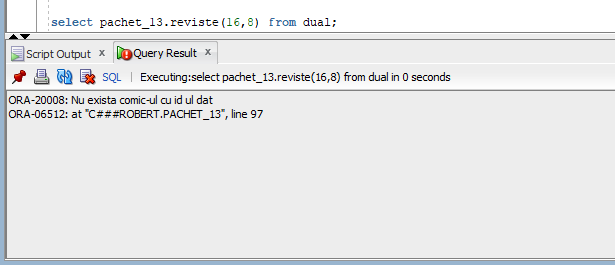
select pachet\_13.reviste(9,2) from dual;



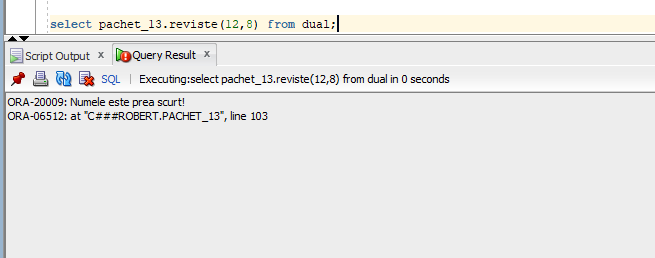
select pachet\_13.reviste(11,2) from dual;



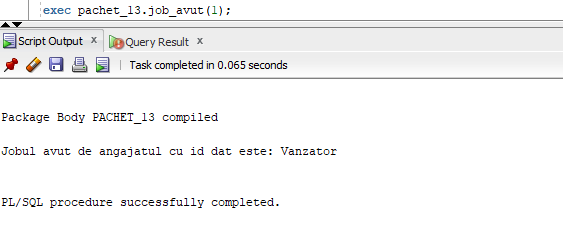
select pachet\_13.reviste(16,8) from dual;



select pachet\_13.reviste(12,8) from dual;



exec pachet\_13.job\_avut(1);



exec pachet\_13.job\_avut(242);

