Stan Sebastian-Daniel

Seria E , anul II, grupa 1056

**Proiect**

**Baze de date**

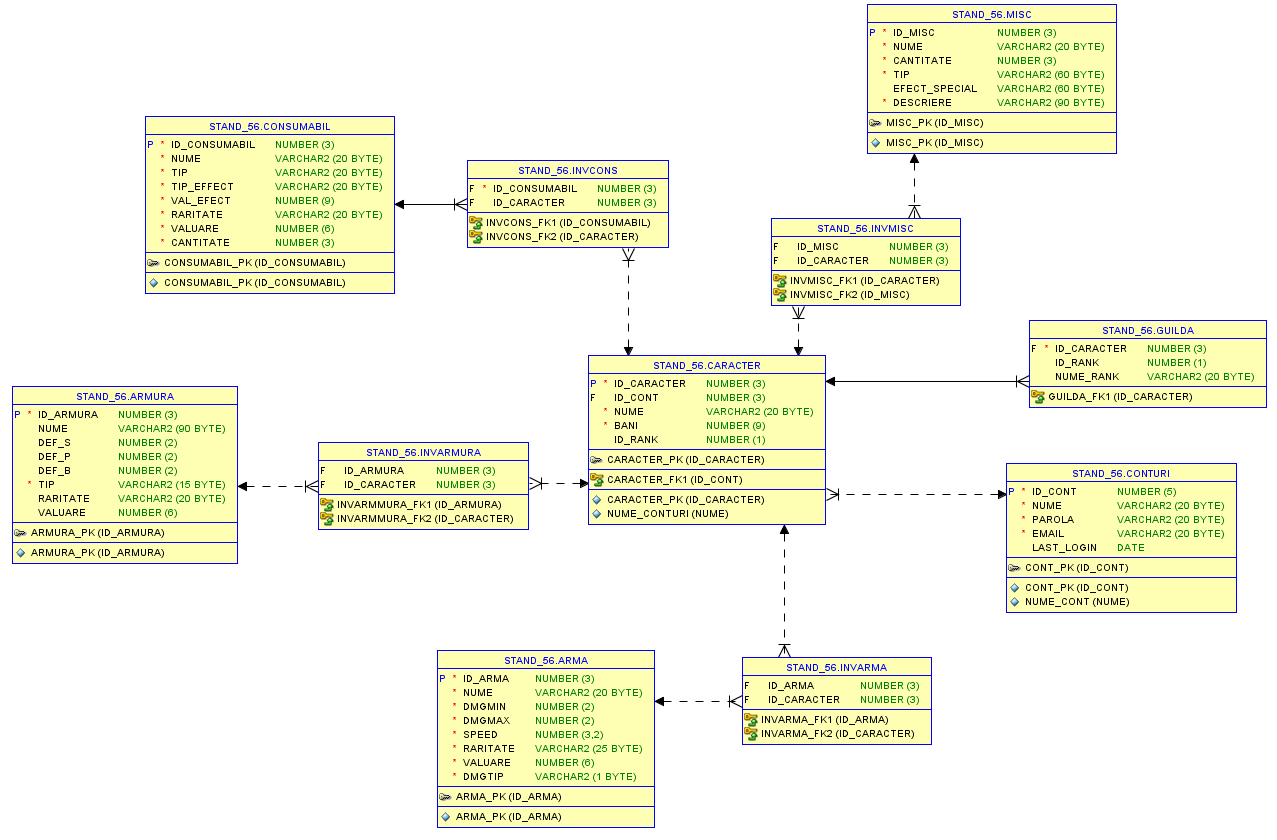
DESCRIEREA BAZEI DE DATE

Baza de date a fost creeata cu scopul stocarii datelor depre un conturile playerilor dintr-un joc video de tip MMORPG. Aceasta are rolul de atine diverse detalii despre cont cat si despre itemele accumulate de diversii avatari ai jucatorului

Aceasta baza de date contiine informatii despre:

* contul jucatorului avand in el diferite informatii ca id-ul contului ,numele contului, emailul jucatorului , parola si banii introdusi in joc
* Caracterele care sunt affiliate contului jucatorului, aceasta tabela contine date ca id\_caracter,id\_cont, numele caracterului si numerarul modei vitoale a jocului
* Itemele detinute de jucator , aici se gasesc diverse tabele care stocheaza date despre diverse tipuri de inteme detinute de caracterul jucatorului ca si ,dauna provocata de arme, tipul armei, tipul armuri, valoarre de protective a armurii ,raritatea diverselor iteme precum si valoarea lor si multe alte date

Aceasta baza de date are scopul de a fi folositia pt stocarea infortaiilor legetate de caracterele unui jucator pentru jocuri de tip RPG, acestea fiind legate de itemele pe care caracterul le detine cu toate detaliile despre acestea, infortatiile fiind folosite ca date de intrare pentru diversele variabile folosite in joc.



* Cursori
* Implicit

1) Marirea costului unei armuri date de la tastatura cu 10%

begin

update armura

set valuare=valuare+valuare\*01

where nume='&x';

IF SQL%NOTFOUND THEN

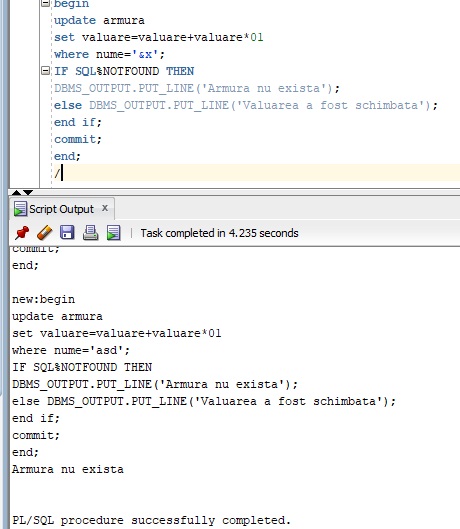
DBMS\_OUTPUT.PUT\_LINE('Armura nu exista');

else DBMS\_OUTPUT.PUT\_LINE('Valuarea a fost schimbata');

end if;

commit;

end;/



2) Adaugare de armura si verificare cu sql%found

begin

insert into armura (id\_armura,nume,def\_s,def\_P,def\_b,tip,raritate,valuare)

values (7,'breastplate',5,5,3,'medium','common',30);

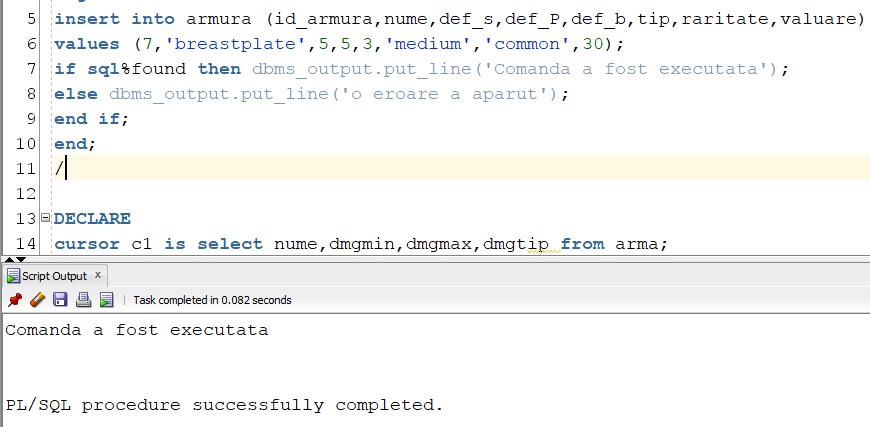
if sql%found then dbms\_output.put\_line('Comanda a fost executata');

else dbms\_output.put\_line('o eroare a aparut');

end if;

end;

/



* Explicit

1)Folosirea cursorului pentru afisarea tabelei arma

DECLARE

cursor c1 is select nume,dmgmin,dmgmax,dmgtip from arma;

a\_nume arma.nume%type

a\_dmin arma.dmgmin%type;

a\_dmax arma.dmgmax%type;

a\_tip arma.dmgtip%type;

begin

OPEN c1;

loop

fetch c1 into a\_nume,a\_dmin,a\_dmax,a\_tip;

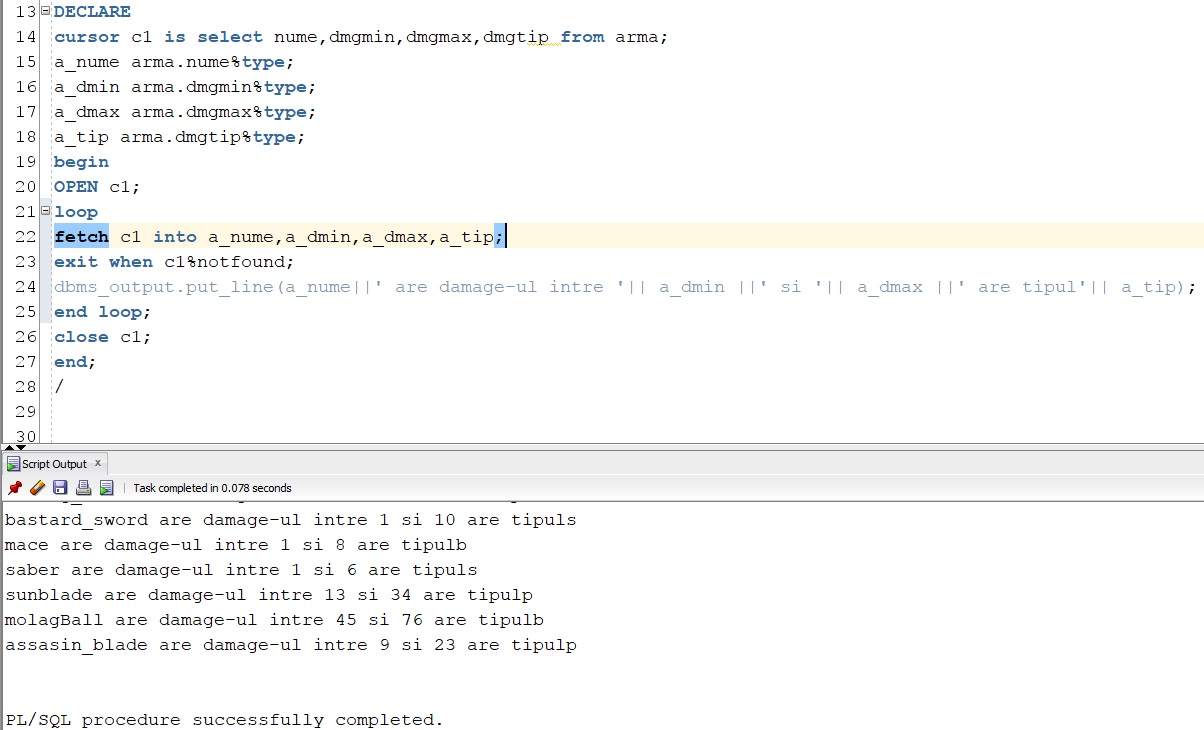
exit when c1%notfound;

dbms\_output.put\_line(a\_nume||' are damage-ul intre '|| a\_dmin ||' si '|| a\_dmax ||' are tipul'|| a\_tip);

end loop;

close c1;

end;



2)Citirea de la tastatura a unei arme pentru a verifica daca exista in tabela de arme folosind cursori:

declare

cursor c1 is select nume from arma;

a\_nume arma.nume%type;

cauta arma.nume%type := '&x';

found number :=0;

begin

open c1;

loop

fetch c1 into a\_nume;

IF cauta=a\_nume then dbms\_output.put\_line('itemul exista');

found:=1;

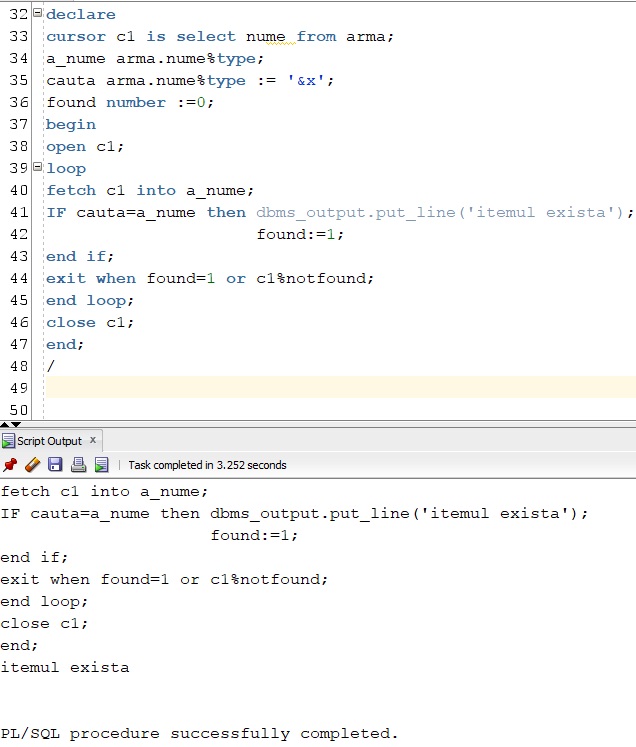
end if;

exit when found=1 or c1%notfound;

end loop;

close c1;

end;

/

* Functii

1) Functia calcul al daunelor pe secunde in functie de id-ul armei date

de la tastatura

set SERVEROUTPUT on;

CREATE OR REPLACE function DPS(id\_selectat VARCHAR2) RETURN NUMBER

AS

--v\_min arma.dmgmin%row;

--v\_max arma.dmgmax%row;

--v\_speed arma.speed%row;

damage NUMBER;

begin

--select dmgmin,dmgmax,speed into v\_min,v\_max,v\_speed from arma where id\_arma=x;

select (dmgmin+dmgmax)/2\*speed into damage from arma where id\_arma=id\_selectat;

--damage:=(v\_min+v\_max)/2\*v\_speed;

RETURN damage;

exception

when no\_data\_found then return -1;

end ;

/

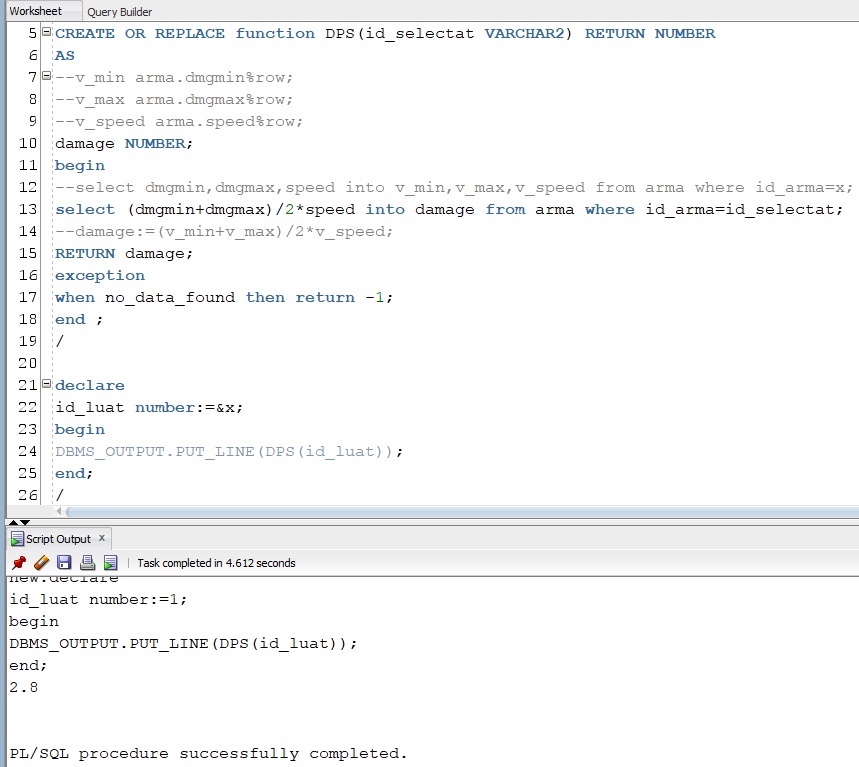
declare

id\_luat number:=&x;

begin

DBMS\_OUTPUT.PUT\_LINE(DPS(id\_luat));

end;

/

2) Functia testeaza daca itemul dat in functie are defensiva pe tipul cerut mai mare decat valoarea data in functie. Daca indeplineste conditiile va retruna 1 daca nu 0 si -1 daca nu exista itemul sau apar erori

CREATE OR REPLACE function DmgTreshold(armura\_id Number,minumum\_dmg number,tip\_dmg VARCHAR2) RETURN NUMBER

as

invalid EXCEPTION;

tst number;

ds number;

dp number;

db number;

begin

select def\_s into ds from armura where id\_armura=armura\_id;

select def\_p into dp from armura where id\_armura=armura\_id;

select def\_b into db from armura where id\_armura=armura\_id;

IF (tip\_dmg='s') then if (ds>=minumum\_dmg) then tst:=1;

end if;

ELSIF (tip\_dmg='p') then if (dp>=minumum\_dmg) then tst:=1;

end if;

ELSIF (tip\_dmg='b') then if (db>=minumum\_dmg) then tst:=1;

end if;

else raise invalid;

end if;

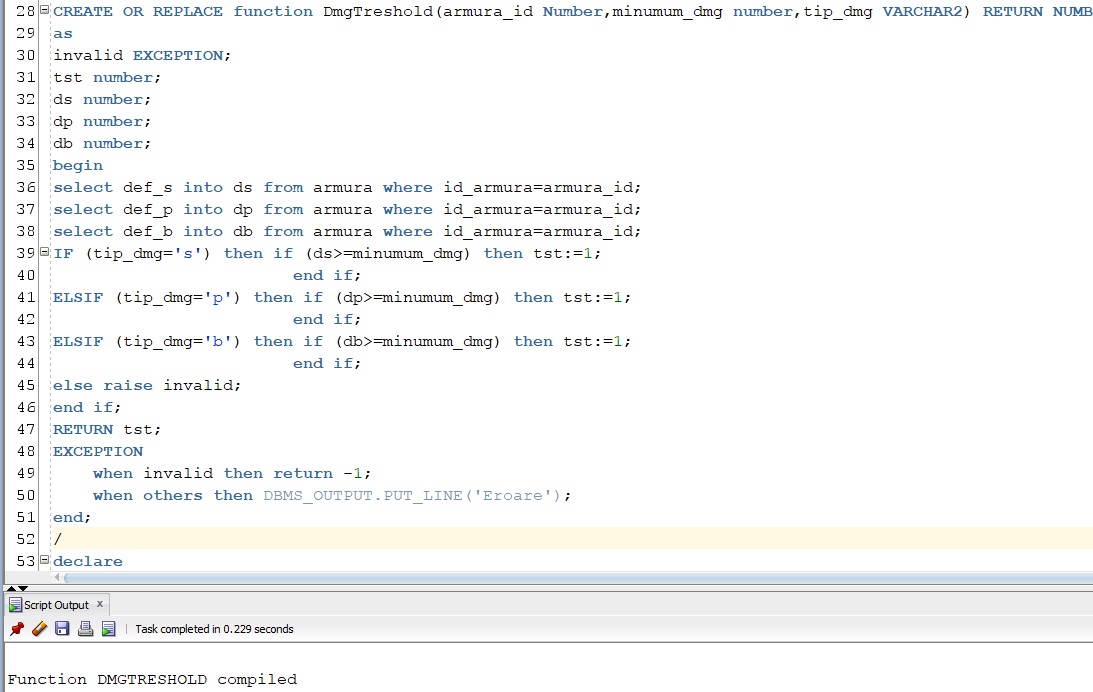
RETURN tst;

EXCEPTION

when invalid then return -1;

when others then DBMS\_OUTPUT.PUT\_LINE('Eroare');

end;

/

declare

v\_a number;

v\_b number;

v\_c number;

begin

v\_a:=DmgTreshold(2,5,'s');

v\_b:=DmgTreshold(1,5,'s');

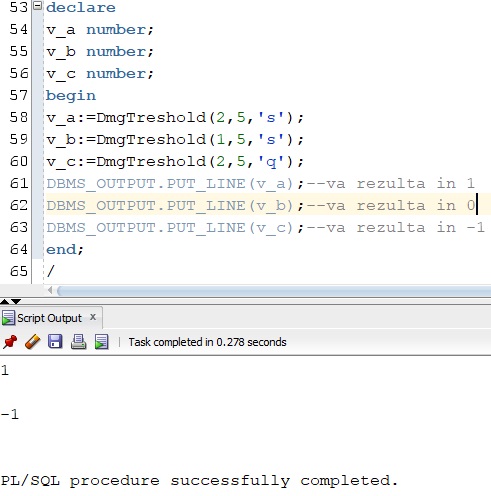
v\_c:=DmgTreshold(2,5,'q');

DBMS\_OUTPUT.PUT\_LINE(v\_a);--va rezulta in 1

DBMS\_OUTPUT.PUT\_LINE(v\_b);--va rezulta in 0

DBMS\_OUTPUT.PUT\_LINE(v\_c);--va rezulta in -1

end;

/

3)Crearea unei functii care genereaza dmg-ul dat de o arma in functie de id-ul sau, dmg-ul va fi generat radom in functie de dmgmin si dmgmax.

create or replace function atack(id\_cerut arma.id\_arma%TYPE) return number as

v\_min arma.dmgmin%type;

v\_max arma.dmgmax%type;

dmg\_dealt number;

negativ EXCEPTION;

begin

select dmgmin,dmgmax into v\_min, v\_max from arma where id\_arma=id\_cerut;

SELECT TRUNC(DBMS\_RANDOM.VALUE(v\_min,v\_max+1)) into dmg\_dealt FROM dual;

if dmg\_dealt>0 then return dmg\_dealt;

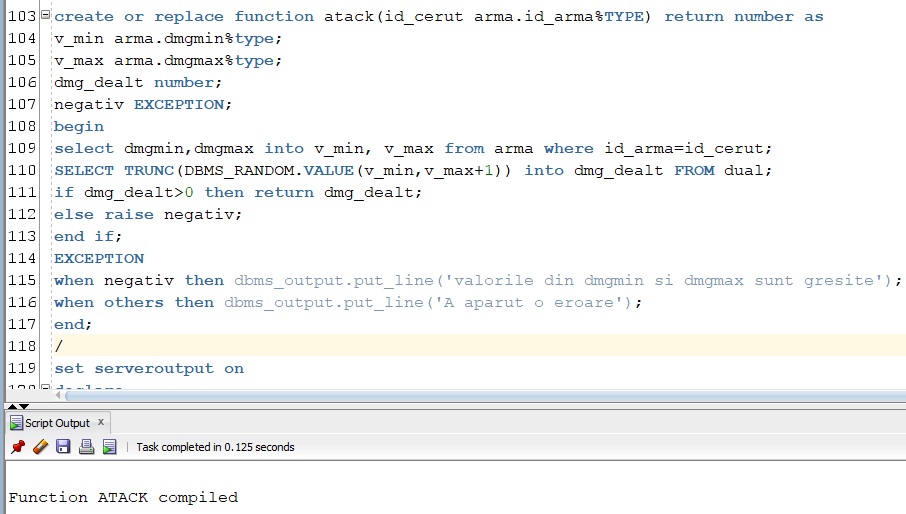
else raise negativ;

end if;

EXCEPTION

when negativ then dbms\_output.put\_line('valorile din dmgmin si dmgmax sunt gresite');

when others then dbms\_output.put\_line('A aparut o eroare');

end;

set serveroutput on

declare

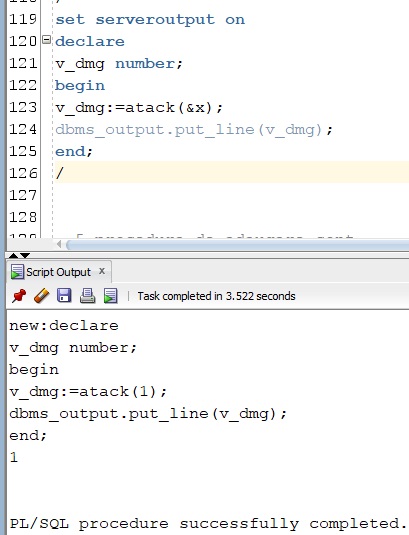
v\_dmg number;

begin

v\_dmg:=atack(&x);

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

end;

/

* Proceduri

1) Procedura de listare a caracterelor din cont

CREATE OR REPLACE PROCEDURE Listare(idCont Number)

as

cursor c is (select \* from caracter where id\_cont=idCont);

v\_c c%rowtype;

begin

open c;

loop

fetch c into v\_c;

exit when c%notfound;

dbms\_output.put\_line('Numele caracterului este: '||v\_c.nume||' si are '||v\_c.bani||' bani.');

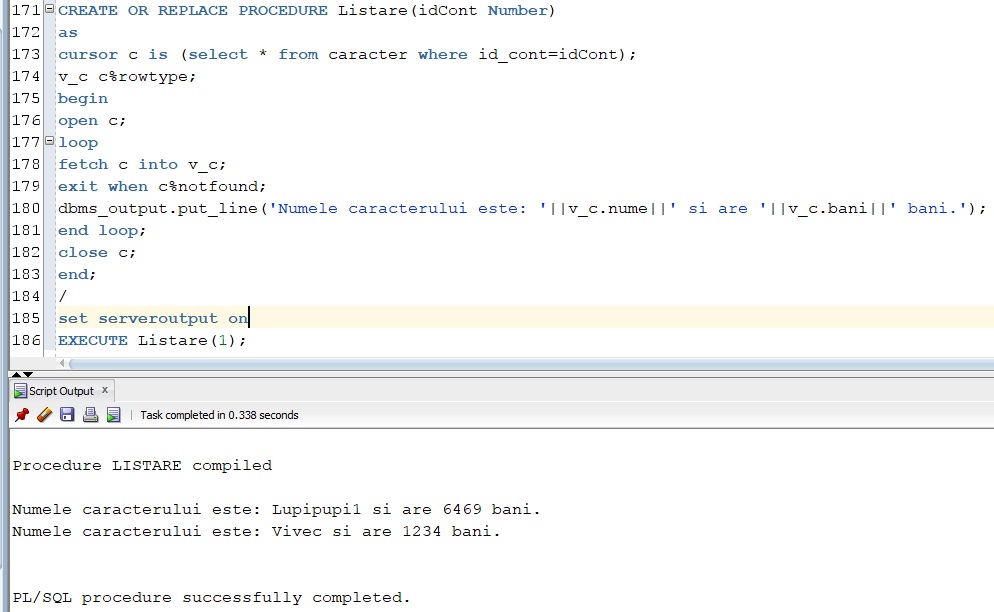
end loop;

close c;

end;

/

EXECUTE Listare(1);



2)Procedura de adaugare user

CREATE OR REPLACE PROCEDURE Adaugare(idCont number,Name VARCHAR2,pass VARCHAR2,email VARCHAR2)

as

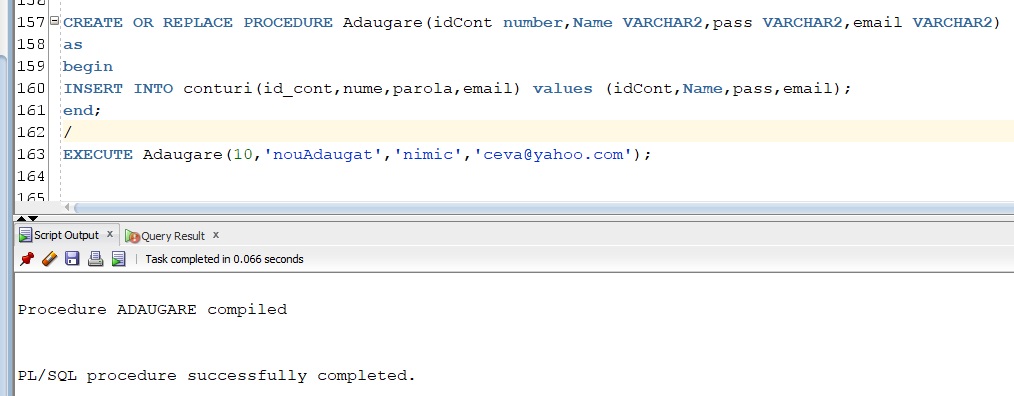
begin

INSERT INTO conturi(id\_cont,nume,parola,email) values (idCont,Name,pass,email);

end;

/

EXECUTE Adaugare(10,'nouAdaugat','nimic','ceva@yahoo.com');



* Pachet

1)Creeare de pachet pentru managerierea conurilor ,in acesta se gasc atiuni ca gasirea unui cont, stergerea, adaugarea unui cont

CREATE OR REPLACE PACKAGE manage\_conturi

as

function exista\_cont (v\_id number) RETURN BOOLEAN;

function afla\_parola (v\_nume varchar2) RETURN VARCHAR2;

procedure add\_cont (v\_id number,v\_nume varchar2, v\_parola varchar2, v\_email varchar2);

procedure sterge\_cont (v\_id number);

end;

/

CREATE OR REPLACE PACKAGE BODY manage\_conturi as

function exista\_cont (v\_id number) RETURN BOOLEAN is

exista boolean;

id\_exista number;

begin

SELECT id\_cont into id\_exista FROM conturi where id\_cont=v\_id;

if(id\_exista is NOT NULL)

then exista:=true;

else exista:=false;

end if;

RETURN exista;

end exista\_cont;

function afla\_parola (v\_nume varchar2) RETURN VARCHAR2 IS

par conturi.parola%type;

begin

select parola into par from conturi where nume like v\_nume;

RETURN par;

end afla\_parola;

procedure add\_cont (v\_id number,v\_nume varchar2, v\_parola varchar2, v\_email varchar2) as

id\_exista NUMBER;

exceptie EXCEPTION;

BEGIN

SELECT id\_cont into id\_exista FROM conturi where id\_cont=v\_id;

if(id\_exista is NOT NULL)

then raise exceptie;

else

INSERT

INTO FUNCTII values(v\_id, v\_nume, v\_parola, v\_email);

end if;

EXCEPTION

when exceptie then dbms\_output.put\_line('Contul exista deja ');

END add\_cont;

procedure sterge\_cont (v\_id Number) as

id\_exista NUMBER;

exceptie EXCEPTION;

BEGIN

delete from conturi where id\_cont=v\_id;

SELECT id\_cont into id\_exista FROM conturi where id\_cont=v\_id;

if(id\_exista is NOT NULL)

then raise exceptie;

end if;

EXCEPTION

when exceptie then dbms\_output.put\_line('Contul nu a fost sters');

END sterge\_cont;

end manage\_conturi;

declare

exista BOOLEAN;

id number:=&x;

n\_id NUMBER:=&xd;

n\_nume varchar(30):=&num;

n\_parola varchar(30):=&par;

n\_email varchar(30):=&mail;

--f\_parola conturi.parola;

begin

exista:=manage\_conturi.exista\_cont(id);

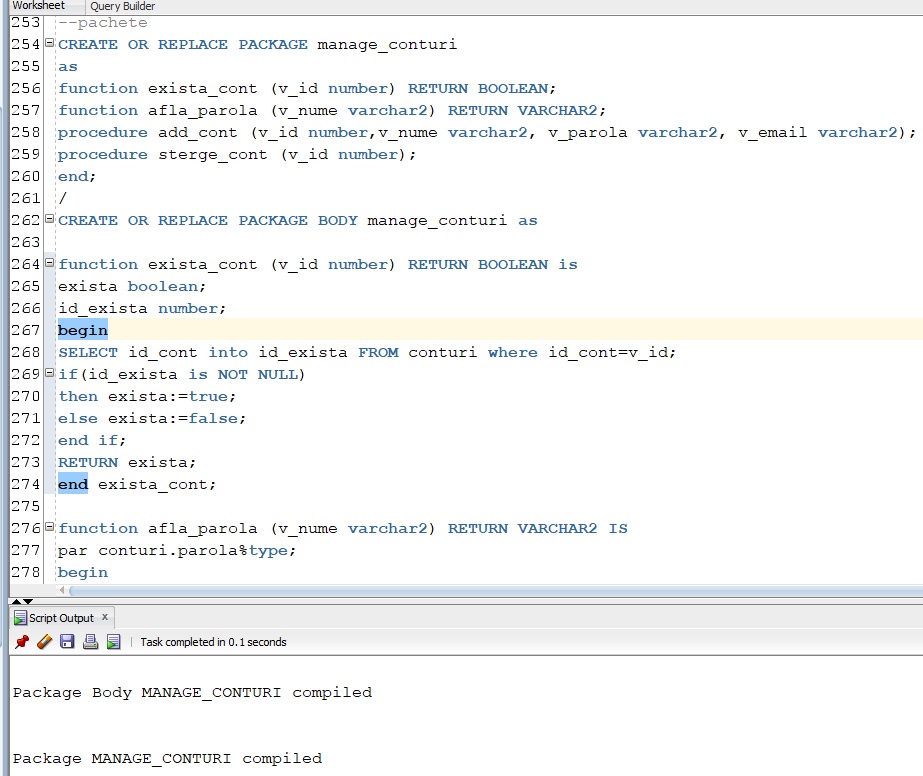
if (exista=true) then dbms\_output.put\_line(manage\_conturi.afla\_parola(id));

end if;

manage\_conturi.add\_cont(n\_id,n\_parola,n\_email);

manage\_conturi.sterge\_cont(n\_id);

end;



* Triggeri

1)Creearea unui trigger ce impiedica schibarea datei de last\_login cu una mai veche decat cea existenta

create or replace trigger schimbare\_last\_login

before update of last\_login on conturi

for each row

when (new.last\_login<old.last\_login)

begin

raise\_application\_error(-2000,'Nu se poate schimba data logarii cu una mai veche');

end;

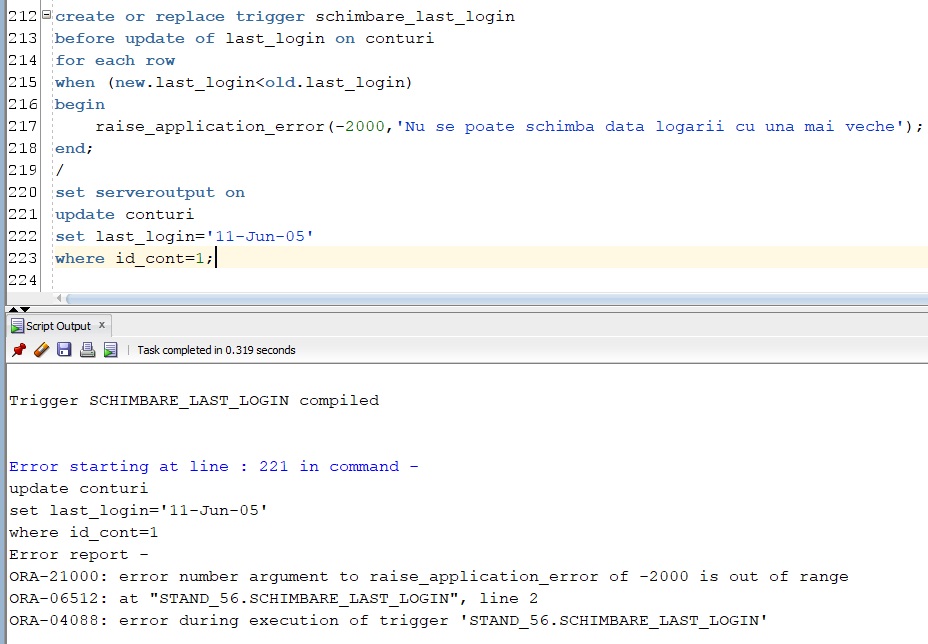
/

set serveroutput on

update conturi

set last\_login='11-Jun-05'

where id\_cont=1;



2)Creearea unui trigger care updateaza last\_loginul la data curenta atunci cand se adauga un nou cont sau unul este modificat

create or replace trigger actualizeza\_data\_login

AFTER INSERT or update ON conturi

for each row

begin

update conturi

set last\_login=sysdate

where :old.id\_cont=:new.id\_cont;

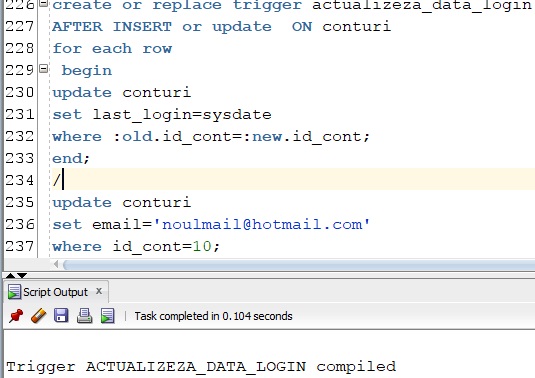
end;

/

update conturi

set email='noulmail@hotmail.com'

where id\_cont=10;



3)Sa se creeze un trigger care opreste stergerea armurilor legendare

create or replace view armura\_v as select \* from armura;

create or replace trigger descompunere\_item

instead of delete on armura\_v

for each row

begin

if :old.raritate='legendary' then raise\_application\_error(-2000,'Itemele legendare nu pot fi sterse');

end if;

end;

/

delete armura\_v where id\_armura=7;

