

Proiect PSGBD

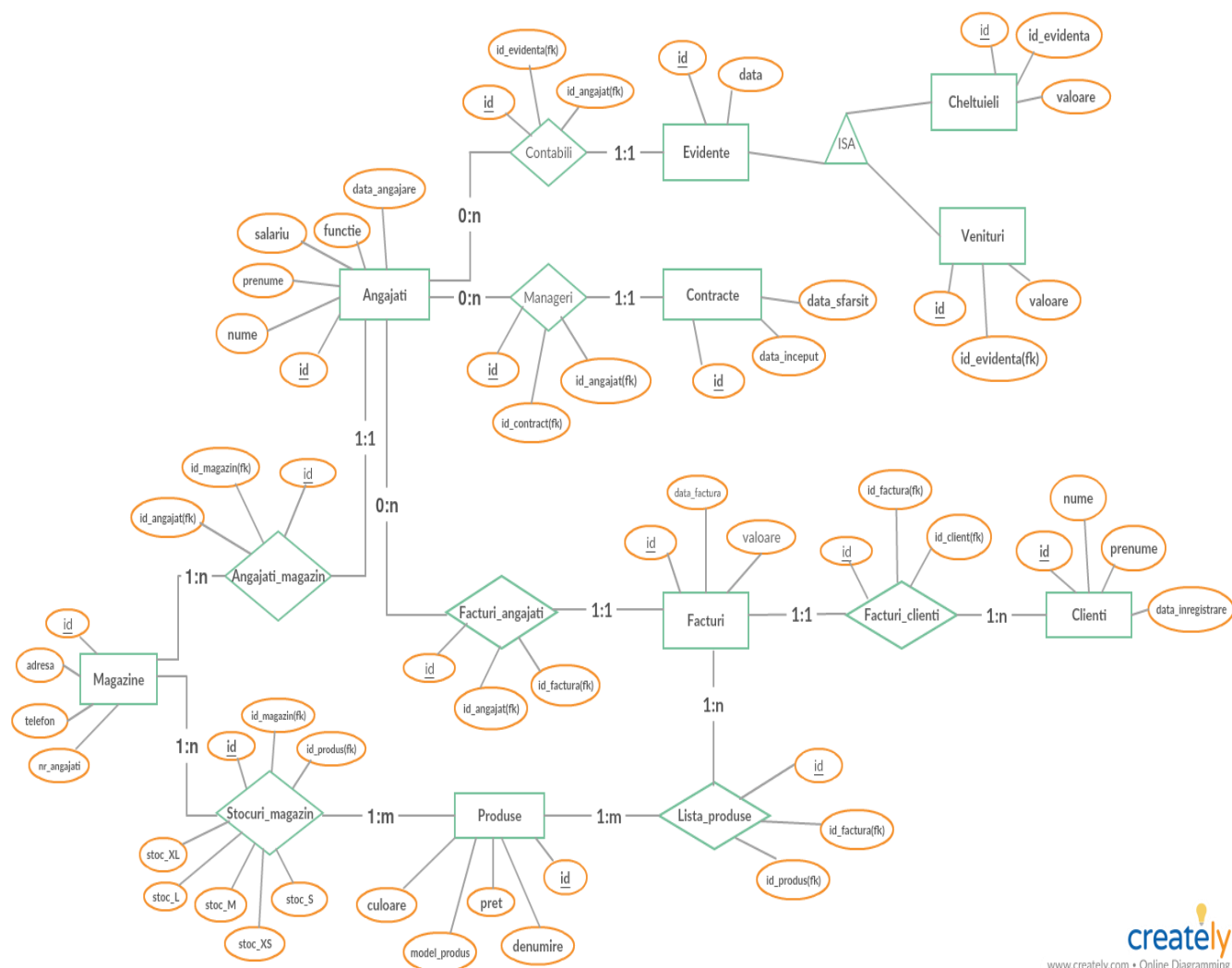
Descriere:

Tema proiectului este reprezentată de un lanț de magazine cu articole de îmbrăcăminte.

Aplicatia ofera angajatilor gestiunea usoara a produselor, interactiunea eficienta cu clientii, dar si evidenta clara a veniturilor si cheltuielilor.

Schema bazei de date: este alcatuita din 16 tabele:

-magazine	-manageri	-venituri	-clienti
-angajati_magazin	-evidente	-facturi_angajati	-stocuri_magazin
-angajati	-contracte	-facturi	-produs
-contabili	-cheltuieli	-facturi_clienti	-lista_produce



Forme normale:

Schema bazei de date este in:

- ✓ **1NF \Leftrightarrow domeniile atributelor sunt indivizibile si fiecare valoare a fiecarui atribut este atomica.**

- a) Eliminarea grupurilor repetitive din fiecare relatie
- b) Identificarea tuplelor ce ar putea avea duplicate in coloana printr-o cheie.
- c) Crearea unei noi scheme de relatie (aceasta are identificatorul de la b. + valoarea repetata ca atribut atomic).

Nu avem in schema bazei de date attribute care ar putea fi divizate in mai multe attribute diferite (nu avem mai multe valori in acelasi camp)

Nu avem mai multe coloane intr-un tabel care reprezinta acelasi tip de date/fapte/relatii(nu avem 2 coloane pt prenume1 si prenume2).

- ✓ **2NF -este in 1NF
-orice atribut neprim este dependent plin de orice cheie**

*atribut neprim-nu face parte din nicio cheie candidat

*dependent plin $X \rightarrow A$ plina daca nu exista X' inclus in X astfel incat $X' \rightarrow A$ apartine multimii de dependente functionale.

Cheia tabelelor noastre este "id"

Nu avem chei multivaluate \Rightarrow schema bazei de date este in 2NF (avem numai dependente pline de la chei, adica nu putem gasi submultimi de attribute atunci cand cheia este formata dintr un singur atribut).

Ex: Tabela Angajati

id	nume	prenume	salariu	functie	data_angajare	id_magazin
A	B	C	D	E	F	G

$\sigma = \{ A \rightarrow B, A \rightarrow C, A \rightarrow D, A \rightarrow E, A \rightarrow F, A \rightarrow G \}$

S	M	D

A	BCDEFG	

$A^+ = \{A, B, C, D, E, F, G\}$ cheie \Rightarrow attribute prime: A attribute neprime: B, C, D, E, F, G

$A \rightarrow B,$
 $A \rightarrow C,$
 $A \rightarrow D,$ dependente pline
 $A \rightarrow E,$
 $A \rightarrow F,$
 $A \rightarrow G$

Ex: Tabela Magazine

id	adresa	telefon	nr_angajati
A	B	C	D

$\sigma = \{ A \rightarrow B, A \rightarrow C, A \rightarrow D \}$

S	M	D

A	BCD	

$A^+ = \{A, B, C, D\}$ cheie \Rightarrow attribute prime: A attribute neprime: B, C, D

$A \rightarrow B,$
 $A \rightarrow C,$ dependente pline
 $A \rightarrow D$

Ex: Angajati_magazin

id	id_magazin	id_angajati
A	B	C

$\sigma = \{ A \rightarrow B, A \rightarrow C \}$

S	M	D

A	BC	

$A^+ = \{A, B, C\}$ cheie \Rightarrow attribute prime: A attribute neprime: B, C

$A \rightarrow B,$
 $A \rightarrow C$ dependente pline

Ex: Produe

id	denumire	pret	model_produ	culoare
A	B	C	D	E

$\sigma = \{A \rightarrow B, A \rightarrow C, A \rightarrow D, A \rightarrow E\}$

S	M	D

A		BCDE

$A^+ = \{A, B, C, D, E\}$ cheie \Rightarrow attribute prime: A attribute neprime: C, D, E

A \rightarrow B,
A \rightarrow C,
A \rightarrow D, dependente pline
A \rightarrow E

- ✓ **3NF** -este in 2NF
-orice atribut neprim NU este tranzitiv dependent de nicio cheie. (orice atribut neprim depinde de chei si nu de un alt atribut sau grupare de attribute neprime)

Ex: Tabela Angajati

id	nume	prenume	salariu	functie	data_angajare	id_magazin
A	B	C	D	E	F	G

$\sigma = \{A \rightarrow B, A \rightarrow C, A \rightarrow D, A \rightarrow E, A \rightarrow F, A \rightarrow G\}$

S	M	D

A		BCDEFG

$A^+ = \{A, B, C, D, E, F, G\}$ cheie \Rightarrow attribute prime: A attribute neprime: B, C, D, E, F, G

$A \rightarrow B,$
 $A \rightarrow C,$
 $A \rightarrow D,$ attributele neprime depind doar de cheia A
 $A \rightarrow E,$
 $A \rightarrow F,$
 $A \rightarrow G$

Ex: Tabela Magazine

id	adresa	telefon	nr_angajati
A	B	C	D

$\sigma = \{ A \rightarrow B, A \rightarrow C, A \rightarrow D \}$

S	M	D
A		BCD

$A^+ = \{A, B, C, D\}$ cheie \Rightarrow attribute prime: A attribute neprime: B, C, D

$A \rightarrow B,$
 $A \rightarrow C,$ attributele neprime depind doar de cheia A
 $A \rightarrow D$

Ex: Angajati_magazin

id	id_magazin	id_angajati
A	B	C

$\sigma = \{ A \rightarrow B, A \rightarrow C \}$

S	M	D
A		BC

$A^+ = \{A, B, C\}$ cheie \Rightarrow attribute prime: A attribute neprime: B, C

$A \rightarrow B,$
 $A \rightarrow C$ attributele neprime depind doar de cheia A

Ex: Produce

id	denumire	pret	model_produc	culoare
A	B	C	D	E

$\sigma = \{A \rightarrow B, A \rightarrow C, A \rightarrow D, A \rightarrow E\}$

S	M	D

A		BCDE

$A^+ = \{A, B, C, D, E\}$ cheie \Rightarrow attribute prime: A attribute neprime: C, D, E

A \rightarrow B,
A \rightarrow C,
A \rightarrow D, attributele neprime depind doar de cheia A
A \rightarrow E

- ✓ **BCNF** - este 1NF
- pt orice dependenta functionala netriviala $X \rightarrow A$, X este supercheie.

*triviala- $X \rightarrow Y$ este triviala daca si numai daca Y inclus in X

Ex: Tabela Angajati

id	nume	prenume	salariu	functie	data_angajare	id_magazin
A	B	C	D	E	F	G

$\sigma = \{A \rightarrow B, A \rightarrow C, A \rightarrow D, A \rightarrow E, A \rightarrow F, A \rightarrow G\}$

S	M	D

A		BCDEFG

$A^+ = \{A, B, C, D, E, F, G\}$ cheie \Rightarrow attribute prime: A attribute neprime: B, C, D, E, F, G

A->B,
A->C,
A->D, A este **supercheie** - atribut care identifica unic un tuplu intr-o relatie
A->E,
A->F,
A->G

La fel este si pentru restul tabelelor.

- ✓ **4NF** - este 1NF
- pt orice dependenta multivaluata netriviala $X \twoheadrightarrow A$, X este supercheie.

Noi nu avem dependente multivaluate.

Utilitate structuri de date:

Tabele:

Tabelele sunt folosite pentru a stoca informatiile despre produse, clienti, angajati, despre orice informatie necesara pentru a usura munca angajatilor dintr-un lant de magazine.

Tabelele evidente, contracte si facturi sunt folosite pentru a tine evidenta cheltuielilor si veniturilor astfel incat angajatii sa poate afla cu usurinta ce contracte sunt expirate sau vor expira in curand, care este cifra afacerii etc.

Tabela Stocuri_magazin este folosita pentru a tine evidenta produselor care se gasesc intr-un anumit magazin, precum si marimile in care sunt disponibile.

Tabela Clienti este folosita pentru a tine evidenta clientilor.

Tabela Angajati este folosita pentru a tine evidenta angajatilor. Contine campurile: id(cheie primara), nume, prenume, salariu, functie, data_angajare. Astfel va fi usor de aflat care este salariu fiecaruia, data de angajare daca se doreste o marire de salariu.

Tabelele contabili si manageri sunt folosite pentru a tine evidenta contabililor/managerilor si a fiecarei evidenta contabila sau contract pe care le-au realizat.

Triggere:

```
create or replace trigger delete_produs
after delete on produse
for each row
declare
begin
    delete from stocuri_magazin where id_produs = :old.id;
end
```

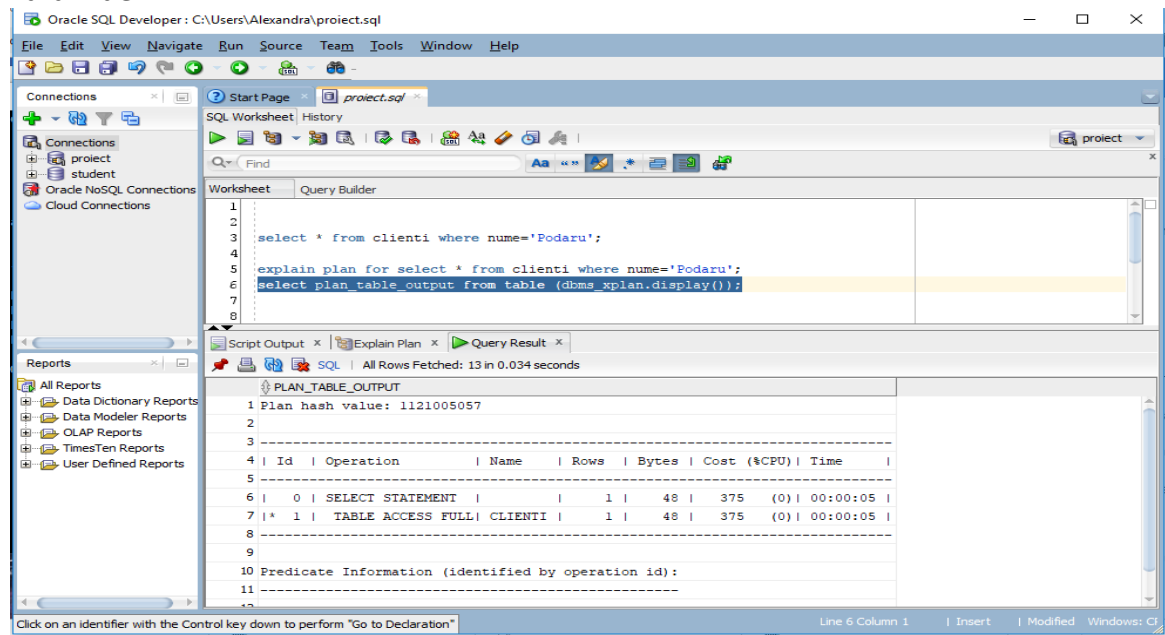
Acest trigger se va declansa automat in momentul in care un produs este sters din baza de date, acesta fiind sters si din tabela stocuri_magazin.

Indeksi:

Pentru a usura cautarile in baza de date a fost creat index-ul:

- **create index idx on clienti (nume);** -cautarea unor clienti in baza de date este mai rapida si eficienta datorita acestui index. Cautarea se poate face dupa nume.

Fara index



The screenshot displays the Oracle SQL Developer interface. The main window shows a SQL worksheet with the following query:

```
1 select * from clienti where nume='Podaru';
2
3
4
5 explain plan for select * from clienti where nume='Podaru';
6 select plan_table_output from table (dbms_xplan.display());
7
8
```

The bottom pane shows the execution plan output, titled "PLAN_TABLE_OUTPUT". It indicates that the plan hash value is 1121005057. The plan details are as follows:

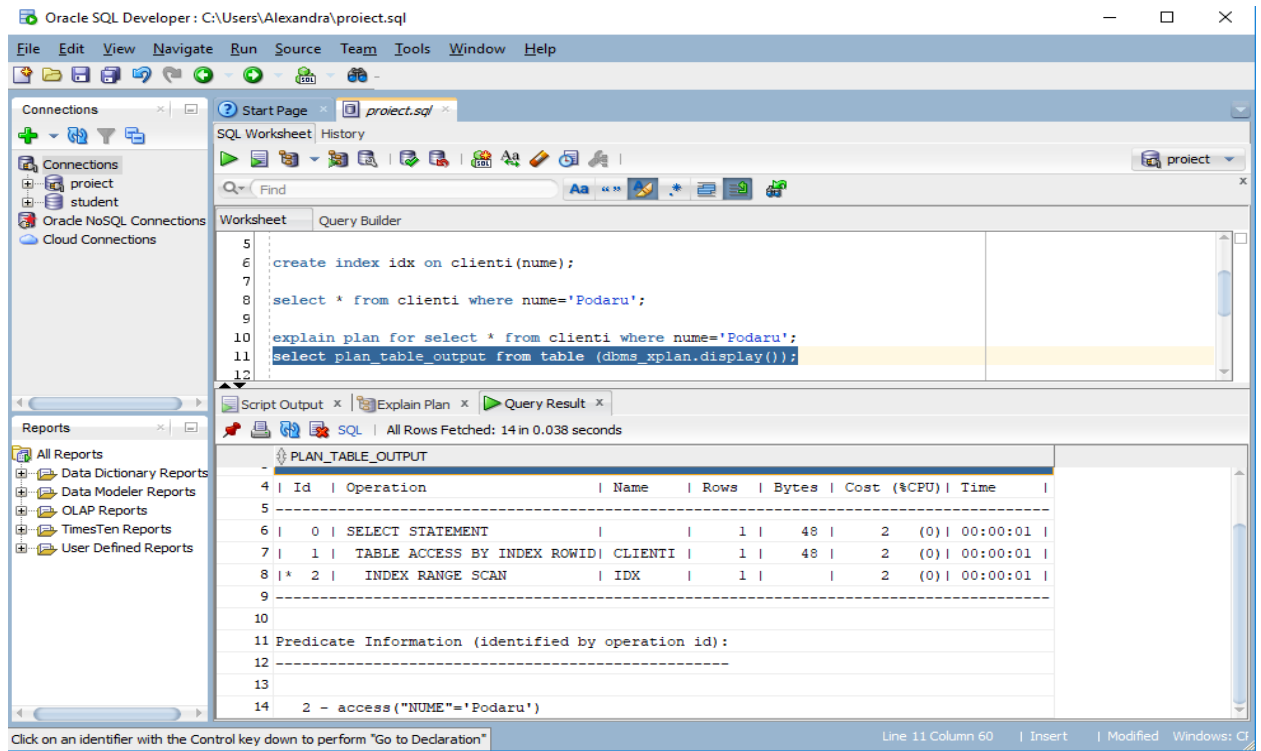
Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		1	48	375 (0)	00:00:05
1	TABLE ACCESS FULL	CLIENTI	1	48	375 (0)	00:00:05

The output also includes predicate information, which is identified by operation id:

```
10 Predicate Information (identified by operation id):
11
```

The status bar at the bottom indicates "Line 6 Column 1 | Insert | Modified | Windows: C".

Cu index



Constrangeri:

- chei primare: toate tabelele au id – cheie primara.
- chei straine: id_magazin, id_produs, id_angajat, id_evidenta
- not null: campurile id din toate tabelele sunt de tip not null deoarece fiecare inregistrare trebuie sa fie identificata intr-un mod diferit.

Proceduri si functii:

- ✓ **function adauga_produs**(p_categorie produse.categorie%type , p_denumire produse.denumire%type, p_pret produse.pret%type, p_model_produs produse.model_produs%type, p_culoare produse.culoare%type) return varchar2;
- introduce un nou produs in baza de date, generand automat un id unic.
- ✓ **function sterge_produs**(p_id produse.id%type) return varchar2;

- se sterge un produs din baza de date, primind la iesire un mesaj sugestiv.
 - ✓ **function actualizare_stoc** (p_id_produs produse.id_produs%type, p_stocS stocuri_magazin.stocS%type, p_stocXS stocuri_magazin.stocXS%type, p_stocM stocuri_magazin.stocM%type, p_stocL stocuri_magazin.stocL%type, p_stocXL stocuri_magazin.stocXL%type) return varchar2;
 - se verifica apartenenta produsului la baza de date, iar in caz afirmativ se actualizeaza stocurile XS,S,M,L,XL.
 - ✓ **function verifica_disponibilitate_produs**(p_denumire produse.denumire%type) return varchar2;
 - se verifica apartenenta produsului la baza de date, iar in caz afirmativ se afiseaza informatiile aferente;
 - ✓ **function discount_produs**(p_id produse.id%type, p_valoare_discount number) return varchar2;
 - se verifica apartenenta produsului la baza de date si se aplica discountul respectiv.
 - ✓ **function discount_categorie**(p_categorie produse.categorie%type, p_valoare_discount number) return varchar2;
 - se verifica existenta categoriei in baza de date si se aplica discountul respectiv.
 - ✓ **function afisare_total_produse** return varchar2;
 - afiseaza pe ecran id-ul, denumirea, categoria, pretul, modelul, culoarea si stocul tuturor produselor existente in baza de date.
-
- ✓ **function adauga_client**(p_id_client clienti.id%TYPE, p_nume clienti.nume %TYPE, p_prenume clienti.prenume%TYPE) return varchar2;
 - se verifica existenta id-ului noului client. Daca nu exista, se face inserarea in baza de date + data curenta;
 - ✓ **function actualizare_client**(p_id_client clienti.id%TYPE, p_nume clienti.nume %TYPE, p_prenume clienti.prenume%TYPE) return varchar2;
 - se verifica existenta id-ulul. Daca exista, se face actualizarea in baza de date;
 - ✓ **function sterge_client**(p_id_client clienti.id%TYPE) return varchar2;
 - se verifica existenta id-ulul. Daca exista, se face stergerea din baza de date;
 - ✓ **function cautare_client**(p_nume clienti.nume %TYPE, p_prenume clienti.prenume%TYPE) return varchar2;
 - se verifica existenta numelui si a prenumelui in baza de date,iar apoi se afiseaza datele clientului.
 - ✓ **function afisare_client**(p_id_client clienti.id%TYPE) return varchar2;

-se verifica existenta id-ulul. Daca exista, se face afisarea datelor clientului

✓ **function adauga_lista_produce**(p_id_product lista_produce.id_product%type, p_id_factura lista_produce.id_factura%type)

-se verifica existenta produselor si creeaza lista de produse pentru o anumita factura.

✓ **function adaugare_factura**(p_id_client clienti.id%type, p_id_angajat angajati.id_%type, p_valoare facturi.valoare%type) return varchar2;

- se verifica id-ul clientului, id-ul angajatului, valoare si lista produselor, iar apoi urmeaza emiterea facturii.

✓ **function sterge_factura**(p_id_factura facturi.id%type) return varchar2;

- se verifica existenta facturii, iar apoi se efectueaza stergerea.

✓ **function informatii_factura**(p_id_factura facturi.id%type) return varchar2;

- se verifica existenta facturii, iar apoi vor fi afisate toate informatiile despre aceasta(id, id_angajat, id_client, data_factura, valoare, lista_produce).

✓ **function adauga_angajat**(p_id_angajat angajati.id%TYPE, p_nume_angajat angajati.nume%TYPE, p_prenume_angajat angajati.prenume%TYPE, p_functie angajati.functie%TYPE, p_salariu angajati.salariu%type) return varchar2;

- se verifica existenta id-ului noului angajat. Daca nu exista, se face inserarea in baza de date;

✓ **function actualizare_angajat**(p_id_angajat angajati.id%TYPE, p_nume_angajat angajati.nume%TYPE, p_prenume_angajat angajati.prenume%TYPE) return varchar2;

- se verifica existenta id-ului angajatului. Daca exista, se face actualizarea in baza de date;

✓ **function sterge_angajat**(p_id_angajat angajati.id%TYPE) return varchar2;

- se verifica existenta id-ului angajatului. Daca exista, se face stergerea din baza de date;

✓ **function afisare_angajat**(p_id_angajat angajati.id%TYPE) return varchar2;

- se verifica existenta id-ului angajatului. Daca exista, se face afisarea datelor.

- ✓ **function listare_angajati** return varchar2;
- se afiseaza id-ul, nume, prenumele, salariul, functia si data_angajarii tuturor angajatilor existenti in baza de date;
- ✓ **function listare_clienti** return varchar2;
- se afiseaza id-ul, numele, prenumele si datei de inregistrare a tuturor clientilor existenti in baza de date.
- ✓ **function cheltuieli_impozit**(p.luna VARCHAR2(10), p.id magazine.id%TYPE) return VARCHAR2;
-returneaza suma data la impozit in luna respectiva pentru magazin;
- ✓ **function cheltuieli_taxe**(p.luna VARCHAR2(10), p.id magazine.id%TYPE) return VARCHAR2;
-returneaza taxele din luna respectiva pentru magazin;
- ✓ **function cheltuieli_intretinere**(p.luna VARCHAR2(10), p.id magazine.id%TYPE) return VARCHAR2;
-returneaza suma data la intretinere in luna respectiva pentru magazin;
- ✓ **function cheltuieli_materiale**(p.luna VARCHAR2(10), p.id magazine.id%TYPE) return VARCHAR2;
-returneaza suma data pentru materiale in luna respectiva pentru magazin;
- ✓ **function cheltuieli_chirie**(p.luna VARCHAR2(10), p.id magazine.id%TYPE) return VARCHAR2;
-returneaza suma data pentru chirie in luna respectiva pentru magazin;
- ✓ **function cheltuieli_salarii**(p.luna VARCHAR2(10), p.id magazine.id%TYPE) return VARCHAR2;
-returneaza suma rezervata pentru salarii in luna respectiva pentru magazin;
- ✓ **function maxim_cheltuieli**(p.luna VARCHAR2(10), p.id magazine.id%TYPE) return VARCHAR2;
-returneaza pentru ce se alocă cea mai mare suma de bani in luna respectiva;
- ✓ **procedure raport_produce**(p.luna VARCHAR2(10)) ;
-returneaza ce produse s-au vandut cel mai bine in luna respectiva;
- ✓ **function bestseller**(p.luna VARCHAR2(10)) return VARCHAR2;
-returneaza ce produs s-a vandut cel mai bine in luna respectiva;
- ✓ **procedure raport_vanzari**(p.luna_inceput VARCHAR2(10),p.luna_inceput VARCHAR2(10)) ;
-returneaza care au fost produsele cu vanzarile cele mai mare/cele mai mici ;
- ✓ **procedure raport_clienti**(p.id magazin.id%TYPE);
-returneaza clienti existenti in baza de date intr-un magazin;
- ✓ **procedure raport_contracte_expire**(p.id magazin.id%TYPE) ;
-returneaza ce contracte sunt expirate;

- ✓ **procedure raport_contracte_expira**(p.id magazin.id%TYPE) ;
-returneaza ce contracte vor expira in luna viitoare;