PROIECT BAZE DE DATE

-EVIDENTA POPULATIEI LA SERVICIUL DE ADMINISTRATIE PUBLICA-



Universitatea POLITEHNICA din Bucuresti



Facultatea de automatica si calculatoare

NUME STUDENT: IORDACHE TIBERIU-STEFAN

GRUPA: 331AB

1 Tema proiectului

Tema proiectului se numeste: "Evidenta populatiei la Serviciul de Administratie Publica (proprietati, taxe, datorii etc.)", iar numarul biletului ce o continea in momentul extragerii a fost 6.

2 Definirea cerintelor pentru aplicatia aleasa: lista functionalitatilor, identificarea componentelor

Evidenta Populatiei este o unitate componenta a Serviciului de Administratie Publica si are ca scop asigurarea colectarii impozitelor si a taxelor locale de la cetateni, prin intermediul Directiei Impozite si Taxe Locale.

Pentru modelarea activitatii institutiei, proiectul are ca scop realizarea bazei de date **Evidenta Populatiei.**

Intrucat obiectul muncii il constituie **cetatenii**, acestia reprezinta un element esential, necesitatea stocarii datelor de identificare pentru fiecare in partedevenind foarte clara. Asadar, datele de interes vor fi: nume, prenume, CNP, serie CI, numar CI, statut persoana, numar rol nominal unic.

Plata impozitelor si a taxelor locale se realizeaza individual, in functie de numarul de bunuri pe care un locuitor le are pe raza de competenta a Directiei Impozite si Taxe Locale. Astfel, pe langa datele de identificare a cetatenilor, este util ca in baza de date sa se regaseasca informatiidespre **bunurile** pe care acestia le poseda, din care sa se poata obtine o clasificare si o detaliere a acestora, cum ar fi: **categoria** din care un bun face parte, **caracteristicile**acestuia si o descriere succinta.

De asemenea, fiecarui cetatean si fiecarui bun ii este asociata o **adresa**. La randul sau, o adresa se compune din: strada, numar, bloc, apartament, judet/sector, oras.

Din descrierea functionalitatilor rezulta urmatoarele componente ce vor fi modelate in baza de date *Evidenta Populatiei*, si anume: **cetateni**, **adresa**, **bunuri**, **categorie** si **caracteristici**.

3 Proiectarea bazei de date

3.1 Identificarea tabelelor

Avand in vedere lista functionalitatilor prezentata mai sus, pentru fiecare dintre entitati (cetateni, adresa, bunuri, categorie si caracteristici) voi crea un tabel

in baza de date *Evidenta Populatiei*. Vor exista urmatoarele 5 tabele: Cetateni, Adrese, Bunuri, Categorie Bun si Caracteristici.

O precizare importanta ce trebuie mentionata este legata de tabelul **Adrese**. Entitatea adresa a fost modelata in baza de date printr-un tabel de sine statator pentru a reduce pe cat posibil retinerea de date duplicate. Altfel spus, includerea datelor despre adresa atat in tabelul Cetateni, cat si in tabelul Bunuri ar fi incalcat principiul Formei Normale de ordin 2, deoarece in situatia in care adresa unui cetatean coincidea cu cea a unui bun care se afla in posesia sa aparea fenomenul de stocarea in baza de date de doua ori a aceleasi adrese (duplicarea datelor).

3.2 Stabilirea campurilor si a tipurilor de date

Facand analogia cu situatia reala, fiecare entitate este caracterizata de o serie de atribute. Pe baza interactiunilor pe care eu, dar si parintii mei le-am avut cu Evidenta Populatiei, impreuna cu efectuarea unei documentari pe internet, voi lua in considerare urmatoarele atribute caracteristice pentru fiecare entitate, care in structura tabelelor vor deveni campuri, dupa cum urmeaza:

CETATENI	ADRESE	BUNURI	CATEGORIE BUN	CARACTERISTICI
ID Cetatean	ID Adresa	ID Bun	ID Categorie	ID Caracteristica
Nume	Strada	Descriere	Denumire	Denumire
Prenume	Numar			
CNP	Bloc			
Serie CI	Apartament			
Numar CI	Judet/Sector			
Statut persoana	Oras			
Nr. rol				
nominal				
unic				

Pentru identificarea in mod unic a fiecarei inregistrari, am folosit o cheie primara artificiala. Astfel, pentru fiecare tabel identificat am introdus un camp identificator de tip intreg, caruia i-am atribuit rolul de cheie primara si poate fi regasit in tabelele de mai sus pe un fundal galben.

Tipurile de date asociate campurilor din tabelele de mai sus sunt urmatoarele:

Pentru tabelul Cetateni:

- o ID Cetatean: int
- Nume: nvarchar(50)
- o Prenume: nvarchar(50)
- o CNP: char(13)
- o Serie CI : char(2)
- o Numar CI : char(6)
- o Statut persoana : char(1)
- o Nr. rol nominal unic: nvarchar(25)

♣ Pentru tabelul Adrese:

- o ID Adresa: int
- Strada : nvarchar(50)
- Numar :nvarchar(10)
- o Bloc: nvarchar(10)
- Apartament : nvarchar(5)
- Judet/Sector : nvarchar(50)
- o Oras : nvarchar(50)

Pentru tabelul Bunuri:

- o ID Bun: int
- Descriere : nvarchar(50)

♣ Pentru tabelul **Categorie Bun**:

- o ID Categorie: int
- o Denumire: nvarchar(50)

♣ Pentru tabelul Caracteristici:

- o ID Caracteristica: int
- o Denumire: nvarchar(50)

3.3 Identificarea relatiilor dintre tabele

\Rightarrow Cetateni – Bunuri 1:N + 1:N = N:N

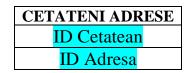
- O Un cetatean poate detine unul sau mai multe bunuri. Un bun poate fi detinut de unul sau mai multi cetateni (de exemplu: un teren este detinut de mai multi cetateni in urma unei succesiuni; o locuinta poate fi trecuta pe numele a doi soti).
- Relatia este de tip many-to-many si se modeleaza in baza de date prin introducerea tabelului de legatura, pe care l-am denumit sugestiv Impozit. In structura tabelului Impozit se vor regasi copii ale cheilor primare din tabelele Cetateni (ID Cetatean), respectiv Bunuri (ID Bun)care vor deveni chei straine. In plus, tabelul Impozit va contine si un camp denumit Valoare de plata si reprezinta suma pe care un cetatean trebuie sa o achite pentru un anumit bun ce se afla in posesia sa. Mai jos se afla structura tabelului de legatura.



Tipul campului Valoare de plata este decimal(10, 2).

\bigcirc Cetateni – Adrese 1:N+1:N=N:N

- Un cetatean poate locui la una sau mai multe adrese (de exemplu: un cetatean detine o viza de flotant). La o adresa pot locui unul sau mai multi cetateni.
- Relatia este de tip many-to-many si se modeleaza in baza de date prin introducerea tabelului de legatura denumitCetateniAdrese. In structura tabelului Cetateni Adrese se vor regasi copii ale cheilor primare din tabelele Cetateni (ID Cetatean), respectiv Adrese (ID Adresa)care vor deveni chei straine.



Danuari – **Adrese** 1:1 + 1:N = 1:N

- O Un bun poate fi asociat unei singure adrese. La o adresa pot fi asociate unul sau mai multe bunuri.
- O Relatia este de tip one-to-many. "Partea unu" a relatiei este Adrese, iar "partea mai multi" este Bunuri. Relatia se modeleaza in baza de date astfel: se ia o copie a cheii primare a tabelului din "partea unu" (ID Adresa) si se introduce in structura tabelului din "partea mai multi" (Bunuri), unde aceasta va deveni cheie straina.

Description Bun 1:1 + 1:N = 1:N

- Un bun poate apartine unei singure categorii. O categorie poate contine unul sau mai multe bunuri.
- O Relatia este de tip one-to-many. "Partea unu" a relatiei este Categorie Bun, iar "partea mai multi" este Bunuri. Relatia se modeleaza in baza de date astfel: se ia o copie a cheii primare a tabelului din "partea unu" (ID Categorie) si se introduce in structura tabelului din "partea mai multi" (Bunuri), unde aceasta va deveni cheie straina.



Dunuri – Caracteristici 1:N + 1:N = N:N

- O Un bun poate avea una sau mai multe caracteristici. O caracteristica poate fi intalnita la unul sau mai multe bunuri.
- Relatia este de tip many-to-many si se modeleaza in baza de date prin introducerea tabelului de legatura denumit Bunuri Caracteristici. In structura tabelului Bunuri Caracteristici se vor regasi copii ale cheilor primare din tabelele Bunuri (ID Bun), respectiv Caracteristici (ID Caracteristica) care vor deveni chei

straine. In plus, tabelul **Bunuri Caracteristici** va contine si un camp denumit *Valoare asociata*si reprezinta un numar ce descrie marimea caracteristicii unui bun.

BUNURI CARACTERISTICI						
ID Bun						
ID Caracteristica						
Valoarea asociata						

Tipul campului Valoare asociata este nvarchar(10).

3.4 Impunerea unor constrangeri de integritate

Constrangerile impuse campurilor din tabelele din baza de date sunt urmatoarele:

Pentru tabelul Cetateni:

- o ID Cetatean: PRIMARY KEY cu autoincrementare
- o Nume: NOT NULL
- o Prenume: NOT NULL
- o CNP: NOT NULL, UNIQUE
- o Serie CI : NOT NULL
- o Numar CI: NOT NULL
- o Statut persoana: NOT NULL, CHECK('F' OR 'J')
- o Nr. rol nominal unic : NOT NULL, UNIQUE

♣ Pentru tabelul Adrese:

- o ID Adresa: PRIMARY KEY cu autoincrementare
- o Strada: NOT NULL
- o Numar: NOT NULL
- o Bloc:
- o Apartament:
- o Judet/Sector: NOT NULL
- o Oras : NOT NULL

Pentru tabelul Bunuri:

o ID Bun: PRIMARY KEY cu autoincrementare

o Descriere: NOT NULL

o ID Adresa: FOREIGN KEY

o ID Categorie: FOREIGN KEY

♣ Pentru tabelul **Categorie Bun**:

o ID Categorie: PRIMARY KEY cu autoincrementare

o Denumire: NOT NULL

Pentru tabelul Caracteristici:

o ID Caracteristica: PRIMARY KEY cu autoincrementare

o Denumire: NOT NULL

Pentru tabelul Impozit:

o ID Cetatean: FOREIGN KEY

o ID Bun: FOREIGN KEY

o Valoare de plata : NOT NULL

♣ Pentru tabelul **Bunuri Caracteristici**:

o ID Bun: FOREIGN KEY

o ID Caracteristica: FOREIGN KEY

Valoare asociata : NOT NULL

4. Extragerea selecturilor

\$\rightarrow\$ SELECTURILE **SIMPLE**

- 1. Interogare Simpla 1: Afiseaza Nume, Prenume, Strada, Numar, Bloc, Apartament, Judet, Oras pentru inregistrarile din tabelul Cetateni.
- \$sql = "SELECT Nume, Prenume, Strada, Numar, Bloc, Apartament, Judet, Oras FROM cetateni C

INNER JOIN cetateniadrese CA ON (C.ID_Cetatean = CA.ID_Cetatean) INNER JOIN adrese A ON (CA.ID_Adresa = A.ID_Adresa)";

2. Interogare Simpla 2: Afiseaza Nume, Prenume, CNP, Denumire, NumarBunuri (mobile si imobile) pentru *TOATE* inregistrarile din tabelul Cetateni.

\$sql = "SELECT Nume, Prenume, CNP, Denumire, COUNT(B.ID_Categorie) AS NumarBunuri

FROM cetateni C

LEFT OUTER JOIN impozit I ON (C.ID_Cetatean = I.ID_Cetatean)
LEFT OUTER JOIN bunuri B ON (I.ID_Bun = B.ID_Bun)
LEFT OUTER JOIN categoriebun CB ON (B.ID_Categorie = CB.ID_Categorie)
GROUP BY Nume, Prenume, CNP, Denumire";

- 3. Interogare Simpla 3: Permite introducerea unei valori pentru campul CNP (camp variabil). Pe baza CNP-ului introdus, afiseaza Nume, Prenume, TotalDePlata asociate cetateanului cu CNP-ul respectiv.
- \$sql = "SELECT Nume, Prenume, SUM(ValoareDePlata) AS TotalDePlata
 FROM cetateni C INNER JOIN impozit I ON (C.ID_Cetatean = I.ID_Cetatean)
 WHERE CNP = \$cnp
 GROUP BY Nume, Prenume";
 - 4. Interogare Simpla introducerea 4: Permite unei valori pentru campul NrRolNominalUnic (camp variabil). Pe baza NrRolNominalUnic introdus, afiseaza adresa asociata (Strada, Numar, ...) si bunurile (Descriere) de la acea adresa, impreuna cu caracteristicile (Denumire) si valorile asociate acestora (ValoareAsociata).

\$sql = "SELECT Strada, Numar, Bloc, Apartament, Judet, Oras, Descriere, Denumire, ValoareAsociata

FROM cetateni C

INNER JOIN cetateniadrese CA ON (C.ID_Cetatean = CA.ID_Cetatean)

INNER JOIN adrese A ON (CA.ID_Adresa = A.ID_Adresa)

INNER JOIN bunuri B ON (A.ID_Adresa = B.ID_Adresa)

INNER JOIN bunuricaracteristici BC ON (B.ID_Bun = BC.ID_Bun)

INNER JOIN caracteristici CR ON (BC.ID_Caracteristica =

CR.ID_Caracteristica)

WHERE NrRolNominalUnic = \$nrrolnominalunic

GROUP BY Strada, Numar, Bloc, Apartament, Judet, Oras, Descriere, Denumire, ValoareAsociata";

5. Interogare Simpla 5:Permite introducerea unei adrese (valori pentru Strada, Numar, ...) (camp variabil). Pe baza adresei introduse, afiseaza Nume, Prenume, SerieCI, NumarCI ale cetatenilor care locuiesc (sunt asociati) la acea adresa.

\$sql = "SELECT Nume, Prenume, SerieCI, NumarCI

FROM cetateni C

INNER JOIN cetateniadrese CA ON (C.ID_Cetatean = CA.ID_Cetatean)

INNER JOIN adrese A ON (CA.ID_Adresa = A.ID_Adresa)

WHERE Strada = '\$strada'

AND Numar = '\$numar'

AND (Bloc = '\$bloc' OR Bloc IS NULL)

AND (Apartament = '\$apartament' OR Apartament IS NULL)

AND Judet = '\$judet'

AND Oras = '\$oras''';

6. Interogare Simpla 6:Permite introducerea pentru unei valori campul **NumarInmatriculare** (pentru campul ValoareAsociata care corespunde caracteristicii cu denumirea NumarInmatriculare) (camp variabil). Pe baza NumarInmatriculare introdus, afiseaza Nume. Prenume ale cetateanului care detine bunul (care poate fi masina. motocicleta sau barca) cu numarul de inmatriculare respectiv.

\$sql = "SELECT Nume, Prenume

FROM cetateni C

INNER JOIN cetateniadrese CA ON (C.ID_Cetatean = CA.ID_Cetatean)

INNER JOIN adrese A ON (CA.ID_Adresa = A.ID_Adresa)

INNER JOIN bunuri B ON (A.ID_Adresa = B.ID_Adresa)

INNER JOIN bunuricaracteristici BC ON (B.ID_Bun = BC.ID_Bun)

WHERE ValoareAsociata = '\$numarinmatriculare'";

SELECTURILE *COMPLEXE*

7. Interogare Complexa 1:Afiseaza **Nume, Prenume, TotalDePlata** pentru cetateanul a carui valoare totala de plata este cea mai mare dintre toti cetatenii.

\$sql = "SELECT Nume, Prenume, SUM(I.ValoareDePlata) AS TotalDePlata

FROM cetateni C, impozit I

WHERE C.ID_Cetatean = I.ID_Cetatean

GROUP BY C.Nume, C.Prenume

HAVING TotalDePlata >= ALL (SELECT SUM(I2.ValoareDePlata)

FROM impozit I2

GROUP BY ID_Cetatean)";

8. Interogare Complexa 2: Afiseaza Nume, Prenume, StatutPersoana, NrRolNominalUnic, Oras pentru cetatenii care nu au bunuri.

\$sql = "SELECT Nume, Prenume, StatutPersoana, NrRolNominalUnic, Oras

FROM cetateni C, cetateniadrese CA, adrese A

WHERE C.ID_Cetatean = CA.ID_Cetatean AND

CA.ID_Adresa = A.ID_Adresa AND

C.ID_Cetatean NOT IN (SELECT I.ID_CetateanFROM impozit I)";

9. Interogare Complexa 3: Pentru fiecare **Oras**, **Judet** afiseaza **valoarea toala a capacitatilor cilindrice** corespunzatoare.

\$sql = "SELECT Oras, Judet,

(SELECT SUM(BC.ValoareAsociata)
FROM bunuricaracteristici BC, bunuri B
WHERE BC.ID_Caracteristica = 1 AND

BC.ID_Bun = B.ID_Bun AND B.ID_Adresa = A.ID_Adresa) AS TotalCapacitatiCilindrice

FROM adrese A ORDER BY TotalCapacitatiCilindrice DESC;";

10.Interogare Complexa 4: Permite introducerea unui numar care reprezinta **numarul minim de bunuri dupa care facem cautarea** (*camp variabil*). Afiseaza **Nume, Prenume, NrRolNominalUnic** pentru cetatenii care detin un numar de bunuri mai mare sau egal cu cel introdus.

\$sql = "SELECT Nume, Prenume, NrRolNominalUnic

FROM cetateni C, (SELECT C2.ID_Cetatean

FROM cetateni C2, impozit I

WHERE C2.ID_Cetatean = $I.ID_Cetatean$

GROUP BY C2.ID_Cetatean

HAVING COUNT(I.ID_Bun) >= \$nrminimdebunuri) AS CB

WHERE C.ID_Cetatean = CB.ID_Cetatean;";

5. Functionarea aplicatiei

Aplicatia se deschide cu o pagina principala care contine butoane cu anumite functionalitati, dupa cum se poate observa si in imaginea de mai jos:

Evidenta Populatiei -pagina de pornire-Operatii Modificare Cetatean Permite realizarea operatiilor de insert, update, delete asupra datelor din tabelul Cetateni. Operatii Modificare Adresa Permite realizarea operatiilor de insert, update, delete asupra datelor din tabelul Adrese. Mai jos urmeaza INTEROGARILE SIMPLE Interogare simpla 1 Afiseaza Nume, Prenume, Strada, Numar, Bloc, Apartament, Judet, Oras pentru inregistrarile din tabelul Cetateni. Interogare simpla 2 Afiseaza Nume, Prenume, CNP, Denumire, NumarBunuri (mobile și imobile) pentru TOATE inregistrarile din tabelul Cetateni. Interfata interogare 3 Permite introducerea unei valori pentru campul CNP (camp variabil). Pe baza CNP-ului introdus, afiseaza Nume, Prenume, TotalDePlata asociate cetateanului cu CNP-ul respectiv. Interfata interogare 4 Permite introducerea unei valori pentru campul NrRolNominalUnic (camp variabil). Pe baza NrRolNominalUnic introdus, afiseaza adresa asociata (Strada, Numar, ...) si bunurile (Descriere) de la acea adresa, impreuna cu caracteristicile (Denumire) si valorile asociate acestora (ValoareAsociata) Interfata interogare 5 Permite introducerea unei adrese (valori pentru Strada, Numar, ...) (camp variabil). Pe baza adresei introduse, afiseaza Nume, Prenume, SerieCI, NumarCI ale cetatenilor care locuiesc (sunt asociati) la acea adresa Interfata interogare 6 Permite introducerea unei valori pentru campul NumarInmatriculare (pentru campul ValoareAsociata care corespunde caracteristicii cu denumirea NumarInmatriculare) (camp variabil) Pe baza NumarInmatriculare introdus, afiseaza Nume, Prenume ale cetateanului care detine bunul (care poate fi masina, motocicleta sau barca) cu numarul de inmatriculare respectiv. Mai jos urmeaza INTEROGARILE COMPLEXE: Interogare Complexa 1 Afiseaza Nume, Prenume, TotalDePlata pentru cetateanul a carui valoare totala de plata este cea mai mare dintre toti cetatenii Interogare Complexa 2 Afiseaza Nume, Prenume, StatutPersoana, NrRolNominalUnic, Oras pentru cetatenii care nu au bunuri. Interogare Complexa 3 Pentru fiecare Oras, Judet afiseaza valoarea toala a capacitatilor cilindrice corespunzatoare Interfata Interogare Complexa 4 Permite introducerea unui numar care reprezinta numarul minim de bunuri dupa care facem cautarea (camp variabil).

Afiseaza Nume, Prenume, NrRolNominalUnic pentru cetatenii care detin un numar de bunuri mai mare sau egal cu cel introdus

Pentru a executa operatiile de insert, update sau delete, puteti apasa pe butoanele Operatii Modificare Cetatean, respectiv Operatii Modificare Adresa. Diferenta dintre cele doua este tabelul din baza de date asupra caruia se efectueaza aceste operatii. Apasarea butonului Operatii Modificare Cetatean va va redirectiona catre pagina de mai jos:

Lista Cetatenilor										
Insert Cetatean										
Iordache	Tiberiu	5010209460087	RK	286914	F	1574869031	Update Delete			
Constantin	Vlad	1890828180570	TR	912475	F	4269710254	Update Delete			
Popa	Alexandru	6020530221075	XB	429748	J	7443809165	Update Delete			
Ungureanu	Dominic	2950721098845	KZ	147023	F	7461359233	Update Delete			
Constantin	Mara	5001228334388	XT	624738	J	3557140119	Update Delete			
			Pagina p	orincipala						

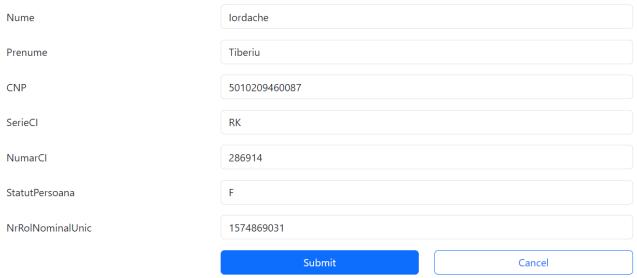
Denumirile butoanelor din imaginea de mai sus sunt sugestive, iar rolul lor este usor de inteles. Astfel, pentru a introduce un cetatean nou in baza de date trebuie apasat butonul Insert Cetatean, apoi va aparea urmatoarea pagina:

Nume Prenume CNP SerieCl NumarCl StatutPersoana NrRolNominalUnic Submit Cancel

Dupa completarea campurilor si apasarea butonului Submit, datele nou introduse ar trebui sa apara in Lista Cetatenilor ca fiind ultima inregistrare.

Daca se doreste actualizarea datelor unui cetatean, apasarea butonului Update din dreptul acestuia va va redirectiona la pagina urmatoare:

Update Cetatean



Se regasesc asadar vechile valori ale cetateanului. In functie de ce dorim, putem schimba (actualiza) orice valoare din campurile de mai sus. Dupa apasarea butonului de Submit, modificarile produse inregistrarii corespunzatoare cetateanului se vor regasi in Lista Cetatenilor.

Operatia de Delete se executa similar cu cea de Update, adica se navigheaza in dreptul cetateanului pe care dorim sa il stergem din baza de date si se apasa butonul Delete, iar inregistrarea corespunzatoare acestuia va disparea. Modul de functionare pentru Operatii Modificare Adresa este similar cu cel prezentat mai sus, Operatii Modificare Cetatean.

Pentru vederea rezultatului produs de o interogare, fie ea simpla sau complexa, se procedeaza astfel:

• Daca interogarea este fara camp variabil: Se va apasa pe butonul Interogare Simpla/Complexa 'x', iar apoi va aparea pe ecran rezultatul acesteia. Mai jos, este un exemplu cu rezultatul Interogarii Simple 2:



• Daca interogarea este cu camp variabil: Se va apasa pe butonul Interfata Interogare 'x' sau Interfata Interogare Complexa 4. Vom lua ca exemplu, mai jos, Interfata Interogare Complexa 4. La apasarea acestui buton vom vedea urmatoarea pagina:



Introducem valoarea pe care o dorim in campul liber si apasam Trimite. Ulterior, se va afisa pe ecran rezultatul interogarii complexe, care va arata in felul urmator:



Campul variabil presupune introducerea unei valori de catre utilizator si folosirea acesteia in cadrul Cererilor Simple sau Cererilor Complexe.

Va multumesc foarte mult!