**Lucrare de atestare profesională la INFORMATICĂ**

PROGRAMĂRI

**Realizator,**LUPU DAVID-ȘTEFAN

**Profesor coordonator,**

BUTYKA EDMOND

Cuprins

[1. Definirea Problemei 4](#_Toc133258152)

[2. Baze de date 5](#_Toc133258153)

[2.1. Tabele 5](#_Toc133258154)

[2.1.1. Indecși 5](#_Toc133258155)

[2.1.2. Câmpuri 5](#_Toc133258156)

[2.1.3. Relații de tabele 6](#_Toc133258157)

[2.2. Interogări 7](#_Toc133258158)

[3. Descriere Microsoft Access 9](#_Toc133258159)

[3.1. Formulare 10](#_Toc133258160)

[3.2. Macrocomenzi 11](#_Toc133258161)

[3.3. Module 11](#_Toc133258162)

[3.4. Rapoarte 13](#_Toc133258163)

[4. Descrierea tabelelor și a interogărilor folosite 14](#_Toc133258164)

[4.1. Tabele 14](#_Toc133258165)

[4.1.1. Tabela PuncteLucru 14](#_Toc133258166)

[4.1.2. Tabela ZileLibere 14](#_Toc133258167)

[4.1.3. Tabela Judete 16](#_Toc133258168)

[4.1.4. Tabela Localitati 17](#_Toc133258169)

[4.1.5. Tabela Persoane 18](#_Toc133258170)

[4.1.6. Tabela TMPPerioade 19](#_Toc133258171)

[4.1.7. Tabela Programari 20](#_Toc133258172)

[4.2. Interogări 21](#_Toc133258173)

[4.2.1.Interogarea QProgramări/Qprogramări0 21](#_Toc133258174)

[4.2.2. Interogarea QProgramariGrupate 22](#_Toc133258175)

[4.2.3. Interogarea QLocalități 25](#_Toc133258176)

[4.2.4. Interogarea QPersoane 25](#_Toc133258177)

[5. Descrierea formelor 26](#_Toc133258178)

[5.1. Formular “PuncteLucru” 26](#_Toc133258179)

[5.2. Formular “ZileLibere” 26](#_Toc133258180)

[5.3. Formular “Judete” 27](#_Toc133258181)

[5.4. Formular “Localitati” 28](#_Toc133258182)

[5.5. Formular “Persoane” 28](#_Toc133258183)

[5.6. Formular “Programari” 29](#_Toc133258184)

[5.6.1. Descriere procedura “Aplica”: 30](#_Toc133258185)

[5.6.2. Adaugare programare noua 33](#_Toc133258186)

[6. Viitoare îmbunătățiri ale aplicației 38](#_Toc133258187)

[7. Bibliografie 38](#_Toc133258188)

# Definirea Problemei

Problema propusă pentru rezolvare este programarea paciențiilor pentru un cabinet medical cu multiple puncte de lucru. Din cadrul fiecărui punct de lucru, trebuie permisă programarea paciențiilor în oricare punct de lucru respectiv vizualizarea programărilor din toate locațiile. Punctele de lucru sunt conectate într-o rețea virtuală private, astfel încât se pot partaja resurse între calculatoare.

Aplicația care ar rezolva această problem ar trebui sa gestioneze următoarele date:

-Puncte de lucru;

-Persoane (cu localitate și județe);

-Zile libere (Naționale);

-Programări;

Am ales să dezvolt această aplicație în Microsoft Access datorită faptului că în acesta se pot defini ușor tabele și se pot dezvolta la fel de ușor ferestre de editare a datelor. Aplicația va fi instalată pe cel puțin un calculator la fiecare punct de lucru și vor partaja un fișier de date comun.

# Baze de date

O bază de date este un instrument pentru colectarea și organizarea informațiilor. Bazele de date pot stoca informații despre persoane, produse, comenzi sau orice altceva. Bazele de date ne permit să stocăm și să consultăm date structurate respectiv nestructurate.

În aplicația de față ne vom referii doar la date structurate și vom exemplifica folosirea lor în mediul Microsoft Access.

O bază de date conține tabele, acestea pot fi relaționate între ele. În baza de date se pot defini interogări (queries) pe baza tabelelor. În marea majoritate a bazelor de date moderne se permite definirea unor proceduri și funcții care pot fi utilizate în accesarea/stocarea datelor.

## Tabele

O tabelă stochează datele pentru un tip bine structurat de date. Tabelele conțin înregistrări cu același tip structurat de date. Structura înregistrării este definită de câmpuri, câmpurile reprezintă proprietăți ale datelor pe care le reprezintă.

Tabelele sunt relaționate între ele, de exemplu: o localitate face parte dintr-un județ respectiv o persoană este născută într-o localitate.

O tabelă are un câmp de cheie primară, aceasta permite identificarea unică a unei inregistrări pe baza valorii acestei chei.

### Indecși

Pe tabelă putem defini indecși care facilitează căutarea mai ușoară a datelor. Există indecși simpli pe un câmp din tabelă respectiv indecși compuși definiți pe mai multe câmpuri din tabelă. De exemplu într-o tabelă care conține date despre firme, se poate defini un index pe baza câmpului CUI (cod unic de identificare), index care va ajuta la regăsirea eficientă a unei firme pe baza acestui câmp.

Pe o tabelă putem defini indecși unici, care nu permit introducerea a două înregistrări cu aceeași valoare pentru câmpurile din index. Pentru câmpurile care definesc o relație cu o altă tabelă se va defini un index pe acel câmp pentru a optimiza operațiunile de căutare.

### Câmpuri

Câmpurile au un tip de date, o dimensiune și anumite reguli de consistență.

Mai jos am enumerat o parte din tipurile de date care pot fi definite in Access:

Text – Short/Long;

Number – Byte, Integer, Long integer, Single, Double, Decimal;

Date/Time – General Date, Long Date/Time, Medium Date/Time, Short Date/Time;

Yes/No – reprezintă valorile adevărat/fals;

AutoNumber – este un număr care se auto-incrementeză începând de la 1. De obicei acesta se folosește la cheia primară.

Pentru un câmp se pot seta urmatoarele proprietăți:

câmpul este sau nu obligatoriu – adica permite sau nu valoare NULL;

Valoare implicită (conform cu tipul câmpului)

* La un câmp numeric pot pune o valoare 0;
* La un câmp DateTime pot pune funcția Now (Data și timpul curent)

Reguli de validare

* La un câmp numeric pot pune “> 0”, adică valoare trebuie sa fie pozitivă;
* La un câmp DateTime pot pune “ > Now”, adică valoare introdusă trebuie sa fie mai mare decât timpul actual;

Mesajul de validare este mesajul afișat când nu se respectă regula de mai sus.

### Relații de tabele

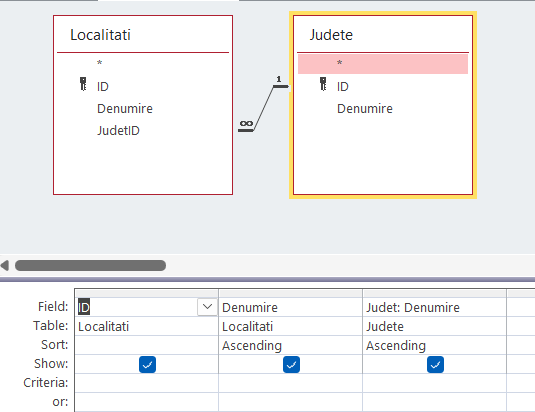
Deși o tabelă conține date structurate referitoare la un anumit subiect, poate conține de asemenea și date legate de altă tabelă. De exemplu o tabelă care conține date despre persoane poate să conțină un câmp de legătură spre o tabelă de localități.

"O relație este o conexiune logică între două tabele care specifică câmpurile pe care tabelele le au în comun"**[[1]](#footnote-1)**.

## Interogări

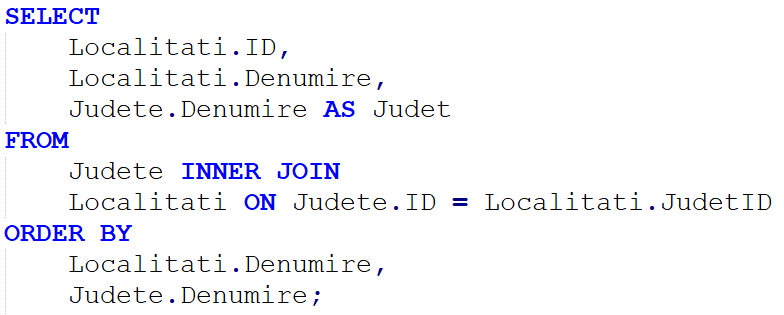
În marea majoritate a cazurilor, interogările sunt folosite pentru a regăsi date specifice din tabele. Acestea permit utilizarea unor criterii pentru a filtra datele și a le utiliza într-un mod mai eficient.

Interogările în Access sunt de două tipuri: selecție și acțiune. În proiect am folosit doare interogări de selecție. Acestea preiau datele și le furnizează intr-un rezultat pentru a putea fi ultilizat. Rezultatul poate fi vizualizat pe ecran, copiat în memorie, tipărit la imprimantă sau utilizat ca sursă de date pentru un o formă sau raport.



De exemplu, interogarea de mai sus (reprezentată în format grafic) furnizează ca și rezultat localitățiile și județele corespunzătoare acestora. Ca și coloane de ieșire avem cheia primară din tabela de localități, denumirea și denumirea județului.

Interogarea poate fi specificată inclusiv în forma de text:



O interogare poate conține următoarele secțiuni:

* Zona de „SELECT” unde se specifică câmpurile care apar în rezultat;
* Zona de „FROM” unde se specifică tabelele și legăturile dintre ele;
* Zona de “WHERE” unde se specifică criteriile de filtrare;
* Zona de “ORDER BY” unde se specifică modalitatea de ordonare a rezultatelor;

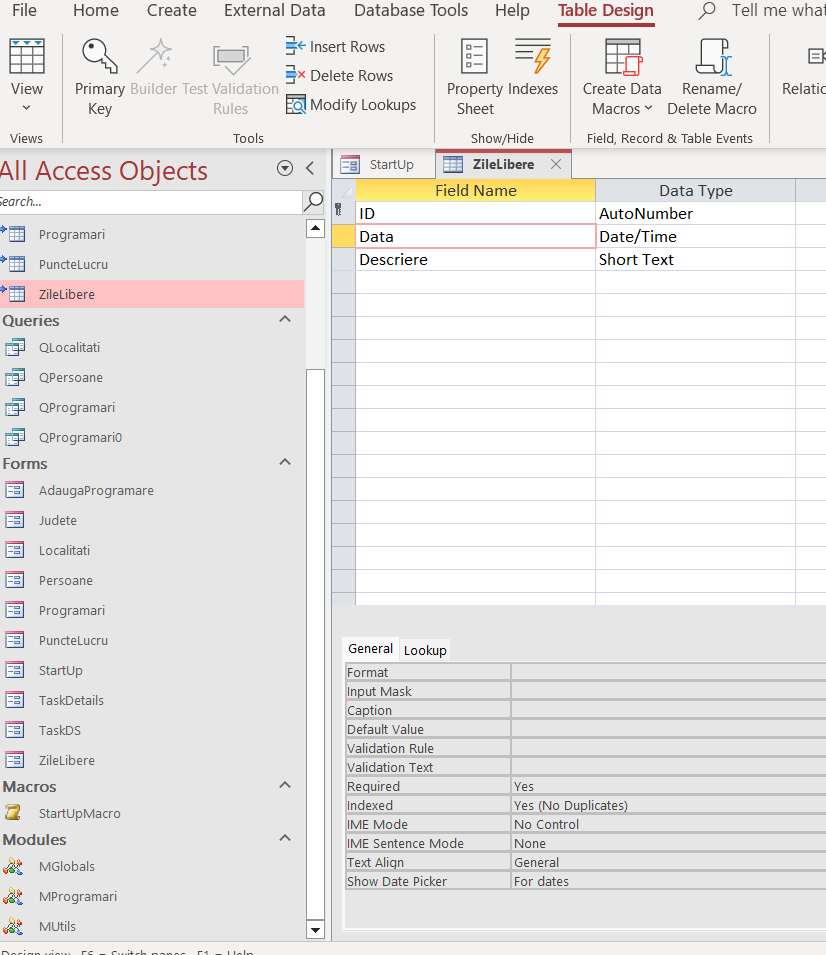
Într-o interogare trebuie să specificăm câmpurile și tabelele de ieșire precum și legăturile între tabele. Nu pot exista două câmpuri cu aceeași denumire, dar există posibilitatea unui alias. Revenind la exemplul de mai sus, există o coloană “Județ” care este aliasul coloanei “Denumire” din tabela Județe.

În Access interogările pot fi folosite ca și sursă de date pentru formulare, rapoarte, alte interogări și respectiv ca sursă de date pentru listbox-uri.

# Descriere Microsoft Access

Mediu Microsoft Access permite definirea de tabele, interogări, forme, macrouri și module.

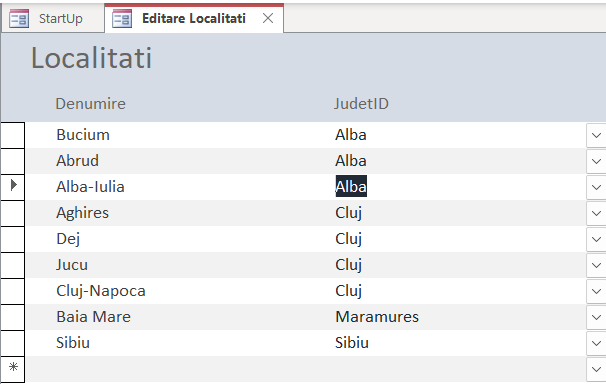
În partea stangă se poate observa lista lor grupate pe categorii iar în partea de sus se pot observa diverse unelte cu ajutorul cărora putem edita, adăuga sau elimina diverse lucruri în aplicație. În partea de jos de multe ori se pot folosi modalități de editare a formatării.



## Formulare

Formularele permit utilizarea unei interfațe în care pot fi introduse date. Se pot folosi butoane de comandă și alte controale pentru a efectua diferite cerințe. Folosirea formelor nu este obligatorie însă este preferabilă folosirea lor pentru o vizualizare mai ușoară a datelor.

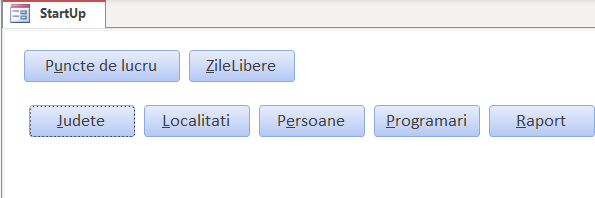




Butoanele de comandă se pot programa să determine ce date apar pe formular, să deschidă alte formulare, să șteargă elemente etc. Formularele permit în special modul în care alți utilizatori interacționează cu baza de date. Acest lucru contribuie la protejarea datelor și asigură introducerea corectă a datelor.

## Macrocomenzi

Macrocomenzile pot fi considerate un fel de limbaj de programare simplificat care este utilizat pentru a adăuga funcționalitate bazei de date. Acestea conțin acțiuni care efectuează activități, de exemplu deschiderea unui raport, rularea unei interogări, închiderea bazei de date sau în cazul acesta un panou de comandă care trimite către toate formele pentru ca datele sa poată fi editate.

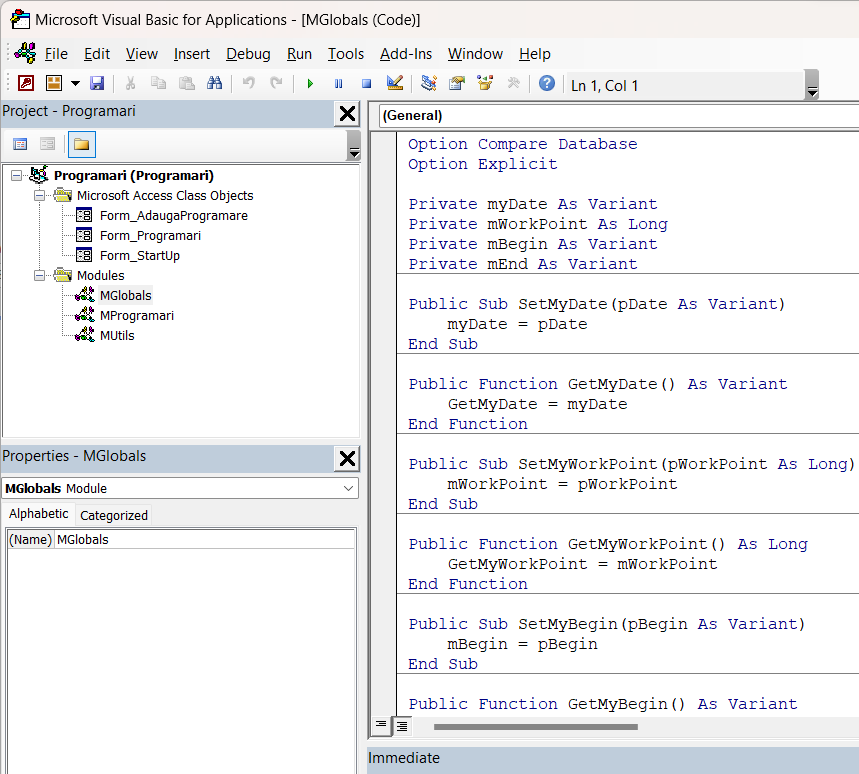


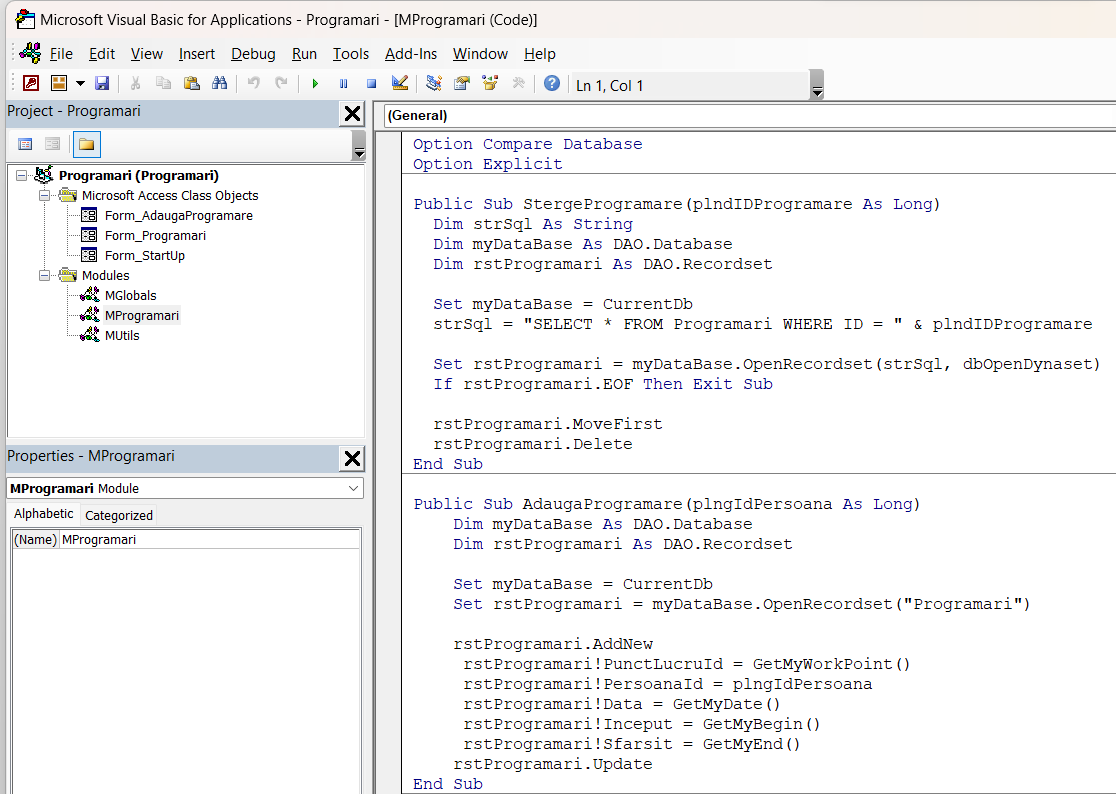
Acest macro se deschide întodeauna la rularea aplicației.

## Module

Modulele sunt obiecte care se pot utiliza pentru diverse funcționalități ale bazei de date. Modulele sunt scrise în Visual Basic for Applications (VBA). Un modul este o mulțime de declarații, instrucțiuni și proceduri stocate împreună. Acesta poate fi modul de clasă sau standard. Modulele de clasă sunt atașate la formulare sau rapoarte și conțin proceduri specifice formularului/raportului la care sunt atașate. Modulele standard conțin proceduri standard care nu sunt atașate altui obiect.

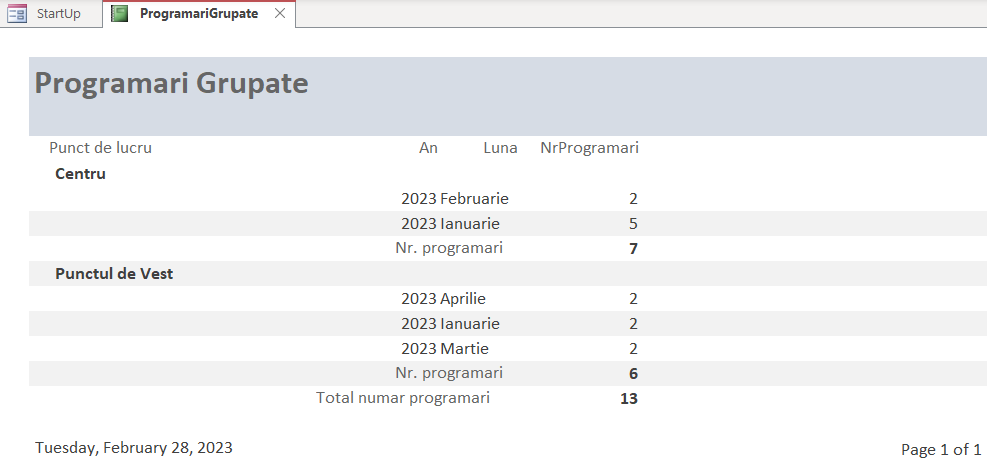






## Rapoarte

Rapoartele sunt folosite pentru a rezuma, formata și prezenta date. Un raport poate fi rulat oricând și va reflecta întodeauna datele din baza de date folosită, dar pot fi vizualizate și pe ecran sau pot fi exportate în alt program.

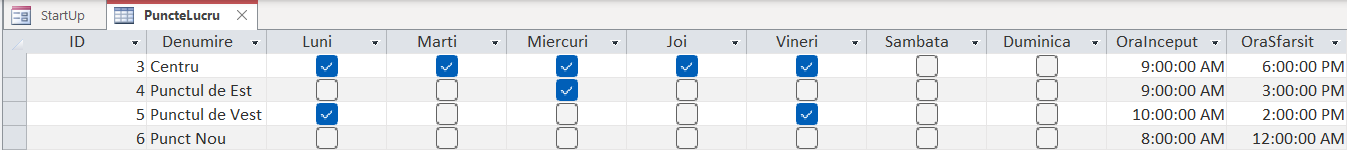


# Descrierea tabelelor și a interogărilor folosite

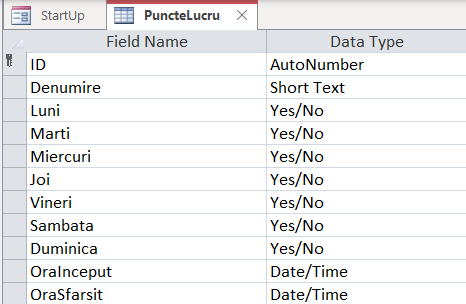
## Tabele

### Tabela PuncteLucru

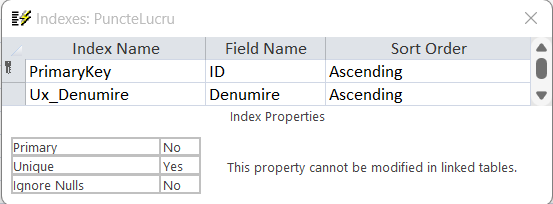
Tabela PucteLucru reține numele punctelor de lucru, zilele în care se pot face programări și intervalul te timp în care se pot face programările.



Aceasta folosește câmpurile Id (de tip AutoNumber de lungime Long Integer), Denumire (de tip Short Text de lungime 50), Luni/Marti/Miercuri/Joi/Vineri/Sambata/Duminica (de tip Yes/No de format Yes/No) și OraInceput/OraSfarsit (de tip Date/Time de format Long Time).

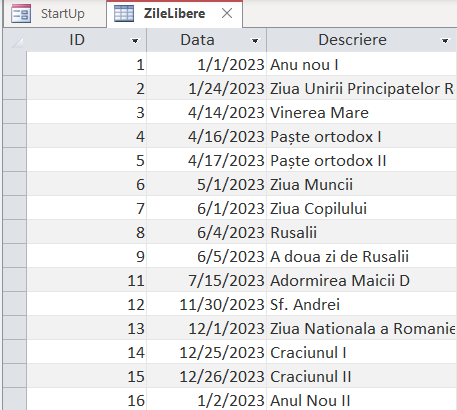


Indexul unic definit este Ux\_Denumire definit pentru câmpul Denumire.

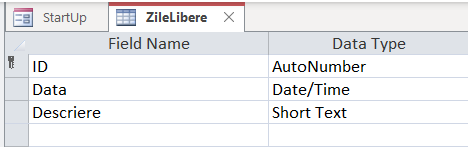


### Tabela ZileLibere

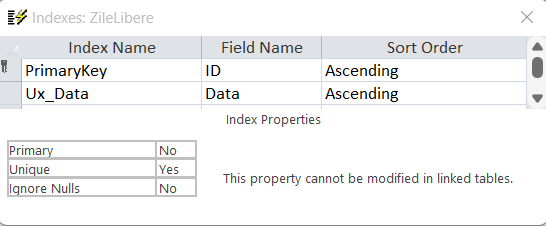
Tabela ZileLibere reține datele tuturor zilelor libere naționale dintr-un an și denumirea lor. Aceste date trebuie actualizate în fiecare an deoarece unele se schimbă de la an la an.



Aceasta se folosește de câmpurile ID (de tip Autoumber de lungime Long Integer), Data (de tip Date/Time) și Descriere de tip (Short Text de lungime 50).



Indexul unic definit este Ux\_Data definit pentru câmpul Data.

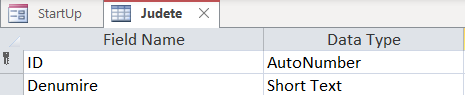


### Tabela Judete

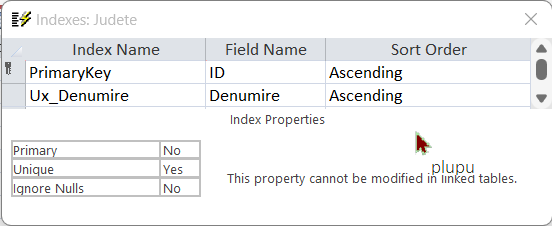
Tabela Judete reține numele tuturor județelor din care fac parte persoanele pentru care se fac programări.



Aceasta folosește câmpurile ID (de tip AutoNumber de lungime Long Integer) și Denumire (de tip Short Text de lungime 50).

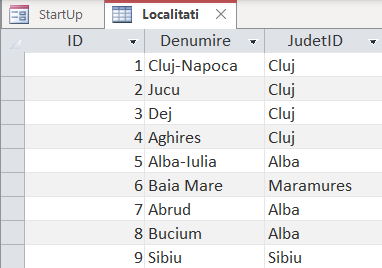


Pe tabelă este definit indexul unic Ux\_Denumire care nu permite introducerea a doua Județe cu același nume.

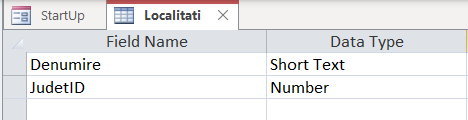


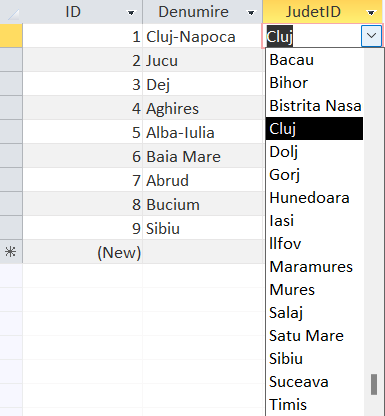
### Tabela Localitati

Tabela Localitati reține numele tuturor localităților corespunzător județelor acestora.

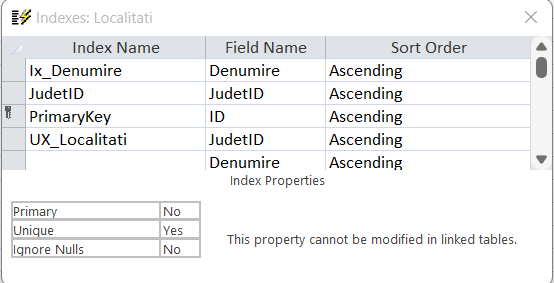


Aceasta folosește câmpurile ID (de tip AutoNumber de lungime Long Integer), Denumire (de tip Short Text de lungime 50) și JudetID (de tip Number de lungime Long Integer). La adăugarea unui nume de localitate selectezi din dreapta denumirii județul din care face parte.



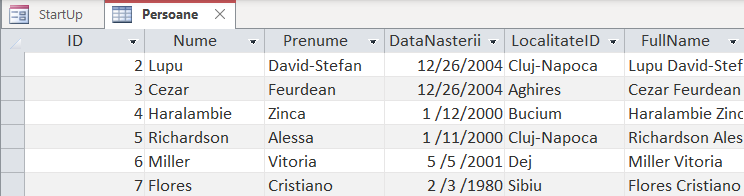


Pe tabelă este definit indexul unic UX\_Localitati, definit ca și index compus pe câmpurile de județ și denumire. Acest index nu permite adăugarea a doua localități cu același nume în cadrul aceluiași județ.

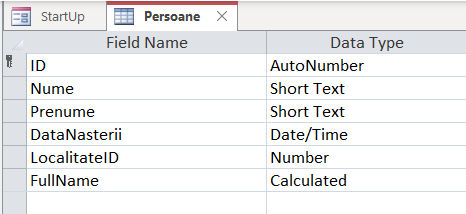


### Tabela Persoane

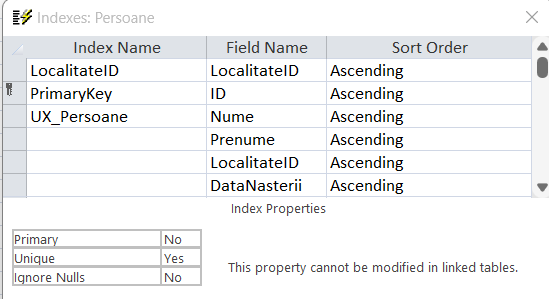
Tabela Persoane reține numele, prenumele, data nașterii și localitatea în care are domiciliu persoana respectivă.



Aceasta folosește câmpurile ID (de tip AutoNumber de lungime Long Integer), Nume/Prenume (de tip Short Text de lungime 50), DataNasterii (de tip Date/Time), LocalitateID (de tip Number de lungime Long Integer și FullName (de tip Calculated definită de expresia [Nume] & " " & [Prenume]).

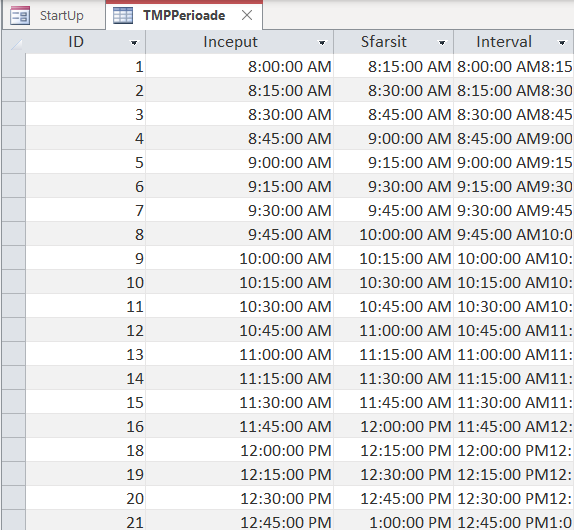


Indexul unic pe tabelă este UX\_Persoane definit pentru câmpurile Nume,Prenume,LocalitateID și DataNasterii.

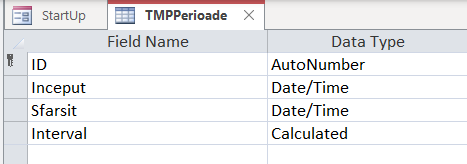


### Tabela TMPPerioade

Tabela TMPPerioade definește intervale de timp pentru care se pot face programări între orele de lucru de la 8.00 AM până la 8.00 PM, de câte 15 minute.

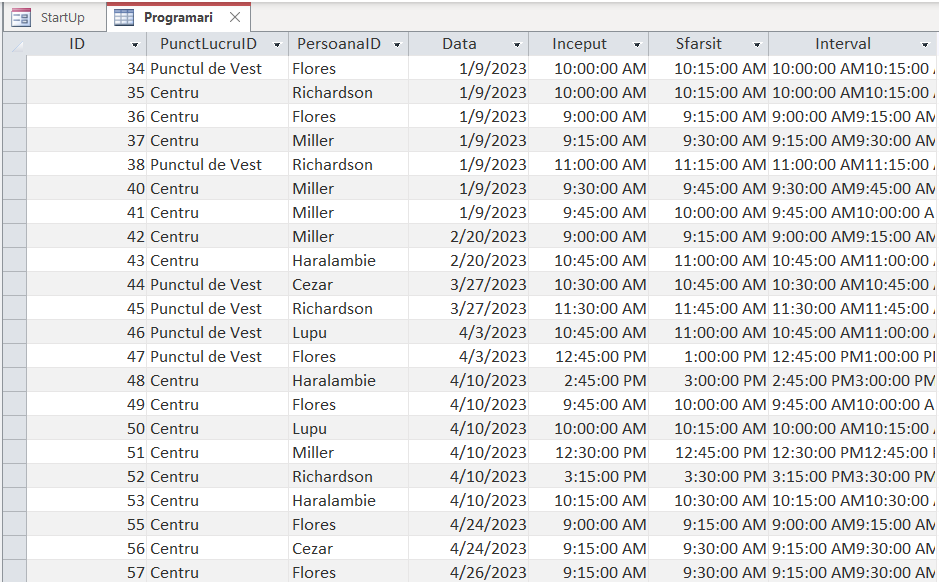


Aceasta folosește câmpurile ID (de tip Auto Number de lungime Long Integer), Inceput/Sfarsit (de tip Date/Time de format Long Time) și Interval (de tip Calculated definită de expresia [Inceput] & [Sfarsit]).

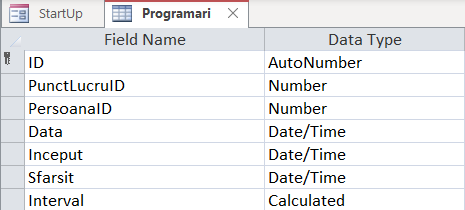


### Tabela Programari

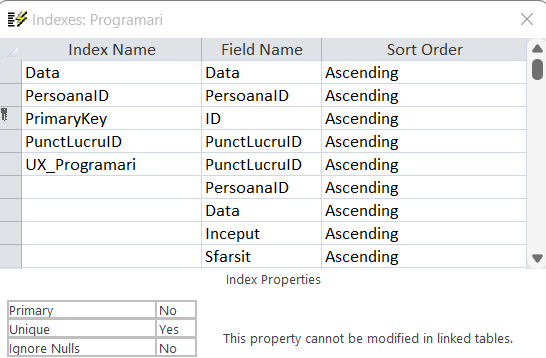
Tabela Programări reține toate programările adăugate și se reactualizează de fiecare dată când este adăugată o programare nouă.



Aceasta folosește câmpurile ID (de tip AutoNumber de lungime Long Integer), PunctLucruID/PersoanaID (de tip Number de lungime Long Integer), Data (de tip Date/Time de format General Date), Inceput/Sfarsit (de tip Date/Time de format Long Time) și Interval (de tip Calculated definit de expresia [Inceput] & [Sfarsit]).



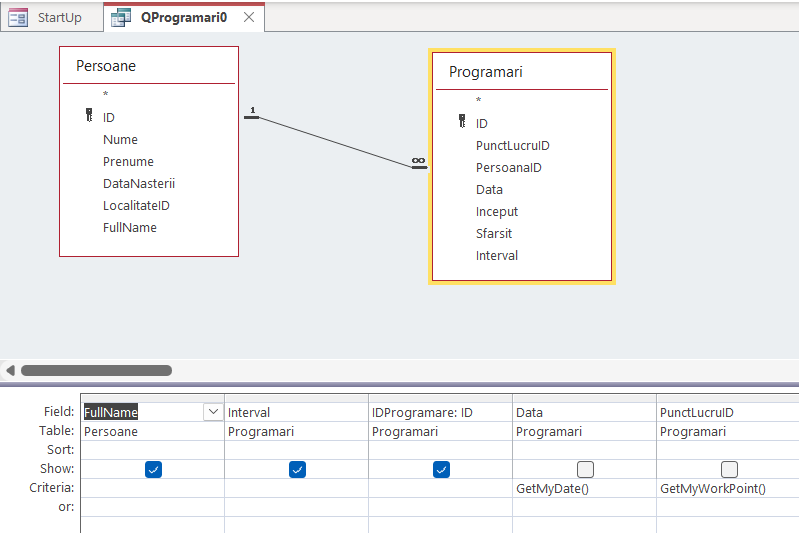
Indexul unic pe tabelă este UX\_Programari definit pentru câmpurile PunctLucruID, PersoanaID, Data, Inceput și Sfarsit



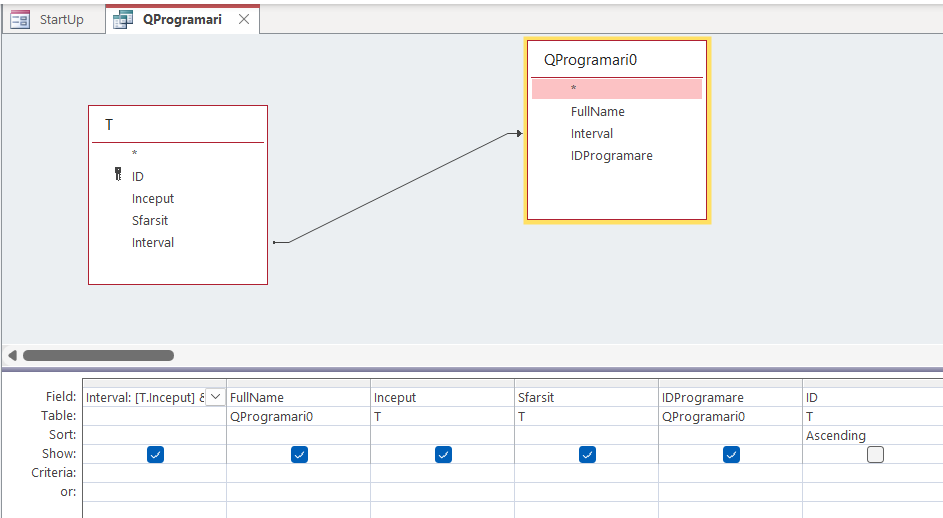
## 4.2. Interogări

### 4.2.1.Interogarea QProgramări/Qprogramări0

Interogarea QProgramari0 afișează date din tabela de persoane și tabela de programări. Se selectează câmpurile FullName, Interval, IDProgramare. La criterii se filtrează data cu funcția GetMyDate() și punctul de lucru cu funcția GetMyWorkingPoint(). Setarea valorilor din spatele acestor funcții se face din formularul Programari.

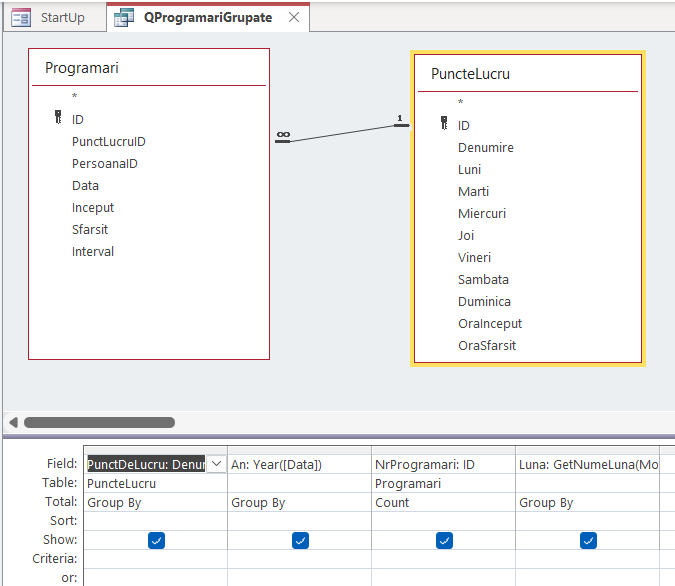


Interogarea QProgramări iși ia datele din tabelele TMPPerioade alias T și QProgramări0. Se selectează câmpurile T.Interval, FullName, T.Inceput, T.Sfarsit, IDProgramare.



### 4.2.2. Interogarea QProgramariGrupate

Interogarea QprogramariGrupate este folosită în generarea datelor pentru raportul ‘ProgramariGrupate’ unde se afișează numarul de programări grupat după lună și puncte de lucru. Intergoarea selectează câmpurile PunctDeLucru, An, Luna și NrProgramări.



Pentru a folosi numele lunilor în română am folosit câmpul GetNumeLuna care folosește funcția cu aceeași denumire pentru a-și furniza valorile.

Public Function GetNumeLuna(luna As Integer) As String

GetNumeLuna = "???"

If luna = 1 Then

GetNumeLuna = "Ianuarie"

Else

If luna = 2 Then

GetNumeLuna = "Februarie"

Else

If luna = 3 Then

GetNumeLuna = "Martie"

Else

If luna = 4 Then

GetNumeLuna = "Aprilie"

Else

If luna = 5 Then

GetNumeLuna = "Mai"

Else

If luna = 6 Then

GetNumeLuna = "Iunie"

Else

If luna = 7 Then

GetNumeLuna = "Iulie"

Else

If luna = 8 Then

GetNumeLuna = "August"

Else

If luna = 9 Then

GetNumeLuna = "Septembrie"

Else

If luna = 10 Then

GetNumeLuna = "Octombrie"

Else

If luna = 11 Then

GetNumeLuna = "Noiembrie"

Else

If luna = 12 Then

GetNumeLuna = "Decembrie"

End If

End If

End If

End If

End If

End If

End If

End If

End If

End If

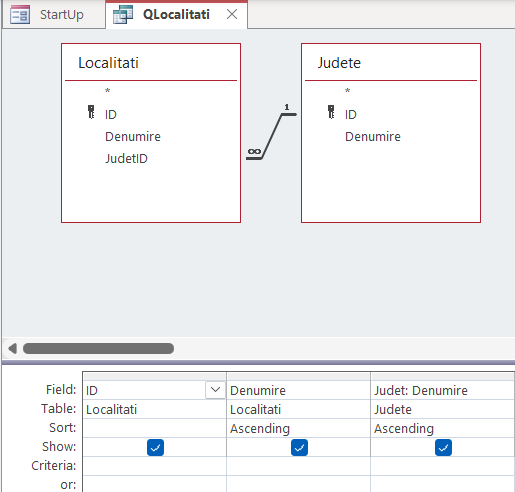
End If

End If

End Function

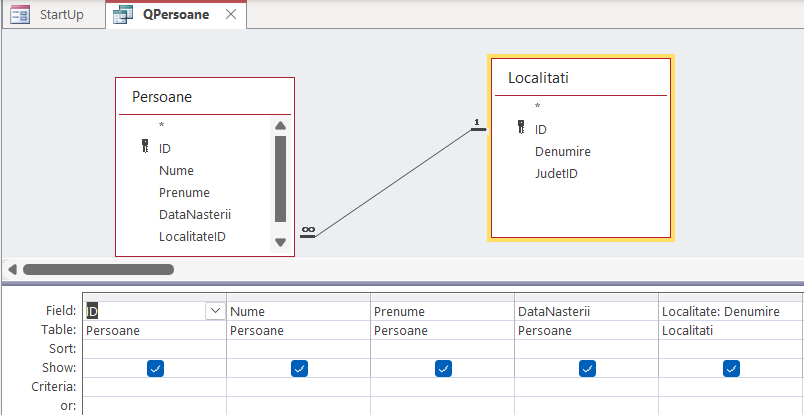
### 4.2.3. Interogarea QLocalități

Interogarea QLocalități afișează date din tabela de localități și județe. Aceasta face legătura între cele două, unde include doar linii în care toate valorile sunt egale. Se folosesc câmpurile ID, Denumire și Judet.



### 4.2.4. Interogarea QPersoane

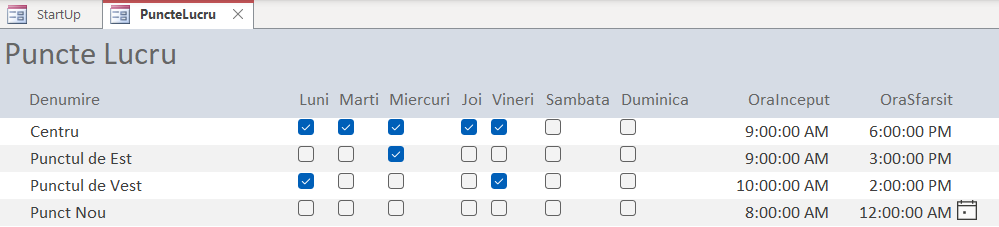
Interogarea QPersoane afișează date din tabela de persoane și tabela de localități. Aceasta face legătura între cele două, unde include doar linii în care toate valorile sunt egale. Se folosesc câmpurile ID, Nume, Prenume, DataNasterii și Localitate.



# Descrierea formelor

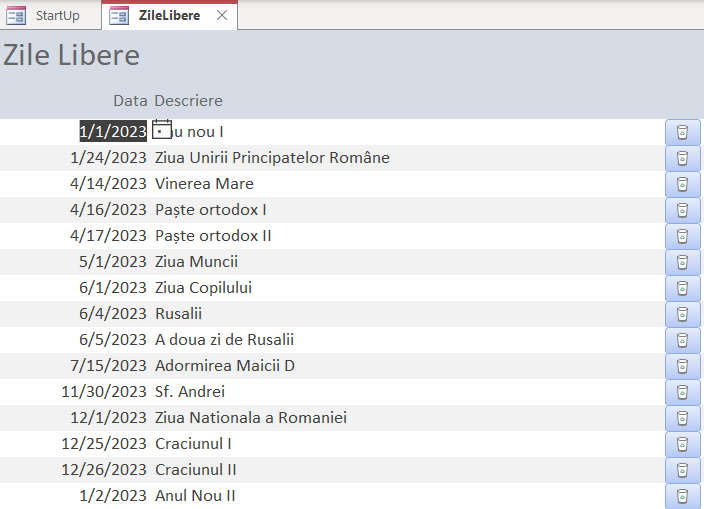
## Formular “PuncteLucru”

Prin acest formular se permite editarea punctelor de lucru, nu se permite ștergerea, record source-ul este tabela “PuncteLucru”. Punctele de lucru au o denumire și un program de lucru. Pentru fiecare se vor specifica zilele în care are activitate respectiv ora de început și cea de sfârșit.



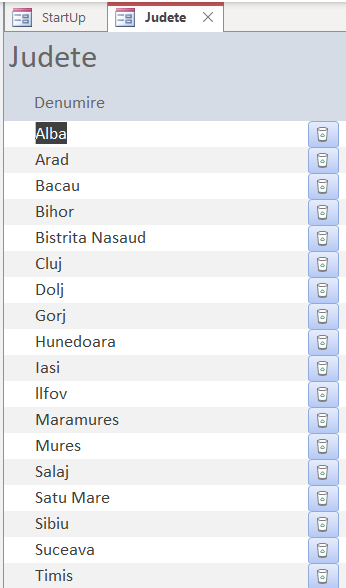
## Formular “ZileLibere”

Prin acest formular se permite introducerea și ștergerea zilelor libere dintr-un an, în aceste zile nu se vor putea face prgramări, record source-ul este tabela “ZileLibere”. Apasând butonul din dreapta se permite ștergerea înregistrării selectate. Tabela este editată în avans înainte de a face orice programare pe anul respectiv.



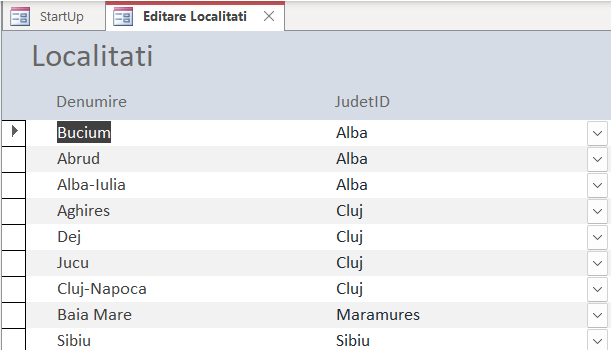
## Formular “Judete”

Prin acest formular se permite editarea unui catalog de județe, record source-ul lui este tabela “Judete”. Apasând butonul din dreapta se permite ștergerea înregistrării selectate, ștergerea județului nu este permisă dacă este folosit în tabela de localități (cu care are relație).



## Formular “Localitati”

Prin acest formular se permite adăugarea și editarea unor localitati, record source-ul este tabela “Localități”. În partea stângă a formularului se adaugă numele localității, iar în partea dreaptă este o săgeată care se va folosi pentru a selecta județul din care face parte localitatea.



## Formular “Persoane”

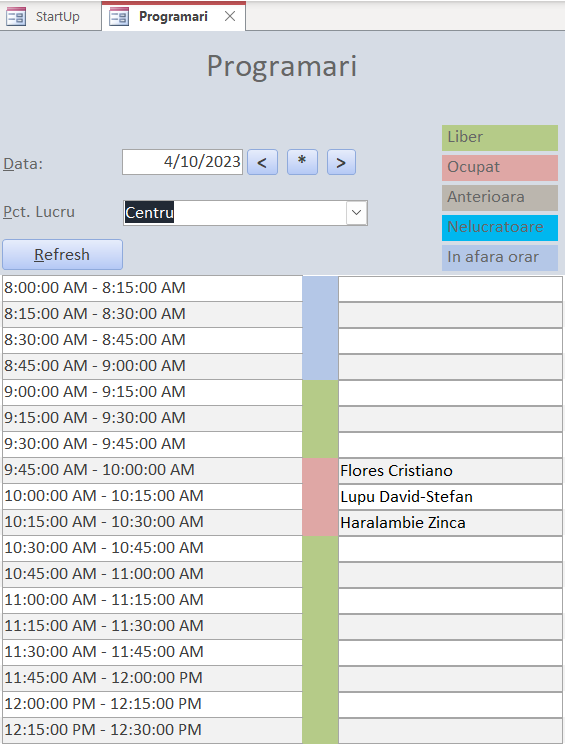
Prin acest formular se permite adăugarea și editarea persoanelor care sunt programate, înclusiv orașul de proveniență și data nașterii, record source-ul este tabela “Persoane”. Primele doua coloane sunt numele, urmat de prenume, iar ultimele două sunt data nașterii și o săgeată cu care se adauga localitatea în care se află domiciliul persoanei respective.



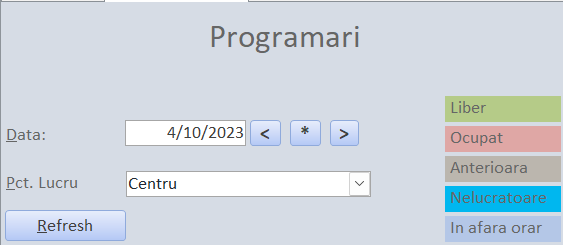
## Formular “Programari”

Prin acest formular se permite adăugarea și editarea programărilor în tabelul de programări unde se pot observa numele persoanelor, data programării, centrul programării și orele programării.

Sursa de date a formularului se bazează pe interogarea QProgramări și afișează datele din ziua specificată în controlul txtDate respectiv cmbPunctLucru.

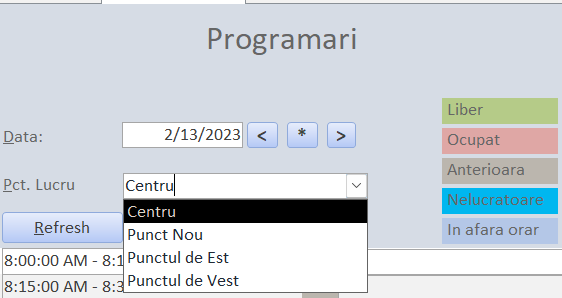


În partea de sus a formularului se pot selecta zilele și punctele de lucru în care se pot face programări.



Butonul cu simbolul “<” va schimba data (prin schimbarea valorii controlului txtDate) cu o zi anterioară. Butonul cu simbolul “\*” sare la data zilei curente. Butonul cu simbolul “>” va schimba data (prin schimbarea valorii controlului txtDate) cu o zi următoare. De asemenea este posibilă selectarea dății de dinaintea acestor caractere și se poate introduce data dorită de la tastatură.

Pentru punctele de lucru se poate selecta dintr-o listă deja existentă.



La schimbarea valorii controlului txtDate sau cmbPunctLucru respectiv apăsarea butonului Refresh(cmdAplica) se va apela funcția Aplica.

Private Sub cmbPunctLucru\_AfterUpdate()

Call Aplica

End Sub

Private Sub cmdAplica\_Click()

Call Aplica

End Sub

Private Sub txtDate\_AfterUpdate()

Call Aplica

End Sub

### Descriere procedura “Aplica”:

Private Sub Aplica()

On Error GoTo Err

SetMyDate (CVDateErr(Me.txtDate))

SetMyWorkPoint (CLngErr(Me.cmbPunctLucru.Value))

mIsFreeDate = IsFreeDay(CVDateErr(Me.txtDate.Value))

Call LoadInterval

Me.Requery

Me.Recalc

Exit Sub

Err:

MsgBox Error

End Sub

Se setează variabilele pentru dată și punctul de lucru care sunt folosite în interogarea QProgramări.

Public Sub SetMyDate(pDate As Variant)

myDate = pDate

End Sub

Public Sub SetMyWorkPoint(pWorkPoint As Long)

mWorkPoint = pWorkPoint

End Sub

În variabila mIsFreeDay setăm daca data curentă este zi liberă sau nu, și folosim funcția IsFreeDay.

Public Function IsFreeDay(pDate As Variant) As Boolean

Dim strSql As String

Dim myDataBase As DAO.Database

Dim rstZileLibere As DAO.Recordset

Set myDataBase = CurrentDb

strSql = "SELECT \* FROM ZileLibere WHERE Data = " + GetFilterDate(pDate)

Set rstZileLibere = myDataBase.OpenRecordset(strSql, dbReadOnly)

If rstZileLibere.EOF Then

IsFreeDay = False

Else

IsFreeDay = True

End If

End Function

Funcția se uită în tabela de zile libere și verifică daca ziua dată ca și parametru există sau nu. Daca există returnează true sau fals în caz contrar.

Funcția LoadInterval pregătește în variabilele private mStartSchedule, mEndSchedule respectiv mIsDayInSchedule în funcție de setările punctului de lucru selectat.

Private Sub LoadInterval()

Dim daysInweek(7) As Boolean

Dim dayInWeek As Integer

Dim strSql As String

Dim myDataBase As DAO.Database

Dim RS As DAO.Recordset

Set myDataBase = CurrentDb

strSql = "SELECT \* FROM PuncteLucru WHERE ID = " + CStrErr(GetMyWorkPoint())

Set RS = myDataBase.OpenRecordset(strSql, dbReadOnly)

RS.MoveFirst

daysInweek(1) = RS!Luni

daysInweek(2) = RS!Marti

daysInweek(3) = RS!Miercuri

daysInweek(4) = RS!Joi

daysInweek(5) = RS!Vineri

daysInweek(6) = RS!Sambata

daysInweek(7) = RS!Duminica

mStartSchedule = RS!OraInceput

mEndSchedule = RS!OraSfarsit

dayInWeek = Weekday(CVDateErr(Me.txtDate.Value), vbMonday)

mIsDayInSchedule = daysInweek(dayInWeek)

End Sub

La apelul de Me.Recalc respectiv Me.Requery se apelează de către sistem Detail\_Paint(formatarea detaliului formularului). În această funcție se va seta culoarea de la controlul lblOcupat.

Private Sub Detail\_Paint()

On Error GoTo Err

If IsPreviousDate(CVDateErr(Me.txtDate.Value)) Then

Me.lblOcupat.BackColor = Me.lblPrevious.BackColor

Exit Sub

End If

If mIsFreeDate = True Then

Me.lblOcupat.BackColor = Me.lblFreeDay.BackColor

Exit Sub

End If

If mIsDayInSchedule = False Then

Me.lblOcupat.BackColor = Me.lblSchedule.BackColor

Exit Sub

End If

If (IsInScheduleTime() = False) Then

Me.lblOcupat.BackColor = Me.lblSchedule.BackColor

Exit Sub

End If

If IsNull(Me.txtFullName.Value) Then

Me.lblOcupat.BackColor = Me.lblFree.BackColor

Else

Me.lblOcupat.BackColor = Me.lblBusy.BackColor

End If

Exit Sub

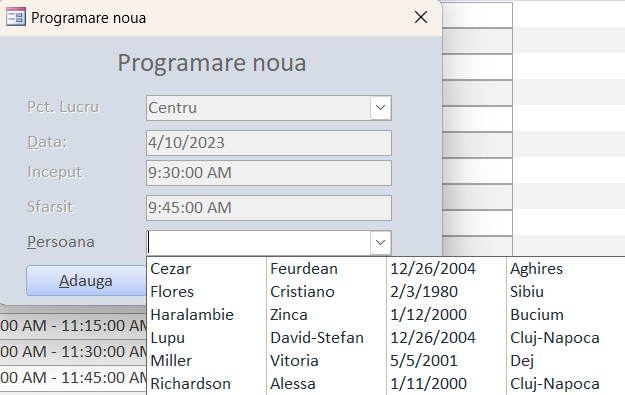
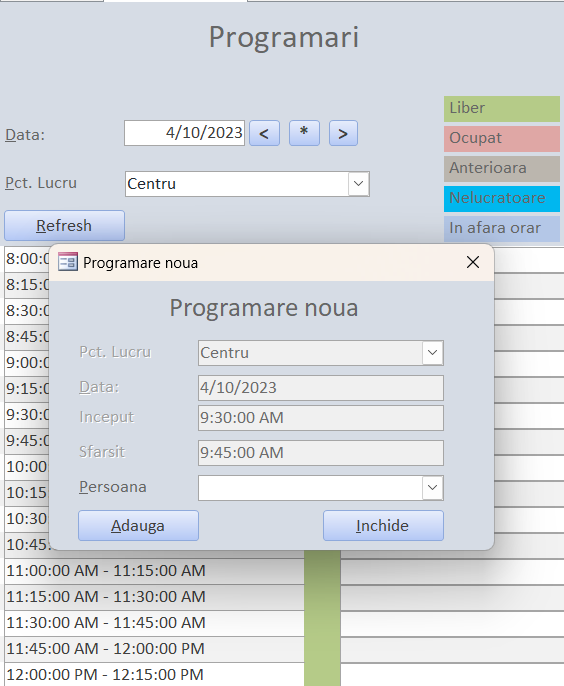
Err:

MsgBox Error

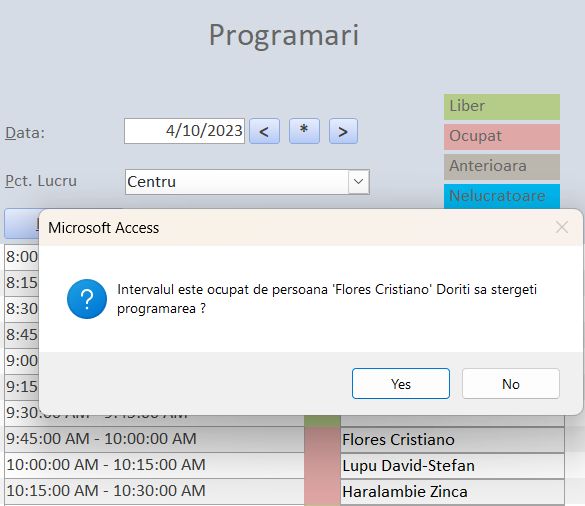
End Sub

### Adăugarea unei noi programări

Pentru a adăuga o programare se dă click de doua ori pe cadranul din stânga în dreapta intervalului de timp. După ce apare un nou formular, AdaugaProgramare, se alege persoana pentru care se face programarea. Programările se pot face doar în zilele și orele stabilite, marcate cu culoarea verde.



Apăsând dublu click pe o zona deja ocupata de cineva va rezulta în apariția unui mesaj.



Dacă utilizatorul apasă butonul “Yes” se apelează procedura StergereProgramare.

Public Sub StergeProgramare(plndIDProgramare As Long)

Dim strSql As String

Dim myDataBase As DAO.Database

Dim rstProgramari As DAO.Recordset

Set myDataBase = CurrentDb

strSql = "SELECT \* FROM Programari WHERE ID = " & plndIDProgramare

Set rstProgramari = myDataBase.OpenRecordset(strSql, dbOpenDynaset)

If rstProgramari.EOF Then Exit Sub

rstProgramari.MoveFirst

rstProgramari.Delete

End Sub

În procedura de mai sus se va șterge programarea care are ID-ul trimis ca și parametru în variabila plndIDProgramare.

Codul care se apelează la dublu click este în procedura txtInterval\_DblClick.

Private Sub txtInterval\_DblClick(Cancel As Integer)

On Error GoTo Err

If (IsPreviousDate(CVDateErr(Me.txtDate.Value)) Or mIsFreeDate = True Or mIsDayInSchedule = False Or IsInScheduleTime() = False) Then

Exit Sub

End If

If IsNull(Me.txtFullName.Value) = False Then

Dim result

result = MsgBox("Intervalul este ocupat de persoana '" & Me.txtFullName.Value & "' Doriti sa stergeti programarea ?", vbQuestion + vbYesNo)

If (result = vbYes) Then

Call StergeProgramare(CLngErr(Me.txtIdProgramare.Value))

Call Aplica

End If

Exit Sub

End If

SetMyBegin (Me.txtInceput.Value)

SetMyEnd (Me.txtSfarsit.Value)

DoCmd.OpenForm "AdaugaProgramare", acNormal, , , , acDialog

Call Aplica

Exit Sub

Err:

MsgBox Error

End Sub

La apăsarea butonului “Adauga” (cmdAdauga) din formularul AdaugaProgramare se apelează funcția cmdAdauga\_Click.

Private Sub cmdAdauga\_Click()

On Error GoTo Err

If (IsNull(Me.cboPersoana)) Then

MsgBox "Alegeti persoana!", vbInformation

Exit Sub

End If

If ExistsAlreadyASchedule() Then

MsgBox "Exista deja o programare in acelas loc și acceasi perioada!", vbInformation

Exit Sub

End If

If (IsSamePersonIn2Places(CLngErr(Me.cboPersoana.Value))) Then

MsgBox "Aceeasi persoana in 2 locuri in aceeasi perioada!", vbInformation

Exit Sub

End If

Call AdaugaProgramare(CLngErr(Me.cboPersoana.Value))

MsgBox "Programarea a fost adaugata cu succes!", vbInformation

DoCmd.Close acForm, Me.Name

Exit Sub

Err:

MsgBox Error

End Sub

În această procedură, înainte de adăugarea efectivă, se vor efectua anumite validări:

* Dacă persoana a fost completată;
* Daca există deja o programare în același interval și același punct de lucru (ExistsAlreadySchedule);
* Daca aceeași persoană este în doua puncte diferite în același interval de timp (IsSamePersonIn2Places);

Funcția ExistsAlreadySchedule verifică, în tabela de programări, dacă există deja o înregistrare cu același interval de început respectiv sfârșit care se vrea a fi adăugat.

Public Function ExistsAlreadyASchedule()

Dim myDataBase As DAO.Database

Dim rstProgramari As DAO.Recordset

Dim strSql As String

Set myDataBase = CurrentDb

strSql = "SELECT \* FROM Programari WHERE Data = " + GetFilterDate(GetMyDate()) + \_

" AND Inceput = " + GetFilterTime(GetMyBegin()) + " AND Sfarsit = " + GetFilterTime(GetMyEnd()) + \_

" AND PunctLucruId = " + CStrErr(GetMyWorkPoint())

Set rstProgramari = myDataBase.OpenRecordset(strSql, dbReadOnly)

If rstProgramari.EOF Then

ExistsAlreadyASchedule = False

Else

ExistsAlreadyASchedule = True

End If

End Function

Funcția IsSamePersonIn2Places verifică în tabela de programări daca aceeași persoană, în același interval de timp apare în alt punct de lucru.

Public Function IsSamePersonIn2Places(plngIdPersoana As Long)

Dim myDataBase As DAO.Database

Dim rstProgramari As DAO.Recordset

Dim strSql As String

Set myDataBase = CurrentDb

strSql = "SELECT \* FROM Programari WHERE Data = " + GetFilterDate(GetMyDate()) + \_

" AND Inceput = " + GetFilterTime(GetMyBegin()) + " AND Sfarsit = " + GetFilterTime(GetMyEnd()) + \_

" AND PersoanaId = " + CStrErr(plngIdPersoana) + " AND PunctLucruId <> " + CStrErr(GetMyWorkPoint())

Set rstProgramari = myDataBase.OpenRecordset(strSql, dbReadOnly)

If rstProgramari.EOF Then

IsSamePersonIn2Places = False

Else

IsSamePersonIn2Places = True

End If

End Function

Aceste validări sunt necesare deoarece programul poate rula de pe mai multe stații de lucru și partajează același set de date.

Procedura AdaugaProgramare inserează o nouă înregistrare în tabela de programări.

Public Sub AdaugaProgramare(plngIdPersoana As Long)

Dim myDataBase As DAO.Database

Dim rstProgramari As DAO.Recordset

Set myDataBase = CurrentDb

Set rstProgramari = myDataBase.OpenRecordset("Programari")

rstProgramari.AddNew

rstProgramari!PunctLucruId = GetMyWorkPoint()

rstProgramari!PersoanaId = plngIdPersoana

rstProgramari!Data = GetMyDate()

rstProgramari!Inceput = GetMyBegin()

rstProgramari!Sfarsit = GetMyEnd()

rstProgramari.Update

End Sub

# Viitoare îmbunătățiri ale aplicației

* Adăugarea unui buton pentru ștergerea unei programări.
* Alte rapoarte pentru a vedea încărcarea pe punctele de lucru.
* Definirea medicilor și a specializărilor lor.
* Permiterea programărilor pentru o specializare anume.

# Bibliografie

1. [https://support.microsoft.com/ro-ro/office/noțiuni-elementare-despre-bazele-de-date-a849ac16-07c7-4a31-9948-3c8c94a7c204#:~:text=O%20bază%20de%20date%20este,sau%20o%20foaie%20de%20calcul.](https://support.microsoft.com/ro-ro/office/noțiuni-elementare-despre-bazele-de-date-a849ac16-07c7-4a31-9948-3c8c94a7c204%23:~:text=O%20bază%20de%20date%20este,sau%20o%20foaie%20de%20calcul.)
2. [https://www.accessrepairnrecovery.com/blog/access-best-practices-for-performance#a1](https://www.accessrepairnrecovery.com/blog/access-best-practices-for-performance%23a1)
3. [https://www.accessrepairnrecovery.com/blog/access-best-practices-for-performance#a3](https://www.accessrepairnrecovery.com/blog/access-best-practices-for-performance%23a3)
4. [https://www.accessrepairnrecovery.com/blog/access-best-practices-for-performance#a9](https://www.accessrepairnrecovery.com/blog/access-best-practices-for-performance%23a9)
5. [https://www.accessrepairnrecovery.com/blog/access-best-practices-for-performance#a7](https://www.accessrepairnrecovery.com/blog/access-best-practices-for-performance%23a7)
6. <https://support.microsoft.com/en-us/office/guide-to-table-relationships-30446197-4fbe-457b-b992-2f6fb812b58f>
7. <https://www.oreilly.com/library/view/fixing-access-annoyances/059600852X/ch04.html>

1. <https://support.microsoft.com/> - Relații despre tabele [↑](#footnote-ref-1)