Crearea structurii Bazei de Date

1. Pornirea SQL Server Management Studio-ului (SSMS)

Trebuie reținut că SQL Server-ul este serverul de baze de date, iar Management Studio-ul este un instrument pentru conectarea la server și pentru manipularea bazelor de date gestionate de către server. La un server se pot conecta mai mulți clienți, deci, implicit, ne putem conecta cu mai multe sesiuni de Management Studio.

După instalarea SQL Server-ului și a Management Studio-ului, în funcție de versiunea de Windows, veți găsi ceva similar cu imaginea de mai jos.

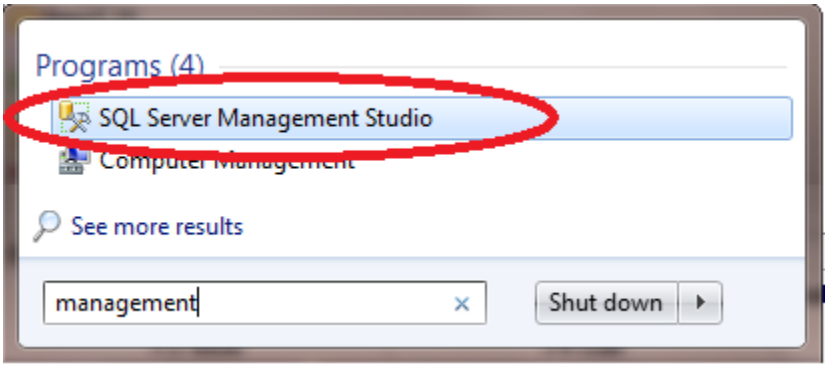

Fig. 1

Dacă instalarea a avut loc cu succes, la pornirea Management Studio-ului apare fereastra de conectare la server. Parametrii sunt cei din imagine. De aici trebuie să reținem numele serverului (MOBIL\SQLEXPRESS în cazul de față) și faptul că ne vom conecta cu Windows Authentification. Pentru pornire, clic pe Connect.

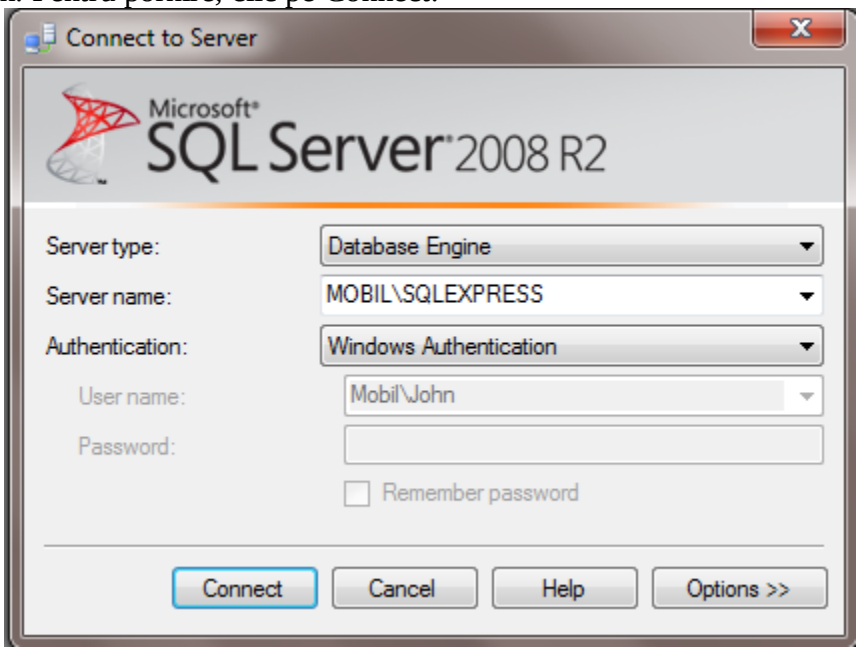

Fig. 2.

În cazul în care nu apare numele serverului, e posibil ca instalarea să fi avut probleme sau serviciul să nu fie pornit. Mai întâi vom verifica serviciul. Scriem services.msc în căsuța de căutare a Windows-ului, ca mai jos.

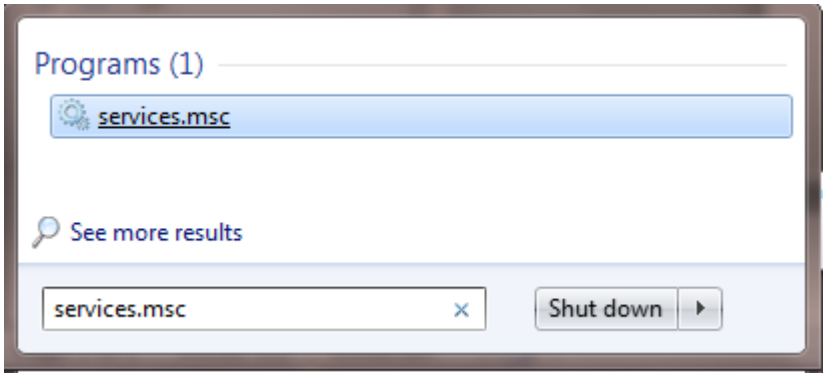

Fig. 3.

În fereastra Services căutăm serviciul care trebuie să fie pornit, așa cum se vede în imagine.

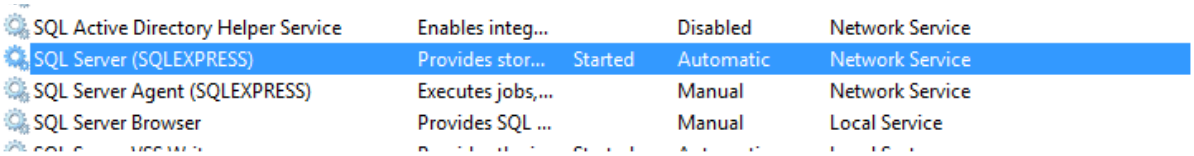

Fig. 4.

În cazul în care nu e pornit, cu dublu clic pe SQL Server întrăm în fereastra de Proprietăți, unde putem porni manual serviciul (serverul) sau putem să-l setăm să pornească automat la pornirea Windows-ului.

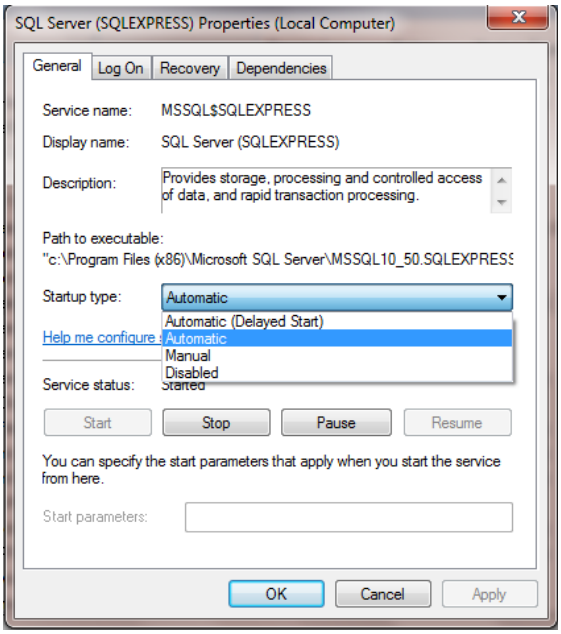

Fig. 5.

Dacă nu există serviciul SQL Server, aproape sigur instalarea a avut probleme și cel mai indicat este să dezinstalăm și apoi să reinstalăm SQL Server-ul și SSMS-ul.

Pentru instalare vezi https://simaioan.files.wordpress.com/2016/10/t1-instalare-sql-server.pdf

2. Crearea bazei de date.

Atunci când vorbim de o bază de date ne referim pe de o parte la crearea, citirea modificarea și ștergerea **structurii** sale, pe de altă parte, la manipularea **conținutului** său (inserare de date noi, modificarea, respectiv ștergerea datelor existente, precum și la extragerea datelor după anumite criterii într-un timp fezabil – operații **CRUD**).

Presupunând că ne-am conectat deja și că totul e în regulă, o să apară fereastra principală a SSMS-ului. În general, în partea stângă apar elementele serverului: Bazele de date, Securitate, etc.

O bază de date (cu toate obiectele sale) poate fi creată direct în SSMS în designer sau apelând la scrierea de cod SQL.

Varianta 1 – Creare BD în Designer

a. Crearea bazei de date

Dacă deschidem folderul Databases o să observăm că există baze de date sistem și bazele de date ale utilizatorilor.

Pentru exemplificare o să considerăm o bază de date care conține tabelele: Persoane, Telefoane, .

Pentru crearea bazei de date, facem clic dreapta pe *Databases*, apoi clic pe *New Database...* Pot să existe mici diferențe de la o versiune la alta de SQL Server.

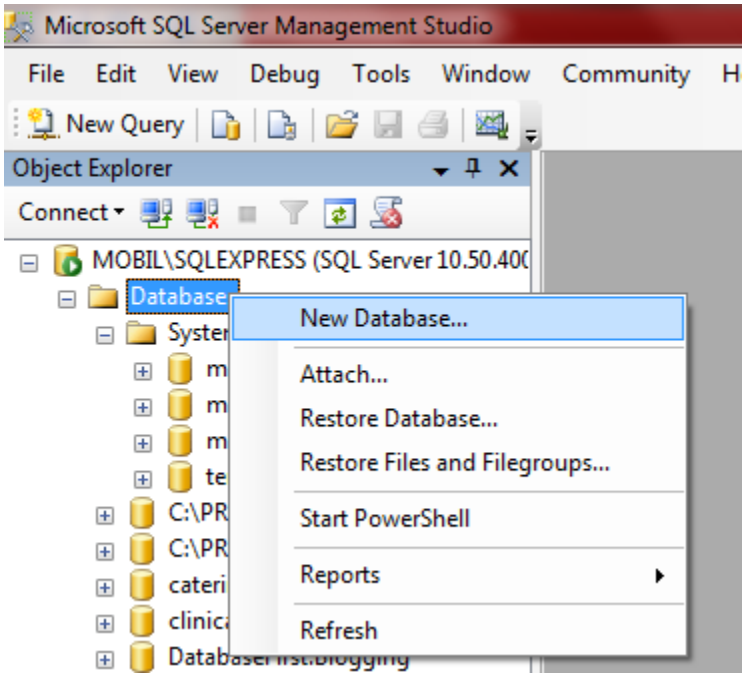

Fig. 6.

În fereastra New Database, scriem numele (denumirea) bazei de date (*TestStudenti* în exemplul de mai jos) și dăm clic pe *OK*. Ceilalți parametri rămân impliciți.

Trebuie reținut că se creează un fișier cu extensia mdf (ex: TestStudenti.mdf) și un fișier cu extensia ldf (ex: TestStudenti_log.ldf), care vor fi stocate în folderul DATA din SQL Server.

Exemplu: c:\Program Files (x86)\Microsoft SQL Server\MSSQL10_50.SQLEXPRESS\MSSQL\DATA

Datele vor fi stocate în fișierul mdf

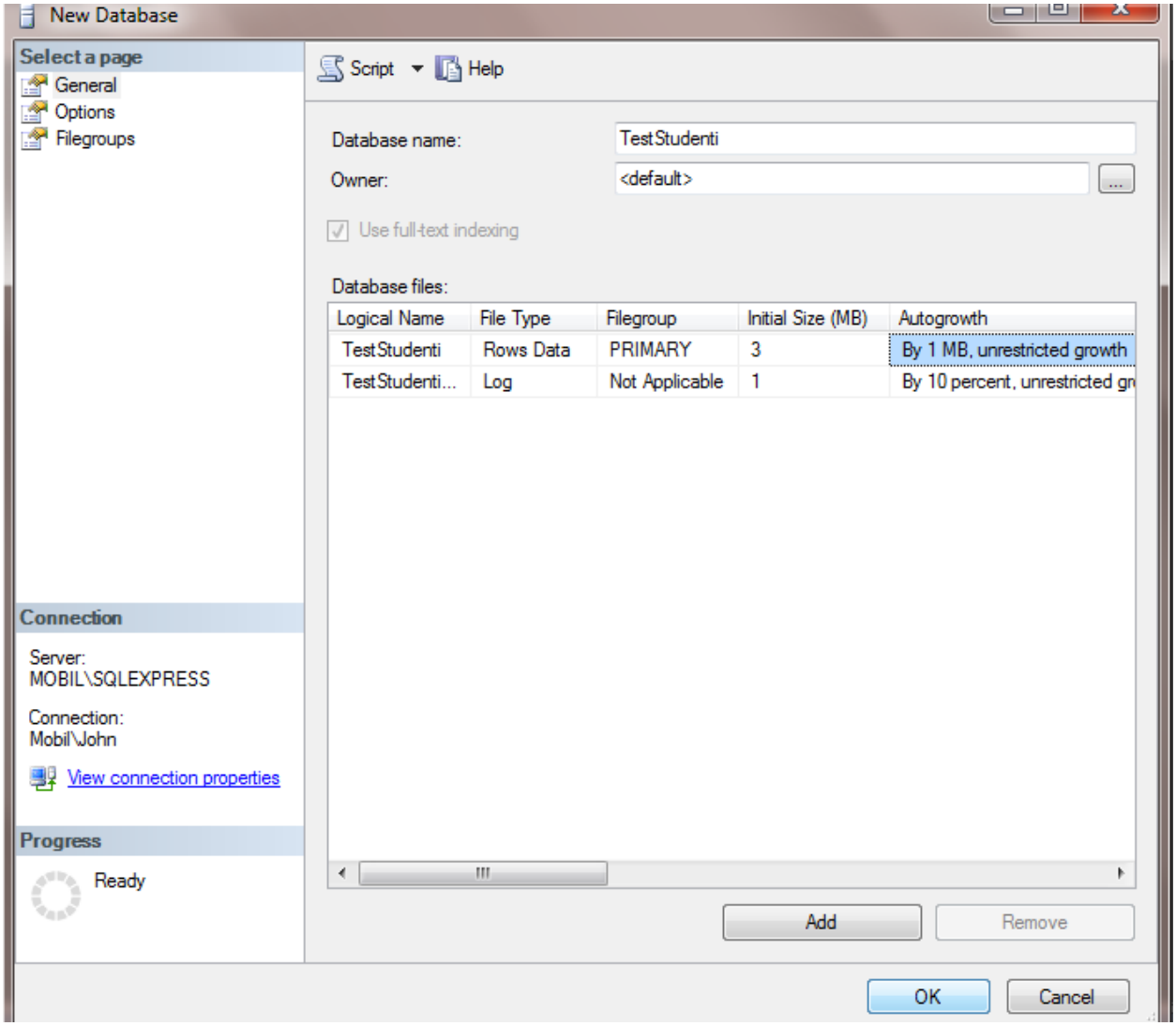

Fig. 7.

În partea stângă o să observăm că a apărut noua bază de date. Dacă extindem cu clic pe + -ul din fața denumirii o să vedem elementele importante ale unei baze de date SQL Server: Diagramele, Tabelele, Vederile, aspectele de programare (procedurile stocate, trigerele, funcțiile), precum și aspectele privind securitatea.

Următorul pas este reprezentat de crearea tabelelor și a câmpurilor acestora – structura tabelelor.

b. Crearea tabelelor

Similar bazelor de date, și în cazul tabelelor există tabele sistem pentru fiecare bază de date și tabele utilizator pe care le creăm noi, așa cum se vede în fig. 8.

Clic dreapta pe *Tables*

Clic pe *New Table*

În zona de lucru a SSMS, se deschide o fereastră pentru crearea tabelelor. Aici vom introduce **denumirea** câmpurilor (coloanelor, impropriu zis) și a **tipurilor** de date. Există o asemănare cu denumirea variabilelor și a tipurilor acestora din limbajele de programare.

Înainte de a crea efectiv tabela și câmpurile acesteia trebuie să știm câteva lucruri despre tipurile de date și constrângerile aplicate asupra câmpurilor.

În SQL Server există aprox de **tipuri de date**. Cele mai uzuale sunt:

Tipuri șir de caractere ASCII.

- char (n) – șir de caractere de dimensiune fixă de maxim 8000 caractere. Este folosit în câmpurile cu număr fix de caractere (EX: CNP – char (13));
- varchar (n) - șir de caractere de dimensiune variabilă de maxim 8000 caractere. Este cel mai des folosit în bazele de date care nu stochează șiruri de caractere ce conțin diacritice (ș,ț,â,ă,î, etc). Avantajul folosirii sale este că nu se stochează inuitil spații goale și, deci, nu crește mărimea bazei date (necesită explicație);
- varchar (MAX) - șir de caractere de dimensiune variabilă de maxim 1073741824 caractere. Firește, e folosit rar și doar atunci când e necesară stocarea unor volume foarte mari de date;
- text - șir de caractere de dimensiune variabilă de max 2 Gb. După cum se vede în acest caz dimensiunea este dată byti și nu în număr de caractere.

Tipuri șir de caractere Unicode. Pentru cele 4 tipuri de mai sus există câte un omolog pentru șiruri de date de tip Unicode (care pot stoca și diacritice) (nchar(n), nvarchar(n), nvarchar(MAX), ntext). Deoarece un caracter Unicode este dtocat pe 2 byti, spre deosebire de Ascii care e stocat pe 1 byte, numărul de caractere se reduce la jumătate față de tipurile ASCII. Nu contează acest lucru pentru ntext, care stochează, normal, tot 2 Gb.

În cazul bazelor de date care stochează date în limba română sau alte limbi care au diacritice (maghiară, germană, franceză, etc), precum și alte seturi de caractere (chineză, rusă, greacă, japoneză) vom folosi tipuri din această categorie, cel mai des *nvarchar(n)*.

Tipuri binare.

- bit – permite stocarea lui 0, 1 și NULL. E folosit în general pentru valori de tip boolean. (Ex: Activ – 0 sau 1). Alte tipuri binare sunt folosite pentru stocarea unor date diverse, de la imagini, documente, filme, etc.

Tipuri numerice.

- tinyint – nr întregi cuprinse între 0 și 255 (1 byte)
- smallint - nr întregi cuprinse între -32768 și 32767 (2 byte)
- int – nr întregi cuprinse între aprox -2 mld și 2 mld (4 octeți)
- bigint - nr întregi cuprinse între aprox -9 mld de mld și 9 mld de mld (10^{18}) (8 octeți)
- decimal(p,s) – stochează numere cu precizie și magnitudine fixă, unde **p** este precizia, **s** – nr maxim de cifre zecimale
- numeric(p,s)

- float
- real

Tipuri folosite în calculele financiare.

- smallmoney – stocat pe 4 octeți, stochează valori de până la 214 mii, puțin utilizabil
- money – stocat pe 8 octeți, stochează date de până a 922 de mii de mld, suficient pentru cele mai multe tipuri de calcule contabile, financiare, etc.

Tipuri pentru stocarea timpului și a datei calendaristice.

- datetime – de la 1 ian 1753 la 31 dec 9999, cu o rezoluție (acuratețe) de 3,33 ms). Este cel mai des utilizat tip de date pentru date calendaristice, suficient pentru majoritatea aplicațiilor
- datetime2 – 1 ian 0001 la 31 dec 9999, rezoluție 100 ns. Este folosit pentru stocarea unor date din teste inginerești în care se culeg date cu frecvență mare
- date – 1 ian 0001 la 31 dec 9999, rezoluție: 1 zi. E folosit atunci când dorim să stocăm doar data și nu ne interesează ore, min, etc. F frecvent utilizat în aplicații uzuale
- time – stochează orele cu o rezoluție de 100 ns. Folosit pentru câmpuri în care suntem interesați numai de stocarea timpului fără a interesa data.
- timestamp – amprentă de timp. E folosit atunci când dorim să stocăm momentul în care cineva a făcut o modificare, ștergere, etc din baza de date, precum și în alte cazuri similare. Într-o tabelă putem avea un singur câmp de acest tip.

Deasemenea tot în acest moment putem preciza tipurile de constrângeri la care e supus fiecare câmp în parte. Exemple de **constrângeri**:

- PRIMARY KEY – precizează câmpul (câmpurile) care va fi cheie primară. Cheia primară poate fi simplă (formată dintr-un singur câmp), sau compusă (formată din mai multe câmpuri). Foarte important în cazul cheii primare e faptul că în acest câmp (câmpuri) se stochează valor unice la nivelul întregii tabele). Scopul său este să se identifice „în mod unic” fiecare rând din tabelă. Orice câmp cheie primară va fi setat la NOT NULL și la UNIQUE.
- NOT NULL – nu permite introducerea de valor null în câmpul respectiv. Cu excepția câmpurilor care sunt obligatorii (ID, Nume, în exemplul de mai sus, de obicei celelalte câmpuri sunt setate la NULL).
- UNIQUE – nu permite introducerea de valori egale în câmpul unic pentru rânduri diferite.
- CHECK – forțează introducerea într-un câmp a unui domeniu precizat de valori. Ex: valori pozitive; pentru un câmp numit Luna am putea forța să permită numai valori cuprinse între 1 și 12.
- DEFAULT – este o valoare specifică unui câmp, pe care o precizăm în timpul creării tabelei. De fiecare dată când vom introduce un rând nou în tabelă, dacă pentru câmpul respectiv nu precizăm nicio valoare, atunci în câmpul respectiv va fi stocată implicit valoarea precizată.
- FOREIGN KEY – precizează câmpul (câmpurile) cheie străină pentru a putea crea asocieri între tabele.

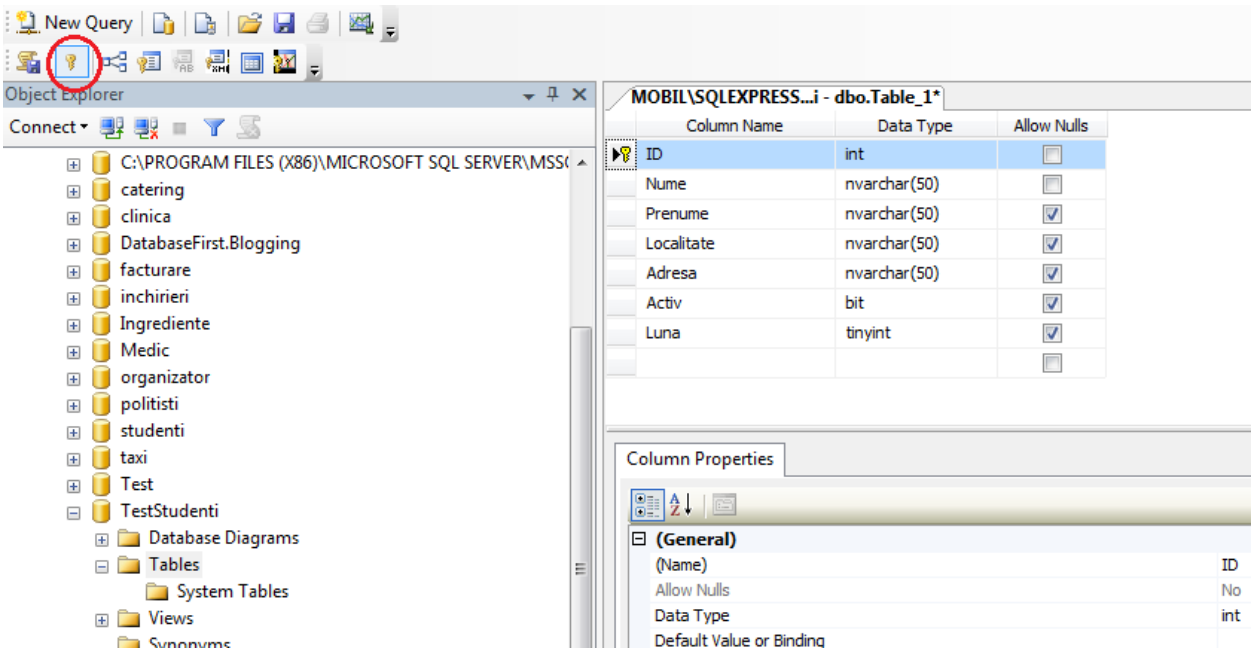

Fig. 8.

În imagine se vede că am creat 7 câmpuri.

Câmpul ID, de tip int, a fost setat să fie cheie primară. Setarea, respectiv desetarea se face prin clic pe butonul cheie primară din bara Table Designer, buton încercuit cu roșu în imagine. În momentul în care setăm un câmp cheie primară, este automat setat să nu permită NULL-uri și de asemenea este setată constrângerea UNIQUE. Câmpul fiind de tip int putem să gestionăm noi valorile sale sau putem să setăm ca în momentul inserării unui rând nou valorile să fie gestionate automat (autonumber). Pentru aceasta, în partea de jos, la Proprietăți (*Column Properties*), căutăm *Identity Specification* pe care îl expandăm cu clic pe plus și setăm *Is Identity* la *Yes*. Se vor activa *Seed* și *Increment*. Seed reprezintă valoarea de start, iar Increment pasul cu care va crește valoarea cheii primare.

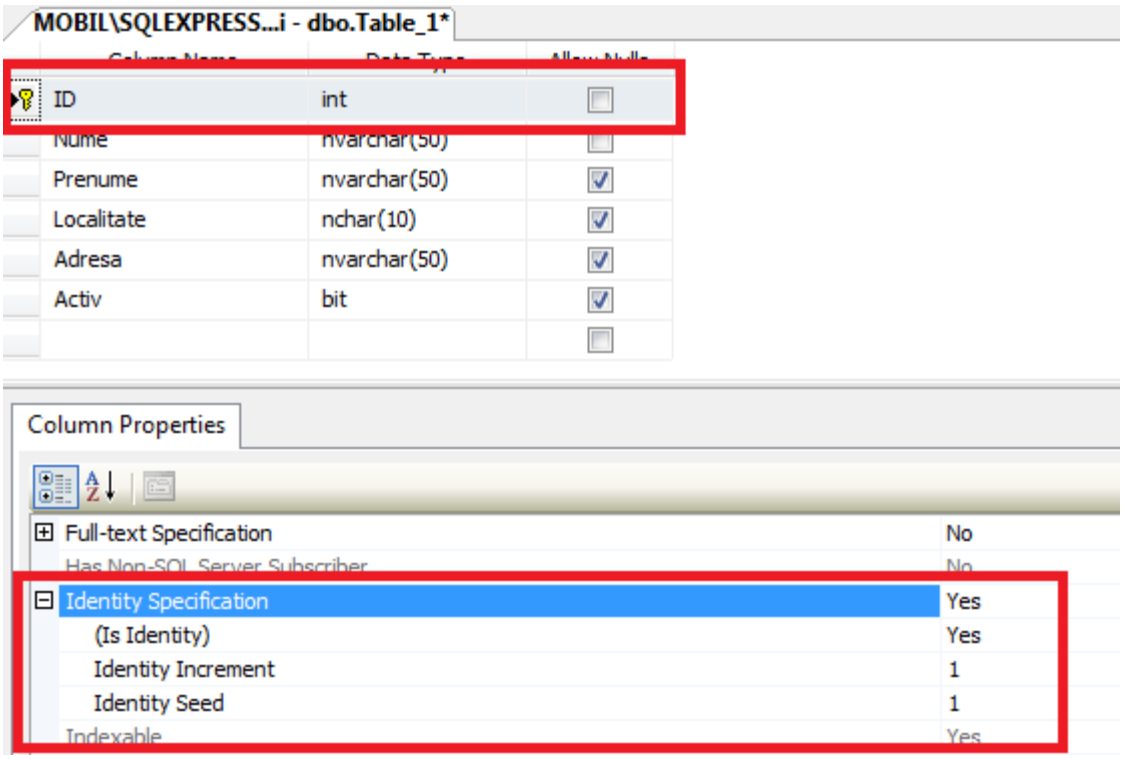

Fig. 9.

Pe câmpurile Nume și Prenume le vom face de tip nvarchar de 50 de caractere. Pentru că dorim să nu existe rânduri fără nici un fel de date (cu excepția cheii primare care e un câmp surogat, deci nu are nici un sens fizic), vom stabili ca cel puțin Numele să fie obligatoriu. Pentru aceasta vom debifa Permite valori Null (Allow Nulls). Observăm că toate câmpurile sunt implicit SET NULL. În cazul câmpului din cheia primară (ID), debifarea aceasta s-a făcut automat.

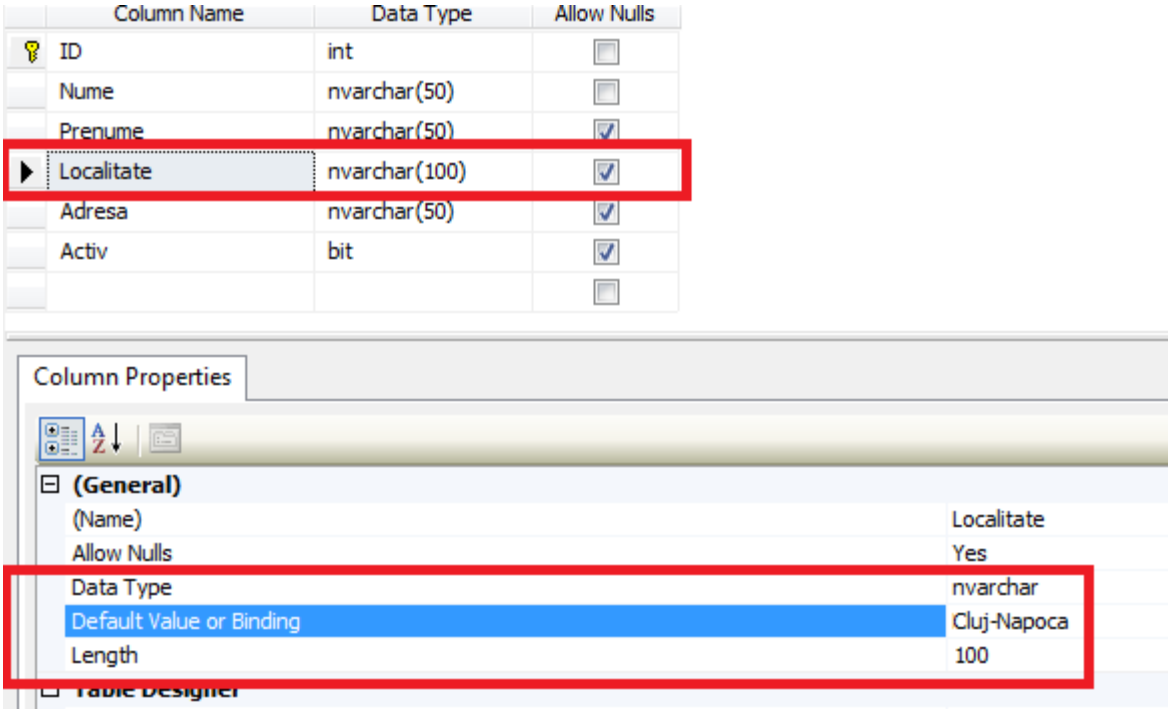

Fig. 10.

În cazul câmpului Localitate vom stabili tipul tot nvarchar, dar de 100 de caractere și suplimentar vom stabili o constrângere DEFAULT la *Cluj-Napoca* așa cum se vede în fig. 10.

Deoarece tipul este Unicode, valoarea *Cluj-Napoca* va fi transformată în *N'Cluj-Napoca'*. Dacă tipul ar fi fost ASCII, de ex varchar, valoarea ar fi fost transformată în *'Cluj-Napoca'*.

Deoarece câmpul Activ este boolean l-am setat la tipul bit.

În cazul câmpului Luna (pe care l-am adăugat doar pentru exemplificare) vom crea o constrângere CHECK în care forțăm să permită numai numere de lună valide (de la 1 la 12).

Clic dreapta pe câmp, apoi clic pe Check Constraints...
În fereastra care apare clic pe Add
Schimbăm numele (Name) ca în fig. 11
În Expression introducem: *Luna>0 and Luna<13*
Apoi clic pe Close

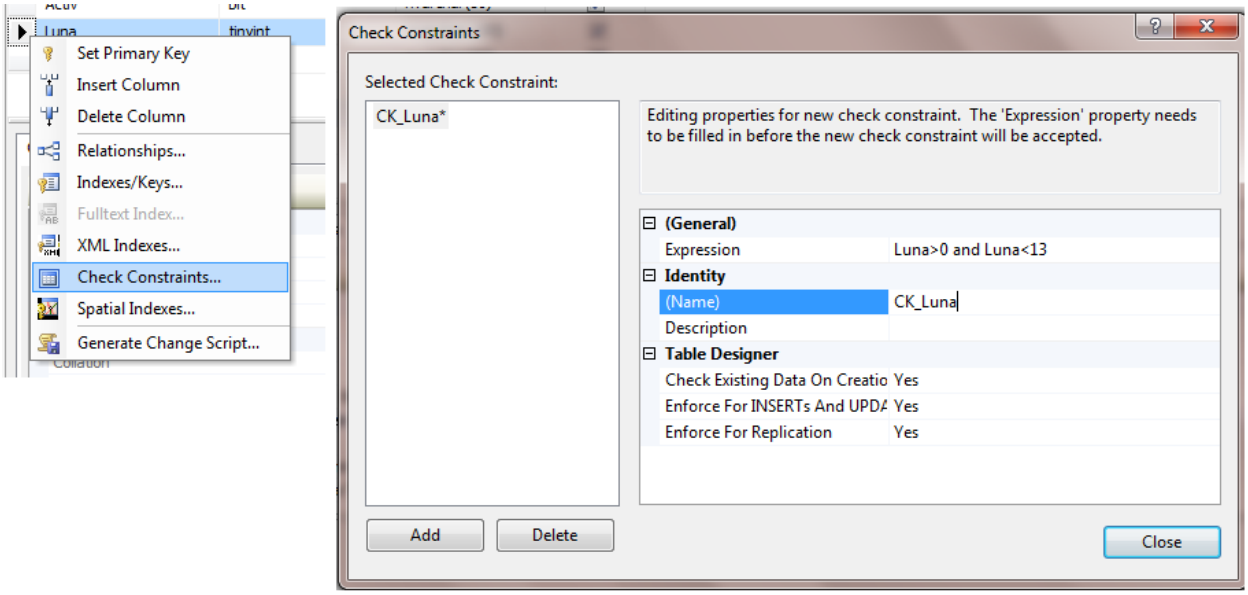

Fig. 11.

Pentru salvarea tabelei închidem fereastra tabelă de pe x-ul din dreapta sus (nu închidem tot SSMS-ul), dăm clic pe Yes, iar în fereastră scriem numele tabelei, apoi OK:

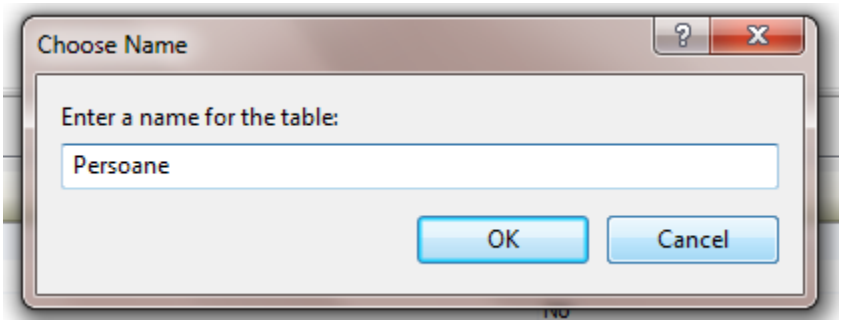

Fig. 12.

O să observăm, în partea stângă, o nouă intrare care e tabela nou creată. Obiectele sale sunt:

- **Coloane (Columns)** – dacă expandăm observăm denumirea coloanelor, tipul și dacă permite NULL sau nu. Deasemnea vedem câmpul cheie primară.
- **Chei (Keys)** – aici vedem cheile tabelei. În cazul de față e cheia primară
- **Constrângeri (Constraints)** – vedem constrângerea de tip CHECK stabilită asupra câmpului Luna și constrângerea de tip DEFAULT stabilită asupra câmpului Localitate
- **Declanșatori (Triggers)** – nedefinit nici un trigger
- **Indecși (Indexes)**- în momentul definirii cheii primare a fost automat definit un index de tip cluster. Orice alt index care va fi definti în această tabelă trebuie obligatoriu să fie de tip non-cluster.

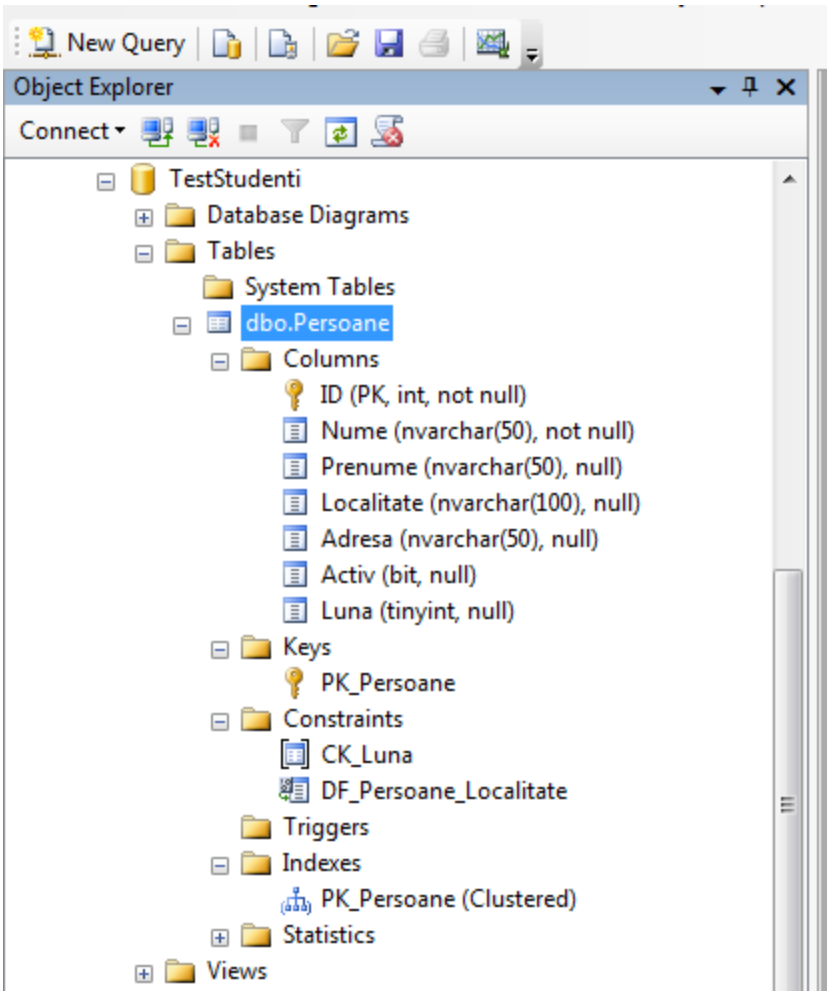

Fig. 13.

c. Modificarea structurii tabelelor

Clic dreapta pe denumirea tabelei, apoi clic pe *Design*

Pentru a vedea modificările: clic dreapta pe numele tabelei, apoi clic pe *Refresh*

d. Ștergerea tabelelor

Clic dreapta pe denumirea tabelei, apoi clic pe *Delete*

e. Introducerea datelor în tabelă

Clic dreapta pe denumirea tabelei, apoi clic pe *Edit Top 200 Rows*

De reținut.În bazele de date salvarea datelor introduse într-un rând se face automat atunci când se trece pe alt rând (cu Enter, cu tastele săgeți sau cu clic de mouse).

Vom **introduce un rând (C din CRUD)** ca în fig. 14. În câmpul ID scriem 1 (doar dacă nu e de tip Identity), apoi dăm ENTER (trecere la rândul următor). Dacă ID este de tip Identity introducem un prenume, apoi ENTER. Vom primi un mesaj în care ni se spune că Numele nu permite valori NULL. În consecință, vom introduce un nume, apoi ENTER. De această dată are loc salvarea datelor.

Observăm că, deși nu am introdus nimic în câmpul Localitate, a apărut *Cluj-Napoca*, ceea ce e firesc atâta timp cât am definit constrângerea DEFAULT.

Pe randul 2 introducem un nume și o localitate. De această dată se reține în tabelă numele localității introduse, ceea ce este normal.

Modificare date –U din CRUD

Revenim pe oricare rând și introducem în câmpul Luna valoarea 14. O să primim un mesaj în care ni se spune că avem o constrângere CHECK pe coloana Luna. Clic pe OK, apoi scriem o valoare validă (ex: 5).

În cazul în care dorim să nu continuăm editarea, renunțăm cu tasta ESC.

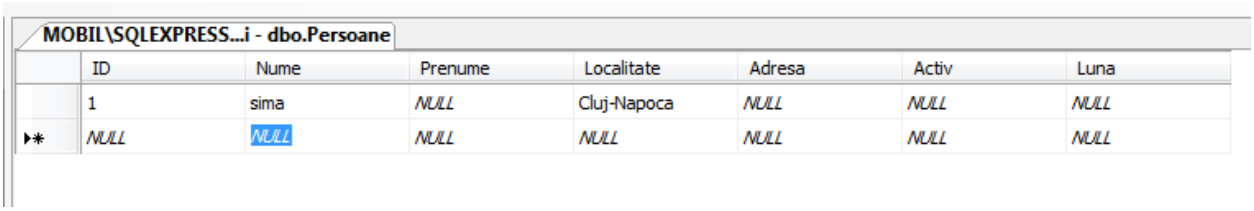

Fig. 14.

Pentru a **șterge (D din CRUD)** unul sau mai multe rânduri se selectează cu mouse-ul pe antet rând (în stânga unde apare săgeata și steluța), clic dreapta, apoi clic pe Delete.

f. Vizualizarea datelor din tabelă

Clic dreapta de denumirea tabelei, apoi clic pe *Select Top 1000 Rows*. **R din CRUD.**

g. Crearea tabelei Telefoane

În mod similar creăm o tabelă pe care o numim *Telefoane.*

Observăm că am creat și aici un câmp ID, tot cu autoincrementare. În cazul acestei tabele acest câmp nu e strict necesar.

Deasemenea nu permitem NULL în cazul câmpului telefon. Aceasta ar putea duce la crearea de rânduri cu conținut vid.

Însă cel mai important lucru care trebuie reținut aici este faptul că am introdus un câmp care nu aparține acestei tabele: id_persoana, motiv pentru care îl numim cheie străină. Singurul motiv pentru care îl întrebuințăm este ca să putem lega cele 2 tabele 1 la m (o persoană poate deține 0, 1 sau mai multe telefoane). Foarte important este ca cele 2 câmpuri (ID-ul – cheia primară din tabela Persoane și id_persoana – cheia străină din tabela Telefoane) să fie de același tip (int, în cazul de față). După cum se vede numele lor nu trebuie să fie identic.

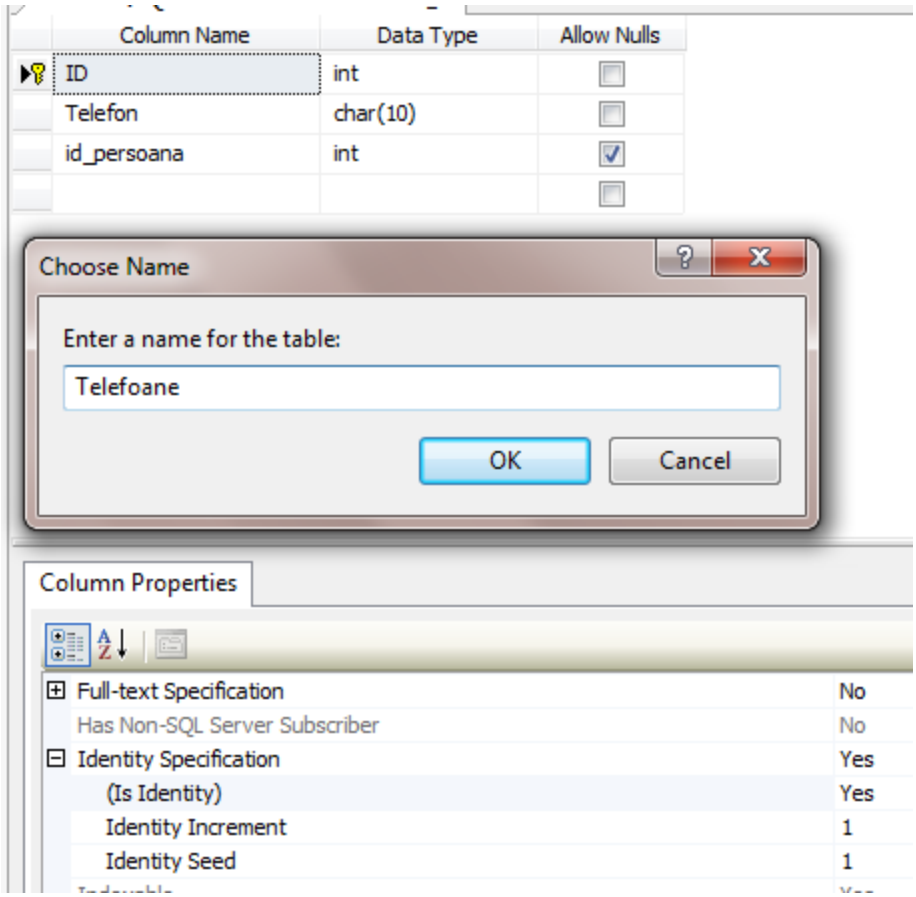

Fig. 14.

În cazul acestei tabele, deocamdată nu avem nicio constrângere. Avem doar o cheie primară și un index corespunzător acesteia.

h. Crearea diagramelor

Legarea celor două tabele se poate face în mai multe modeuri. Aici vom construi o diagramă în care vom face legătura.

Pentru aceasta:

- clic dreapta pe *Database diagrams* din baza noastră de date
- Clic pe *New Database Diagrams*
- Clic pe OK
- Selectăm cele două tabele din ferastră, clic pe *Add*, apoi clic pe *Close*.
- Prindem cu mouse-ul ID-ul din Persoane și îl tragem peste id_persoana din Telefoane
- Vor apărea 2 ferestre: verificăm să fie numele tabelelor și a câmpurilor corecte apoi le închidem cu clic pe OK

Ar trebui să vedem legătura ca în fig. 15 (sau similar)

Dacă am creat o legăturără greșită, pentru ștergerea sa (implicit și a constrângerii de cheie străină) dăm clic dreapta pe legătură apoi clic pe *Delete Relationships from Database*.

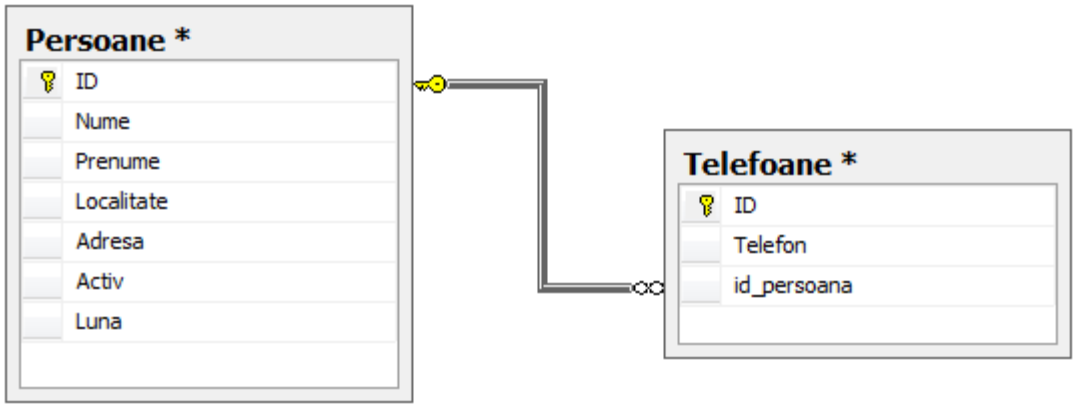

Fig. 15.

Închidem diagrama, îi dăm un nume și o salvăm.

Dacă dăm clic dreapta pe numele tabelei și Refresh o să vedem că avem o cheie suplimentară (care e și constrângerea de tip FOREIGN KEY).

Spor la muncă!