



UNIVERSITATEA TEHNICĂ „GHEORGHE ASACHI” IAȘI
FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
SPECIALIZAREA CALCULATOARE ȘI TEHNOLOGIA
INFORMAȚIEI



DISCIPLINA BAZE DE DATE PROIECT

GESTIONAREA CENTRELOR DE RECOLTARE DE SÂNGE

(pentru analize)

Coordonator,
Cătălin Mironeanu

Student,
Mihalache Nicoleta-Ecaterina

Iași, 2020

Titlu proiect: Gestionarea centrelor de recoltare de sânge (pentru analize)

Analiza, proiectarea și implementarea unei baze de date și a aplicației aferente care să modeleze activitatea unui centru de recoltare de sânge cu privire la gestiunea analizelor realizate, a pacienților și a personalului medical.

Descrierea cerințelor și modul de organizare al proiectului

Volumul mare de informații existente în cazul unui centru de recoltare de sânge cu numeroși pacienți determină necesitatea fluidizării fluxurilor de date, gestiunea acestora fiind o adevărată provocare. Activitatea de gestiune a unui centru de recoltare de sânge implică o muncă intensă în ceea ce privește numeroasele documente ce țin de înregistrarea pacienților, a analizelor pe care aceștia le fac, de personalul medical, precum și de realizarea analizelor în urma cărora cabinetul medical oferă pacienților rezultatul acestora.

Informațiile de care avem nevoie sunt cele legate de:

-cabinete: ne interesează să identificăm cabinetul ce va gestiona datele (de exemplu, posibilitatea pacienților de a alege un anumit cabinet unde își vor face analizele medicale). Acesta va avea o denumire, o adresă și un număr de telefon.

-contracte: ne interesează să identificăm contractele semnate de fiecare cabinet cu angajații săi, ce vor avea o dată când au fost semnate, o dată de final și un număr de luni ce reprezintă durata acestora.

-pacienți: datele referitoare la pacienți se află în tabela Pacienti și ne interesează numele și prenumele pacientului, data nașterii, CNP-ul, vârsta, genul și adresa acestuia.

-analize: în cazul informațiilor legate de analize, informațiile se vor stoca în două tabele: Analize și Detalii. Astfel, în tabela Analize vom stoca informații legate de tipul analizei, prețul ei, data când a fost recoltată și data rezultatului, urmând ca în tabela Detalii să fie stocate informații legate de numele analizei ce face parte dintr-o anumită categorie data de tipul analizei din tabela Analize, valoarea minimă și valoarea maximă a acesteia, precum și rezultatul ei.

-angajații: în cazul informațiilor legate de angajați, informațiile se vor stoca în două tabele: Asistente și Medici_lab. Astfel, în tabela Asistente vom stoca informații legate de asistentele angajate care recoltează sânge de la pacienți, numele și prenumele lor, iar în tabela Medici_lab vom stoca informațiile legate de medicii de laborator, cei care realizează rezultatul analizei, numele și prenumele acestora.

Descrierea funcțională a aplicației

Principalele funcții care se pot întâlni într-un centru de recoltare de sânge sunt:

- Evidența pacienților
- Evidența angajaților
- Evidența analizelor efectuate
- Evidența perioadei de angajare pentru fiecare angajat

Descrierea detaliată a entităților și a relațiilor dintre tabele

Tabelele din această aplicație sunt:

- ✓ Cabinete;
- ✓ Contracte;
- ✓ Analize;
- ✓ Detalii;
- ✓ Pacienți;
- ✓ Asistente;
- ✓ Medici_lab;
- ✓ cab_a_FK;
- ✓ pac_a_FK;
- ✓ pac_det_FK;
- ✓ med_lab_a_FK;
- ✓ asis_paci_FK;

În proiectarea acestei baze de date s-au identificat tipurile de **relații** 1:1, 1:n, n:n.

Între tabelele **Cabinete** și **Contracte** este o relație **one-to-many** deoarece un cabinet poate încheia mai multe contracte. Reciproca însă nu este valabilă, deoarece un contract poate fi semnat doar de un singur cabinet. Legătura dintre cele două este realizată prin câmpul **Cabinete_id_cabinet**.

Între tabelele **Cabinete** și **Analize** este o relație **many-to-many** deoarece un cabinet pot realiza una sau mai multe tipuri de analize, iar un tip de analiză poate fi realizată de unul sau mai multe cabinete. Pentru ca tabela să se afle în 3FN această relație se va sparge în două, rezultând două relații **1:n** și legătura dintre cele două se va realiza cu ajutorul unei alte tabele **cab_a_FK** care va conține cheia primară a fiecărei tabele. Altfel spus legătura se face prin două câmpuri **Cabinete_id_cabinet** și **Analize_id_analiza** reunite într-o tabelă comună.

Între tabelele **Analize** și **Detalii** este o relație **one-to-one** deoarece unei analize îi corespunde un set de detalii, iar un set de detalii este deținut de o singură analiză. Tabela Detalii este concepută în special pentru a descrie intervalul de valori pe care le poate avea o analiză. Legătura dintre cele două este data de câmpul **Analize_id_analiza**.

Între tabelele **Analize** și **Pacienți** este o relație **many-to-many** deoarece o analiză pot fi făcută de unul sau mai mulți pacienți, iar un pacient poate face una sau mai multe analize. Pentru ca tabela să se afle în 3FN această relație se va sparge în două, rezultând două relații **1:n** și legătura dintre cele două se va realiza cu ajutorul unei alte tabele **pac_a_FK** care va conține cheia primară a fiecărei tabele. Altfel spus legătura se face prin două câmpuri **Pacienti_id_pacient** și **Analize_id_analiza** reunite într-o tabelă comună.

Între tabelele **Pacienți** și **Detalii** este o relație **many-to-many** deoarece un pacient pot avea una sau mai multe detalii ale analizelor, iar un detaliu este corespondent unui sau mai multor pacienți. Pentru ca tabela să se afle în 3FN această relație se va sparge în două, rezultând două relații **1:n** și legătura dintre cele două se va realiza cu ajutorul unei alte tabele **pac_det_FK** care va conține cheia primară a fiecărei tabele. Altfel spus legătura se face prin două câmpuri **Detalii_id_detalii** și **Pacienti_id_pacient** reunite într-o tabelă comună.

Între tabelele **Analize** și **Medici_lab** este o relație many-to-many deoarece o analiză poate fi realizată de unul sau mai mulți medici, iar un medic poate realiza una sau mai multe analize. Pentru ca tabela să se afle în 3FN această relație se va sparge în două, rezultând două relații **1:n** și legătura dintre cele două se va realiza cu ajutorul unei alte tabele **med_lab_a_FK** care va conține cheia primară a fiecărei tabele. Altfel spus legătura se face prin două câmpuri **Medici_lab_id_medic** și **Analize_id_analiza** reunite într-o tabelă comună.

Între tabelele **Pacienți** și **Asistente** este o relație **many-to-many** deoarece unui pacient i se poate recolta sânge de către una sau mai multe asistente, iar o asistentă poate recolta sânge la rândul ei unui sau mai multor pacienți. Pentru ca tabela să se afle în 3FN această relație se va sparge în două, rezultând două relații **1:n** și legătura dintre cele două se va realiza cu ajutorul unei alte tabele **asis_paci_FK** care va conține cheia primară a fiecărei tabele. Altfel spus legătura se face prin două câmpuri **Asistente_id_asistenta** și **Pacienti_id_pacient** reunite într-o tabelă comună.

Între tabelele **Contracte** și **Medici_lab** este o relație **one-to-one** deoarece un contract este semnat cu un medic, iar un medic poate semna un singur contract. Legătura dintre cele două este data de câmpul **Contracte_id_contract**.

Între tabelele **Contracte** și **Asistente** este o relație **one-to-one** deoarece un contract poate fi semnat cu o asistentă, iar o asistentă poate semna un singur contract. Legătura dintre cele două este data de câmpul **Contracte_id_contract**.

În proiectarea acestei baze de date s-au identificat următoarele tipuri de **constrângeri de integritate referențială**: primary key și foreign key, precum și **alte tipuri** de constrângeri: check, unique, not null.

Primary key-urile sunt generate de baza de date prin mecanismul de tip **autoincrement** și sunt atribuite tuturor id-urilor din tabele, identificând în mod unic fiecare înregistrare dintr-un tabel. Ele trebuie să conțină doar valori unice și nu pot avea valori NULL, iar un tabel poate avea doar o singură cheie primară. Aici avem totodată și constrângerile de tip **unique** și **not null**.

Constrângerea **unique** este atașată atributului CNP din tabela Pacienți, deoarece nu pot exista doi pacienți cu același cnp. Punând această constrângere impunem ca toate CNP-urile introduse în baza de date să fie unice.

Foreign key-urile sunt utilizate pentru a lega două tabele împreună, fiind utilizate în toate relațiile dintre tabelele bazei de date. Acestea sunt câmpuri din tabele care se referă la Primary key-ul dintr-un anumit tabel. Un exemplu este Cabinete_id_cabinet ce este foreign key și leagă tabela copil Contracte de tabela părinte Cabinete. Pot exista și mai multe foreign key-uri și un exemplu este în cazul în care avem o relație many-to-many. Aceasta este spartă în alte două relații one-to-many rezultând astfel 2 foreign key-uri. Un exemplu este apariția tablei copil med_lab_a_FK între tabelele Medici_lab și Analize, având 2 foreign key-uri Medici_lab_id_medic și Analize_id_analiza.

Constrângerea de tip **check** este utilizată pentru atributul nr_telefon din tabela Cabinete deoarece se impune un anumit format pentru numărul de telefon pe care îl poate avea acel cabinet (să conțină 10 cifre, prima cifră să fie 0 și a doua cifră să fie 7, 2 sau 3). Mai este utilizată pentru atributul rezultat_an din tabela Detalii, impunându-i acestuia să fie în intervalul dat de val_min și val_max, pentru attributele val_min și val_max din aceeași tabelă pentru ca val_min să nu conțină valori mai mici decât 0 și pentru

val_max să conțină valori mai mari decât valorile pe care le conține val_min, pentru atributul CNP din tabela Pacienți ca lungimea lui să fie egală cu 13.

Constrângerea **check** este utilizată pentru a impune unei valori să aparțină unui interval de valori. Un exemplu este reprezentat prin atributul preț din tabela Analize al cărei valoare trebuie să fie în intervalul (10, 250). Aceasta mai este utilizată pentru a impune unei valori să aparțină unei liste de valori existente. Un exemplu este dat de către atributul gen din tabela Pacienți a cărei valoare trebuie să existe în lista de valori data, dacă este femeie gen=F și dacă este bărbat gen=B. Ea este utilizată și pentru attributele tip și nume_d din tabelele Analize și Detalii pentru a specifica categoria și numele analizei.

Diagrama Logică

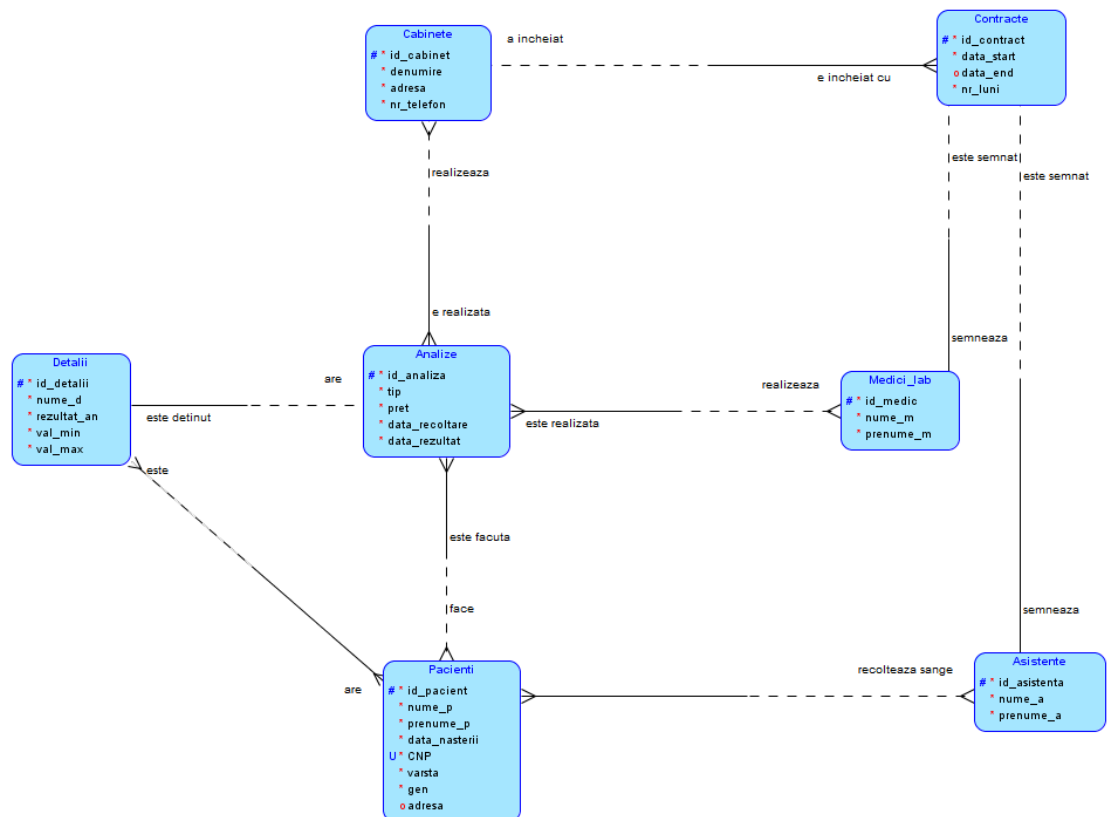


Fig.1 Model logic

Diagrama Relațională

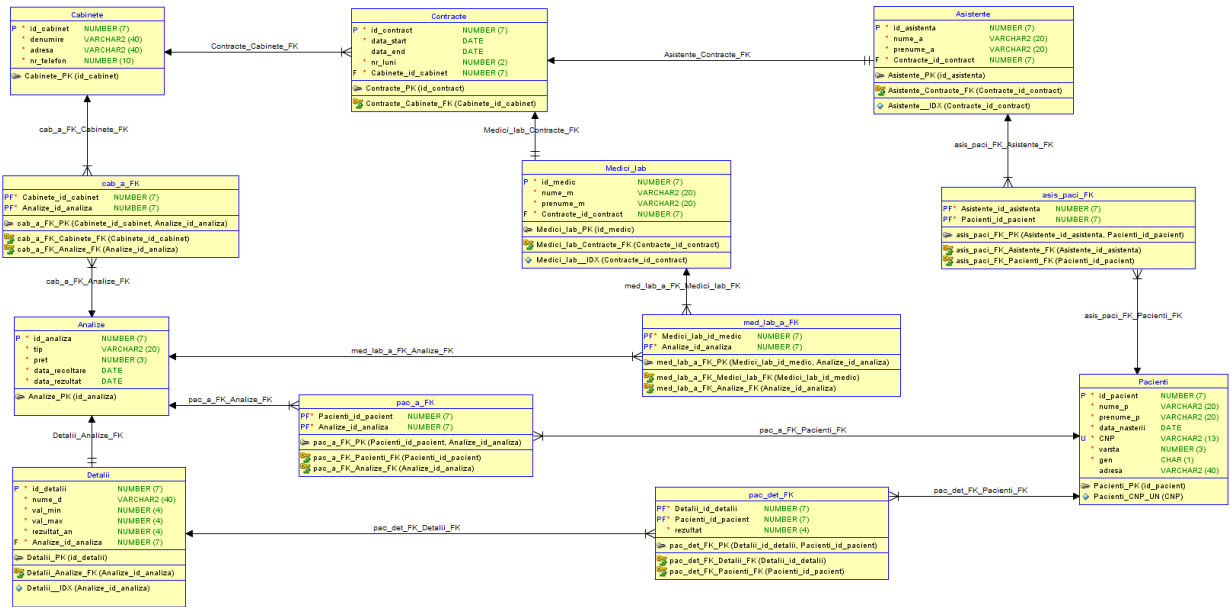


Fig.2.Model relațional