

**UNIVERSITATEA TEHNICĂ “GHEORGHE
ASACHI” IAȘI**
FACULTATEA AUTOMATICĂ ȘI CALCULATOARE
SPECIALIZAREA CALCULATOARE
DISCIPLINA BAZE DE DATE PROIECT

Gestiunea activității într-un penitenciar

Coordonator,

Asist. Drd. Ing. Cătălin Mironeanu

Student,

Mucedu Georgeta-Iuliana, 1408A

Iași, 2020

Titlu proiect: Gestionare modul de funcționare și organizare a unui penitenciar

Analiza, proiectarea și implementarea unei baze de date care să modeleze activitatea unui penitenciar în ceea ce privește gestionarea deținuților, cât și a persoanelor responsabile cu supravegherea acestora.

Descrierea cerintelor si modul de organizare al proiectului

Coordonarea unui penitenciar constituie o adevărată provocare deoarece necesită o organizare foarte bine stabilită în ceea ce privește atât angajații, cât și deținuții deoarece este necesar să avem numeroase informații despre ambele categorii.

Astfel, se disting următoarele aspecte:

-Fișe: conține informații despre deținuți precum id-ul fișei (un deținut poate avea mai multe fișe în cazul în care acesta a mai fost încarcerat anterior, adică recidivă), data la care un deținut a fost încarcerat

-Deținuți: conține câte un id_detinut unic pentru fiecare deținut, și informații personale ce ne ajută la identificare, precum numele și prenumele acestuia, cnp-ul unic, adresa de domiciliu

-Celule: conține informații cu privire la organizarea spațiului, precum id-ul celulei, capacitatea acesteia (câți deținuți încap în aceasta), și etajul la care aceasta se află pentru a facilita accesul în cazul necesității unei intervenții.

-Deținuți-Celule: leagă tabela Deținuți de tabela Celule-**Fapte_deținuți:** leagă tabela Deținuți de tabela Fapte. Aceasta tabelă a fost creată pentru a elimina relația m:n și a o transforma în alte două relații 1:n

-Fapte: conține informații distinctive cu privire la faptele comise identificate printr-un id_fapta unic, numărul de ani corespunzători comiterii acesteia și numele.

-Angajați: conține o serie de informații personale în ceea ce privește angajații, astfel, în cazul în care dorim să aflăm informații dintr-un anumit interval orar să știm la cine să apelăm. Fiecărui angajat îi este atribuit un id_angajat unic, numele și prenumele, cnp-ul și funcția.

-**Detalii_Angajați:** conține adresa, numărul de contact la care îl putem găsi, email-ul și numărul de telefon.

-**Ture:** conține informații cu privire la programul angajaților pentru a stabili cu ușurință în cazul producerii unui eveniment cine este responsabil. Așadar avem tura angajat și data tura.

Descrierea funcțională a aplicației

Principalele funcții pe care le putem utiliza sunt:

- Evidența Deținuților și fișele acestora
- Evidența Angajaților și turelor de lucru
- Evidența Celulelor și distribuirea Deținuților în ele

Descrierea detaliată a entităților și a relațiilor dintre tabele

Tabelele din această aplicație sunt:

- Angajați
- Detalii_Angajați
- Ture
- Celule
- Deținuți
- Fișe
- Deținuți_celule
- Fapte
- Relational_10

Descrierea a relațiilor dintre tabele

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

Între tabela Fișe și tabela Deținuți avem o relație de tip **n:1** deoarece un deținut poate avea mai multe fișe(se vizează cazul în care un deținut a mai fost încarcerat anterior, iar acum a recidivat), însă o fișă face referire la un singur deținut. Cele două tabele sunt legate prin intermediu atributului id_deținut.

Între tabela Deținuți și tabela Fapte avem o relație de tip **m:n** deoarece mai mulți deținuți pot comite mai multe fapte. Această legătură se realizează prin intermediul unei alte tabele numită Relational_10 ce transformă legătura many-to-many în două legături de tip n:1. Relational_10 conține id_deținut din tabela Deținuți și id_faptă din tabela Fapte.

Între Deținuți și Deținuți_celule avem o relație de tip **1:n** deoarece un deținut poate să fi locuit anterior într-o altă celulă, cele două tabele fiind legate prin atributul id_deținut.

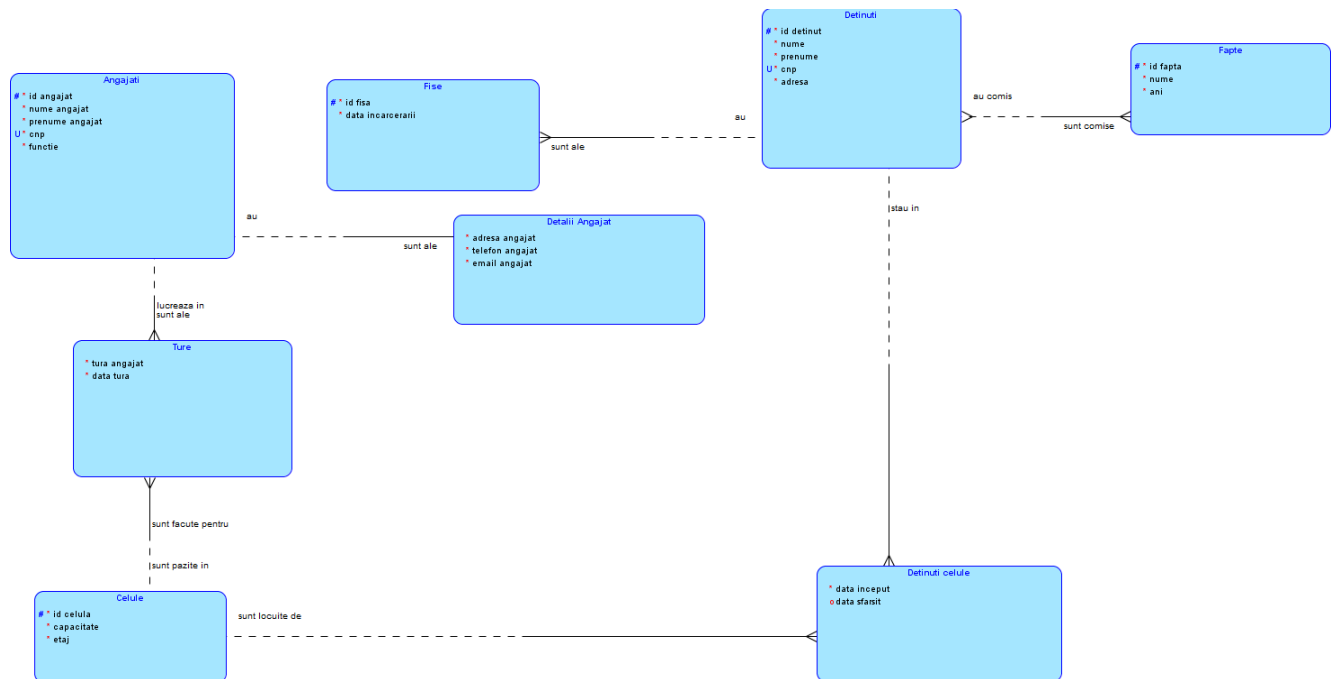
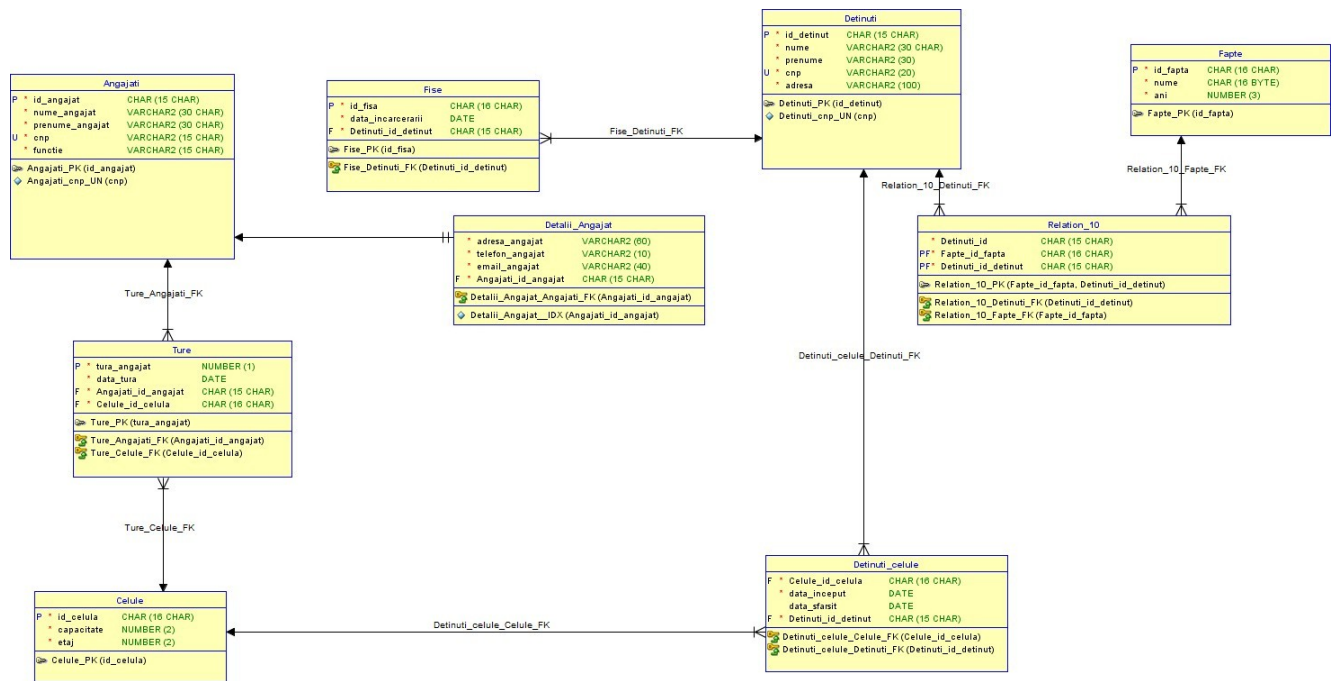
Între tabela Celule și Deținuți_celule avem o relație de tip 1:n deoarece într-o celulă pot să conviețuiască mai mulți deținuți. Cele două tabele sunt legate prin atributul id_celulă.

Practic tabela Deținuți_celule are rolul de a intermedia relația dintre tabela Deținuți și tabela Celule.

Între tabela Celule și tabela Ture avem o relație de tip 1:n deoarece o celulă este păzită în mai multe ture pentru a acoperi toate cele 24 ore ale unei zile. Cele două tabele sunt legate prin id_celulă.

Între tabela Angajați și tabela Ture avem o relație de tip 1:n deoarece un angajat poate lucra în mai multe ture în zile diferite. Cele două tabele sunt legate prin intermediul atributului id_angajat.

Între tabela Angajați și tabela Detalii_Angajat avem o relație de tip 1:1 deoarece fiecare angajat are un câmp rezervat pentru detalii. Cele două tabele sunt interconectate prin intermediul atributului id_angajat.



Constrângeri utilizate:

1. DE TIP NOT NULL

Toate atributele sunt de tip NOT NULL, cu excepția atributului data_sfârșit

2. DE TIP UNIQUE

Este necesar ca numărul de telefon și email-ul și cnp-ul de la orice deținut sau angajat să fie unic, întrucât nu pot exista două astfel de attribute identice.

Toate primay-key-urile sunt unice predefinite.

3. DE TIP CHECK

CNP să aibă 14 cifre

Numele și prenumele Angajaților și Deținuților să nu conțină caractere speciale.

Numărul de telefon al angajaților să înceapă cu 07 și să continue cu alte 8 cifre (trebuie să fie neapărat un nr de mobil)

Emailul să aibă caracterul @.

Testele ce pot fi rulate pentru atingerea scopului propus

Am verificate:

- eroare datorata constrângerii dată de numărul de telefon
- eroare datorata constrângerii generate de cnp
- eroare datorata constrângerii email-ului
- eroare datorata constrângerii legate de nume
- eroare datorata constrângerii legate de prenume
- eroare datorata unicitatii cnp-ului
- eroare datorata datei de inceput fiind mai mare decat data de sfârșit
- eroare datoara angajatului avand deja o tura in acea zi
- ștergand un angajat, se vor șterge prin trigger si toate turele acestuia si datele angajatului (delete în cascada)
- testare package procedura creeaza_detinut
- testare package functie salar angajat pe tura 3
 - medicii & gardienii au salar 150, inrijitorii 50
 - tura 3 se plateste dublu
- testare package functie identifica luna

În plus, avem o funcție ajutătoare identificarea unica a unei luni, (MMYYYY).

Descrierea logicii stocate

Am creat un pachet, `PACKAGE DETINUTI_PKG` ce înglobează o procedură și două funcții. Acestea sunt: `CREEAZA_DETINUT`, `IDENTIFICA_LUNA`, `SALARIU_ANGAJAT`.

În interiorul body-ului pachetului am creat procedura `CREEAZA_DETINUT`. Aceasta primește un nume, prenume, cnp și adresa ca tipuri `varchar` și le înserează în tabela `detinuti`.

În continuare am declarat funcția ajutătoare (spun ajutătoare deoarece se apelează într-o altă funcție), `IDENTIFICA_LUNA` ce preia o lună dintr-o anumită dată calendaristică.

Funcția `SALARIU_ANGAJAT` apelează funcția precedentă și pe baza id-ului angajatului și a unei date va calcula salariul angajatului în luna specificată pe baza funcției angajatului și a nr de ture lucrate. Tot în cadrul acesteia folosim un cursor pentru parcurgerea fiecărei ture pe care o are angajatul.

De asemenea, am implementat și 3 trigger-uri. Acestea deservește următoarele sarcini:

- verifică la inserarea în tabela `ture` ca un angajat nu poate avea mai mult decât o tură în aceeași zi
- verifică ca o dată de început adăugată în tabela `detinuti_celule` nu este după data de sfârșit
- la ștergerea unui angajat îi șterge și detaliile și turele din tabelele `detalii_angajati` și `ture`