



**UNIVERSITATEA TEHNICĂ “GH ASACHI” IAȘI**  
**FACULTATEA AUTOMATICĂ ȘI CALCULATOARE**  
**SPECIALIZAREA CALCULATOARE ȘI TEHNOLOGIA**  
**INFORMAȚIEI**

**DISCIPLINA BAZE DE DATE PROIECT**

# **Gestiunea angajaților unei firme de curierat**

**Coordonator,  
Prof. Mironeanu Cătălin**

**Student,  
Pintilie Răzvan-Nicolae  
Grupa 1306B**

**Iași, 2022**

## Titlul proiectului

### Gestiunea angajaților unei firme de curierat

Proiectul își propune să ofere o modalitate de gestionare a angajaților unei firme de curierat și, mai mult, o modalitate de a ține evidența fiecărei adrese din oraș în parte pentru o bună organizare internă.

### Descrierea proiectului

Numărul mare de angajați al unei astfel de firme și multitudinea zonelor orașului ce trebuie acoperite de aceștia determină necesitatea unei bune organizări și a unei bune cunoașteri (din partea superiorilor) a îndatoririlor fiecărui subordonat; numărul de străzi al orașului fiind foarte mare, e nevoie de o grupare a acestora pe zone, fiecărui curier revenindu-i, de obicei, o singură zonă. Informațiile de care avem nevoie corespund următorilor factori:

- **Managerii depozitului:** aceștia se ocupă de gestionarea coletelor la nivel de depozit, în vederea distribuirii lor fiecărui curier; dacă un manager este responsabil de o anumită zonă, atunci acesta îi este superior angajatului ce livrează pe acea zonă;
- **Curierii (angajații):** un angajat poate avea mai multe **roluri** (șofer, ajutor șofer, manipulator lift); angajaților le sunt atribuite zone;
- **Adresele și gruparea lor în zone:** mai multe adrese formează o „zonă mică”, iar mai multe „zone mici” formează o „zonă cargo” (aici revin coletele mari/grele); unele adrese pot avea **observații** legate de starea traficului, starea carosabilului, etc;

### Descrierea funcțională a aplicației

Principalele funcții ale aplicației sunt:

- evidența angajaților (referitor la zonele pe care livrează)
- evidența zonelor
- evidența adreselor (în ce zone sunt distribuite)
- furnizarea de informații și despre alți manageri

### Structura și relațiile dintre entități

Entitățile din această aplicație sunt:

- Angajați
- Zone
- Adrese
- Manageri\_depozit
- Tipuri\_rol
- Observații

În proiectarea acestei baze de date s-au identificat următoarele tipuri de relații:  
1:1 (one-to-one), 1:n (one-to-many).

Între entitatea **Zone** și entitatea **Angajați** se stabilește o relație de 1:n. O zonă poate să corespundă mai multor angajați, dar un angajat va livra pe o singură zonă.

Între entitatea **Zone** și entitatea **Adrese** se stabilește o relație de 1:n. O zonă cuprinde mai multe adrese, iar o adresă aparține unei singure zone.

Între entitatea **Manageri\_depozit** și entitatea **Zone** se stabilește o relație de 1:n. Un manager este responsabil de mai multe zone, iar o zonă îi va reveni unui singur manager.

Între entitatea **Tipuri\_rol** și entitatea **Angajați** se stabilește o relație de 1:n. Mai mulți angajați pot avea același rol, iar un angajat este responsabil doar de o singură sarcină generală.

Între entitatea **Adrese** și entitatea **Observații** se stabilește o relație de 1:1. O observație corespunde unei singure adrese, iar o adresă poate avea o singură observație (desigur, aceasta poate să cuprindă mai multe informații).

## Descrierea constrângerilor

Constrângerile de tip **check** se găsesc în toate entitățile. Cu ajutorul lor verificăm dacă valorile introduse pentru nume de persoane sunt corecte sau dacă valorile numerice respectă intervalele impuse.

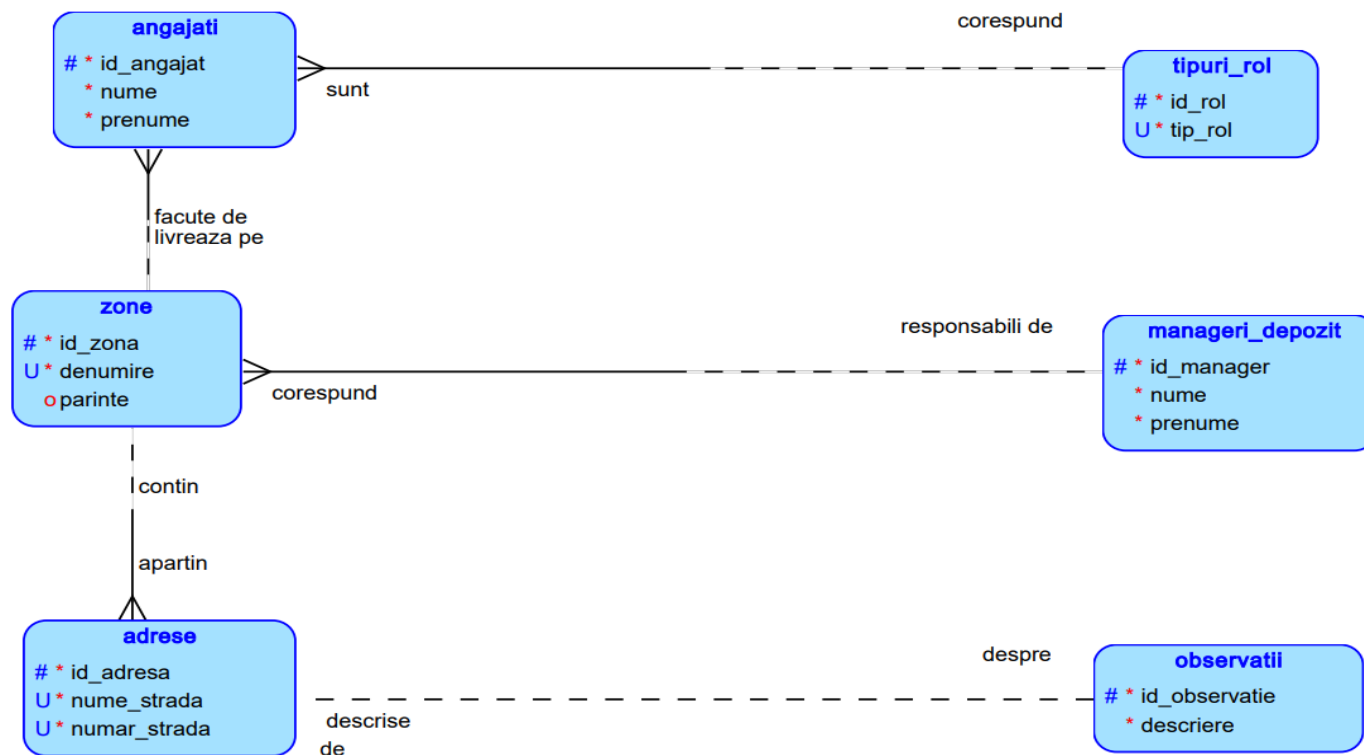
Constrângerile de tip **unique** sunt folosite pentru attributele *tip\_rol*, *denumire* (în cadrul unei zone), dar și în gruparea de attribute *nume\_strada* // *numar\_strada* (pentru a nu avea două adrese identice).

Constrângerile de tip **not null** se găsesc pe marea majoritate a atributelor din entități.

Primary key-urile sunt generate de baza de date pe baza unui mecanism de **autoincrement**.

## Modelul logic

Pe baza entităților mai sus descrise, a rezultat următorul model logic:



O descriere amănunțită a atributelor fiecărei entități se va face la prezentarea *modelului relațional*, după ce vor fi fost convertite în coloane ale tabelelor.

## Aspecte legate de normalizare

Baza de date a fost normalizată, deoarece îndeplinește următoarele condiții:

**A)** Tabelele respectă condițiile primei forme normale:

- un atribut conține valori atomice din domeniul său (și nu grupuri de astfel de valori)
- nu conține grupuri care se repetă

**B)** Tabelele respectă condițiile celei de-a doua forme normale:

- este în prima formă normală
- toate attributele non-cheie depind în totalitate de toate cheile candidat

**C)** Tabelele respectă a treia formă normală:

- este în a doua formă normală
- toate attributele non-cheie sunt direct (non-tranzitiv) dependente de toate cheile candidat.

Prima formă normală este îndeplinită în cazul tuturor tabelelor; în fiecare tabelă, un câmp poate conține doar o singură valoare din domeniul acestuia. De exemplu, pentru tabela **Angajați**, dacă s-ar dori inserarea a încă o zonă unui angajat, acest lucru este imposibil și se va recurge la o înregistrare nouă în baza

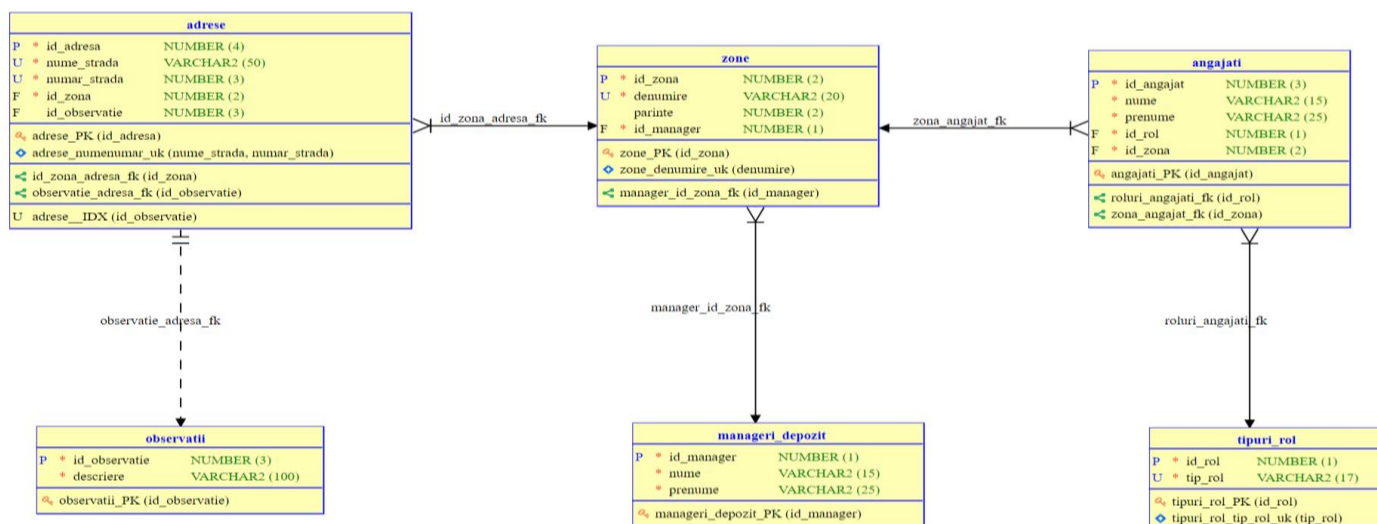
de date, cu două zone diferite.

A doua formă normală este îndeplinită de toate tabelele; nicio cheie candidat nu este de tipul cheie multiplă, deci fiecare tabelă are doar o cheie primară.

A treia formă normală este îndeplinită în cazul relației dintre tabela **Tipuri\_rol** și tabela **Angajați**; atunci când specificăm un *id\_rol*(FK) pentru un angajat, rolul acestuia va fi preluat cu ajutorul cheii primare din prima tabelă menționată.

## Modelul relațional

Pe baza modelului logic, respectiv și a relațiilor dintre entitățile acestuia, dar și în urma procesului de normalizare, a rezultat următorul model relațional:



## Descriere a tabelor finale și a coloanelor acestora

Tabelele rezultate sunt:

- Angajați
- Zone
- Adrese
- Manageri\_depozit
- Tipuri\_rol
- Observații

Tabela **Angajați** are următoarele coloane:

- *id\_angajat*: primary-key;
- *nume*;
- *prenume*;
- *id\_rol*: foreign-key; realizează legătura 1:n dintre tabela **Tipuri\_rol** și tabela

**Angajați**;

- *id\_zona*: foreign-key; specifică zona pe care va livra curierul și realizează legătura 1:n dintre tabela **Zone** și tabela **Angajați**; pot avea același *id\_zona* doar angajați cu roluri diferite.

Tabela **Tipuri\_rol** are următoarele coloane:

- *id\_rol*: primary\_key;
- *tip\_rol*: roluri ce vor fi atribuite angajaților.

Tabela **Adrese** are următoarele coloane:

- *id\_adresa*: primary-key;
- *nume\_strada*;
- *numar\_strada*;
- *id\_zona*: foreign-key; specifică în ce zonă este localizată o adresă și realizează

legătura 1:n dintre tabela **Zone** și tabela **Adrese**;

- *id\_observație*: foreign-key specifică ce observație are o anumită adresă (dacă are) și realizează legătura 1:1 dintre tabela **Adrese** și tabela **Observații**.

Tabela **Observații** are următoarele coloane:

- *id\_observație*: primary-key;
- *descriere*: specifică informații despre starea traficului, starea carosabilului, etc.

Tabela **Zone** are următoarele coloane:

- *id\_zona*: primary-key;
- *denumire*;
- *părinte*: dacă o zonă are părinte, este de tip „zonă mică”; dacă această coloană

rămâne *NULL*, zona este de tip „Cargo”;

- *id\_manager*: foreign-key; managerul responsabil de coletele ce trebuiesc livrate la adresele dintr-o anumită zonă și, implicit, responsabil și de angajații ce livrează pe acea zonă; realizează legătura 1:n dintre tabela **Manager\_depozit** și tabela **Zone**.

Tabela **Manageri\_depozit** are următoarele coloane:

- *id\_manager*: primary-key;
- *nume*;
- *prenume*.