

UNIVERSITATEA DIN BUCUREȘTI
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ
DEPARTAMENTUL DE CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI

PROIECT BAZE DE DATE

COORDONATOR:
LECT. DR. SILVIU LAURENȚIU VASILE

STUDENT:
MARCU ALEXANDRA-ELENA

BUCUREȘTI 2025

UNIVERSITATEA DIN BUCUREȘTI
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ
DEPARTAMENTUL DE CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI

SISTEM DE GESTIUNE AL CĂILOR FERATE

COORDONATOR:
LECT. DR. SILVIU LAURENȚIU VASILE

STUDENT:
MARCU ALEXANDRA-ELENA

BUCUREȘTI 2025

CUPRINS

INTRODUCERE	4
Motivarea alegerii temei	4
Software utilizat	4
Prezentarea modelului din viața reală	5
MODELUL RELAȚIONAL.....	6
Diagrama Entitate-Relație (E/R).....	6
Descrierea entităților, atributelor, cheilor	7
Descrierea relațiilor și a cardinalităților dintre entități	9
Diagrama conceptuală.....	11
Descrierea constrângerilor de integritate	13
Schemele relaționale	15
IMPLEMENTARE	16
Crearea tabelelor și definirea constrângerilor	16
Inserarea datelor.....	21
Ștergerea tabelelor	27
INCHEIERE	28
BIBLIOGRAFIE.....	28

INTRODUCERE

Motivarea alegerii temei

Am ales să realizez un proiect de baze de date despre gestiunea căilor feroviare, deoarece acest subiect a devenit o parte importantă din rutina mea săptămânală. Fiind un utilizator frecvent al serviciilor de transport feroviar, am fost mereu curioasă să înțeleg mai bine ce se află în spatele proceselor care asigură funcționarea acestui sistem complex.

Software utilizat

Pentru implementarea proiectului, am ales MYSQL ca Sistem de Gestiune a Bazelor de Date (SGBD). Ca mediu de dezvoltare, am folosit MySQL Workbench pentru a crea și gestiona baza de date. Pentru partea de interfață web, am utilizat limbajul de programare Python, împreună cu framework-ul Flask, care gestionează logica aplicației și interacțiunea cu utilizatorul. Conexiunea între aplicația web (realizată în HTML și CSS) și baza de date a fost realizată folosind librăria mysql.connector.

Prezentarea modelului din viața reală

Modelul după care a fost realizat acest proiect se bazează pe acela al operatorilor feroviari privați din România, care se ocupă de transportul persoanelor. Vom lua Transferoviar Călători drept exemplu pentru enumerarea anumitor reguli.

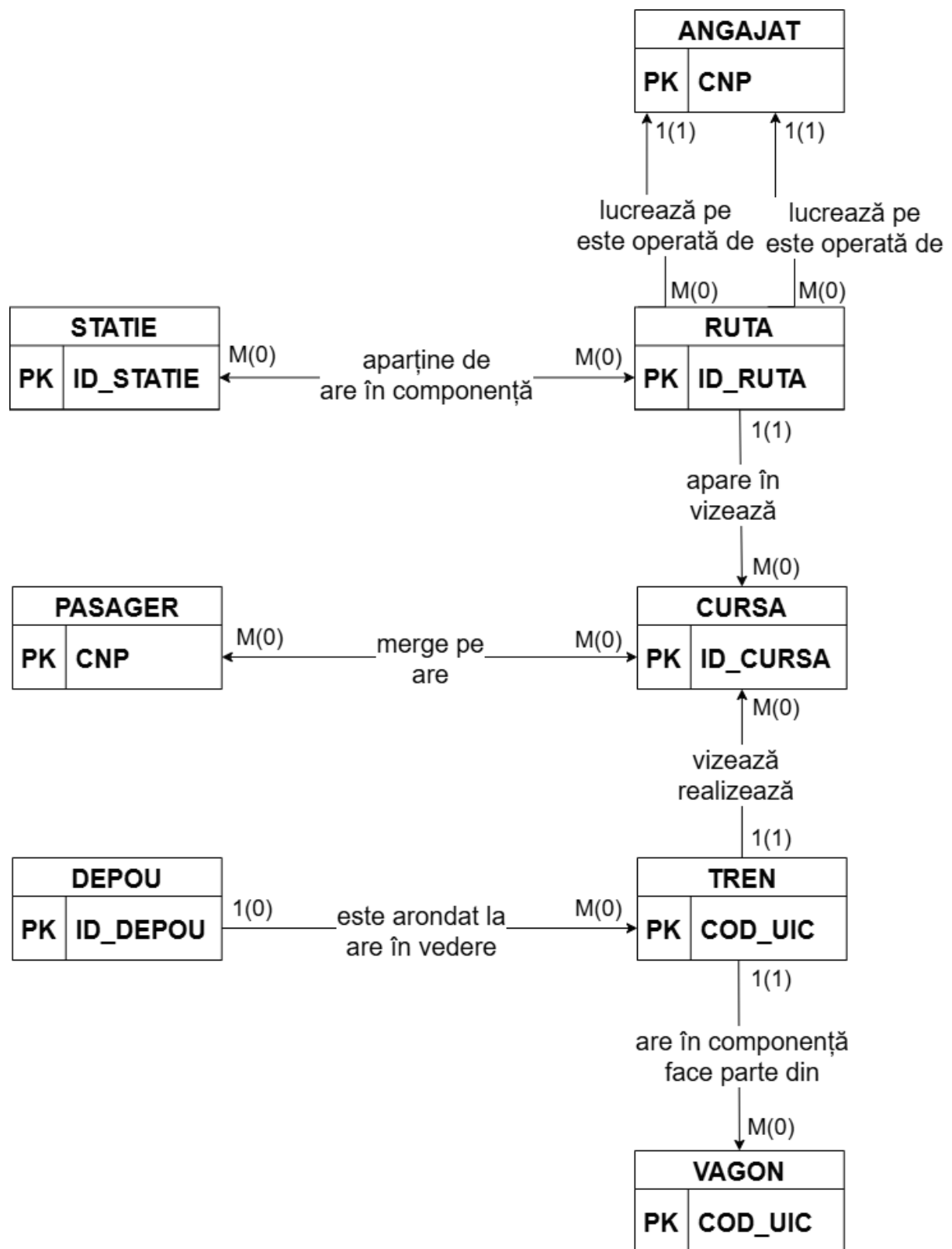
- Aceasta firmă nu are ghișee în toate stațiile, călătorul fiind obligat, în majoritatea situațiilor, să își cumpere bilet din trenul în care se urcă, de la controlorul de bilete.
- Indiferent dacă acesta cumpără bilet de la un ghișeu sau de la controlor, călătorul va primi așa numitul „bilet fără loc”.
- Dată fiind această situație anterior menționată, el își poate alege pe loc vagonul în care stă la momentul îmbarcării.
- Călătorul poate primi o reducere de 50%, 90%, respectiv 100%, în urma dovedirii statutului său (pensionar, student, elev) prin legitimare.
- În mare parte din trenuri, vagoanele puse la dispoziție sunt numai de clasa a II-a.

Regulile acestei baze de date sunt derivate din cele anterior menționate, fiind ușor simplificate, cu specificația că:

- Un călător își va cumpăra bilet pe toată ruta (implicit pentru toate stațiile), indiferent de stația la care coboară, acest sistem de plată al serviciilor funcționând similar cu cel al operatorilor de transport public din orașele mari.
- Un tren este posibil asociat unui depou (nu este automat staționat acolo, dar acolo este locul unde se va retrage, spre exemplu, în cazul unei defectiuni). Se subînțelege din diagramă că, dacă un tren nu apare cu nicio cursă asociată, trenul nu mai este în circulație și se regăsește la depoul desemnat. Este posibil ca trenul să se defecteze pe drum și să fie retras la depou, caz în care bilete asociate sunt anulate/șterse definitiv.
- În cazul în care depoul asociat trenului este desființat/scos din uz, trenul va putea merge la orice alt depou dispus să îl primească.
- Toate vagoanele sunt de clasa a II-a, deci aspect nu se mai regăsește menționat drept atribut. În schimb, ele pot fi clasificate după alte criterii.
- Vagoanele nu pot fi mutate de la un tren la altul, doar eliminate/casate și înlocuite, astfel că un vagon aparține unui singur tren pe toata durata utilizării acestuia.
- O rută este operată de un conductor și un controlor. În cazul în care unul dintre posturi nu va mai fi ocupat de un angajat, ruta va dispărea.
- Ruta unei curse nu poate fi schimbată atâta timp cât au fost bilete cumpărate pentru ea.
- În mod similar, cursa unui bilet nu poate fi schimbată după emiterea biletului și necesită cumpărarea unui alt bilet.

MODELUL RELAȚIONAL

Diagrama Entitate-Relație (E/R)



Descrierea entităților, atributelor, cheilor

DEPOU – reprezintă tabelul de informații despre depouri din toată țara.

1. **ID_DEPOU** – *INT* - identificatorul unic al unui depou.
2. **JUDET** – *VARCHAR(30)* – numele județului în care se află depoul.
3. **ORAS** – *VARCHAR(30)* – numele orașului în care se află depoul.
4. **TIP** – *ENUM* – scopul pe care îl deservește depoul: adăpostire sau mentenanță.

TREN - conține informațiile despre toate trenurile aflate la dispoziția operatorului care administrează căile ferate. În acest context, termenul „tren” se referă exclusiv la locomotivă, considerată elementul principal al ansamblului. Locomotiva definește și caracterizează întregul tren, având un rol similar cu cel al șasiului în raport cu o mașină.

1. **COD_UIC** – *VARCHAR(12)* – codul unic asociat unei locomotive, atribuit de Uniunea Internațională a Căilor Ferate. Codurile au fost preluate de pe pagina https://ro.wikipedia.org/wiki/Lista_depourilor_CFR
2. **TIP_LOCOMOTIVA** – *ENUM* – tipul de locomotivă, informație preluată din codul UIC. Două dintre cele mai întâlnite tipuri regăsite la noi în țară sunt: Diesel și Electrică, așadar tipul poate fi unul dintre acestea.
3. **MODEL_LOCOMOTIVA** – *VARCHAR(6)* – modelul/clasificarea locomotivei, care descrie locomotiva în termeni de putere și viteză maximă. Din punct de vedere al structurii acronimului, poate conține câteodată și numele producătorului, cât și tipul de locomotivă.
<https://forum.metroousor.com/ViewTopic?topicId=3228&pageNum=1>

VAGON – deține informații despre fiecare vagon asociat unui anume tren.

1. **COD_UIC** – *VARCHAR(12)* – asemenea codului pentru tren, este un cod unic asociat unui vagon, atribuit de Uniunea Internațională a Căilor Ferate. Codurile au fost generate aleatoriu în contextul prezentat pe această pagină web: <https://forum.metroousor.com/ViewTopic?topicId=3239&pageNum=1>.
2. **ETAJAT** – *ENUM* – descrie structura vagonului, respectiv dacă acesta este etajat (ia valoarea „Y”) sau nu (ia valoarea „N”) și corespunde cu a 6-a cifră din codul UIC.
3. **TIP** – *ENUM* – descrie scopul vagonului; poate lua valorile „Cușetă”, „Restaurant”, „Dormit” sau poate fi NULL (desemnând un vagon normal, doar cu locuri obișnuite), corespunzând cu a 5-a cifră din codul UIC.
4. **INALTIME** – *DECIMAL (5,2)* – conține înălțimea în metri a vagonului.

STATIE – conține informații despre stațiile din toată țara.

1. **ID_STATIE** – *VARCHAR(5)* – identificator unic, prima literă conținând mărimea stației.
2. **NUME** - *VARCHAR(30)* – numele stației (poate coincide cu numele orașului).
3. **JUDET** - *VARCHAR(30)* – județul în care se află stația.
4. **MARIME** - *ENUM* – mărimea stației; poate fi „Halta” sau „Gara”.

ANGAJAT

1. **CNP** – *VARCHAR(13)* – codul numeric personal al angajatului.
2. **NUME** – *VARCHAR(20)* – numele de familie al angajatului.
3. **PRENUME** – *VARCHAR(50)* – prenumele angajatului (poate stoca mai multe prenume).
4. **TELEFON** – *VARCHAR(10)* – numărul de telefon mobil al angajatului, fără prefixul specific țării.
5. **DATA_ANGAJARII** – *DATE* – data angajării.

UTA

1. **ID_UTA** – *INT* – identificator unic al fiecărei rute oferite de operator.
2. **DISTANTA** – *DECIMAL(6,2)* – distanța în kilometri.
3. **DURATA_ESTIMATA** – *INT* – durata temporală a rutei, exprimată în minute.
4. **PRET** – *DECIMAL(6,2)* – prețul care trebuie plătit pentru a călători dintr-un capăt în celălalt al rutei. (implicit prețul pe care trebuie să îl plătească un călător pentru a merge pe ruta respectivă, conform regulilor).

CURSA – reprezintă tabelul cu informații despre cursele pe care diverse trenuri le realizează pe anumite rute.

1. **ID_CURSA** – *INT* – identificator unic al cursei realizate.
2. **TIMP_PLEcare** – *TIMESTAMP* – ora și data la care trenul pleacă din capătul rutei.
3. **TIMP_SOSIRE** – *TIMESTAMP* – ora și data la care trenul sosește la destinație.

PASAGER

1. **CNP** – *VARCHAR(13)* - codul numeric personal al pasagerului.
2. **NUME** – *VARCHAR(20)* - numele de familie al pasagerului.
3. **PRENUME** – *VARCHAR(50)* - prenumele pasagerului (poate stoca mai multe prenume).
4. **TELEFON** – *VARCHAR(10)* – numărul de telefon mobil al pasagerului, fără prefixul specific țării.
5. **STATUT** - *VARCHAR2(20)* – poate lua valori precum „Elev”, „Student”, „Adult”, „Pensionar”.
6. **REDUCERE** – *INT* – reducerea calculată în baza statutului, aplicată pe prețul rutei. Reducerea este exprimată în procente.

Descrierea relațiilor și a cardinalităților dintre entități

DEPOU – TREN (Minimal 0:0, Maximal 1:M)

1. Un depou poate avea în vizor mai multe trenuri.
2. Un depou poate să nu aibă în vedere niciun tren.
3. Un tren poate să nu fie arondat la niciun depou.
4. Un tren poate fi repartizat la exact un depou.

TREN – VAGON (Minimal 1:0, Maximal 1:M)

1. Un tren poate să nu aibă în componență vagoane.
2. Unui tren îi pot fi atașate mai multe vagoane.
3. Un vagon trebuie să fie atașat exact unui tren.
4. Un vagon nu poate aparține de mai multe trenuri de-a lungul timpului.

TREN – CURSA (Minimal 1:0, Maximal 1:M)

1. Un tren poate să nu aibă curse.
2. Un tren poate să realizeze mai multe curse (își schimbă ruta).
3. O cursă trebuie să vizeze un singur tren.
4. O cursă nu poate deservei mai multe trenuri în același timp.

CURSA - PASAGER (Minimal 0:0, Maximal M:M)

1. Un pasager poate să nu aibă niciun bilet pentru o cursă (nu a cumpărat încă un bilet).
2. Un pasager poate să cumpere mai multe bilete (pentru curse diferite).
3. O cursă poate să nu aibă pasageri.
4. O cursă poate avea mai mulți pasageri.

RUTA - CURSA (Minimal 1:0, Maximal 1:M)

1. O ruta poate să nu aibă o cursă asociată.
2. O ruta poate avea mai multe curse asociate.
3. O cursă trebuie să vizeze o rută.
4. O cursă nu poate viza mai multe rute în același timp.

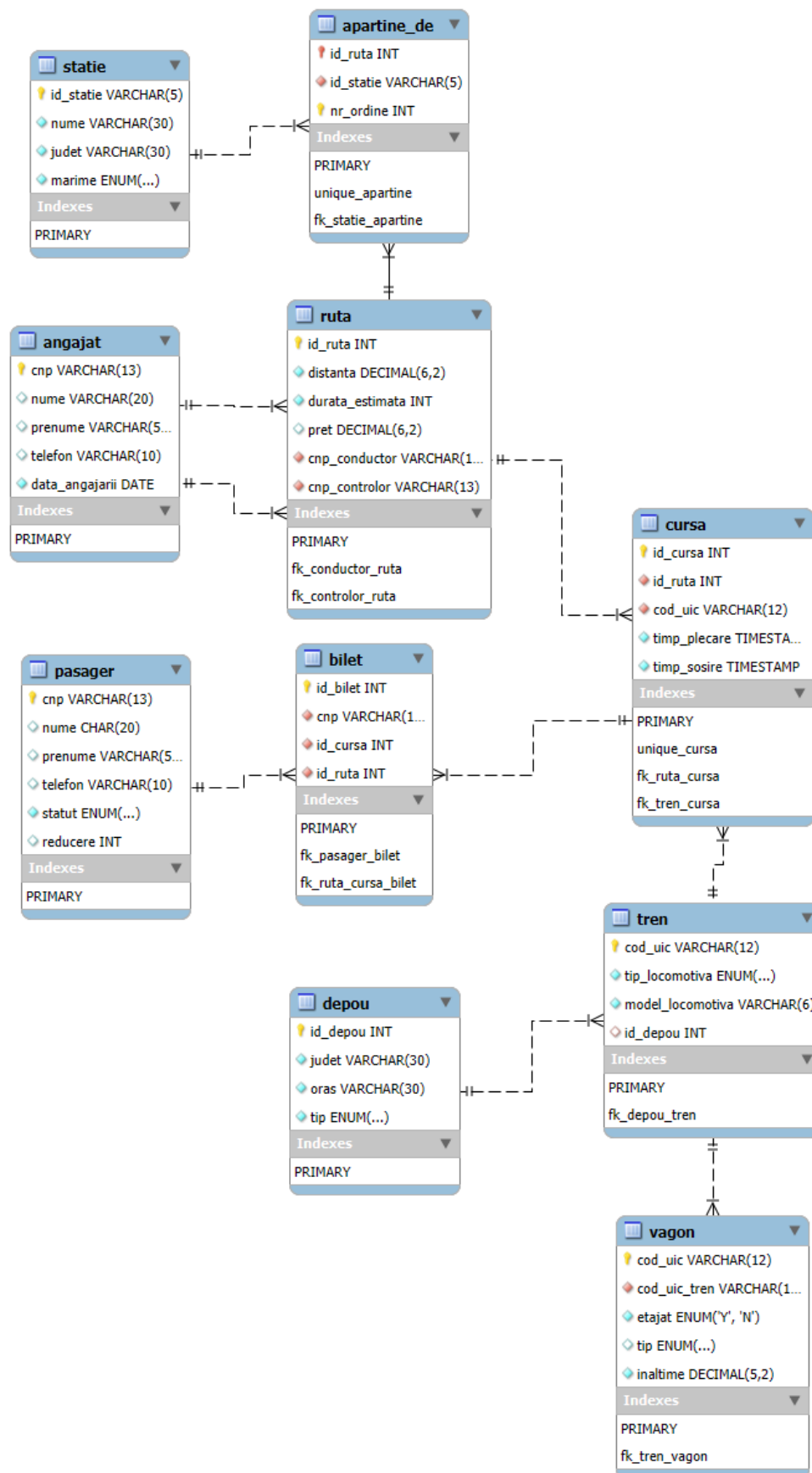
RUTA – STATIE (Minimal 0:0, Maximal M:M)

1. O stație poate să nu facă parte din nicio rută.
2. O stație poate apărea în mai multe rute.
3. O ruta poate să nu aibă stații în componență.
4. O ruta poate fi alcătuită din mai multe stații.

ANGAJAT – RUTA (Minimal 1:0, Maximal 1:M)

1. O rută trebuie să fie operată de exact doi angajați. (un conductor și un controlor)
2. O rută nu poate să fie operată de mai mult de doi angajați.
3. Un angajat poate să nu fie asociat unei rute.
4. Un angajat poate fi asociat mai multor rute.

Diagrama conceptuală



În plus pe lângă entitățile inițiale regăsim și tabelele asociative BILET și APARTINE_DE:

BILET

1. **ID_BILET** – *INT* – cheie primară a tabelului.
2. **CNP** – *VARCHAR(13)* – cheie străină.
3. **ID_CURSA** – *INT* – cheie străină.
4. **ID_RUTA** – *INT* – cheie străină.

APARTINE_DE

1. **ID_RUTA** – *INT* – cheie străină, care compune cheia primară a tabelului.
2. **ID_STATIE** – *VARCHAR(5)* – cheie străină.
3. **NR_ORDINE** – *INT* – cheie străină, care compune cheia primară a tabelului. Descrie ordinea stației în cadrul rutei.

Descrierea constrângerilor de integritate

DEPOU

1. **ID_DEPOU** – cheie primară.
2. **JUDET** – NOT NULL.
3. **ORAS** – NOT NULL.
4. **TIP** – NOT NULL.

TREN

1. **COD_UIC** – cheie primară.
2. **TIP_LOCOMOTIVA** – NOT NULL, cu valoarea „Electrica” dacă primele două cifre din COD_UIC sunt 91, sau valoarea „Diesel” pentru 92.
3. **MODEL_LOCOMOTIVA** - NOT NULL.
4. **ID_DEPOU** – cheie străină, ON DELETE SET NULL, ON UPDATE CASCADE.

VAGON

1. **COD_UIC** – cheie primară.
2. **COD_UIC_TREN** – cheie străină, ON DELETE CASCADE, ON UPDATE CASCADE.
3. **ETAJAT** – NOT NULL, cu valoarea „Y” dacă a șasea cifră din COD_UIC este 6, sau „N” dacă diferă.
4. **TIP** – cu valoarea „Cuseta” dacă a cincea cifră din COD_UIC este 4 sau 5, „Dormit” pentru cifra 7, „Restaurant” pentru cifra 8 sau nicio valoare (NULL) pentru orice altă cifră.
5. **INALTIME** – NOT NULL, mai mare ca 0.

STATIE

1. **ID_STATIE** – cheie primară.
2. **NUME** – NOT NULL.
3. **JUDET** – NOT NULL.
4. **MARIME** – cu valoarea „Halta” dacă primul caracter din cheia primară este „H”, sau „Gara” pentru caracterul „G”.

ANGAJAT

1. **CNP** – cheie primară, cu lungime de 13 caractere și prima cifră de forma 1, 2, 5 sau 6.
2. **NUME** – nicio constrângere.
3. **PRENUME** – nicio constrângere.
4. **TELEFON** – ia numai valori începând cu „07”.

5. **DATA_ANGAJARII** – NOT NULL, trebuie să fie o dată calendaristică la care angajatul ar fi avut mai mult de 18 ani, conform CNPului.

RUTA

1. **ID_RUTA** – cheie primară.
2. **DISTANTA** – NOT NULL.
3. **DURATA_ESTIMATA** – NOT NULL.
4. **PRET** – generat pe baza distanței. (implicit NOT NULL)
5. **CNP_CONDUCTOR** – cheie străină, NOT NULL, ON DELETE CASCADE, ON UPDATE CASCADE.
6. **CNP_CONTROLOR** – cheie străină, NOT NULL, ON DELETE CASCADE, ON UPDATE CASCADE.

CURSA

1. **ID_CURSA** – cheie primară, într-o combinație unică cu ID_RUTA (ea este regăsită într-o combinație unică implicit, fiind cheie primară care identifică fiecare intrare, dar această constrângere redundantă susține structura constrângerii cheii străine din tabelul BILET).
2. **ID_RUTA** – cheie străină, NOT NULL, ON DELETE CASCADE, ON UPDATE CASCADE.
3. **COD_UIC** – cheie străină, NOT NULL, ON DELETE CASCADE, ON UPDATE CASCADE.
4. **TIMP_PLECARE** – NOT NULL, data și ora de plecare trebuie să fie în trecut față de TIMP_SOSIRE
5. **TIMP_SOSIRE** – NOT NULL, data și ora de sosire trebuie să fie în viitor față de TIMP_PLECARE

PASAGER

1. **CNP** – cheie primară, cu lungime de 13 caractere și prima cifră de forma 1, 2, 5 sau 6.
2. **NUME** – nicio constrângere.
3. **PRENUME** – nicio constrângere.
4. **TELEFON** - ia numai valori începând cu „07”.
5. **STATUT** – NOT NULL, ia numai valorile „Elev”, „Student”, „Adult” sau „Pensionar”.
6. **REDUCERE** – generată pe baza valorii STATUT, 100 pentru „Elev”, 90 pentru „Student”, 50 pentru „Pensionar”, 0 în rest (implicit pentru „Adult”).

BILET

1. **ID_BILET** – cheie primară.
2. **CNP** - cheie străină, NOT NULL.
3. **ID_CURSA** – NOT NULL, parte dintr-o cheie străină alături de ID_RUTA, care la rândul său are constrângerile ON DELETE CASCADE, ON UPDATE RESTRICT.
4. **ID_RUTA** – NOT NULL, parte din cheia străină alături de ID_CURSA cu constrângerile ON DELETE CASCADE, ON UPDATE RESTRICT.

Cheia străină (ID_CURSA, ID_RUTA) asigură corectitudinea combinației coloanelor la care face referire. Constrângerea ON UPDATE RESTRICT nu permite modificarea rutei (ID_RUTA) după ce o cursă a fost înregistrată și există bilete asociate acesteia.

APARTINE_DE

1. **ID_RUTA** – NOT NULL, cheie străină, parte din cheia primară, trebuie să se regăsească într-o combinație unică cu ID_STATIE, ON DELETE CASCADE, ON UPDATE CASCADE.
2. **ID_STATIE** – NOT NULL, cheie străină, trebuie să se regăsească într-o combinație unică cu ID_RUTA, ON DELETE CASCADE, ON UPDATE CASCADE.
3. **NR_ORDINE** – NOT NULL, parte din cheia primară.

Schemele relaționale

DEPOU (ID_DEPOU#, JUDET, ORAS, TIP);

TREN (COD_UIC#, TIP_LOCOMOTIVA, MODEL_LOCOMOTIVA, ID_DEPOU [FK]);

VAGON (COD_UIC#, COD_UIC_TREN [FK], ETAJAT, TIP, INALTIME);

STATIE (ID_STATIE#, NUME, JUDET, MARIME);

ANGAJAT (CNP#, NUME, PRENUME, TELEFON, DATA_ANGAJARII);

RUTA (ID_RUTA#, DISTANTA, DURATA_ESTIMATA, PRET, CNP_CONDUCTOR [FK], CNP_CONTROLOR [FK]);

APARTINE_DE (ID_RUTA# [FK], NR_ORDINE#, ID_STATIE [FK]);

CURSA (ID_CURSA#, ID_RUTA [FK], COD_UIC [FK], TIMP_PLECARE, TIMP_SOSIRE);

PASAGER (CNP#, NUME, PRENUME, TELEFON, STATUT, REDUCERE);

BILET (ID_BILET#, CNP [FK], (ID_RUTA, ID_CURSA)[FK]);

IMPLEMENTARE

Crearea tabelelor și definirea constrângerilor

```
create table depou (  
    id_depou int primary key,  
    judet varchar(30) not null,  
    oras varchar(30) not null,  
    tip enum('Mentenanta', 'Adapostire') not null  
);
```

```
create table tren (  
    cod_uic varchar(12) primary key,  
    tip_locomotiva enum('Electrica', 'Diesel') not null,  
    model_locomotiva varchar(6) not null,  
    id_depou int,  
    constraint vf_cod_tren check (substr(cod_uic, 1, 2) in ('91', '92')  
and substr(cod_uic, 3, 2) = '53'),  
    constraint vf_tip_locomotiva check (  
        (substr(cod_uic, 1, 2) = '91' and tip_locomotiva = 'Electrica') or  
        (substr(cod_uic, 1, 2) = '92' and tip_locomotiva = 'Diesel')  
    ),  
    constraint fk_depou_tren foreign key (id_depou) references  
depou(id_depou) on delete set null on update cascade  
);
```

```
create table vagon (  
    cod_uic varchar(12) primary key,  
    cod_uic_tren varchar(12) not null,  
    etajat enum('Y', 'N') not null,  
    tip enum('Cuseta', 'Dormit', 'Restaurant'),  
    inaltime decimal(5, 2) not null,  
    constraint fk_tren_vagon foreign key (cod_uic_tren) references  
tren(cod_uic) on delete cascade on update cascade,
```



```

constraint vf_etajat_vagon check (
    (substr(cod_uic, 6, 1) = '6' and etajat = 'Y') or
    (substr(cod_uic, 6, 1) != '6' and etajat = 'N')
),
constraint vf_tip_vagon check (
    (substr(cod_uic, 5, 1) in ('4', '5') and tip = 'Cuseta') or
    (substr(cod_uic, 5, 1) = '7' and tip = 'Dormit') or
    (substr(cod_uic, 5, 1) = '8' and tip = 'Restaurant') or
    null
),
constraint vf_inaltime_vagon check (inaltime > 0)
);

create table statie (
    id_statie varchar(5) primary key,
    nume varchar(30) not null,
    judet varchar(30) not null,
    marime enum('Gara', 'Halta') not null,
    constraint vf_marime_statie check (
        (upper(substr(id_statie, 1, 1)) = 'G' and marime = 'Gara') or
        (upper(substr(id_statie, 1, 1)) = 'H' and marime = 'Halta')
    )
);

create table angajat (
    cnp varchar(13) primary key,
    nume varchar(20),
    prenume varchar(50),
    telefon varchar(10),
    data_angajarii date not null,
    constraint vf_lungime_cnp_ang check (length(cnp) = 13),

```

```

    constraint vf_nr_cnp_ang check (substr(cnp, 1, 1) in ('1', '2', '5',
'6')),
    constraint vf_tlf_ang check (telefon like '07%'),
    constraint vf_angajare_ang check (
        timestampdiff(year,
            str_to_date(
                concat(
                    case
                        when substr(cnp, 1, 1) in ('1', '2') then '19'
                        else '20'
                    end,
                    substr(cnp, 2, 6)
                ), '%Y%m%d'), data_angajarii) >= 18
    )
);

```

```

create table ruta (
    id_ruta int primary key,
    distanta decimal(6, 2) not null,
    durata_estimata int not null,
    pret decimal(6, 2) as ((distanta * 6) / 10) stored,
    cnp_conductor varchar(13) not null,
    cnp_controlor varchar(13) not null,
    constraint fk_conductor_ruta foreign key (cnp_conductor) references
angajat(cnp) on delete cascade on update cascade,
    constraint fk_controlor_ruta foreign key (cnp_controlor) references
angajat(cnp) on delete cascade on update cascade
);

```

```

create table apartine_de (
    id_ruta int not null,
    id_statie varchar(5) not null,
    nr_ordine int not null,

```

```

    constraint fk_ruta_apartine foreign key (id_ruta) references
ruta(id_ruta) on delete cascade on update cascade,

    constraint fk_statie_apartine foreign key (id_statie) references
statie(id_statie) on delete cascade on update cascade,

    constraint pk_apartine primary key(id_ruta, nr_ordine),

    constraint unique_apartine unique (id_ruta, id_statie)
);

```

```

create table cursa (
    id_cursa int primary key,
    id_ruta int not null,
    cod_uic varchar(12) not null,
    timp_plecure timestamp not null,
    timp_sosire timestamp not null,
    constraint fk_ruta_cursa foreign key (id_ruta) references
ruta(id_ruta) on delete cascade on update cascade,
    constraint fk_tren_cursa foreign key (cod_uic) references
tren(cod_uic) on delete cascade on update cascade,
    constraint vf_timp_cursa check (timp_plecure < timp_sosire)
);

```

```

create table pasager (
    cnp varchar(13) primary key,
    nume char(20),
    prenume varchar(50),
    telefon varchar(10),
    statut enum('Elev', 'Student', 'Adult', 'Pensionar') not null,
    reducere int as (
        case
            when statut = 'Elev' then 100
            when statut = 'Student' then 90
            when statut = 'Pensionar' then 50
            else 0
        end
    )
);

```

```

        end
    ) stored,
    constraint vf_lungime_cnp_pas check (length(cnp) = 13),
    constraint vf_nr_cnp_pas check (substr(cnp, 1, 1) in ('1', '2', '5',
'6')),
    constraint vf_tlf_pas check (telefon like '07%')
);

```

```

alter table cursa
add constraint unique_cursa unique (id_cursa, id_ruta);

```

```

create table bilet (
    id_bilet int primary key,
    cnp varchar(13) not null,
    id_cursa int not null,
    id_ruta int not null,
    constraint fk_pasager_bilet foreign key (cnp) references
pasager(cnp) on delete cascade on update cascade,
    constraint fk_ruta_cursa_bilet foreign key (id_cursa, id_ruta)
references cursa(id_cursa, id_ruta) on delete cascade on update
restrict
);

```

Inserarea datelor

În baza de date există mai multe inserări, însă în documentație au fost incluse doar 10 per tabel pentru exemplificare. Restul inserărilor pot fi consultate în fișierul .sql.

```
insert into depou (id_depou, judet, oras, tip) values (1, 'Bucuresti',  
'Bucuresti', 'Mentenanata');
```

```
insert into depou (id_depou, judet, oras, tip) values (2, 'Cluj',  
'Cluj-Napoca', 'Adapostire');
```

```
insert into depou (id_depou, judet, oras, tip) values (3, 'Timis',  
'Timisoara', 'Mentenanata');
```

```
insert into depou (id_depou, judet, oras, tip) values (4, 'Arad',  
'Arad', 'Adapostire');
```

```
insert into depou (id_depou, judet, oras, tip) values (5, 'Prahova',  
'Ploiesti', 'Mentenanata');
```

```
insert into depou (id_depou, judet, oras, tip) values (6, 'Arges',  
'Pitesti', 'Adapostire');
```

```
insert into depou (id_depou, judet, oras, tip) values (7, 'Sibiu',  
'Sibiu', 'Mentenanata');
```

```
insert into depou (id_depou, judet, oras, tip) values (8, 'Brasov',  
'Brasov', 'Adapostire');
```

```
insert into depou (id_depou, judet, oras, tip) values (9, 'Galati',  
'Galati', 'Mentenanata');
```

```
insert into depou (id_depou, judet, oras, tip) values (10, 'Iasi',  
'Iasi', 'Adapostire');
```

```
insert into tren (cod_uic, tip_locomotiva, model_locomotiva, id_depou)  
values ('915353040863', 'Electrica', '060EA', 2);
```

```
insert into tren (cod_uic, tip_locomotiva, model_locomotiva, id_depou)  
values ('915353041925', 'Electrica', '060EA', 2);
```

```
insert into tren (cod_uic, tip_locomotiva, model_locomotiva, id_depou)  
values ('925353062630', 'Diesel', '060DA', 2);
```

```
insert into tren (cod_uic, tip_locomotiva, model_locomotiva, id_depou)  
values ('925353062653', 'Diesel', '060DA', 2);
```

```
insert into tren (cod_uic, tip_locomotiva, model_locomotiva, id_depou)  
values ('915353046102', 'Electrica', '040EC', 3);
```

```
insert into tren (cod_uic, tip_locomotiva, model_locomotiva, id_depou)  
values ('915353046103', 'Electrica', '040EC', 3);
```

```
insert into tren (cod_uic, tip_locomotiva, model_locomotiva, id_depou)
values ('925353089309', 'Diesel', '040DHD', 3);
```

```
insert into tren (cod_uic, tip_locomotiva, model_locomotiva, id_depou)
values ('925353080437', 'Diesel', '040DHC', 3);
```

```
insert into tren (cod_uic, tip_locomotiva, model_locomotiva, id_depou)
values ('925353064092', 'Diesel', 'LDEGM', 4);
```

```
insert into tren (cod_uic, tip_locomotiva, model_locomotiva, id_depou)
values ('915353047755', 'Electrica', '060EA', 4);
```

```
insert into vagon (cod_uic, cod_uic_tren, etajat, tip, inaltime)
values ('505346051234', '915353041925', 'Y', 'Cuseta', 3.20);
```

```
insert into vagon (cod_uic, cod_uic_tren, etajat, tip, inaltime)
values ('505323040863', '915353040863', 'N', null, 3.20);
```

```
insert into vagon (cod_uic, cod_uic_tren, etajat, tip, inaltime)
values ('505323041925', '915353041925', 'N', null, 3.50);
```

```
insert into vagon (cod_uic, cod_uic_tren, etajat, tip, inaltime)
values ('505353046102', '915353046102', 'N', 'Cuseta', 3.10);
```

```
insert into vagon (cod_uic, cod_uic_tren, etajat, tip, inaltime)
values ('505353046103', '915353046103', 'N', 'Cuseta', 3.25);
```

```
insert into vagon (cod_uic, cod_uic_tren, etajat, tip, inaltime)
values ('505373089309', '925353089309', 'N', 'Dormit', 3.40);
```

```
insert into vagon (cod_uic, cod_uic_tren, etajat, tip, inaltime)
values ('505373080437', '925353080437', 'N', 'Dormit', 3.35);
```

```
insert into vagon (cod_uic, cod_uic_tren, etajat, tip, inaltime)
values ('505383040209', '915353040209', 'N', 'Restaurant', 3.50);
```

```
insert into vagon (cod_uic, cod_uic_tren, etajat, tip, inaltime)
values ('505383041210', '915353041210', 'N', 'Restaurant', 3.60);
```

```
insert into statie (id_statie, nume, judet, marime) values ('G0001',
'Bucuresti Nord', 'Bucuresti', 'Gara');
```

```
insert into statie (id_statie, nume, judet, marime) values ('G0002',
'Ploiesti Vest', 'Prahova', 'Gara');
```

```
insert into statie (id_statie, nume, judet, marime) values ('H0001',
'Mizil', 'Prahova', 'Halta');
```

```
insert into statie (id_statie, nume, judet, marime) values ('G0003',
'Iasi', 'Iasi', 'Gara');
```

```
insert into statie (id_statie, nume, judet, marime) values ('G0004',
'Cluj-Napoca', 'Cluj', 'Gara');
```

```

insert into statie (id_statie, nume, judet, marime) values ('H0002',
'Fieni', 'Dambovita','Halta');

insert into statie (id_statie, nume, judet, marime) values ('G0005',
'Brasov', 'Brasov','Gara');

insert into statie (id_statie, nume, judet, marime) values ('H0003',
'Targoviste', 'Dambovita','Halta');

insert into statie (id_statie, nume, judet, marime) values ('G0006',
'Sibiu', 'Sibiu','Gara');

insert into statie (id_statie, nume, judet, marime) values ('H0004',
'Fagaras', 'Brasov','Halta');

```

```

insert into angajat (cnp, nume, prenume, telefon, data_angajarii)
values ('1970501050018', 'Popescu', 'Ion', '0723123456',
str_to_date('2022-12-15', '%Y-%m-%d'));

insert into angajat (cnp, nume, prenume, telefon, data_angajarii)
values ('2950102034567', 'Ionescu', 'Maria', '0724123456',
str_to_date('2020-03-10', '%Y-%m-%d'));

insert into angajat (cnp, nume, prenume, telefon, data_angajarii)
values ('1960709054321', 'Vasilescu', 'Andrei', '0725123456',
str_to_date('2018-01-20', '%Y-%m-%d'));

insert into angajat (cnp, nume, prenume, telefon, data_angajarii)
values ('2990808067890', 'Dumitrescu', 'Ana', '0726123456',
str_to_date('2021-05-15', '%Y-%m-%d'));

insert into angajat (cnp, nume, prenume, telefon, data_angajarii)
values ('1961107043215', 'Pop', 'George', '0727123456',
str_to_date('2019-09-10', '%Y-%m-%d'));

insert into angajat (cnp, nume, prenume, telefon, data_angajarii)
values ('1980605045432', 'Marin', 'Elena', '0728123456',
str_to_date('2023-02-12', '%Y-%m-%d'));

insert into angajat (cnp, nume, prenume, telefon, data_angajarii)
values ('2000303032109', 'Iliescu', 'Paul', '0729123456',
str_to_date('2021-11-01', '%Y-%m-%d'));

insert into angajat (cnp, nume, prenume, telefon, data_angajarii)
values ('1950901023987', 'Rahmet', 'Cristina', '0730123456',
str_to_date('2020-08-20', '%Y-%m-%d'));

insert into angajat (cnp, nume, prenume, telefon, data_angajarii)
values ('2980406041234', 'Constantin', 'Ioana', '0731123456',
str_to_date('2022-04-10', '%Y-%m-%d'));

```

```
insert into angajat (cnp, nume, prenume, telefon, data_angajarii)
values ('1960201056789', 'Grigore', 'Vlad', '0732123456',
str_to_date('2019-07-25', '%Y-%m-%d'));
```

```
insert into ruta (id_ruta, distanta, durata_estimata, cnp_conductor,
cnp_controlor) values (1, 150.5, 120, '1970501050018',
'2950102034567');
```

```
insert into ruta (id_ruta, distanta, durata_estimata, cnp_conductor,
cnp_controlor) values (2, 320.0, 240, '1960709054321',
'2990808067890');
```

```
insert into ruta (id_ruta, distanta, durata_estimata, cnp_conductor,
cnp_controlor) values (3, 180.0, 150, '1961107043215',
'1980605045432');
```

```
insert into ruta (id_ruta, distanta, durata_estimata, cnp_conductor,
cnp_controlor) values (4, 500.0, 420, '2000303032109',
'1950901023987');
```

```
insert into ruta (id_ruta, distanta, durata_estimata, cnp_conductor,
cnp_controlor) values (5, 250.0, 180, '2980406041234',
'1960201056789');
```

```
insert into ruta (id_ruta, distanta, durata_estimata, cnp_conductor,
cnp_controlor) values (6, 400.0, 360, '2950102034568',
'2970102012345');
```

```
insert into ruta (id_ruta, distanta, durata_estimata, cnp_conductor,
cnp_controlor) values (7, 650.0, 600, '2990102045678',
'2960102045432');
```

```
insert into ruta (id_ruta, distanta, durata_estimata, cnp_conductor,
cnp_controlor) values (8, 320.5, 260, '2940102023423',
'2910102036574');
```

```
insert into ruta (id_ruta, distanta, durata_estimata, cnp_conductor,
cnp_controlor) values (9, 450.0, 390, '2900102034421',
'2980102027654');
```

```
insert into ruta (id_ruta, distanta, durata_estimata, cnp_conductor,
cnp_controlor) values (10, 600.0, 540, '2930102021289',
'2990102048592');
```

```
insert into apartine_de (id_ruta, id_statie, nr_ordine) values (1,
'G0001', 1);
```

```
insert into apartine_de (id_ruta, id_statie, nr_ordine) values (1,
'G0002', 2);
```



```
insert into apartine_de (id_ruta, id_statie, nr_ordine) values (1,
'H0001', 3);
```

```
insert into apartine_de (id_ruta, id_statie, nr_ordine) values (1,
'G0003', 4);
```

```
insert into apartine_de (id_ruta, id_statie, nr_ordine) values (2,
'G0004', 1);
```

```
insert into apartine_de (id_ruta, id_statie, nr_ordine) values (2,
'H0002', 2);
```

```
insert into apartine_de (id_ruta, id_statie, nr_ordine) values (2,
'H0003', 3);
```

```
insert into apartine_de (id_ruta, id_statie, nr_ordine) values (3,
'G0005', 1);
```

```
insert into apartine_de (id_ruta, id_statie, nr_ordine) values (3,
'H0004', 2);
```

```
insert into apartine_de (id_ruta, id_statie, nr_ordine) values (3,
'G0006', 3);
```

```
insert into cursa (id_cursa, id_ruta, cod_uic, timp_plecare,
timp_sosire) values (1, 1, '915353040863', '2024-12-29 22:00:00',
'2024-12-30 00:00:00');
```

```
insert into cursa (id_cursa, id_ruta, cod_uic, timp_plecare,
timp_sosire) values (2, 2, '915353041925', '2024-12-30
06:00:00', '2024-12-30 10:00:00');
```

```
insert into cursa (id_cursa, id_ruta, cod_uic, timp_plecare,
timp_sosire) values (3, 3, '915353040209', '2024-12-30
14:00:00', '2024-12-30 16:30:00');
```

```
insert into cursa (id_cursa, id_ruta, cod_uic, timp_plecare,
timp_sosire) values (4, 4, '915353046102', '2024-12-31
08:15:00', '2024-12-31 15:15:00');
```

```
insert into cursa (id_cursa, id_ruta, cod_uic, timp_plecare,
timp_sosire) values (5, 5, '925353062630', '2024-12-31
11:30:00', '2024-12-31 14:30:00');
```

```
insert into cursa (id_cursa, id_ruta, cod_uic, timp_plecare,
timp_sosire) values (6, 16, '925353080541', '2024-12-29
20:00:00', '2024-12-30 01:30:00');
```

```
insert into cursa (id_cursa, id_ruta, cod_uic, timp_plecare,
timp_sosire) values (7, 7, '925353062653', '2024-12-30 18:00:00',
'2024-12-30 23:00:00');
```

```
insert into cursa (id_cursa, id_ruta, cod_uic, timp_plecare,
timp_sosire) values (8, 8, '925353064092', '2024-12-31 05:30:00',
'2024-12-31 09:00:00');
```

```
insert into cursa (id_cursa, id_ruta, cod_uic, timp_plecare,
timp_sosire) values (9, 9, '925353062768', '2024-12-31 09:45:00',
'2024-12-31 16:00:00');
```

```
insert into cursa (id_cursa, id_ruta, cod_uic, timp_plecare,
timp_sosire) values (10, 10, '925353064925', '2024-12-31
14:00:00', '2024-12-31 23:00:00');
```

```
insert into pasager (cnp, nume, prenume, telefon, statut) values
('5020202123456', 'Carut', 'Maria', '0722000002', 'Student');
```

```
insert into pasager (cnp, nume, prenume, telefon, statut) values
('5010101123456', 'Marcu', 'Andrei-Radu', '0722000001', 'Adult');
```

```
insert into pasager (cnp, nume, prenume, telefon, statut) values
('5030303123456', 'Dumitrescu', 'Alexandru', '0722000003',
'Pensionar');
```

```
insert into pasager (cnp, nume, prenume, telefon, statut) values
('5040404123456', 'Stanescu', 'Ioana-Ana-Maria', '0722000004',
'Elev');
```

```
insert into pasager (cnp, nume, prenume, telefon, statut) values
('5050505123456', 'Mihai', 'Cristian', '0722000005', 'Adult');
```

```
insert into pasager (cnp, nume, prenume, telefon, statut) values
('6010101123456', 'Radu', 'Elena', '0722000006', 'Student');
```

```
insert into pasager (cnp, nume, prenume, telefon, statut) values
('6020202123456', 'Tudor', 'Vlad', '0722000007', 'Pensionar');
```

```
insert into pasager (cnp, nume, prenume, telefon, statut) values
('6030303123456', 'Neagu', 'Bianca', '0722000008', 'Elev');
```

```
insert into pasager (cnp, nume, prenume, telefon, statut) values
('2970417123456', 'Dragomirisc', 'Sorin', '0722000009', 'Pensionar');
```

```
insert into pasager (cnp, nume, prenume, telefon, statut) values
('2860901123456', 'Vasile', 'Alina', '0722000010', 'Adult');
```

```
insert into pasager (cnp, nume, prenume, telefon, statut) values
('5040913123456', 'Pruna', 'Andrei', '0722000047', 'Student');
```

```

insert into bilec (id_bilet, cnp, id_cursa, id_ruta) values (1,
'5020202123456', 1, 1);

insert into bilec (id_bilet, cnp, id_cursa, id_ruta) values (2,
'5010101123456', 5, 5);

insert into bilec (id_bilet, cnp, id_cursa, id_ruta) values (3,
'5030303123456', 3, 3);

insert into bilec (id_bilet, cnp, id_cursa, id_ruta) values (4,
'5040404123456', 2, 2);

insert into bilec (id_bilet, cnp, id_cursa, id_ruta) values (5,
'5040404123456', 7, 7);

insert into bilec (id_bilet, cnp, id_cursa, id_ruta) values (6,
'5040404123456', 14, 14);

insert into bilec (id_bilet, cnp, id_cursa, id_ruta) values (7,
'6010101123456', 6, 16);

insert into bilec (id_bilet, cnp, id_cursa, id_ruta) values (8,
'6010101123456', 15, 3);

insert into bilec (id_bilet, cnp, id_cursa, id_ruta) values (9,
'6030303123456', 4, 4);

insert into bilec (id_bilet, cnp, id_cursa, id_ruta) values (10,
'6020202123456', 10, 10);

```

Ștergerea tabelelor

```

drop table bilec;
drop table pasager;
drop table cursa;
drop table apartine_de;
drop table ruta;
drop table angajat;
drop table statie;
drop table vagon;
drop table tren;
drop table depou;

```

INCHEIERE

Mi s-a părut interesant să explorez modul în care datele legate de trenuri, rute, orare, bilete și pasageri pot fi organizate și gestionate eficient. Proiectarea unei baze de date pentru căile feroviare reprezintă pentru mine o oportunitate de a combina utilul cu plăcutul: pe de o parte, să-mi dezvolt abilitățile în domeniul bazelor de date, iar pe de altă parte, să creez un model care să reflecte un sistem cu care interacționez atât de des.

BIBLIOGRAFIE

- https://ro.wikipedia.org/wiki/Lista_depourilor_CFR
- https://en.wikipedia.org/wiki/UIC_identification_marking_for_tractive_stock
- https://en.wikipedia.org/wiki/List_of_UIC_country_codes
- https://docs.oracle.com/cd/E17952_01/mysql-5.7-en/create-table-generated-columns.html
- <https://transferoviarcalatori.ro/ro/pentru-calatori/tarife/tarife-generale>
- <https://forum.metroususor.com/ViewTopic?topicId=3228&pageNum=1>
- <https://forum.metroususor.com/ViewTopic?topicId=3239&pageNum=1>