

Gestionarea unei flote de transport

Coordonator:

Prof. Mironeanu Cătălin

Student,

Stoian Mario-Daniel

Grupa 1312A

Iasi, 2024

Titlul proiectului

Gestionarea unei flote de transport

Analiza, proiectarea și implementarea unei baze de date care să modeleze gestiunea unei flote de transport.

Descrierea proiectului

Informațiile de care avem nevoie sunt legate de: vehiculele pe care flota le are la dispoziție, șoferii, orașele din care pleacă și unde trebuie să ajungă mașinile, departamentele la care trebuie să se întoarcă mașinile din flotă după ce au fost luate, rutele pe care se deplasează, salariile șoferilor și momentele când fiecare vehicul este ocupat.

În prima parte vom avea nevoie de informații legate de vehiculul care o să fie folosit. Fiecare vehicul este indentificat prin: id, marca vehiculului (flota deține vehicule cu marca Mercedes-Benz, Volvo, Scania, MAN, DAF), modelul acestora (modelele deținute de această flotă sunt Actos, FH16, R500, TGX, XF105), anul de fabricare, VIN-ul (Vehicle Identification Number) și locația în care acest vehicul se află la un moment dat (fiecare mașină inițial se afla la câte unul din departamentele acestei flote). Fiecare vehicul este particularizat printr-o serie de detalii: id-ul său și VIN-ul acestuia. Inițial un client o să își aleagă o anumită mașină care circulă pe o anumită rută.

În a doua parte ne vom ocupa de șoferul care o să conducă vehiculul

selectat. Fiecare șofer este identificat printr-un id, numele complet al acestuia, numărul permisului de conducere, statusul (care reprezintă dacă șoferul este disponibil sau nu) și acțiunile desfășurate (care reprezintă cursele făcute în luna respectivă). Fiecare șofer este particularizat printr-o serie de detalii: id-ul acestuia și numărul permisului de conducere. După selectarea unei mașini, clientul va alege un șofer care să conducă pe ruta selectată. Se vor putea selecta doar șoferii liberi, adică cei care nu au o cursă în desfășurare sau au adus vehiculul înapoi după efectuarea unei curse.

În a treia parte vom rezolva problemele ce pot apărea în cadrul gestionării flotei. Acestea pot să fie legate de datele închirierii unui vehicul, legate de indisponibilitatea unui anumit șofer și legate de rutele disponibile momentan pentru transport. Problemele pot apărea atunci când un anumit vehicul trebuie să fie reparat sau verificat pentru o anumită dată, lucru care îl va face indisponibil pentru închiriere. Altă problemă poate să reprezinte dorința de a se închiria un vehicul care încă nu s-a întors, lucru care îl va face și de această dată indisponibil. În cazul șoferilor, indisponibilitatea va apărea atunci când încă nu s-au întors dintr-o cursă anterioară, iar în cazul rutelor se va prezenta indisponibilitatea în cazul în care se dorește închirierea pe rute indisponibile momentan.

Descrierea funcțională a aplicației

Principalele funcții ale aplicației sunt:

- Evidența vehiculelor care se află în flotă
- Evidența șoferilor
- Evidența rutelor de deplasare
- Evidența zilelor în care vehiculele sunt indisponibile
- Evidența plăților efectuate către șoferi
- Evidența departamentelor din orașele în care trebuie să se întoarcă vehiculele

Structura și relațiile dintre tabele

Tabelele din aceasta aplicație sunt:

❖ Departments:

- Department_id
- Department_Name
- Location
- Route_id

❖ Drivers:

- Driver_id
- Name
- License_number
- Status
- Actions
- Department_id

❖ Maintenance:

- Maintenance_id

- Vehicle_id
- Maintenance_date
- Details
- ❖ Payments:
 - Payment_id
 - Driver_id
 - Amount
 - Payment_date
- ❖ Routes:
 - Route_id
 - Start_location
 - End_location
 - Distance
- ❖ Vehicles:
 - Vehicle_id
 - Make
 - Model
 - Year
 - VIN
 - Location
 - Driver_id

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

Între tabela **Departments** și tabela **Routes** se stabilește o relație de 1:1. Fiecare oraș de plecare pentru o anumită rută are câte un departament al companiei care gestionează flota pentru a dezvolta firma care gestionează flota în mai multe orașe. Legătura dintre cele două tabele se face prin câmpul **ROUTE_ID**.

Între tabela **Departments** și tabela **Drivers** se stabilește o relație 1:n. Un departament poate avea mai mulți șoferi, dar fiecare șofer este asociat cu un singur departament. Legatura dintre cele două tabele se face prin campul **DEPARTMENT_ID**.

Între tabela **Drivers** și tabela **Vehicles** se stabilește o relație 1:n. Un șofer poate conduce mai multe vehicule, dar fiecare vehicul este asociat cu un singur șofer. Legatura dintre cele două tabele se face prin campul **DRIVER_ID**.

Între tabela **Vehicles** și tabela **Managers** se stabilește o relație 1:n. Un vehicul poate avea mai multe sesiuni de întreținere, dar fiecare sesiune de întreținere este asociată unui singur vehicul. Legatura dintre cele două tabele se face prin campul **VEHICLE_ID**.

Între tabela **Drivers** și tabela **Payments** se stabilește o relație 1:n. Un șofer poate avea mai multe plăți, dar fiecare plată se face pentru un singur șofer. Legatura dintre cele două tabele se face prin campul **DRIVER_ID**.

Explicarea normalizării:

Prima formă normal: Structura de tabele se află în prima formă normală pentru că la mine toate coloanele reprezintă o singură valoare, nu există coloane care să reprezinte două valori în același timp. Și de

asemenea nici nu am coloane separate care pot reprezenta același lucru, fiecare reprezintă un lucru diferit.

A doua formă normal: Putem observa de mai sus că toate tabelele sunt în prima formă normală deci mai rămâne doar a doua cerință. Deoarece nu există dependențe parțiale, fiecare câmp non-cheie depinde complet de cheia primară, astfel a doua cerință este și ea verificată.

A treia formă normal: Putem observa de mai sus că toate tabelele sunt în a doua formă normală deci mai rămâne doar a doua cerință. Deoarece nu există dependențe transitive, putem spune că și a doua cerință este verificată.

Forma normală Boyce-Codd: O tabelă este în BCNF dacă, pentru orice dependență funcțională, cheia stângă este o supercheie

- Tabelul Departments: Cheia primară este department_id, iar route_id are constrângere de unicitate, fiind asociat cu cheia primară din tabela Routes. Așadar, respectă BCNF.
- Tabelul Drivers: Cheia primară este driver_id, iar license_number are constrângerea de unicitate. Toate dependențele funcționale sunt corect gestionate.
- Tabelul Vehicles: Cheia primară este vehicle_id, iar vin are constrângerea de unicitate. Relația cu șoferul este gestionată corect.
- Tabelul Routes: Cheia primară este route_id, iar toate dependențele funcționale sunt corect gestionate.
- Tabelul Maintenance: Cheia primară este maintenance_id, iar vehicle_id este o cheie externă. Dependențele sunt corect gestionate.

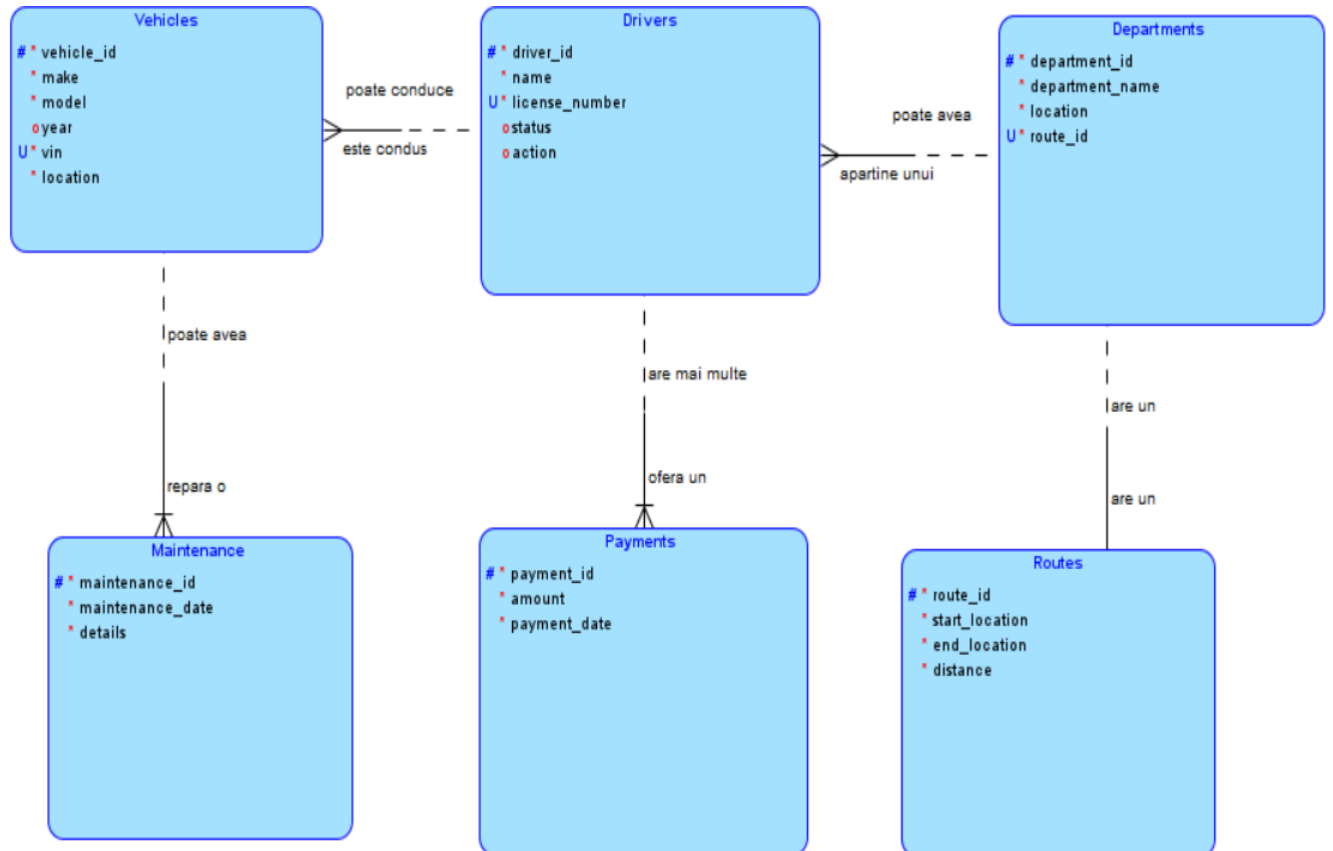
- Tabelul Payments: Cheia primară este payment_id, iar driver_id este o cheie externă. Dependențele sunt corect gestionate.

Toate tabelele respectă BCNF, deoarece fiecare dependență funcțională este corect legată de o supercheie.

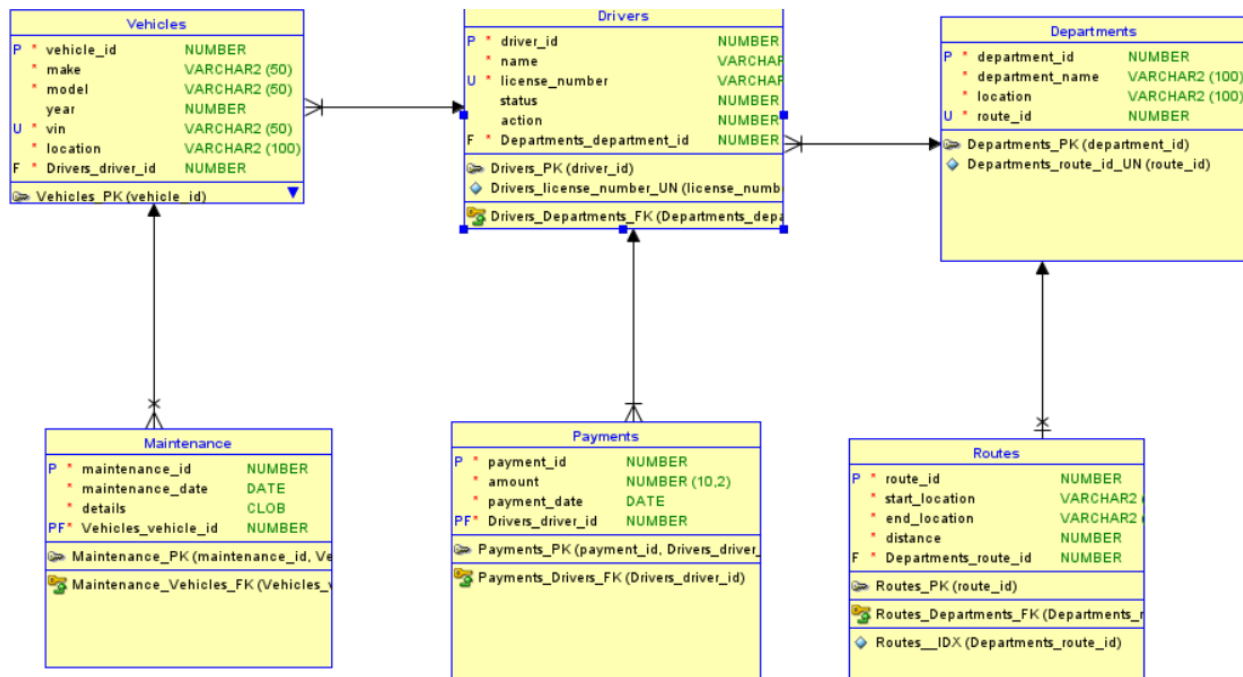
A patra formă normal: Tabelele sunt în a patra formă normal, deoarece nu există dependențe multivalente. Nu există combinații de attribute care să depindă simultan de cheia primară și să formeze seturi de valori independente între ele.

A cincea formă normal: Tabelele sunt în a cincea formă normal, deoarece nu există dependențe de join și fiecare tabel este descompus într-o formă care nu poate fi divizată suplimentar fără a pierde informații.

Modelul logic



Model relațional



Descrierea constrangerilor

- Constrangerile de tip **CHECK** sunt utilizate pentru a valida valorile introduse în tabele și pentru a preveni erorile de scriere sau alte greseli. Aceste constrangeri sunt folosite pentru a verifica distanța dintre locația de plecare și cea de sosire pentru că nu ar avea sens o rută în care vehiculul nu pleacă din punctul în care este. Astfel distanța este verificată să fie mai mare decât 0.
- Constrangerile de tip **UNIQUE** sunt folosite pentru a asigura că anumite atribute din tabelele bazei de date sunt unice. Aceste constrangeri sunt aplicate pentru:

- Numarul de licența al șoferilor: Asigură că fiecare șofer are un numar de licența unic.
 - VIN-ul vehiculelor: Fiecare vehicul din tabela trebuie să aibă un VIN unic pentru a evita duplicarea vehiculelor.
 - Id-ul rutelor: Fiecare rută din tabelă trebuie să aibă un route_id unic pentru a nu duplica traseele.
- Constrangerile de tip **PRIMARY KEY** sunt utilizate pentru a identifica in mod unic fiecare rând din tabelă. Aceste constrangeri sunt aplicate pentru:
 - ID-ul vehiculului: Fiecare vehicul din tabela are un ID unic.
 - ID-ul șoferului: Fiecare șofer din tabela are un ID unic.
 - ID-ul rutei: Fiecare rută din tabela are un ID unic.
 - ID-ul întreținerii: Fiecare întreținere din tabela are un ID unic.
 - ID-ul departamentului: Fiecare departament din tabela are un ID unic.
 - ID-ul plății: Fiecare plată din tabela are un ID unic.
 - Constrangerile de tip **FOREIGN KEY** sunt folosite pentru a stabili relațiile între tabele si pentru a asigura integritatea referențială.
 - Legătura între **Drivers** și **Departments**: Fiecare șofer trebuie să fie asociat cu un departament valid din tabela **Departments**.
 - Legătura între **Vehicles** și **Drivers**: Fiecare vehicul trebuie să fie asociat cu un șofer valid din tabela **Drivers**.
 - Legătura între **Maintenance** și **Vehicles**: Fiecare înregistrare de întreținere trebuie să fie asociată cu un vehicul valid din tabela **Vehicles**.
 - Legătura între **Payments** și **Drivers**: Fiecare plată trebuie să fie asociată cu un șofer valid din tabela **Drivers**.

- Legătura între **Departments** și **Routes**: Fiecare departament poate fi asociat cu o rută din tabela **Routes**.
- Constrangerile de tip **NOT NULL** sunt folosite pentru a asigura ca anumite attribute nu pot fi lasate goale. Aveste constrageri se aplica asupra:
 - department_id: Este folosit ca cheie primară, iar fiecare departament trebuie să aibă un ID unic, deci nu poate fi NULL.
 - location: Un departament trebuie sa fie neaparat repartizat într-un oras.
 - department_name: Numele departamentului este esențial pentru identificarea acestuia. Nu poate fi NULL, deoarece este o valoare fundamentală pentru identificarea unui departament
 - route_id: Fiecare departament trebuie să fie asociat cu o rută, iar route_id trebuie să fie specificat pentru a asocia departamentul cu ruta corespunzătoare.
 - driver_id: Este cheia primară a tabelului, iar fiecare șofer trebuie să aibă un ID unic. Acesta nu poate fi NULL.
 - name: Numele șoferului este esențial pentru identificarea acestuia. Nu poate fi NULL.
 - license_number: Numărul permisului de conducere al șoferului este obligatoriu pentru a identifica șoferul. Nu poate fi NULL, deoarece este necesar pentru validarea legală.
 - department_id: Acest câmp face legătura între șoferi și departamentele lor. Poate fi NULL pentru șoferii care nu sunt încă alocați unui departament, dar se presupune că va fi completat în majoritatea cazurilor.
 - vehicle_id: Este cheia primară a tabelului, iar fiecare vehicul trebuie să aibă un ID unic. Nu poate fi NULL.
 - make: Marca vehiculului este esențială pentru identificarea acestuia. Nu poate fi NULL.

- model: Modelul vehiculului este necesar pentru a distinge între diferitele tipuri de vehicule. Nu poate fi NULL.
- vin: Numărul de identificare al vehiculului (VIN) este unic și necesar pentru a identifica vehiculul în mod oficial. Nu poate fi NULL.
- location: Locația vehiculului este importantă pentru gestionarea și localizarea acestuia. Nu poate fi NULL.
- route_id: Este cheia primară a tabelului, iar fiecare rută trebuie să aibă un ID unic. Nu poate fi NULL.
- start_location: Locația de plecare a rutei este esențială pentru definirea rutei. Nu poate fi NULL.
- end_location: Locația de destinație a rutei este esențială pentru definirea rutei. Nu poate fi NULL.
- distance: Distanța rutei este importantă pentru gestionarea timpilor și costurilor de transport. Nu poate fi NULL.
- maintenance_id: Este cheia primară a tabelului, iar fiecare operațiune de întreținere trebuie să aibă un ID unic. Nu poate fi NULL.
- vehicle_id: Fiecare operațiune de întreținere trebuie să fie asociată cu un vehicul, iar vehicle_id este esențial pentru această asociere. Nu poate fi NULL.
- maintenance_date: Data întreținerii este esențială pentru urmărirea istoricului vehiculului și a reparațiilor efectuate. Nu poate fi NULL.
- details: Detaliile întreținerii sunt importante pentru a documenta ce reparații sau verificări au fost efectuate asupra vehiculului. Nu poate fi NULL.
- payment_id: Fiecare plată trebuie să fie asociată cu un șofer, iar driver_id este esențial pentru această asociere. Nu poate fi NULL.
- amount: Suma plății este esențială pentru a înregistra corect tranzacțiile financiare. Nu poate fi NULL.

- payment_date: Data plății este necesară pentru urmărirea istoricului plăților efectuate șoferilor. Nu poate fi NULL.

Descrierea use-case-urilor

- Ca un prim use-case am ales scopul pentru care a fost creată această bază de date. În acest use-case se dorește să se închirieze o anumită mașină cu un anumit șofer pe o anumită rută. Pentru acest use case se pot găsi suficiente probleme cum ar fi indisponibilitatea șoferului, indisponibilitatea mașinii, mașina selectată să nu existe în baza de date(această selecție având loc după id-ul vehiculului), ruta fiind una nefavorabilă pentru mașina selectată ea fiind în altă locație la momentul actual. După verificarea acestor excepții trebuie să fie adăugate în tabelă indisponibilitatea șoferului și adăugarea unei curse în dreptul acestuia. Pe lângă acestea se adaugă închirierea în tabelul maintenance pentru a oferi indisponibilitate vehiculului în acea dată și modificarea locației mașinii pentru ca aceasta va trebui să fie adusă înapoi pe viitor la locația inițială.
- Al doilea use-case o să fie unul legat strict de șoferi. Fiecare cursă făcută de fiecare șofer este contorizată și asta determină dacă acesta primește salariu sau nu, firma care gestionează flota de transport a decis că după o lună de la primul salariu se va verifica dacă fiecare șofer a făcut minim 3 curse. În cazul în care numărul minim de curse este atins se va mai adăuga o plată pe acea lună, în caz contrar șoferul nu va primi plata luna respectivă.
- Al treilea use-case o să fie specific zilelor de mentenanță. Se va verifica dacă data curentă este una din datele din tabela de mentenanță și în acel caz șoferul responsabil de mașina ce se află în mentenanță va fi trecut indisponibil.

- Al patrulea use-case o să reprezinte întoarcerea înapoi a mașinilor după o cursă. După orice cursă șoferul o să aducă mașina înapoi la departamentul aferent ei. Acesta se va observa prin faptul că repartiția mașinilor este câte una per departament.