A picture containing icon

Description automatically generated

UNIVERSITATEA TEHNICĂ „GHEORGHE ASACHI” IASI

FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE

SPECIALIZAREA CALCULATOARE ȘI TEHNOLOGIA INFORMAȚIEI

DISCIPLINA: BAZE DE DATE

**Sistem de Gestiune pentru Service Auto**

**Coordonator,**

**Prof. Cătălin Mironeanu**

**Student,**

**Petcu Andrei-Dănuț**

**Grupa 1312B**

**Iași, 2024**

### Descrierea Proiectului:

**Scopul aplicației:**  
Proiectul vizează dezvoltarea unei soluții informatice pentru gestionarea eficientă a activității într-un service auto. Acest sistem permite înregistrarea și monitorizarea clienților, vehiculelor și problemelor tehnice într-un mod organizat, dar și satisfacția clienților față de service.

**Funcționalități cheie:**

* Înregistrarea clienților și a vehiculelor lor, inclusiv marca, modelul și problemele raportate.
* Gestionarea și planificarea programărilor, evitând suprapunerea acestora datorită numărului limitat de locuri disponibile.
* Evidența problemelor asociate fiecărui vehicul, incluzând costuri estimate și termene de finalizare.
* Alocarea și evaluarea mecanicilor pentru fiecare vehicul, bazată pe competențe și evaluări anterioare.
* Integrarea unui sistem de feedback pentru clienți, pentru a urmări gradul de satisfacție.

**Obiective:** Aplicația are ca scop principal reducerea erorilor de gestionare, creșterea transparenței față de clienți și optimizarea alocării resurselor, asigurând astfel o experiență mai bună între clienți și personalul service-ului.

### Structura și Inter-relaționarea Tabelelor:

### Modelul logic

### Modelul relational

**Normalizare:**

Tabelele au fost normalizate conform celui de-al treilea normal form (3NF) pentru a evita redundanțele și pentru a asigura o mai bună integrare a datelor.

#### **Inter-relaționarea tabelelor:**

1. **Client - Detalii Client**: Relația este unu-unu (1:1) între client și detalii\_client. Un client poate avea o singură descriere (email, CNP, vârstă).
2. **Client - Mașină**: Relația este unu-unu (1:1) între client și masina. Un client vine cu o singura mașină.
3. **Mașină - Programare**: Relația este unu-unu (1:1) între masina și data\_programare. Fiecare mașină poate avea doar o programare asociata.
4. **Mașină - Mecanic**: Relația este unu-unu (1:1) între masina și mecanic. O mașină poate fi lucrată de un singur mecanic.
5. **Mașină - Problema**: Relația este unul-multiplu (1:n) între masina și problema. O mașină poate avea mai multe probleme asociate.

### Descrierea Coloanelor din Tabele

#### Tabela: Client

* id\_client (NUMBER(4))
  + Tip: Numeric, lungime fixă de 4 cifre
  + Utilizare: Identificator unic pentru fiecare client
* nume\_client (VARCHAR2(50))
  + Tip: Varchar pentru a stoca numele clientului si Not Null
  + Utilizare: Numele complet al clientului, max. 50 caractere
* grad\_satisfactie\_client (NUMBER(2))
  + Tip: Numeric cu restricție între 1 și 10 pentru a înregistra satisfacția clientului

#### Tabela: Detalii\_Client

* cnp (CHAR(13))
  + Tip: Caracter fix pentru CNP, necesar conform reglementărilor legale si Not Null
  + Utilizare: Identificator unic pentru fiecare client din România
* varsta (NUMBER(2))
  + Tip: Numeric, restricție între 18 și 99 pentru vârstă
  + Utilizare: Stocarea vârstei clientului
* email (VARCHAR2(30))
  + Tip: Varchar pentru stocarea adresei de email, max. 30 caractere
* client\_id\_client (NUMBER(4))
  + Tip: Numeric, legătură cu Client (FK)

#### Tabela: Masina

* id\_masina (NUMBER(4))
  + Tip: Numeric, identificator unic pentru fiecare mașină
* marca\_auto (VARCHAR2(20))
  + Tip: Varchar pentru stocarea mărcii auto, Not Null
* model\_auto (VARCHAR2(20))
  + Tip: Varchar pentru stocarea modelului, Not Null
* data\_intrare\_masina\_service (DATE)
  + Tip: Date, pentru a înregistra data intrării mașinii în service, Not Null
* client\_id\_client (NUMBER(4))
  + Tip: Numeric, legătură cu client (FK)

#### Tabela: Data\_Programare

* id\_programare (NUMBER(4))
  + Tip: Numeric, identificator unic pentru programare
* data\_programare (DATE)
  + Tip: Date, pentru stocarea datei programării, Not Null
* masina\_id\_masina (NUMBER(4))
  + Tip: Numeric, legătură cu masina (FK)

#### Tabela: Mecanic

* id\_mecanic (NUMBER(2))
  + Tip: Numeric, identificator unic pentru mecanic
* rating\_mecanic (NUMBER(2))
  + Tip: Numeric, restricție între 1 și 10 pentru ratingul mecanicului
* masina\_id\_masina (NUMBER(4))
  + Tip: Numeric, legătură cu masina (FK)

#### Tabela: Problema

* id\_problema (NUMBER(2))
  + Tip: Numeric, identificator unic pentru problemă
* cost (NUMBER(4))
  + Tip: Numeric, restricție între 0 și 9999 pentru costul problemei, Not Null
* problema (VARCHAR2(100))
  + Tip: Varchar pentru descrierea problemei, max. 100 caractere, Not Null
* termen\_finalizare\_problema (DATE)
  + Tip: Date, pentru a înregistra data termenului de finalizare a problemei, Not Null
* masina\_id\_masina (NUMBER(4))
  + Tip: Numeric, legătură cu masina (FK)

### Descrierea Constrângerilor și Motivul Aplicării Acestora

#### **Constrângeri la nivel de tabele**

* **Primar Key (PK)**: Constrângerile client\_pk, data\_programare\_pk, masina\_pk, mecanic\_pk, și problema\_pk asigură unicitatea identificatoarelor pentru fiecare tabel, prevenind dublările.
* **Foreign Key (FK)**: Legături între tabele, cum ar fi Masina\_Client\_FK, Detalii\_Client\_Client\_FK, Problema\_Masina\_FK, Data\_Programare\_Masina\_FK si Mecanic\_Masina\_FK , asigură integritatea referențială între tabele.
* **Check Constraints**:
  + Client\_GradSatisfactie\_CK (BETWEEN 1 AND 10): Asigură ca ratingul satisfacției clientului să fie valid.
  + Detalii\_Client\_Varsta\_CK (BETWEEN 18 AND 99): Asigură ca vârsta clientului să fie validă.
  + Mecanic\_RatingMecanic\_CK (BETWEEN 1 AND 10) : Verifică validitatea ratingului mecanicului.
  + Problema\_Cost\_CK (BETWEEN 0 AND) 9999: Verifică validitatea costului problemei.
* **Not Null:** Nume\_Client, Marca\_Auto, Model\_Auto, Data\_Intrare\_Masina\_Service, CNP, Cost, Problema, Termen\_Finalizare\_Problema, Data\_Programare asigură că nu pot fi introduse aceleași date.

#### **Motivul Necesității Constrângerilor**

* **FK**: Asigură integritatea datelor și evită duplicarea sau inconsistențele în datele referite.
* **Check Constraints**: Asigură limitarea valorilor la intervalele acceptate, prevenind introducerea de date incorecte.
* **PK**: Este esențial pentru identificarea unică a înregistrărilor din fiecare tabel.
* **UK: Indexurile unice** previn duplicarea înregistrărilor și garantează că fiecare entitate (client, mașină, mecanic, problemă) are un identificator unic, oferind astfel integritatea datelor. **Unicitatea** asigură coerența datelor și previne conflictele sau redundanțele
* **NN:** fiecare rând din tabel trebuie să aibă o valoare specifică pentru acea coloană. Aceasta este utilizată pentru a asigura integritatea datelor, prevenind introducerea rândurilor cu valori lipsă.

Posibile **use-case-uri:**

1. **Adăugarea unui nou client:**

 Receptionerul introduce datele clientului (nume, grad de satisfacție).

 Se generează un ID unic pentru client (folosind secvența client\_id\_client\_seq).

 Datele sunt stocate în tabela client.

 Se pot adăuga detalii suplimentare despre client în tabela detalii\_client.

2. **Programarea unei mașini pentru reparație:**

 Receptionerul introduce detalii despre mașină (marca, model, data intrării în service).

 Se creează o intrare în tabela masina.

 Se adaugă o programare în tabela data\_programare, specificând data și ID-ul mașinii.

3. **Alocarea unui mecanic unei mașini:**

 Managerul selectează un mecanic din baza de date (tabela mecanic).

 Se asociază mecanicul cu o mașină (relația între mecanic și masina prin masina\_id\_masina).

 Se actualizează rating-ul mecanicului după finalizarea reparației.

4. **Gestionarea problemelor și costurilor:**

 Managerul introduce o problemă în tabela problema (ex.: defect motor, costul reparației, termenul de finalizare).

 Se asociază problema cu o mașină prin ID-ul mașinii (masina\_id\_masina).

 Sistemul verifică dacă costul este în limitele permise (constrângerea problema\_cost\_ck).

5. **Analiza datelor despre clienți și performanță:**

 Managerul interoghează tabela client pentru a vedea gradul de satisfacție.

 Compară datele cu istoricul reparațiilor și performanța mecanicilor.

 Creează rapoarte folosind relațiile dintre tabele (client, masina, problema, mecanic).