

Anul universitar 2023-2024

PROIECTAREA UNEI BAZE DE DATE RELAȚIONALE PENTRU UN SERVICE AUTO

Candidat: Adrian, POP

Coordonator științific: Ș.I. dr. ing. Mădălin-Dorin, POP

Ing. Daniela-Nicolia, PĂTRUȚ

Sesiunea: Ianuarie 2024

CUPRINS

1	INTRODUCERE	4
1.1	CONTEXT ȘI DESCRIEREA GENERALĂ A PROIECTULUI	4
1.2	SCOPUL ȘI OBIECTIVELE PROIECTULUI	4
1.3	STRUCTURA LUCRĂRII	5
2	ANALIZA STADIULUI ACTUAL ÎN DOMENIUL PROBLEMEI	6
3	BAZELE TEORETICE	8
3.1	GENERALITĂȚI	8
3.2	PROIECTAREA CONCEPTUALĂ	8
3.3	PROIECTAREA LOGICĂ	10
3.4	PROIECTAREA FIZICĂ	11
4	SOLUTIA PROPUȘĂ ȘI METODOLOGIA DE PROIECTARE	12
4.1	PROIECTAREA CONCEPTUALĂ	12
4.2	PROIECTAREA LOGICĂ	14
4.3	PROIECTARE FIZICĂ	15
5	IMPLEMENTARE	18
5.1	CREAREA TABELELOR	18
5.2	POPULAREA BAZEI DE DATE	19
5.3	VEDERI SIMPLE ȘI COMPLEXE	22
5.4	PROCEDURI	22
6	UTILIZARE ȘI DATE EXPERIMENTALE	24
6.1	REZULTATUL CREĂRII ȘI POPULĂRII BAZEI DE DATE	24
6.2	REZULTATUL VEDERILOR	26
6.3	REZULTATUL PROCEDURILOR	26
7	CONCLUZII ȘI DIRECTII DE DEZVOLTARE	28
8	BIBLIOGRAFIE	31

REZUMAT

Această lucrare a fost concepută pentru a oferi o soluție fezabilă pentru gestionarea datelor necesare pentru buna și eficienta funcționare a unui Service Auto și chiar cu scopul analizării diverselor aspecte din funcționarea acestui tip de firmă. Baza de date va manipula informații despre mașinile care calcă pragul service-ului, proprietarii lor, mecanicii care se ocupă de acestea, precum și piesele de schimb sau reparațiile care se efectuează în cadrul service-ului. Oferind un istoric clar și sigur baza de date va aduce, pe lângă performanțe, un mediu plăcut atât pentru clienți cât și pentru angajați.

Rezultatul lucrării trebuie folosit corespunzător. Actualizarea constantă a bazei de date se va face în vederea situației în care accesibilă fiind angajaților, aceștia vor putea răspunde prompt cerințelor venite din partea clienților. De asemenea, poate fi folosită strict pe plan intern pentru a evalua angajați, lucrări și servicii prestate de aceștia dar și durabilitatea unor piese sau materiale și chiar pentru a face predicții asupra stocurilor aferente service-ului auto. Este necesară monitorizarea bazei de date și introducerea corectă, concisă și veridică a multitudinii de date pentru ca această unealtă să-și îndeplinească scopul pentru care a fost proiectată – creșterea de performanță prin procesarea organizată și în timp reduși a datelor.

1 INTRODUCERE

1.1 CONTEXT ȘI DESCRIEREA GENERALĂ A PROIECTULUI

În contextul actual al dezvoltării continue din punct de vedere al tehnologiei, s-a sesizat nevoia unui sistem informatic avansat de gestiune eficientă a datelor care să modeleze relațiile dintre multiplele entități ale unui Service Auto.

1.2 SCOPUL ȘI OBIECTIVELE PROIECTULUI

Scopul central al lucrării este acela de a gestiona în mod eficient și performant date și relații între diverse entități provenite din mediul unui Service Auto. Astfel, se va proiecta o bază de date care să servească acest scop. În urma analizei în domeniu s-a ajuns la concluzia că este necesară stocarea datelor despre mașinile care ajung în service, despre proprietarii lor, dar și despre mecanicii care lucrează în service și piesele de schimb pe care le folosesc precum și fiecare înregistrare în service a mașinilor în cauză. Pe baza acestor entități și a relațiilor dintre ele se va realiza proiectarea bazei de date. Ca și obiective, produsul finit obținut își propune să ofere o mai bună interacțiune între clienții service-ului și angajați, oferind acces rapid și concludent la informații utile. Un alt țel al proiectului este acela de a oferi management-ului o imagine asupra angajaților reflectată de analiza acestor date. Având în vedere că baza de date creează, în practică, un istoric, intenția este și de a semnaliza persoanele ce se află la administrarea ei de eventuale aprovizionări de stocuri de piese sau materiale sau chiar de posibile viitoare înregistrări în service ale anumitor mașini. Un alt aport pe care lucrarea îl aduce este automatizarea anumitor procese care se desfășoară în cadrul service-ului, reducerea apariției erorilor și bineînțeles diminuarea timpilor de așteptare. Pentru realizarea celor propuse au fost necesare o serie de baze teoretice care vor fi prezentate ulterior și care au constat în diverse tipuri de proiectare (conceptuală, logică, fizică) și desigur de implementare a bazelor de date relaționale. Întrucât se urmărește eficientizarea, se vor analiza și evita prin proiectare potențialele așa-zise capcane sau greșeli ce duc la date cu un grad scăzut de robustețe sau chiar redundante.

1.3 STRUCTURA LUCRĂRII

Documentația proiectului este structurată în 6 capitole (la care se adaugă introducerea și bibliografia) ale căror titluri însumează etapele analizei bazei de date conturate :

- Analiza stadiului actual în domeniul problemei
- Bazele teoretice
- Soluția propusă și metodologia de proiectare/dezvoltare
- Implementare
- Utilizare, rezultate experimentale
- Concluzii și direcții de dezvoltare

2 ANALIZA STADIULUI ACTUAL ÎN DOMENIUL PROBLEMEI

Analiza stadiului actual în domeniul problemei reprezintă o fază absolut necesară în contextul proiectării bazelor de date relaționale întrucât în urma explorării și studiului uneia sau mai multor lucrări care tratează aceeași temă sau același principii se poate spori eficiența noului rezultat obținut. Diverse greșeli care pot fi considerate frecvente pot fi evitate mai ușor în urma acestui studiu. De asemenea, tehnici deja existente pot fi îmbunătățite sau pot oferi o viziune mai amplă asupra subiectului abordat.

Reverse Engineering (termen în limba engleză) reprezintă procesul de a analiza subiectul unui sistem împreună cu componentele și relațiile din sistem, cu scopul de a crea reprezentări sub formă de abstracțiuni ce ajută la înțelegerea programelor. Legătura cu baza de date se face prin evaluarea cheii primare și a celei străine pentru a determina relația dintre tabele. Se va realiza apoi diagrama entitate-relație (Entity Relational Diagram) și odată cu finalizarea diagramei se prezintă concepte ce au trecut prin etapa de normalizare în a treia formă normală. [1]

Sistemul de interogare a bazelor de date Oracle a devenit cel mai des folosit astfel de sistem datorită posibilității sale excelente de a fi compatibil cu mai multe platforme oferind o portabilitate bună. Capacitatea de a gestiona date a acestui sistem este puternică și cuprinzătoare. Acestea fiind spuse, Oracle joacă un rol important în mai multe domenii precum explorarea datelor, comunicații sau tehnologia informațiilor, grație avantajelor pe care le aduce. Este înregistrat un volum tot mai mare al datelor ce trebuie gestionate și bazele de date trec la o structură tot mai complexă, ceea ce poate slăbi performanțele și eficiența. Tot odată, acest lucru conduce la dorințe de creștere a performanțelor de extragere a datelor. Se dorește atât progresul structurii hardware, cât și designul sistemelor software. Așa cum este prezentat în Figura 1 de mai jos, aceasta este rezultatul unui test pentru schema de optimizare construită cu 1 terabyte de date ca suport experimental într-un sistem de interogare al bazei de date Oracle. [2]

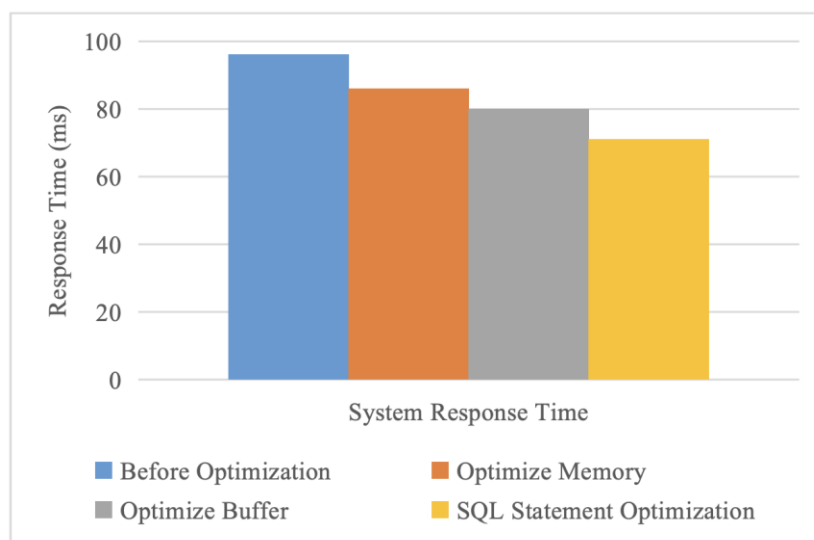


Figura 1 – Rezultatele optimizării unei baze de date [2]

Bazele de date sunt acelea care stau la baza majorității sistemelor informatice. Majoritatea acestor sisteme care se află în uz la momentul actual, au la bază un model relațional, un mod de a reprezenta datele și relațiile folosind tabele bidimensionale. [3]

3 BAZELE TEORETICE

3.1 GENERALITĂȚI

Bazele de date reprezintă sisteme menite să adăpostească și să administreze informații organizate, furnizând utilizatorilor posibilitatea de a accesa și manipula datele cu eficiență. Ele constituie o structură centralizată ce facilitează stocarea și gestionarea coerentă a informațiilor, simplificând astfel procesul de analiză și administrare a datelor. Baza de date relațională este un tip specific de bază de date care se bazează pe modelul relațional.

Proiectarea bazelor de date începe imediat după faza de captare a cerințelor și este formată din 3 etape :

- Proiectare conceptuală;
- Proiectare logică;
- Proiectare fizică.

În urma acestora rezultă structura unei baze de date bazate relaționale și apoi poate fi formulată documentația. În cadrul acestei etape se pot ivi situații în care legătura dintre entități este neclară, ambiguă sau chiar inexistentă. Aceste situații sunt denumite capcane de proiectare și apar cel mai des în urma relațiilor *many-to-many*. În limba engleză ele sunt denumite *fan-traps*, respectiv *chasm-traps*.

Există numeroase sisteme de gestiune relaționale de acest fel. Câteva exemple mai cunoscute ar putea fi : MySQL, Oracle Database, Microsoft SQL Server, PostgreSQL, SQLite sau chiar IBM Db2.

3.2 PROIECTAREA CONCEPTUALĂ

Aceasta este bazată pe modelarea entitate-relație.

Entitatea constă în date ce pot fi modelate ca obiecte independente, unice iar relația în conexiunea dintre două sau mai multe entități de același tip sau de tip diferit. [4]

Asocierea este reprezentată printr-o digramă în care attributele reprezintă proprietățile unei entități.

Entitățile pot fi de două feluri: *slabe* (nu pot exista neavând dependențe față de alte entități) sau *tari* (entități independente).

Relația dintre entități este o asociere bidirecțională și conturează felul în care acestea comunică. Ca și caracteristică se menționează cardinalitatea – numărul de înregistrări. Relațiile dintre aceste entități se pot defini în unul dintre următoarele feluri :

- Relație 1-1 (one-to-one): cheia externă este plasată în tabelul cu mai puține înregistrări;
- Relație 1-N (one-to-many): cheia externă se plasează în tabelul „many”;
- Relație N-M (many-to-many): se introduce un tabel asociat ce conține două chei externe.

Atributele sunt proprietăți ce caracterizează obiectele definite de către entități.

Clasificarea atributelor:

- Atribute cheie – identifică entitatea în mod unic (PK – primary key);
- Atribute de descriere – caracteristici ale entității;
- Atribute multiple – poate reține mai multe valori pentru o înregistrare;
- Atribute compuse – obținute prin combinarea mai multor atribute;
- Atribute derivate – valoarea obținută pe baza unui alt atribut.

În urma acestei etape a procesului rezultă diagrama Entitate-Relație care descrie toate cele menționate mai sus. Cele mai frecvente simboluri folosite în aceasta sunt prezente în figura de mai jos.

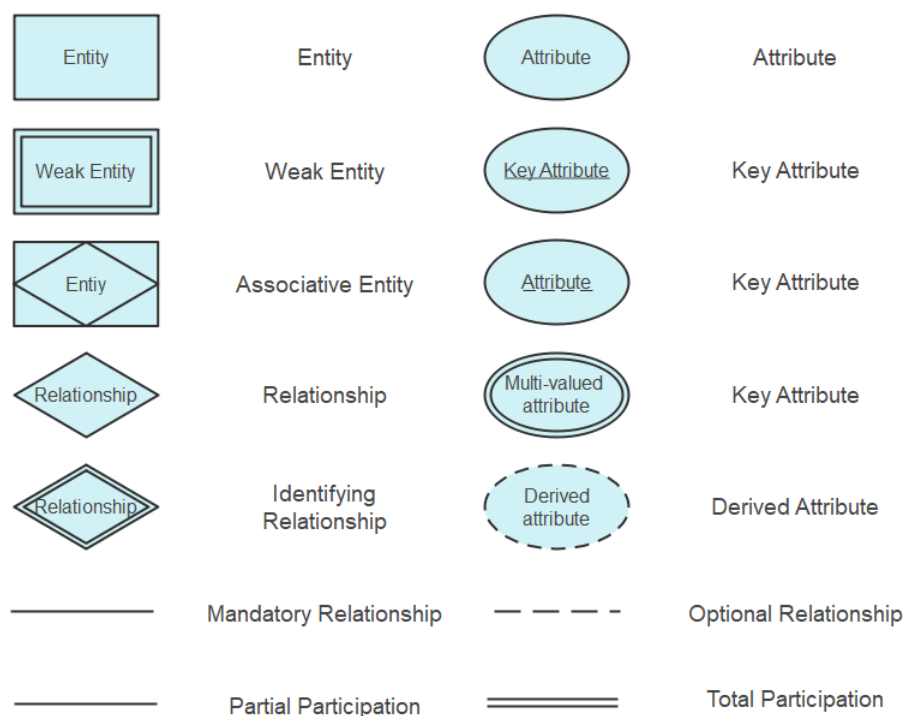


Figura 2 – Simboluri uzuale în proiectarea conceptuală [5]

3.3 PROIECTAREA LOGICĂ

Proiectarea logică constă în îmbunătățirea modelului obținut anterior prin aplicarea unor principii de normalizare. Normalizarea reprezintă descompunerea modelului bazei de date în mai multe relații în scopul reducerii la maxim a redundanței datelor. [4]

În cadrul acestei etape, normalizarea trebuie aplicată deoarece de cele mai multe ori apar anomalii de actualizare, inserare sau ștergere pentru datele introduse. Procesul introduce utilizarea cheilor străine (FK – foreign keys) și debarasarea de valorile nule.

Se folosesc următoarele forme normale :

- Prima formă normală, notată cu 1NF;
- A doua formă normală, notată cu 2NF;
- A treia formă normală, notată cu 3NF;
- A patra formă normală, notată cu 4NF.

Prima formă impune ca orice înregistrare să fie unică și ca fiecare celulă să aibă o singură valoare, în timp ce cea de-a doua adaugă condiția ca orice atribut care nu e cheie primară să fie total dependent de cheia primară a entității. A treia formă normală presupune, în plus celor menționate în primele două forme normale, absența dependențelor tranzitorii

(dependențe între coloanele a două tabele distincte care au indirect un impact asupra altor coloane). A nu se confunda cu dependențele funcționale care sunt determinate într-un tabel de legătura dintre atribute și cheia primară. Pentru a atinge a patra formă normală în procesul de proiectare logică este nevoie în aditie de înlăturarea dependențelor multivalorice. Acestea apar atunci când o coloană poate determina un set de valori pentru o altă coloană, în funcție de o a treia altă coloană.

3.4 PROIECTAREA FIZICĂ

Proiectarea fizică reprezintă procesul de descriere a modului de implementare ce urmează a fi realizată asupra bazei de date. Tot odată, cuprinde implementarea structurilor de date și a modului prin care se pot accesa datele respective.

Se realizează în această etapă o mapare între modelul conceptual obținut în etapele precedente și modelul fizic ca rezultat al acestei faze după cum se poate observa în figura de mai jos.

Model conceptual	Model fizic
entitate	tabelă
atribut	câmp
instanță-entitate	înregistrare
atribut cheie	cheie primară
relație	cheie străină

Figura 3 – Maparea efectuată în proiectarea fizică

Pentru implementarea fizică se va folosi Oracle SQL. Acesta reprezintă limbajul de interogare standard utilizat pentru gestionarea bazelor de date Oracle și este conceput cu scopul de a interacționa cu bazele de date relaționale. Reprezintă în sine un standard în industrie și este folosit pe scară largă în aplicații de dezvoltare a bazelor de date.

Oracle APEX (Application Express) este platforma folosită pentru o dezvoltare rapidă și eficientă a acestor baze de date. Platforma permite dezvoltatorilor, prin intermediul unui limbaj de programare PL/SQL, construirea unor aplicații web complexe și interactive utilizând o interfață grafică plăcută.

4 SOLUȚIA PROPUȘĂ ȘI METODOLOGIA DE PROIECTARE

4.1 PROIECTAREA CONCEPTUALĂ

Baza de date proiectată își propune să gestioneze datele necesare unui Service auto într-un mod cât mai eficient, performant și ușor de folosit. În centrul ideii se află mașinile care trec poarta service-ului. Astfel, se vor stoca date despre mașini și fiecare înregistrare a acestora în service-ul nostru. Desigur, proprietarii mașinilor formează clientela astfel că se vor manipula și date despre aceștia. Următoarea piesă a puzzle-ului este reprezentată de angajați, adică de mecanicii care lucrează pentru a repara mașinile. În cadrul service-ului se aduc și se montează o sumedenie de piese de schimb despre care este bine să avem o evidență. Astfel, vom putea afla la interogarea bazei de date un istoric complet despre lucrările efectuate în service și putem crea și o vedere de ansamblu funcționării service-ului auto.

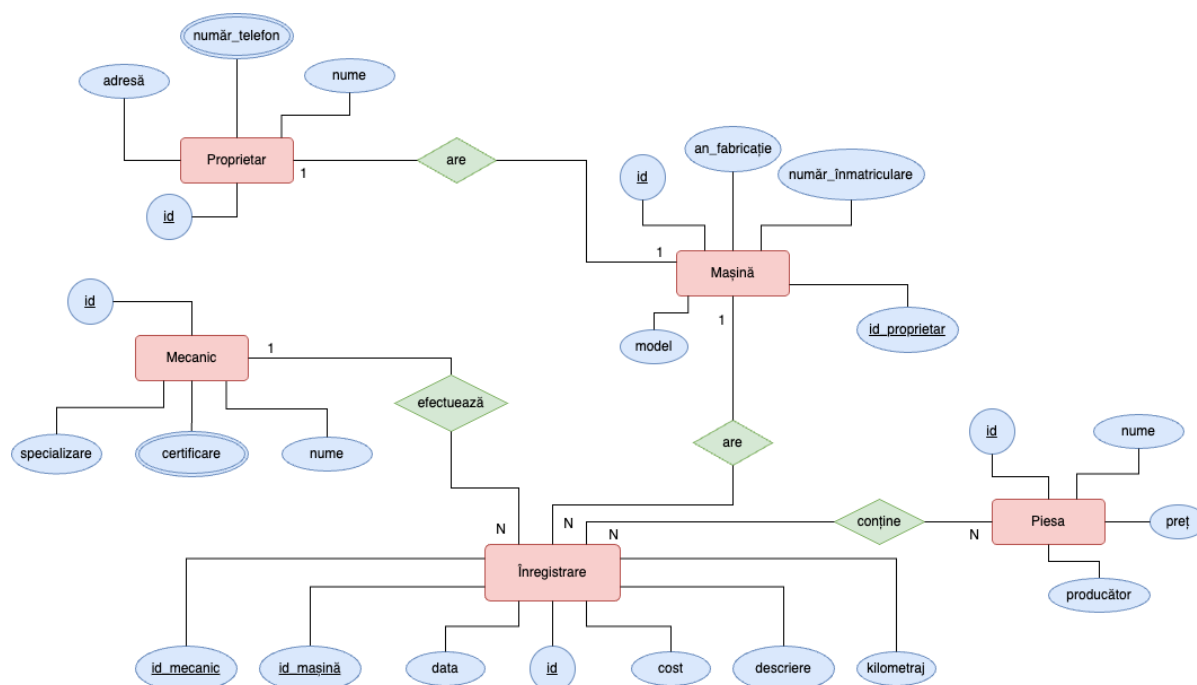


Figura 4 – Diagrama ER în fază inițială

Descrierea entităților și a atributelor lor:

- Mașina : id – cheie primară (PK), model, an_fabricație, număr_inmatriculare, id_proprietar (cheie străină - FK);
- Proprietar : id – cheie primară (PK), nume, adresa, număr_telefon;
- Înregistrare: id – cheie primară (PK), data, cost, descriere, kilometraj, id_masina(cheie străină - FK), id_mecanic (cheie străină - FK);
- Mecanic : id – cheie primară (PK), nume, specializare, certificare;
- Piesă: id – cheie primară (PK), nume, preț, producător.

Descrierea relațiilor dintre entități:

- Mașina-Proprietar relație 1-1 (one-to-one) – evidențiată prin verbul „are” (o mașină are un proprietar);
- Mașina-Înregistrare 1-N (one-to-many) – evidențiată prin verbul „are” (o mașină are mai multe înregistrări în service);
- Înregistrare-Piesă N-N (many-to-many) – evidențiată de verbul „conține” (o înregistrare în service conține mai multe piese auto)
- Mecanic-Înregistrare 1-N (one-to-many) – evidențiată prin verbul „efectuează” (un mecanic efectuează mai multe înregistrări în service);

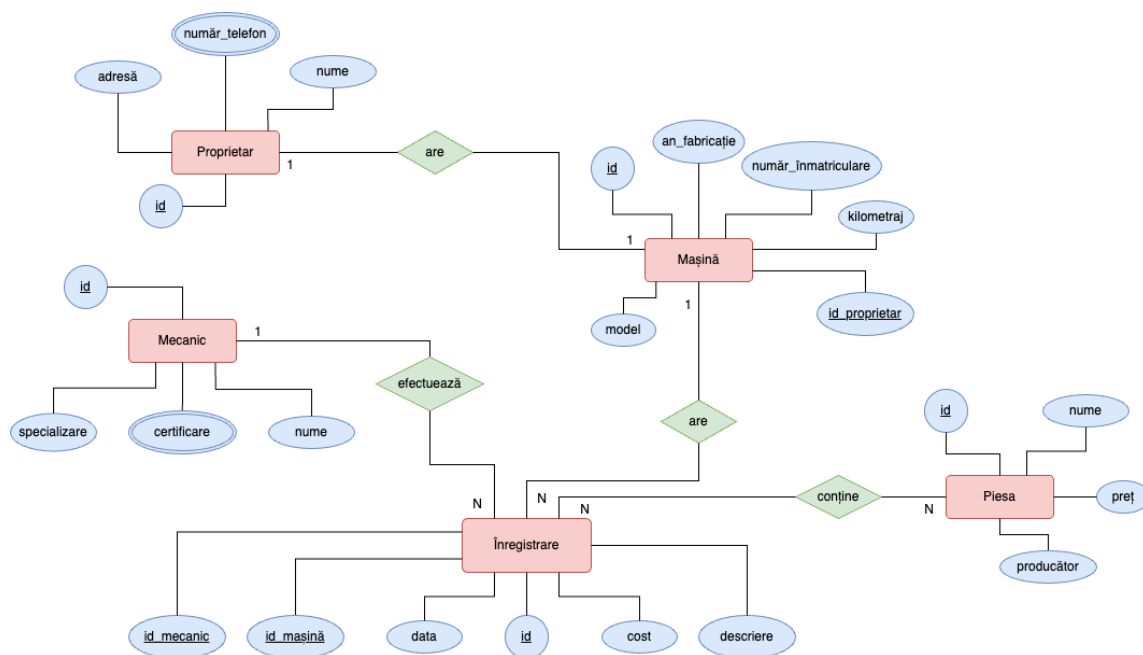


Figura 5 – Diagrama ER în fază finală

4.2 PROIECTAREA LOGICĂ

Normalizarea este procesul aplicat în această etapă și este foarte importantă deoarece elimină posibilitățile ca datele să fie redundante. Se observă că două dintre tabele nu se află în forma normală 1NF. Aceste tabele sunt gazda unor attribute cu valoare multiplă.

Din cauza atributului număr_telefon din tabela Proprietar, se va crea o nouă tabelă, TelefonProprietar. Aceasta va avea ca și attribute număr_telefon, id – cheie primară (PK) și id_proprietar (cheie străină - FK).

În cazul tabelii Mecanic, atributul cu valoare multiplă certificare face ca aceasta să nu se afle într-o forma normală 1NF. Se va crea tabela CertificareMecanic cu attributele certificare, id - cheie primară (PK) și id_mecanic (cheie străină - FK).

La celelalte tabelele deja existente se adaugă cheile străine care consolidează relațiile definite dintre entități :

- Mașina : id_proprietar (cheie străină - FK);
- Înregistrare: id_masina (cheie străină - FK), id_mecanic (cheie străină - FK);

În cazul relației N-N (many-to-many) dintre tabela Înregistrare și Piesă se va crea un nou tabel asociat, ÎnregistrarePiesă cu attributele : id_înregistrare – cheie străină (FK), id_piesă (cheie străină - FK);

Se observă că tabela Înregistrare nu se află în forma normală 2NF din cauza atributului kilometraj care nu este total dependent de cheia primară id_înregistrare a tabelii respective. Acest atribut este dependent și de cheia străină id_mașină. Astfel, atributul kilometraj va fi înlăturat din tabelă și plasat în tabela Mașină unde este total dependent de cheia primară id_mașină.

Urmărind astfel regulile și îndrumările prezentate într-o secțiune anterioară au rezultat 8 tabele care alcătuiesc întreaga bază de date într-un mod robust.

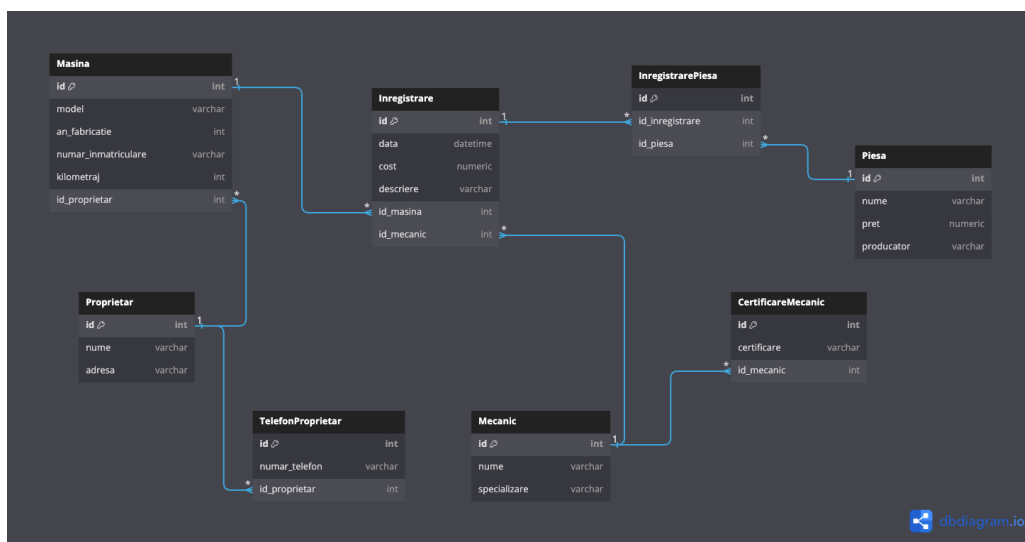


Figura 6 – Diagrama pentru modelul logic

4.3 PROIECTARE FIZICĂ

Se introduce modelul fizic al fiecărei tabele. Niciunul dintre atributele enumerate nu are valoare prestabilită și nu poate reține valoarea NULL.

Tabelul 1. Tabelul Mașină

ATRIBUT	DESCRIERE	TIP
id (PK)	identificator unic în tabelă	NUMBER
model	modelul mașinii	VARCHAR2(255)
an_fabricție	anul fabricației mașinii	NUMBER
număr_înmatriculare	numărul de înmatriculare	VARCHAR2(20)
kilometraj	kilometrajul la momentul dat	NUMBER
id_proprietar (FK)	identificator unic proprietar	NUMBER

Tabelul 2. Tabelul Proprietar

ATRIBUT	DESCRIERE	TIP
id (PK)	identificator unic în tabelă	NUMBER
nume	numele proprietarului	VARCHAR2(255)
adresă	adresa proprietarului	VARCHAR2(500)

Tabelul 3. Tabelul TelefonProprietar

ATRIBUT	DESCRIERE	TIP
id (PK)	identificator unic în tabelă	NUMBER
număr_telefon	numărul de telefon al proprietarului	VARCHAR2(15)
id_proprietar (FK)	identificator unic proprietar	NUMBER

Tabelul 4. Tabelul Mecanic

ATRIBUT	DESCRIERE	TIP
id (PK)	identificator unic în tabelă	NUMBER
nume	numele mecanicului	VARCHAR2(255)
specializare	specializarea mecanicului	VARCHAR2(255)

Tabelul 5. Tabelul CertificareMecanic

ATRIBUT	DESCRIERE	TIP
id (PK)	identificator unic în tabelă	NUMBER
certificare	Certificarea mecanicului	VARCHAR2(255)
id_mecanic (FK)	identificator unic mecanic	NUMBER

Tabelul 6. Tabelul Înregistrare

ATRIBUT	DESCRIERE	TIP
id (PK)	identificator unic în tabelă	NUMBER
data	data înregistrării în service	DATE
cost	costul rezultat în urma înregistrării	NUMBER
descriere	descrierea rezultată în urma înregistrării	VARCHAR2(1000)
id_mașină (FK)	identificator unic mașină	NUMBER
id_mecanic (FK)	identificator unic mecanic	NUMBER

Tabelul 7. Tabelul Piesă

ATRIBUT	DESCRIERE	TIP
id (PK)	identificator unic în tabelă	NUMBER
nume	denumirea piesei	VARCHAR2(255)
preț	prețul piesei	NUMBER
producător	producătorul piesei	VARCHAR2(255)

Tabelul 8. Tabelul ÎnregistrarePiesă

ATRIBUT	DESCRIERE	TIP
id (PK)	identificator unic în tabelă	NUMBER
id_înregistrare (FK)	identificator unic înregistrare	NUMBER
id_piesă (FK)	identificator unic piesă	NUMBER

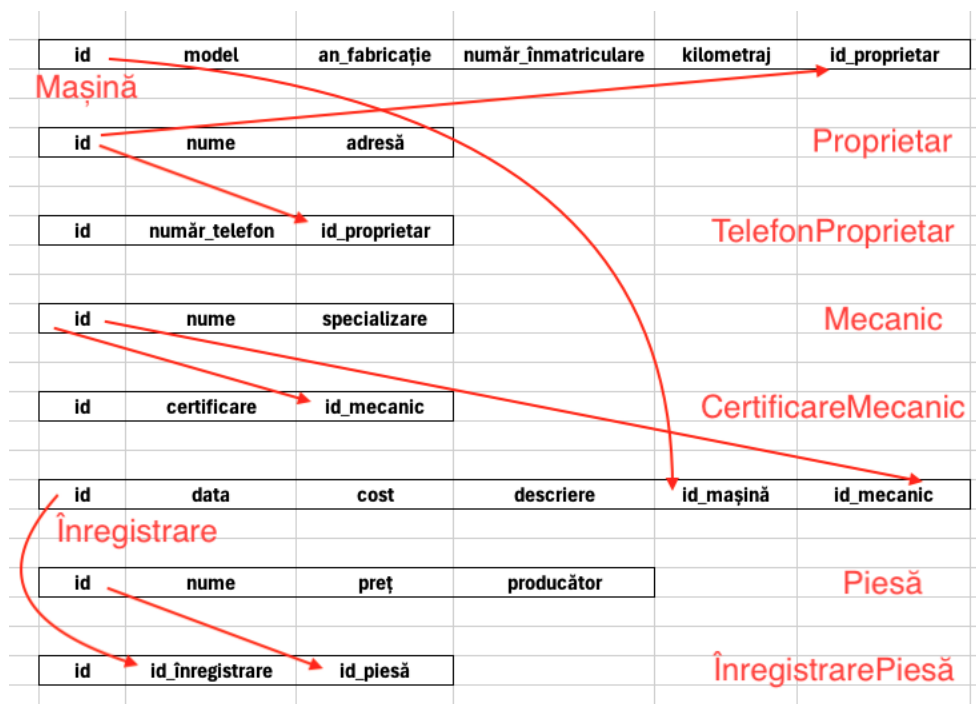


Figura 7 – Tabelul cu relațiile fizice dintre cheile primare și cele străine

5 IMPLEMENTARE

Se implementează modelul obținut în urma proiectării.

5.1 CREAREA TABELELOR

```
CREATE TABLE Proprietar (  
    id NUMBER PRIMARY KEY,  
    nume VARCHAR2(255) NOT NULL,  
    adresa VARCHAR2(500) NOT NULL  
);
```

```
CREATE TABLE TelefonProprietar (  
    id NUMBER PRIMARY KEY,  
    numar_telefon VARCHAR2(15) NOT NULL,  
    id_proprietar NUMBER NOT NULL,  
    FOREIGN KEY (id_proprietar) REFERENCES Proprietar(id)  
);
```

```
CREATE TABLE Mecanic (  
    id NUMBER PRIMARY KEY,  
    nume VARCHAR2(255) NOT NULL,  
    specializare VARCHAR2(255) NOT NULL  
);
```

```
CREATE TABLE CertificareMecanic (  
    id NUMBER PRIMARY KEY,  
    certificare VARCHAR2(255) NOT NULL,  
    id_mecanic NUMBER NOT NULL,  
    FOREIGN KEY (id_mecanic) REFERENCES Mecanic(id)  
);
```

```
CREATE TABLE Masina (  

```

```
id NUMBER PRIMARY KEY,  
model VARCHAR2(255) NOT NULL,  
an_fabricatie NUMBER NOT NULL,  
numar_inmatriculare VARCHAR2(20) NOT NULL,  
kilometraj NUMBER NOT NULL,  
id_proprietar NUMBER NOT NULL,  
FOREIGN KEY (id_proprietar) REFERENCES Proprietar(id)  
);
```

```
CREATE TABLE Inregistrare (  
id NUMBER PRIMARY KEY,  
data DATE NOT NULL,  
cost NUMBER NOT NULL,  
descriere VARCHAR2(1000) NOT NULL,  
id_masina NUMBER NOT NULL,  
FOREIGN KEY (id_masina) REFERENCES Masina(id)  
FOREIGN KEY (id_mecanic) REFERENCES Mecanic(id)  
);
```

```
CREATE TABLE Piesa (  
id NUMBER PRIMARY KEY,  
nume VARCHAR2(255) NOT NULL,  
pret NUMBER NOT NULL,  
producator VARCHAR2(255) NOT NULL  
);
```

```
CREATE TABLE InregistrarePiesa (  
id NUMBER PRIMARY KEY,  
id_inregistrare NUMBER NOT NULL,  
id_piesa NUMBER NOT NULL,  
FOREIGN KEY (id_inregistrare) REFERENCES Inregistrare(id),  
FOREIGN KEY (id_piesa) REFERENCES Piesa(id)  
);
```

5.2 POPULAREA BAZEI DE DATE

```
INSERT INTO Proprietar (id, nume, adresa) VALUES (1, 'Ion Popescu', 'Str.  
Libertatii, Nr. 10, Bucuresti');
```

INSERT INTO Proprietar (id, nume, adresa) VALUES (2, 'Maria Vasilescu', 'Bd. Unirii, Nr. 5, Cluj-Napoca');

INSERT INTO Proprietar (id, nume, adresa) VALUES (3, 'Adrian Ionescu', 'Bd. Victoriei, Nr. 15, Iasi');

INSERT INTO Proprietar (id, nume, adresa) VALUES (4, 'Elena Constantinescu', 'Str. Mihai Bravu, Nr. 25, Timisoara');

INSERT INTO Proprietar (id, nume, adresa) VALUES (5, 'Alexandru Radu', 'Bd. 1 Mai, Nr. 7, Craiova');

INSERT INTO TelefonProprietar (id, numar_telefon, id_proprietar) VALUES (1, '0712345678', 1);

INSERT INTO TelefonProprietar (id, numar_telefon, id_proprietar) VALUES (2, '0723456789', 2);

INSERT INTO TelefonProprietar (id, numar_telefon, id_proprietar) VALUES (3, '0734567890', 3);

INSERT INTO TelefonProprietar (id, numar_telefon, id_proprietar) VALUES (4, '0745678901', 4);

INSERT INTO TelefonProprietar (id, numar_telefon, id_proprietar) VALUES (5, '0756789012', 5);

INSERT INTO Mecanic (id, nume, specializare) VALUES (1, 'Andrei Popa', 'Motor si Transmisie');

INSERT INTO Mecanic (id, nume, specializare) VALUES (2, 'Gabriela Dumitrescu', 'Suspensie si Frane');

INSERT INTO Mecanic (id, nume, specializare) VALUES (3, 'Mihai Vasilescu', 'Electricitate Auto');

INSERT INTO Mecanic (id, nume, specializare) VALUES (4, 'Cristina Radu', 'Caroserie si Vopsitorie');

INSERT INTO Mecanic (id, nume, specializare) VALUES (5, 'Ionel Ionescu', 'Diagnosticare si Service General');

INSERT INTO CertificareMecanic (id, certificare, id_mecanic) VALUES (1, 'Certificat de Diagnosticare Auto', 1);

INSERT INTO CertificareMecanic (id, certificare, id_mecanic) VALUES (2, 'Certificat de Service Tehnic', 2);

INSERT INTO CertificareMecanic (id, certificare, id_mecanic) VALUES (3, 'Certificat de Reparatii Motor', 3);

INSERT INTO CertificareMecanic (id, certificare, id_mecanic) VALUES (4, 'Certificat de Vopsitor Auto', 4);

```
INSERT INTO CertificareMecanic (id, certificare, id_mecanic) VALUES (5, 'Certificat de Expertiza Tehnica', 5);
```

```
INSERT INTO Masina (id, model, an_fabricatie, numar_inmatriculare, kilometraj, id_proprietar) VALUES (1, 'Renault Megane', 2018, 'PH-123ABC', 50000, 1);
```

```
INSERT INTO Masina (id, model, an_fabricatie, numar_inmatriculare, kilometraj, id_proprietar) VALUES (2, 'Volkswagen Golf', 2019, 'CJ-456XYZ', 60000, 2);
```

```
INSERT INTO Masina (id, model, an_fabricatie, numar_inmatriculare, kilometraj, id_proprietar) VALUES (3, 'Ford Focus', 2017, 'IS-789DEF', 75000, 3);
```

```
INSERT INTO Masina (id, model, an_fabricatie, numar_inmatriculare, kilometraj, id_proprietar) VALUES (4, 'Opel Astra', 2020, 'TM-101GHI', 40000, 4);
```

```
INSERT INTO Masina (id, model, an_fabricatie, numar_inmatriculare, kilometraj, id_proprietar) VALUES (5, 'Dacia Logan', 2016, 'DJ-202JKL', 80000, 5);
```

```
INSERT INTO Inregistrare (id, data, cost, descriere, id_masina) VALUES (1, TO_DATE('2023-03-15', 'YYYY-MM-DD'), 300, 'Schimb ulei si filtre', 1, 1);
```

```
INSERT INTO Inregistrare (id, data, cost, descriere, id_masina) VALUES (2, TO_DATE('2023-04-20', 'YYYY-MM-DD'), 500, 'Revizie generala', 2, 2);
```

```
INSERT INTO Inregistrare (id, data, cost, descriere, id_masina) VALUES (3, TO_DATE('2023-05-10', 'YYYY-MM-DD'), 200, 'Schimb placute frana', 3, 3);
```

```
INSERT INTO Inregistrare (id, data, cost, descriere, id_masina) VALUES (4, TO_DATE('2023-06-25', 'YYYY-MM-DD'), 700, 'Reparatii motor', 4, 4);
```

```
INSERT INTO Inregistrare (id, data, cost, descriere, id_masina) VALUES (5, TO_DATE('2023-07-12', 'YYYY-MM-DD'), 100, 'Verificare tehnica', 5, 5);
```

```
INSERT INTO Piesa (id, nume, pret, producator) VALUES (1, 'Placute frana', 100, 'Brembo');
```

```
INSERT INTO Piesa (id, nume, pret, producator) VALUES (2, 'Filtru ulei', 20, 'Mann-Filter');
```

```
INSERT INTO Piesa (id, nume, pret, producator) VALUES (3, 'Bujii', 30, 'NGK');
```

```
INSERT INTO Piesa (id, nume, pret, producator) VALUES (4, 'Disc frana', 150, 'ATE');
```

```
INSERT INTO Piesa (id, nume, pret, producator) VALUES (5, 'Lampa far', 15, 'Philips');
```

```
INSERT INTO InregistrarePiesa (id, id_inregistrare, id_piesa) VALUES (1, 1, 1);
```

```
INSERT INTO InregistrarePiesa (id, id_inregistrare, id_piesa) VALUES (2, 2, 2);
```

```
INSERT INTO InregistrarePiesa (id, id_inregistrare, id_piesa) VALUES (3, 3, 3);
```

```
INSERT INTO InregistrarePiesa (id, id_inregistrare, id_piesa) VALUES (4, 4, 4);  
INSERT INTO InregistrarePiesa (id, id_inregistrare, id_piesa) VALUES (5, 5, 5);
```

5.3 VEDERI SIMPLE ȘI COMPLEXE

Prima vedere simplă afișează datele din tabela Înregistrare pentru înregistrarea cu id-ul 3, iar cea de-a doua vedere afișează numărul de telefon al proprietarului pentru mașina cu id-ul 3:

```
CREATE VIEW InformatiiInregistrare AS  
SELECT *  
FROM Inregistrare  
WHERE id = 3;
```

```
CREATE VIEW InformatiiMasinaProprietar AS  
SELECT  
    m.id AS id_masina,  
    m.model,  
    m.an_fabricatie,  
    m.numar_inmatriculare,  
    p.id AS id_proprietar,  
    p.nume AS nume_proprietar,  
    tp.numar_telefon  
FROM  
    Masina m  
JOIN Proprietar p ON m.id_proprietar = p.id  
JOIN TelefonProprietar tp ON p.id = tp.id_proprietar  
WHERE  
    m.id = 3;
```

5.4 PROCEDURI

Prima procedură este una de inserare. Aceasta introduce o înregistrare în tabela Piesă. A doua procedură este una de afișare. Sunt afișate id-ul, data și costul din tabela Înregistrare dar și numele și prețul pieselor corespunzătoare din tabela Piesă.

```
CREATE OR REPLACE PROCEDURE InsertPiesa(  
    p_id NUMBER,  
    p_nume VARCHAR2,  
    p_pret NUMBER,  
    p_producator VARCHAR2  
)  
AS  
BEGIN  
    INSERT INTO Piesa (id, nume, pret, producator)  
    VALUES (p_id, p_nume, p_pret, p_producator);  
    COMMIT;  
    DBMS_OUTPUT.PUT_LINE('Piesa adaugata cu succes!');  
EXCEPTION  
    WHEN OTHERS THEN  
        ROLLBACK;  
        DBMS_OUTPUT.PUT_LINE('Eroare la adaugarea piesei: ' || SQLERRM);  
END InsertPiesa;  
/
```

```
CREATE OR REPLACE PROCEDURE AfisareInformatiiInregistrare(  
    p_id_inregistrare NUMBER  
)  
AS  
BEGIN  
    FOR rec IN (SELECT i.id AS id_inregistrare,  
        i.data,  
        i.cost,  
        p.nume AS nume_piesa,  
  
        p.pret AS pret_piesa  
        FROM Inregistrare i  
        JOIN InregistrarePiesa ip ON i.id = ip.id_inregistrare  
        JOIN Piesa p ON ip.id_piesa = p.id  
        WHERE i.id = p_id_inregistrare)  
    LOOP  
        DBMS_OUTPUT.PUT_LINE('ID Inregistrare: ' || rec.id_inregistrare ||  
            ', Data: ' || rec.data ||  
            ', Cost: ' || rec.cost ||  
            ', Nume Piesa: ' || rec.nume_piesa ||  
            ', Pret Piesa: ' || rec.pret_piesa);  
    END LOOP;  
END AfisareInformatiiInregistrare;/
```

6 UTILIZARE ȘI DATE EXPERIMENTALE

6.1 REZULTATUL CREĂRII ȘI POPULĂRII BAZEI DE DATE

ID	NUME	ADRESA
1	Ion Popescu	Str. Libertatii, Nr. 10, Bucuresti
2	Maria Vasilescu	Bd. Unirii, Nr. 5, Cluj-Napoca
3	Adrian Ionescu	Bd. Victoriei, Nr. 15, Iasi
4	Elena Constantinescu	Str. Mihai Bravu, Nr. 25, Timisoara
5	Alexandru Radu	Bd. 1 Mai, Nr. 7, Craiova

Figura 8 – Tabelul Proprietar

ID	NUMAR_TELEFON	ID_PROPRIETAR
1	0712345678	1
2	0723456789	2
3	0734567890	3
4	0745678901	4
5	0756789012	5

Figura 9 – Tabelul TelefonProprietar

ID	NUME	SPECIALIZARE
1	Andrei Popa	Motor si Transmisie
2	Gabriela Dumitrescu	Suspensie si Frane
3	Mihai Vasilescu	Electricitate Auto
4	Cristina Radu	Caroserie si Vopsitorie
5	Ionel Ionescu	Diagnosticare si Service General

Figura 10 – Tabelul Mecanic

ID	CERTIFICARE	ID_MECHANIC
1	Certificat de Diagnosticare Auto	1
2	Certificat de Service Tehnic	2
3	Certificat de Reparatii Motor	3
4	Certificat de Vopsitor Auto	4
5	Certificat de Expertiza Tehnica	5

Figura 11 – Tabelul CertificareMecanic

ID	MODEL	AN_FABRICATIE	NUMAR_INMATRICULARE	KILOMETRAJ	ID_PROPRIETAR
1	Renault Megane	2018	PH-123ABC	50000	1
2	Volkswagen Golf	2019	CJ-456XYZ	60000	2
3	Ford Focus	2017	IS-789DEF	75000	3
4	Opel Astra	2020	TM-101GHI	40000	4
5	Dacia Logan	2016	DJ-202JKL	80000	5

Figura 12 – Tabelul Mașină

ID	DATA	COST	DESCRIERE	ID_MASINA	ID_MECHANIC
1	03/15/2023	300	Schimb ulei si filtre	1	1
2	04/20/2023	500	Revizie generala	2	2
3	05/10/2023	200	Schimb placute frana	3	3
4	06/25/2023	700	Reparatii motor	4	4
5	07/12/2023	100	Verificare tehnica	5	5

Figura 13 – Tabelul Înregistrare

ID	NUME	PRET	PRODUCATOR
1	Placute frana	100	Brembo
2	Filtru ulei	20	Mann-Filter
3	Bujii	30	NGK
4	Disc frana	150	ATE
5	Lampa far	15	Philips
6	Baterie auto	120	Varta

Figura 14 – Tabelul Piesă

ID	ID_INREGISTRARE	ID_PIESA
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	1	6

Figura 15 – Tabelul ÎnregistrarePiesă

6.2 REZULTATUL VEDERILOR

ID	DATA	COST	DESCRIERE	ID_MASINA	ID_MECHANIC
3	05/10/2023	200	Schimb placute frana	3	3

Figura 16 – Vedere Simplă

ID_MASINA	MODEL	AN_FABRICATIE	NUMAR_INMATRICULARE	ID_PROPRIETAR	NUME_PROPRIETAR	NUMAR_TELEFON
3	Ford Focus	2017	IS-789DEF	3	Adrian Ionescu	0734567890

Figura 17 – Vedere Complexă

6.3 REZULTATUL PROCEDURILOR

```
BEGIN      InsertPiesa(6, 'Baterie auto', 120, 'Varta'); END;
Piesa adaugata cu succes!
Statement processed. 0.00 seconds
```

Figura 18 – Procedură inserare

```
BEGIN      AfisareInformatiiInregistrare(1); END;
```

```
ID Inregistrare: 1, Data: 03/15/2023, Cost: 300, Nume Piesa: Placute frana, Pret Piesa: 100
```

```
Statement processed. 0.02 seconds
```

Figura 19 – Procedură afișare

7 CONCLUZII ȘI DIRECȚII DE DEZVOLTARE

Proiectul conturat în această lucrare a reușit într-o proporție satisfăcătoare ceea ce și-a propus, și anume, proiectarea unei baze de date care să gestioneze un service auto. Pe parcursul realizării proiectului numeroase noțiuni teoretice, în concordanță cu abilitățile practice de proiectare și implementare au fost antrenate.

Baza de date obținută poate manipula cu succes date despre majoritatea subiectelor de interes majoritar din cadrul unui service al cărei amploare care se reflectă asupra complexității datelor nu este neapărat relevantă. Desigur, însă, că modelul prezentat reprezintă o schemă simplificată pentru service-ul auto.

Avantajele sunt oferite chiar de simplitatea și complexitatea redusă a bazei de date. În urma relațiilor din cadrul ei, se pot vizualiza și introduce date în timpuri reduse.

Dezavantajele sunt constituite din lipsa complexității datelor din baza de date.

Proiectul poate fi extins însă în mai multe direcții deoarece soluția oferită este scalabilă. Cea mai rapidă îmbunătățire care se poate realiza este adăugarea atributelor pentru fiecare dintre entități, adică tratarea detaliilor (de exemplu mai multe detalii despre înregistrarea în service a unei mașini). Tot odată numărul de entități și relații poate crește, adăugând entități mai specifice (de exemplu crearea unor tabele precum electrician, tinichigiu) în locul optării pentru atribute descriptive (specializare, tabelă Mecanic).

În concluzie, proiectul și-a atins scopul și reprezintă o soluție eficientă pentru moment, oferind în același timp posibilitatea dezvoltării din mai multe perspective.

LISTA TABELELOR

- Tabelul 1 - Tabelul entității Mașină
- Tabelul 2 - Tabelul entității Proprietar
- Tabelul 3 - Tabelul entității TelefonProprietar
- Tabelul 4 - Tabelul entității Mecanic
- Tabelul 5 - Tabelul entității CertificareMecanic
- Tabelul 6 - Tabelul entității Înregistrare
- Tabelul 7 - Tabelul entității Piesă
- Tabelul 8 - Tabelul entității ÎnregistrarePiesă

LISTA FIGURILOR

- Figura 1 – Rezultatele optimizării unei baze de date [2]
- Figura 2 – Simboluri uzuale în proiectarea conceptuală [5]
- Figura 3 – Maparea efectuată în proiectarea fizică
- Figura 4 – Diagrama ER în fază inițială
- Figura 5 – Diagrama ER în fază finală
- Figura 6 – Diagrama pentru modelul logic
- Figura 7 – Tabelul cu relațiile fizice dintre cheile primare și cele străine
- Figura 8 – Tabelul entității Mașină în Oracle APEX
- Figura 9 – Tabelul entității Proprietar în Oracle APEX
- Figura 10 – Tabelul entității TelefonProprietar în Oracle APEX
- Figura 11 – Tabelul entității Mecanic în Oracle APEX
- Figura 12 – Tabelul entității CertificareMecanic în Oracle APEX
- Figura 13 – Tabelul entității Înregistrare în Oracle APEX
- Figura 14 – Tabelul entității Piesă în Oracle APEX
- Figura 15 – Tabelul entității ÎnregistrarePiesă în Oracle APEX
- Figura 16 – Vedere Simplă
- Figura 16 – Vedere Complexă
- Figura 18 – Procedură inserare
- Figura 18 – Procedură afișare

8 BIBLIOGRAFIE

[1] Hakim Lubis, Juanda, and Elviawaty Muisa Zamzami. 'Relational Database Reconstruction from SQL to Entity Relational Diagrams'. *Journal of Physics: Conference Series*, vol. 1566, no. 1, June 2020, p. 012072. DOI.org (Crossref), <https://iopscience.iop.org/article/10.1088/1742-6596/1566/1/012072> .

[2] Wu, Hong, and Caijun Cheng. 'Research on Search Optimization Based on Oracle Database'. *Journal of Physics: Conference Series*, vol. 1648, no. 4, Oct. 2020, p. 042047. DOI.org (Crossref), <https://iopscience.iop.org/article/10.1088/1742-6596/1648/4/042047> .

[3] Harrington, Jan L., editor. „Glossary”. *Relational Database Design and Implementation* (Fourth Edition), Morgan Kaufmann, 2016, pp. 669–82. ScienceDirect <https://www.sciencedirect.com/science/article/abs/pii/B9780128043998000399?via%3Dihub> .

[4] Ș.I.dr.ing. Mădălin-Dorin Pop, Cursul „Proiectarea Bazelor de Date”, Universitatea Politehnica Timișoara, 2023-2024

[5] ER Diagram Symbols and Notations, <https://www.edrawsoft.com/er-diagram-symbols.html> Accesat la 12 ianuarie 2024.