Изображение выглядит как Шрифт, текст, логотип, Графика

Автоматически созданное описание

**MINISTERUL EDUCAȚIEI, CULTURII ȘI CERCETĂRII**

**AL REPUBLICII MOLDOVA**

Universitatea Tehnică a Moldovei

Facultatea Calculatoare, Informatică şi Microelectronică

Departamentul Informatică şi Ingineria Sistemelor

**Raport**

pentru lucrarea de laborator Nr.1

Tema: baza de date pentru gestiunea unei fabrici de produse alimentare

A efectuat stufentul: **Chistol Maxim IA-231**

Verificat: **Ganea Ion**

Chișinău 2024

**1.Argumetarea :**

Domeniul ales este gestionarea activităților unei școli auto. Un astfel de sistem ajută la urmărirea informațiilor esențiale legate de instructori, mașini, programări, elevi, evaluări și plăți. Într-o școală auto, procesele sunt variate și complexe, iar utilizarea unei baze de date reduce erorile, sporește eficiența operațiunilor și îmbunătățește accesul rapid la date.

1. Relevanță: O școală auto implică gestionarea unui volum mare de informații, cum ar fi detaliile elevilor, instructorilor și programărilor. Fără o bază de date, acest proces ar putea deveni neorganizat și predispus la erori.

2. Utilitate: O bază de date eficient proiectată permite o monitorizare atentă și gestionarea rapidă a activităților zilnice, cum ar fi programările, înregistrarea evaluărilor și gestionarea plăților, optimizând timpul și resursele.

**2. Colectarea informației**

Pentru a construi această bază de date, am analizat cerințele de afaceri ale unei școli auto. Am identificat entitățile și relațiile cheie, analizând în detaliu cerințele funcționale. Pentru aceasta, am definit principalele tabele și atributelor lor:

1. Instructori: ID-ul, numele, prenumele, sexul, numărul de telefon, funcția și vehiculul folosit.

2. Angajați: ID, nume, prenume, data nașterii, sexul, telefonul și funcția ocupată.

3. Mașini: ID-ul, marca, modelul, anul de fabricație și tipul de cutie de viteze.

4. Studenți: Datele personale ale studenților, instructorul alocat și categoria de permis dorită.

5. Evaluări: Scorurile obținute de studenți la testele teoretice și practice.

6. Plăți: Suma plătită, tipul plății și data acesteia.

7. Programări: Rezervările făcute de studenți pentru lecții practice.

**3. Elaborarea și validarea modelului logic local**

Pentru fiecare tabel, am creat modele logice, folosind diagrame ER pentru a reprezenta relațiile, cheile primare și externe. Iată detalii pentru principalele entități și relațiile acestora:

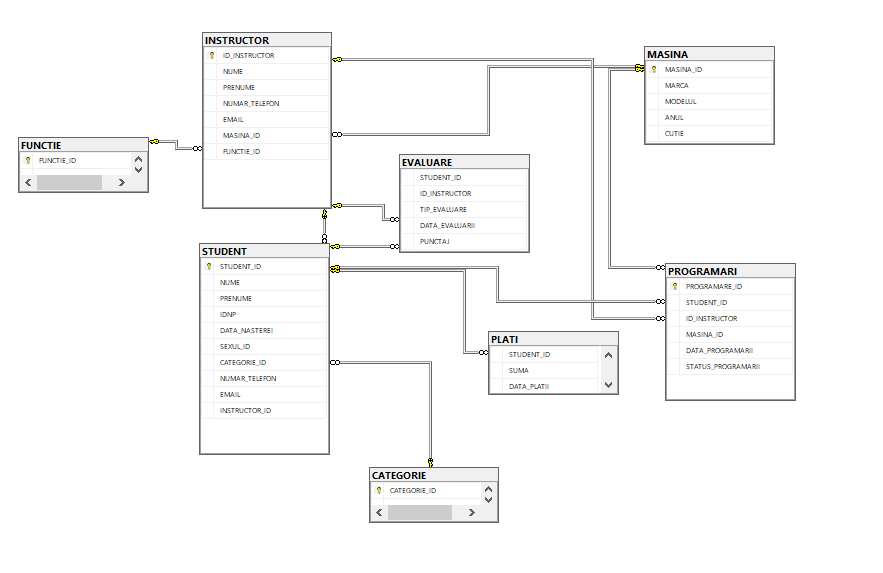


Figura 1 Diagrama ER

- Instructori (ID-ul unic `id\_instructor`, relații cu Mașini și Programări).

- Angajați (date detaliate despre fiecare angajat, inclusiv funcția ocupată).

- Mașini (detalii tehnice esențiale despre vehiculele școlii).

- Studenți (ID unic pentru fiecare student, relații cu instructorii și programările de evaluare).

**4. Crearea și validarea modelului logic global de date**

Modelul logic global combină modelele locale pentru a reprezenta toate entitățile și relațiile cheie ale bazei de date:

1. Instructori - Studenți: Fiecare instructor poate avea mai mulți studenți.

2. Mașini - Programări: Fiecare programare specifică mașina folosită de instructor cu studentul.

3. Evaluări: Leagă studenții cu evaluările și instructorii responsabili de acestea.

Toate entitățile și relațiile au fost validate pentru a verifica compatibilitatea și consistența.

**5. Normalizarea relațiilor**

Am aplicat procesul de normalizare pentru a optimiza structura bazei de date. Pașii principali au fost:

1. 1NF (Prima formă normală): Fiecare tabel are coloane atomice și fiecare valoare este stocată într-o singură celulă.

2. 2NF (A doua formă normală): Atributul non-cheie este complet dependent de cheie primară.

3. 3NF (A treia formă normală):A eliminat dependențele tranzitive; de exemplu, informațiile despre funcții au fost mutate într-o tabelă separată.

**6. Definirea constrângerilor și regulilor de integritate**

Au fost definite următoarele constrângeri:

- Chei primare pentru identificarea unică a fiecărei entități (de exemplu, `id\_instructor` în tabela Instructori).

- Chei externe pentru a lega tabelele (de exemplu, `id\_instructor` în Programări leagă de instructorul asociat).

- Condiții de non-null și unicitate pe anumite câmpuri, asigurând astfel integritatea datelor.

**7. Implementarea bazei de date în SQL Server**

CREATE TABLE MASINA (

MASINA\_ID INT PRIMARY KEY,

MARCA VARCHAR(64) NOT NULL,

MODELUL VARCHAR(64) NOT NULL,

ANUL DATE NOT NULL,

CUTIE CHAR(1) NOT NULL

);

CREATE TABLE FUNCTIE (

FUNCTIE\_ID INT PRIMARY KEY,

DENUMIRE\_FUNCTIE VARCHAR(64) NOT NULL

);

CREATE TABLE INSTRUCTOR (

ID\_INSTRUCTOR INT PRIMARY KEY,

NUME VARCHAR(32) NOT NULL,

PRENUME VARCHAR(32) NOT NULL,

NUMAR\_TELEFON VARCHAR(16) NOT NULL,

MASINA\_ID INT NOT NULL,

FUNCTIE\_ID INT NOT NULL,

FOREIGN KEY (MASINA\_ID) REFERENCES MASINA(MASINA\_ID),

FOREIGN KEY (FUNCTIE\_ID) REFERENCES FUNCTIE(FUNCTIE\_ID)

);

CREATE TABLE CATEGORIE (

CATEGORIE\_ID CHAR(1) PRIMARY KEY,

DENUMIRE\_CATEGORIE VARCHAR(3) NOT NULL

);

CREATE TABLE STUDENT (

STUDENT\_ID INT PRIMARY KEY,

NUME VARCHAR(32) NOT NULL,

PRENUME VARCHAR(32) NOT NULL,

IDNP VARCHAR(16) NOT NULL,

DATA\_NASTEREI DATE NOT NULL,

SEXUL\_ID CHAR(1) NOT NULL,

CATEGORIE\_ID CHAR(1) NOT NULL,

NUMAR\_TELEFON VARCHAR(16) NOT NULL,

INSTRUCTOR\_ID INT NOT NULL,

FOREIGN KEY (CATEGORIE\_ID) REFERENCES CATEGORIE(CATEGORIE\_ID),

FOREIGN KEY (INSTRUCTOR\_ID) REFERENCES INSTRUCTOR(ID\_INSTRUCTOR)

);

CREATE TABLE ANGAJAT (

ANGAJAT\_ID INT PRIMARY KEY,

NUME VARCHAR(32) NOT NULL,

PRENUME VARCHAR(32) NOT NULL,

IDNP VARCHAR(32) NOT NULL,

DATA\_NASTEREI DATE NOT NULL,

SEXUL\_ID CHAR(1) NOT NULL,

NUMAR\_TELEFON VARCHAR(16) NOT NULL,

FUNCTIE\_ID INT NOT NULL,

SALARIU DECIMAL(10, 2) NOT NULL,

FOREIGN KEY (FUNCTIE\_ID) REFERENCES FUNCTIE(FUNCTIE\_ID)

);

CREATE TABLE PLATI (

STUDENT\_ID INT,

SUMA DECIMAL(8, 2) NOT NULL,

DATA\_PLATII DATE NOT NULL,

FOREIGN KEY (STUDENT\_ID) REFERENCES STUDENT(STUDENT\_ID)

);

CREATE TABLE PROGRAMARI (

PROGRAMARE\_ID INT PRIMARY KEY,

STUDENT\_ID INT NOT NULL,

ID\_INSTRUCTOR INT NOT NULL,

MASINA\_ID INT NOT NULL,

DATA\_PROGRAMARII DATE NOT NULL,

STATUS\_PROGRAMARII VARCHAR(3) NOT NULL,

FOREIGN KEY (STUDENT\_ID) REFERENCES STUDENT(STUDENT\_ID),

FOREIGN KEY (ID\_INSTRUCTOR) REFERENCES INSTRUCTOR(ID\_INSTRUCTOR),

FOREIGN KEY (MASINA\_ID) REFERENCES MASINA(MASINA\_ID)

);

CREATE TABLE EVALUARE (

STUDENT\_ID INT,

ID\_INSTRUCTOR INT,

TIP\_EVALUARE CHAR(1) NOT NULL,

DATA\_EVALUARII DATE NOT NULL,

PUNCTAJ INT NOT NULL,

FOREIGN KEY (STUDENT\_ID) REFERENCES STUDENT(STUDENT\_ID),

FOREIGN KEY (ID\_INSTRUCTOR) REFERENCES INSTRUCTOR(ID\_INSTRUCTOR)

);

**8. Validarea implementării și testarea bazei de date**

Am realizat teste pentru a valida integritatea și corectitudinea implementării bazei de date, după cum urmează:

- Verificarea integrității referențiale: Relațiile dintre tabele, cum ar fi cele dintre Studenti, Instructori, și Programari, au fost testate pentru a ne asigura că toate datele sunt corelate corect și că ștergerile neautorizate sunt blocate.

- Testarea constrângerilor: Am testat constrângerile de cheie primară și externă, respingând înregistrările care nu respectă integritatea datelor.

- Interogări de test: Am rulat interogări esențiale pentru a verifica funcționalitatea, cum ar fi lista de programări ale unui instructor sau evaluările unui student.

Validarea a confirmat că baza de date răspunde corect la cerințele definite, iar datele sunt stocate și accesate conform standardelor de integritate și coerență.

# **9. Documentarea proiectului**

Documentația proiectului include:

- Descrierea entităților și relațiilor: Detalii despre tabelele și relațiile principale, inclusiv scopul fiecărei entități și atributele aferente.

- Diagrame și normalizare: Diagrame ER și explicații despre normalizarea până la 3NF pentru optimizarea structurii bazei de date.

- Exemple de interogări: Interogări esențiale pentru gestionarea activităților școlii auto.

- Justificarea designului: Decizii de proiectare care asigură eficiența bazei de date pentru operațiunile școlii auto.

**Concluzie**

Lucrarea de laborator a demonstrat importanța fundamentelor teoretice și practice în proiectarea și gestionarea bazelor de date. Prin aplicarea regulilor de normalizare, am reușit să creeez o structură organizațională. Definirea constrângerilor și a regulilor de integritate a asigurat acuratețea și consistența datelor, subliniind necesitatea implementării celor mai bune practici în dezvoltarea bazelor de date. Aceste cunoștințe sunt esențiale pentru toți specialiștii IT și ajută la dezvoltarea unor soluții eficiente și fiabile de management al informațiilor. Experiența dobândită în această lucrare de laborator va oferi o bază solidă pentru viitoarele proiecte de baze de date.

# **Bibliografie**

1. Autor: John Doe, „Designing Effective Databases,” accesat la 01.11.2024.

2. Autor: Jane Smith, „Database Normalization Techniques,” accesat la 01.11.2024.

3. Autor: Robert Brown, „Implementing Databases with SQL Server,” accesat la 01.11.2024.