**MINISTERUL EDUCAŢIEI ȘI CERCETĂRII AL REPUBLICII MOLDOVA CENTRUL DE EXCELENŢĂ ÎN INFORMATICĂ ŞI**

**TEHNOLOGII INFORMAŢIONALE**

### CATEDRA DE INFORMATICĂ I



LUCRARE INDIVIDUALĂ

**Nr. 1**

### LA DISCIPLINA:

ASISTENȚA PENTRU BAZA DE DATE.

## **TEMA:** BAZA DE DATE A UNUI COMPANIEI PETROM MOLDOVA.

**Grupa: P-2231**

**Elev(ul/a): Suhari Alexei**

**Profesor: Covali Eugenia**

### Chişinău, 2024

**Cuprins**

[Introducere 3](#_Toc181866500)

[Descrierea bazei de date. 5](#_Toc181866503)

[1. Entități: 5](#_Toc181866504)

[2. Schema relațională: 6](#_Toc181866505)

[3. Normalizarea: 6](#_Toc181866506)

[Concluzii 9](#_Toc181866507)

[Bibliografie 9](#_Toc181866508)

# 0

# Introducere

Proiectul „PetromMoldova” este o bază de date creată pentru a ajuta la gestionarea unei rețele de stații de alimentare. Aceasta organizează și stochează informații importante, cum ar fi detalii despre angajați, produse petroliere, clienți, tranzacții, furnizori și comenzi.

Baza de date este proiectată pentru a face mai ușoară monitorizarea activităților zilnice și pentru a oferi o imagine clară a vânzărilor și resurselor. De asemenea, permite actualizarea rapidă a datelor, urmărirea stocurilor de produse și generarea de rapoarte. Astfel, sistemul ajută compania să funcționeze mai eficient și să răspundă mai bine nevoilor clienților.

# Diagrama E-R

## 

# Descrierea bazei de date.

## Entități:

La baza acestei baze de date stau 8 table care permit interacționarea confortă cu toate

datele companiei Petrom Moldova. Totul începe cu tabelul *Stații alimentare.* Tabelul care salveaza filialele companiei, informatia de baza a acestora și pe lângă aceasta permite desparțirea tuturor celorlalte tabel în informatii în funcție cu ce stație are relație.

Dupa merg 2 tabele informaționale care salveaza datele despre *Produsele Petroliere* din diferite stații și despre *angajații* acestora. In entitatea *Angajați* pe lângă datele de baza ca nume, prenume, telefon și salariu mai este prezent și datele pentru conectarea în programa care v-a folosi aceasta baza de date ca sursa de informație. Aceste date sunt special cu tipul de date NVARCHAR deoarece este un tip de date care pe lânga posibilitatea de a salva date de tip text, mai are și proprietatea de a securiza în plus datele pe care le stochează. În entitatea *ProdusePetroliere* sunt descrise produsele cu care se comercializează pe teritoriul fiecarei stații, de la denumire până la preț, cantitatea de produs ce poate fi stocat și cantitatea disponibilă de vânzare.

Pentru procesul de vânzare a producției avem tabelele *Tranzacții*, *DetaliiTranzacții* și *Client*, care permite monitorizarea vânzărilor. În tabela *Client*  sunt salvați toți cei care s-au înregistrat și pot primi reduceri și cumpărători simpli care sunt salvați cu denumirea defaultuser pentru a nu supraîncărca baza de date. Apoi tabela *Tranzacții* unde monitorizam data, stația și angajatul și clientul cu care a avut loc tranzacția. Iar în *DetaliiTranzacție* sunt stocate toate produsele care au fost cumparate în aceasta tranzacție aflând cantitate, prețul și id-ul produsului care a fost cumparat.

Iar în final avem tabelele *Furnizori* și *ComenziFurnizori* unde putem observa unele date despre diferiți furnizori de la care se poate efectua cumpararea diferitor produse petroliere și pe lângă aceasta ne permite să salvam datele despre fiecare comanda efectuat cătr un anumit furnizor, cantitatea, tipul produsului, statia unde v-a fi livrat și nu în ultimul rând data comenzii acestor produse.

Entitățile sunt obiecte, persoane, locuri, evenimente sau concepte despre care dorim să stocăm informații într-o bază de date. Ele devin tabele în baza de date.

Atributele sunt caracteristici sau informații specifice despre o entitate și devin coloane în tabelul corespunzător entității.

## Schema relațională:

 **StatiiAlimentare** ↔ **Angajati**: relație de 1 la M (o stație poate avea mai mulți angajați).

 **StatiiAlimentare** ↔ **Tranzactii**: relație de 1 la M (o stație poate avea mai multe tranzacții).

 **StatiiAlimentare** ↔ **Comenzi**: relație de 1 la M (o stație poate avea mai multe comenzi).

 **Angajati** ↔ **Tranzactii**: relație de 1 la M (un angajat poate efectua mai multe tranzacții).

 **Clienti** ↔ **Tranzactii**: relație de 1 la M (un client poate avea mai multe tranzacții).

 **Tranzactii** ↔ **DetaliiTranzactii**: relație de 1 la M (o tranzacție poate avea mai multe detalii de produse petroliere achiziționate).

 **ProdusePetroliere** ↔ **DetaliiTranzactii**: relație de 1 la M (un produs poate fi în mai multe detalii de tranzacții).

 **Furnizori** ↔ **Comenzi**: relație de 1 la M (un furnizor poate avea mai multe comenzi).

## Normalizarea:

**Normalizarea** este procesul de organizare a datelor într-o bază de date relațională astfel încât să se minimizeze redundanța și să se prevină anomaliile de actualizare. Scopul este de a distribui datele în tabele mai mici și interconectate logic, care să permită stocarea eficientă și coerentă a informațiilor. Normalizarea presupune aplicarea unor reguli specifice pentru a transforma structura inițială a bazei de date într-o formă mai „normalizată” și mai ușor de gestionat.

Etapele normalizării:

* **1NF**: Elimina grupurile de valori multiple și asigură că datele sunt atomice.
* **2NF**: Elimina dependențele parțiale de cheia primară.
* **3NF**: Elimina dependențele tranzitive între atributele non-cheie.
* **4NF**: Elimina dependențele multi-valore.

# 5NF: Asigură că fiecare relație este ireductibilă și că nu există ambiguități în descompunerea datelor.Interogări:

-- 1 Interogare de afisare a informatiei complete despre tranzactiile pe o anumita data

SELECT t.DataTranzactie, c.NumePrenume, s.Locatie, p.Denumire, dt.Cantitate, dt.PretTotal --selectarea coloanelor afisate

FROM Tranzactii t --entitatea din care luam datele

JOIN Clienti c ON t.ClientID = c.ClientID --junctiunea entitatii client cu tranzactii

JOIN StatiiAlimentare s ON t.StatiaID = s.StatiaID --junctiunea entitatii statiialimentare cu tranzactii

JOIN DetaliiTranzactii dt ON t.TranzactieID = dt.TranzactieID --junctiunea entitatii client cu tranzactii

JOIN ProdusePetroliere p ON dt.ProdusID = p.ProdusID --junctiunea entitatii produsepetroliere cu datatranzactii

WHERE t.DataTranzactie BETWEEN '2024-01-01' AND '2024-12-31' --acum efectuam verificarea daca datatranzactiei intre 2 dati la alegere

ORDER BY t.DataTranzactie DESC; --ordonarea datelor dupa datatranzactiei

-- 2 Interogare de afisare produselor petroliere ce au in denumire 'benzina'

GO

CREATE VIEW Benzina AS -- crearea viziunii Benzina

SELECT \* FROM ProdusePetroliere -- selectam tot tabelul din ProdusePetroliere

WHERE Denumire LIKE '%benzina%'; -- verificam daca nenumirea contine stringul benzina

GO

SELECT \* FROM Benzina

-- 3 Interogare de afisare a tuturor tranzactiilor si venitul pentru statiile ce au mai mult de 2 tranzactii

SELECT StatiaID, COUNT(\*) AS TotalTranzactii, SUM(SumaTotala) AS Venit --afisarea coloanelor tabelului afiasat

FROM Tranzactii --alegerea tabelului cu care lucram

GROUP BY StatiaID --grupam informatia primita dupa StatiaID

HAVING COUNT(\*) > 2; --pentru verificarea nr de tranzactii folosim having

-- 4 Interogare de afisare nr de tranzactii a fiecarui client

GO

CREATE VIEW nrTranzactii AS

SELECT c.NumePrenume,

(SELECT COUNT(\*) FROM Tranzactii t --crearea unei coloane pe baza unui select care calculeaza

WHERE t.ClientID = c.ClientID) AS TotalTranzactii --nr de tranzactii pentru fiecare client

FROM Clienti c;

GO

SELECT \* FROM nrTranzactii

-- 5 Interogare de afisare clientilor si sumei medii cheltuite care au o cumparatura mai mare de o suma

SELECT c.NumePrenume, AVG(t.SumaTotala) AS MediaSumelorCheltuite FROM Clienti --afisam clientii si mediul de bani cheltuiti

JOIN Tranzactii t ON c.ClientID = t.ClientID --efectuam jonctiunea cu tabelu tranzactii

WHERE EXISTS ( --afisam tabelele care au macar o returnare din select

SELECT 1

FROM Tranzactii t

WHERE t.ClientID = c.ClientID AND t.SumaTotala > 200 AND c.NumePrenume NOT LIKE 'defaultuser') --verificam daca suma cheltuita e mai mare ca 200

GROUP BY NumePrenume

# Indexi:

CREATE INDEX idx\_nume\_statie ON StatiiAlimentare(NumeStatia)

CREATE INDEX idx\_locatie\_statie ON StatiiAlimentare(Locatie)

CREATE INDEX idx\_angajat\_statie ON Angajati(StatiaID)

CREATE INDEX idx\_nume\_angajat ON Angajati(Nume)

CREATE INDEX idx\_prenume\_angajat ON Angajati(Prenume)

CREATE INDEX idx\_nume\_produs ON ProdusePetroliere(Denumire)

CREATE INDEX idx\_produs\_statie ON ProdusePetroliere(StatiaID)

CREATE INDEX idx\_client\_tranzactie ON Tranzactii(ClientID)

CREATE INDEX idx\_tranzactie\_statie ON Tranzactii(StatiaID)

CREATE INDEX idx\_data\_tranzactie ON Tranzactii(DataTranzactie)

CREATE INDEX idx\_produs\_detaliitranzactie ON DetaliiTranzactii(ProdusID)

CREATE INDEX idx\_nume\_furnizor ON Furnizori(NumeFurnizor)

CREATE INDEX idx\_nume\_comenzifurnizor ON ComenziFurnizori(FurnizorID)

CREATE INDEX idx\_statie\_comenzifurnizor ON ComenziFurnizori(StatiaID)

CREATE INDEX idx\_produs\_comenzifurnizor ON ComenziFurnizori(ProdusID)

# Concluzii

În concluzie, acest proiect de baze de date a fost conceput pentru a asigura o structură organizată și eficientă pentru gestionarea activităților și a datelor esențiale ale unei rețele de stații de alimentare.

Prin utilizarea unor tabele interconectate și respectarea principiilor de normalizare, baza de date permite stocarea și actualizarea informațiilor despre clienți, angajați, tranzacții și produse într-un mod eficient și sigur.

Această structură nu doar optimizează gestionarea datelor, dar contribuie și la facilitarea raportării și analizelor pentru decizii informate. Proiectul demonstrează importanța unei baze de date bine proiectate, capabile să sprijine operațiunile curente și să răspundă cerințelor unei afaceri în dezvoltare.

# Bibliografie

1: Google: <https://www.google.com/>

2: Learn Microsoft: <https://learn.microsoft.com/>

3: Caietul de studii al elevului.

