



**Universitatea Tehnică “Gheorghe Asachi” din Iași**



**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**

# **BAZE DE DATE**

## **proiect**

***Tema: Sistem de Gestionare și Monitorizare a  
Evenimentelor unei Formații***

Student: Pitic Emanuel

Coordonator:

Grupa : 1308A

Asist. Dr. Cătălin Mironeanu

**2024**

## Descrierea proiectului:

Analiza, proiectarea și implementarea unei baze de date care să modeleze activitatea unei formații muzicale, precum și colaboratorii acesteia, concomitent cu evenimentele pe care aceasta le susține, playlist-urile ce vor fi cântate la eveniment (alături de melodiile specifice) și locațiile unde se vor desfășura evenimentele.

## Descrierea cerintelor si modul de organizare al proiectului:

Desfășurarea optimă a activităților unei formații presupune o bună gestiune a resurselor de care aceasta dispune (colaboratorii), cât și de o gestiune bună a evenimentelor pe care aceasta le organizează: locațiile unde evenimentele se vor desfășura, datele de contact pentru clienții formației, tipurile de playlist-uri care se ve cânta la un eveniment, respectiv melodiile care corespund fiecărui playlist.

Astfel, fiind o entitate privată, se dorește să se afle informații care țin de buna funcționare a formației, de pildă:

- Stocarea informațiilor despre clienți, locații și colaboratori;
- Administrarea detaliilor evenimentelor, incluzând tipul, data, durata și costurile;
- Gestionarea playlisturilor muzicale asociate fiecărui eveniment;
- Generarea rapoartelor financiare pentru analiza veniturilor și costurilor asociate.
- Determinarea colaboratorilor care au fost cel mai solocitați
- Determinarea celor mai vizitate locații / celor mai puțin vizitate pentru a realiza reclame în acea zonă mai mult/ mai puțin
- Determinarea categoriilor de evenimente cel mai des realizate
- Determinarea Profitului Total al Formației

În acest context, se dorește implementarea unei baze de date prin intermediul careia să poată fi gestionate toate aceste resurse, astfel încât să se ofere formației un bun management al evenimentelor acesteia.

**Limitări:** Proiectul nu tratează problema rezervării pentru evenimente care nu au legătură cu serviciile formației, cum ar fi gestiunea facturilor.

## **Reguli de gestiune**

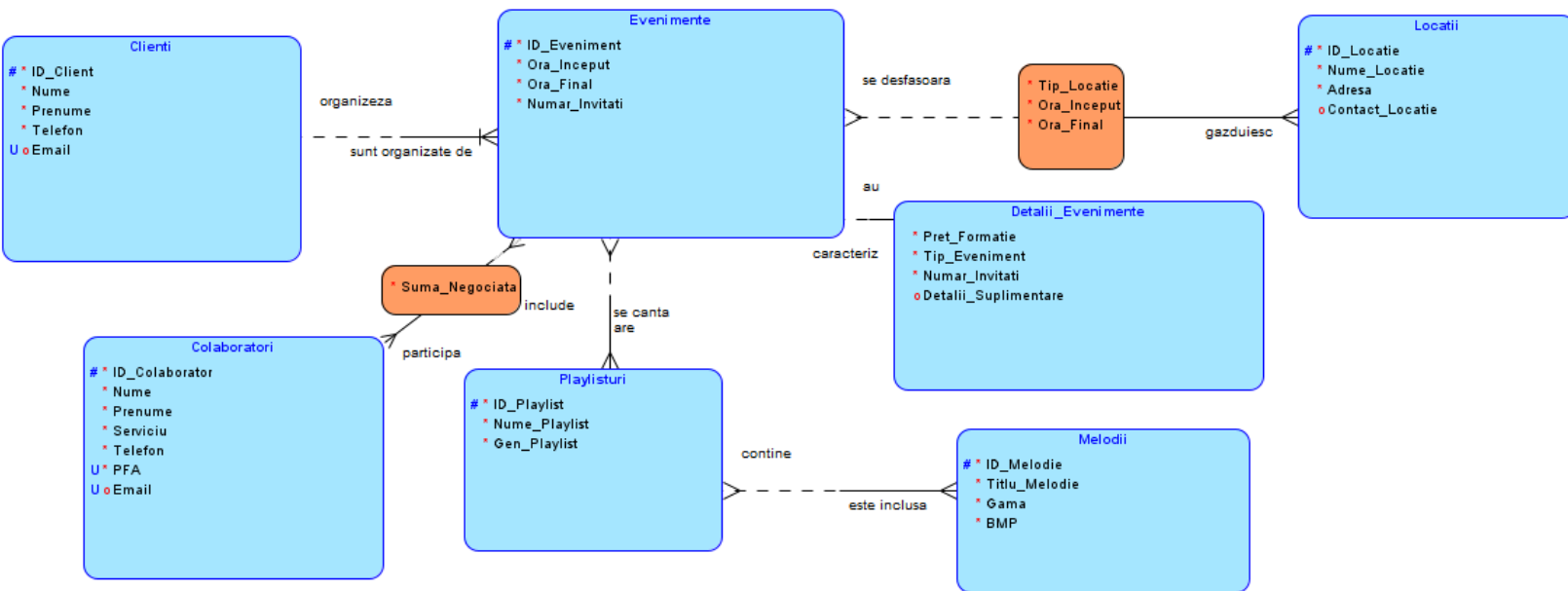
- ✓ Un client poate organiza mai multe evenimente, iar fiecare eveniment are organizator unic
- ✓ Prețul formației variază între 3000-20000 de euro
- ✓ Un eveniment se poate desfășura în mai multe locații
- ✓ La un eveniment participă mai mulți colaboratori, cu PFA unic, cu preț negociat per eveniment
- ✓ Evenimentele trebuie să aibă loc în viitor
- ✓ Melodiile trebuie să aibă o gamă validă, cu un BPM rezonabil
- ✓ Melodiile sunt gestionate pentru a afla cele mai cântate melodii, a.î. să se determine cele mai cântate melodii.

## **Structura tabeli logice:**

Baza de date include următoarele entități:

- Clienti
- Evenimente
- Locații
- Detalii Evenimente
- Colaboratori
- Playlisturi
- Melodii

## Modelul logic:



## Descrierea relațiilor dintre entități:

### Relații:

1. Clienti - Evenimente (1:N): Fiecare client poate participa la mai multe evenimente, dar fiecare eveniment este asociat unui singur client.
2. Colaboratori - Colab\_Event - Evenimente (N:M): Relația many-to-many este modelată prin tabela colab\_event, care include detalii despre costurile asociate colaboratorilor pentru fiecare eveniment.
3. Locații - Event\_Loc - Evenimente (N:M): Relația este gestionată printr-o tabelă intermediară care asociază locațiile utilizate pentru evenimente.
4. Melodii - Playlisturi - Evenimente (N:M): Tabela playlist\_melodie și eveniment\_playlisturi permit asocierea melodiilor și playlisturilor cu evenimentele organizate.

## Normalizare:

- **1NF:** Toate tabelele conțin valori atomice și unice.

Exemplu: Tabela *Cienti* are câmpuri indivizibile (nume, prenume, telefon) și fiecare client este identificat unic prin *id\_client*; totodată, nu există grupuri sau mai coloane care să definească mai multe valori.

- **2NF:** Atributele non-cheie sunt complet dependente de cheia primară.

Exemplu: În tabela *Detalii Evenimente*, toate câmpurile (*pret\_formatie*, *tip\_eveniment*, *numar\_invitati*) sunt dependente de *id\_eveniment*; Totodată, relațiile many-to-many sunt tratate prin intermediul unei tabele intermediare, de pildă tabela *Colab\_Event*, ce conectează tabela *colaboratori* și *evenimente* prin id-urile lor. Aceasta asigură crearea unei tabele noi normalizate, eliminând astfel dependențele parțiale.

- **3NF:** Eliminarea dependențelor tranzitive.

Exemplu: În tabela *Colab\_Event*, câmpurile sunt dependente doar de cheia compusă (*id\_eveniment*, *id\_colaborator*), fără redundanțe. Totodată, câmpul *suma* nu depinde de nici una dintre cele 2 câmpuri-pereche ce alcătuiesc cheia, fiind un element specific doar rândului acestuia.

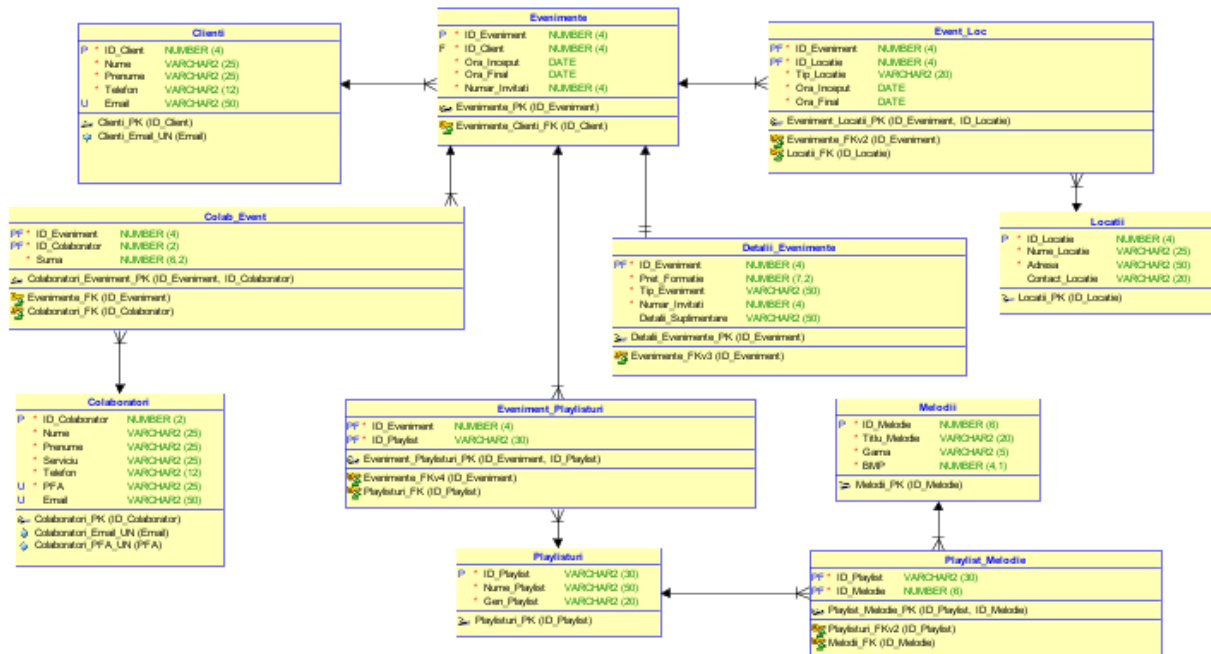
## Beneficiile normalizării

### 1. Eliminarea redundanței:

- Datele nu sunt duplicate, ceea ce reduce spațiul de stocare necesar și minimizează riscul de inconsistență a datelor.
- Exemplu: Detaliile colaboratorilor sunt stocate o singură dată în tabela **Colaboratori** și utilizate prin chei externe în **Colab\_Event**.

### 2. Îmbunătățirea integrității datelor:

- Relațiile bine definite între tabele previn erorile de referință și mențin coerența.
- Exemplu: Cheia externă *id\_client* din tabela **Evenimente** garantează că fiecare eveniment este asociat unui client valid.



- id\_client: Cheie primară, numerică, generată automat;
- nume, prenume: Câmpuri obligatorii, validate pentru a nu conține cifre;
- telefon: Validare pentru format numeric, lungime maximă 12 caractere;
- email: Validare REGEXP pentru format standard;

- id\_colaborator: Cheie primară, numerică;
- nume, prenume: Informații obligatorii;
- serviciu: Domenii valide (DJ, Inginer Sunet, Solist etc.), validate prin constrângerea CHECK;
- telefon, email: Similar cu tabelul Clienti, cu validare REGEXP pentru email;

- id\_eveniment: Cheie primară, numerică, autoincrement;
- id\_client: Cheie externă către tabela Clienti;
- ora\_inceput, ora\_final: Validare cronologică prin trigger;
- numar\_invitati: Numeric, verificat să fie pozitiv;

- id\_locatie: Cheie primară, numerică;
- nume\_locatie, adresa: Detalii esențiale despre locație;

- contact\_locatie: Telefon de contact pentru locație;
- 5. **Melodii:**
  - id\_melodie: Cheie primară, numerică;
  - titlu\_melodie: Nume melodie;
  - gama: Validare pentru notația muzicală standard (A-G, cu sufixe # sau b);
  - bmp: Valori permise: 0-200 (bătăi pe minut);
- 6. **Playlisturi:**
  - id\_playlist: Cheie primară, numerică;
  - nume\_playlist, gen\_playlist: Genuri predefinite (Clasic, Populara, Modern etc.);
- 7. **Playlisturi\_Melodii:**
  - id\_playlist, id\_melodie: Tabel intermediar pentru asocierea melodiilor în playlisturi;
- 8. **Colab\_Event:**
  - id\_eveniment, id\_colaborator, suma: Legătură many-to-many între colaboratori și evenimente, cu detalii financiare;

## Constrângeri utilizate în baza de date

Constrângerile sunt mecanisme utilizate pentru a asigura integritatea și consistența datelor din baza de date. Acestea impun reguli asupra datelor introduse, prevenind erorile și asigurând coerența între tabele. Mai jos sunt detaliate constrângerile aplicate în proiectul de gestionare a evenimentelor.

### 1. Chei Primare (Primary Key)

O cheie primară identifică în mod unic fiecare înregistrare dintr-o tabelă. Aceasta previne existența rândurilor duplicate.

#### Exemple de chei primare:

- id\_client în tabelul Clienti: Asigură că fiecare client are un identificator unic.
  - id\_eveniment în tabelul Evenimente: Garantează unicitatea fiecărui eveniment.
- Beneficii:
- Asigură identificarea rapidă și precisă a unei înregistrări.
- Ajută la stabilirea relațiilor între tabele (de exemplu, ca referință pentru cheile externe).

### 2. Chei Externe (Foreign Key)

Cheile externe asigură integritatea relațională între două tabele, legând o coloană dintr-o tabelă la cheia primară a altei tabele.

#### Exemple de utilizare:

- id\_client din tabelul Evenimente este o cheie externă către tabelul Clienti.
- id\_playlist din tabelul Playlisturi\_Melodii este o cheie externă către tabelul Playlisturi.

Beneficii:

Previne introducerea de date invalide (ex. un eveniment asociat unui client inexistent).

Menține coerența relațiilor între entități.

### 3. UNIQUE

Constrângerea UNIQUE asigură că valorile dintr-o anumită coloană sunt unice în întregul tabel.

#### Exemple de utilizare:

- email în tabelul Clienti și Colaboratori: Previne introducerea a două adrese de email identice.

- nume\_playlist în tabelul Playlisturi: Garantează unicitatea fiecărui playlist.

Beneficii:

Evită duplicatele în coloanele critice.

Crește acuratețea informațiilor stocate.

#### 4. NOT NULL

Constrângerea NOT NULL impune ca o coloană să nu accepte valori lipsă (NULL).

Exemple de utilizare:

- nume și prenume în tabelul Clienți și Colaboratori: Asigură că numele clienților și colaboratorilor sunt întotdeauna disponibile.
- ora\_inceput și ora\_final în tabelul Evenimente: Garantează că fiecare eveniment are specificată durata.

Beneficii:

Asigură completitudinea datelor.

Previne erorile cauzate de câmpuri necompletate.

#### 5. CHECK

Constrângerea CHECK validează valorile dintr-o coloană conform unei reguli specificate.

Exemple de utilizare:

- Pret\_Formatie în tabelul Evenimente: Valoarea trebuie să fie  $\geq 0$ .
- bmp în tabelul Melodii: Valoarea trebuie să fie între 0 și 200.
- gen\_playlist în tabelul Playlisturi: Poate conține doar valori predefinite (ex. „Clasic”, „Pop”, „Populară”).

Beneficii:

Crește calitatea datelor prin prevenirea introducerii de valori incorecte.

Reduce riscul erorilor logice în aplicație.

#### 6. Autoincrement:

Generarea automată a valorilor pentru cheile primare, cum ar fi **client\_id**, elimină necesitatea atribuirii manuale a identificatoarelor unici, reducând riscul erorilor și garantând un flux de date consistent.

## Concluzii

Proiectul prezintă o bază de date bine organizată, care respectă principiile de normalizare pentru a elimina redundanțele și a asigura integritatea datelor. Designul este scalabil și poate fi extins ușor pentru a răspunde nevoilor viitoare, cum ar fi adăugarea de noi funcționalități sau generarea de rapoarte detaliate. Printre posibilele extensii se numără generarea facturilor, managementul taxelor către stat, închirierea echipamentelor pentru evenimente și posibilitatea de a transforma o parte din colaboratori în membri-angajați ai formației, renunțând la forma de PFA.