Universitatea Tehnică "Gheorghe Asachi" din Iaşi

FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE

BAZE DE DATE

Temă de casă

Tema: Sistem de Gestionare și Monitorizare a Evenimentelor Foto-Video

Student: Sfichi Alin-Ionuț

Grupa : 1308A

Coordonator:

ș.l. dr.ing. Buțincu Cristian-Nicolae

2024

Descrierea proiectului:

Scopul proiectului este proiectarea și implementarea unei baze de date pentru gestionarea eficientă a evenimentelor foto-video. Sistemul urmărește să simplifice procesele administrative asociate cu organizarea evenimentelor, cum ar fi nunți, botezuri sau petreceri aniversare, oferind o soluție centralizată pentru stocarea și analiza informațiilor relevante.

Platforma permite:

- Stocarea informațiilor de contact ale clienților și locațiilor, facilitând comunicarea și planificarea eficientă.
- Administrarea colaboratorilor implicați, incluzând detalii despre serviciile lor și costurile asociate.
- Gestionarea detaliilor evenimentelor, cum ar fi tipul, data, durata și starea acestora (confirmat, anulat, în așteptare).
- Urmărirea materialelor realizate pentru fiecare eveniment, cum ar fi fotografii și videoclipuri, cu linkuri pentru acces facil.
- Generarea de rapoarte financiare pentru a analiza veniturile obținute, costurile colaboratorilor și profitul net al fiecărui eveniment.

Limitări: Nu se stochează informații despre echipamentele tehnice necesare.

Use-case-uri implementate:

1. ***Adăugare Client***

Scop: Adăugarea unui client nou în baza de date, asigurând integritatea și unicitatea datelor.
Testare: Se confirmă că nu există deja un client cu același email, iar eventualele erori sunt gestionate pentru a menține consistența bazei de date.

2. ***Adăugare Eveniment***

Scop: Crearea unui eveniment nou asociat unui client existent.
Testare: Validarea existenței clientului și a unicității evenimentului pentru client, asigurând corectitudinea relațiilor între tabele.

3. ***Finalizarea unui Eveniment***

Scop: Actualizarea stării unui eveniment existent la "Finalizat".
Testare: Verificarea existenței evenimentului și a stării acestuia pentru a confirma că nu este deja finalizat.

4. ***Anularea unui Eveniment***

Scop: Setarea stării unui eveniment existent la "Anulat" și resetarea avansului asociat.
Testare: Validarea existenței evenimentului și verificarea stării sale actuale pentru a permite modificarea.

5. ***Actualizare Date Client***

Scop: Modificarea informațiilor unui client existent, cum ar fi numele, prenumele, telefonul sau emailul.
Testare: Confirmarea existenței clientului și validarea noilor date pentru respectarea constrângerilor bazei de date

6. ***Afișarea Evenimentelor Viitoare ale unui Client***

Scop: Obținerea unei liste a evenimentelor viitoare pentru un client specific.
Testare: Verificarea existenței clientului și selectarea evenimentelor programate în viitor.

7. ***Calcularea Venitului Total din Evenimente***

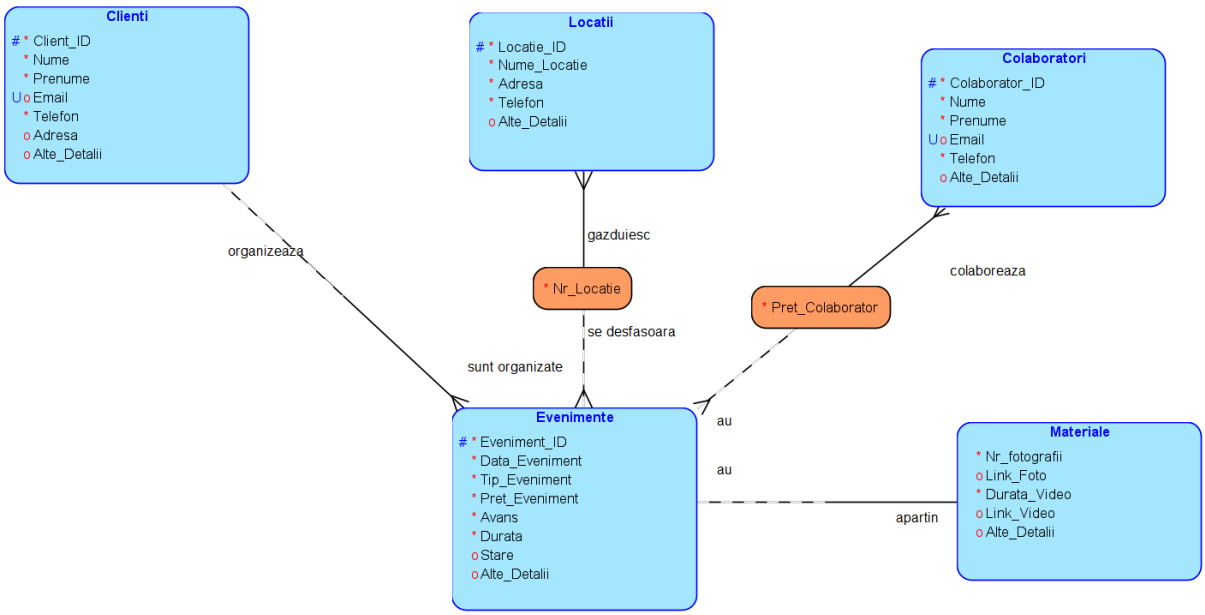
Scop: Calcularea veniturilor totale generate de evenimente într-o anumită perioadă de timp.
Testare: Validarea intervalului de timp specificat și calcularea corectă a veniturilor totale din tabelul Evenimente.

Structura bazei de date:

Baza de date include următoarele tabele:

1. **Clienti**
2. **Colaboratori**
3. **Evenimente**
4. **Locatii**
5. **Colaboratori_Eveniment (Relație Many-to-Many)**
6. **Locatii_Eveniment (Relație Many-to-Many)**
7. **Materiale**

Modelul logic:



Descrierea relațiilor dintre entități:

Relațiile dintre entitățile bazei de date au fost definite pentru a reflecta legăturile logice din cadrul sistemului:

1. **Clienti - Evenimente (1:N):**
   - Fiecare client poate participa la mai multe evenimente, dar fiecare eveniment este asociat unui singur client.
   - Legătura este realizată prin intermediul câmpului *client_id* ca cheie externă în tabelul **Evenimente**.

2. **Colaboratori - Colaboratori_Eveniment - Evenimente (N:M):**
   - Relația many-to-many este modelată cu ajutorul tabelului intermediar **Colaboratori_Eveniment**, care conține detalii despre colaboratori (ex. costul asociat fiecăruia).
   - Fiecare colaborator poate participa la mai multe evenimente, iar fiecare eveniment poate avea mai mulți colaboratori.

3. **Locatii - Locatii_Eveniment - Evenimente (N:M):**
   - Relația many-to-many este modelată cu tabelul intermediar **Locatii_Eveniment**, care detaliază locațiile utilizate pentru evenimente.
   - Fiecare locație poate găzdui mai multe evenimente, iar fiecare eveniment poate avea una sau mai multe locații.

4. **Evenimente - Materiale (1:1):**
    - Fiecare eveniment are asociat un set unic de materiale, cum ar fi fotografii și videoclipuri.
    - Legătura este realizată prin câmpul *eveniment_id* ca cheie primară și externă în tabelul **Materiale**.

Normalizarea bazei de date:

Normalizarea este procesul de eliminare a redundanței și de asigurare a consistenței datelor. S-au aplicat următoarele forme normale:

- **1NF:** Tabelele au coloane cu valori atomice și identificatori unici (chei primare). Exemplu: **Clienti** are un **client_id** unic și câmpuri indivizibile precum **nume** și **telefon**. De asemenea, nu există grupuri de valori sau coloane multiple care să definească mai multe valori într-o singură înregistrare.

- **2NF:** Atributele sunt complet dependente de cheile primare compuse. În **Colaboratori_Eveniment**, toate datele sunt dependente de perechea (**eveniment_id**, **colaborator_id**). Relațiile many-to-many sunt gestionate prin tabele intermediare, eliminând astfel dependențele parțiale.

- **3NF:** Atributele non-cheie depind doar de cheile primare. În **Evenimente**, câmpurile precum **data_eveniment** sau **tip_eveniment** sunt dependente doar de **eveniment_id**. Această abordare garantează că datele non-cheie nu depind de alte atribute non-cheie.

- **4NF:** Eliminarea dependențelor multivalente. Relațiile complexe precum **Evenimente-Locații** și **Evenimente-Colaboratori** sunt tratate prin tabele intermediare (**Locatii_Eveniment** și **Colaboratori_Eveniment**), asigurând separarea completă a dependențelor multivalente. Astfel, fiecare locație sau colaborator poate fi gestionat independent, fără ambiguitate.

- **5NF**(Forma Normală Proiectată)**:** Eliminarea redundanței prin dezagregarea relațiilor complexe. În schema bazei de date, combinațiile complexe dintre **Colaboratori**, **Locații** și **Evenimente** sunt gestionate fără redundanță prin intermediul tabelelor intermediare, garantând o optimizare completă și evitarea redundanțelor cauzate de relații complexe.

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 **Colaboratori_Eveniment**.

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.

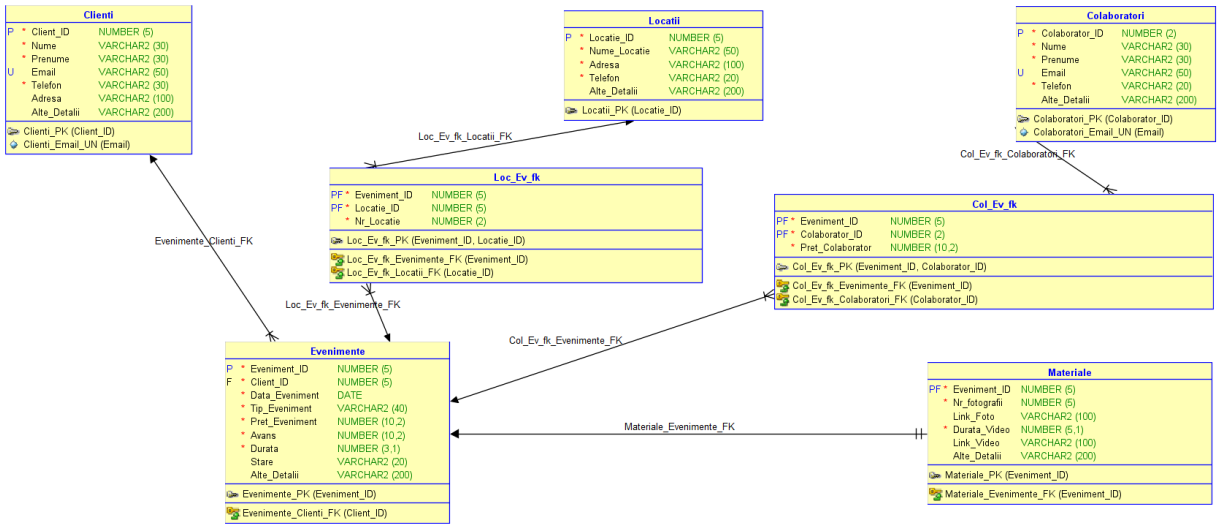
3. Simplificarea actualizărilor

- Orice modificare se face într-un singur loc, fără a verifica mai multe tabele.
- **Exemplu:** Dacă un colaborator își schimbă email-ul, acesta se actualizează doar în tabela **Colaboratori**, iar toate evenimentele rămân corecte.

4. Prevenirea problemelor la ștergere

- Ștergerea unui rând nu va elimina din greșeală informații importante
- **Exemplu:** Ștergerea unei intrări din tabela **Colaboratori_Eveniment** nu va afecta datele din tabelele **Colaboratori** sau **Evenimente** deoarece acestea sunt legate doar prin chei externe.

Modelul relațional:



Descrierea tabelelor:

1. **Clienti**
   - **client_id**: Cheie primară de tip numeric, autoincrement, utilizată pentru identificarea unică a fiecărui client.
   - **nume**: Șir de caractere (VARCHAR2), lungime maximă 30, obligatoriu, pentru a stoca numele clientului.

- **prenume**: Șir de caractere (VARCHAR2), lungime maximă 30, obligatoriu, pentru prenumele clientului.
- **email**: Șir de caractere (VARCHAR2), lungime maximă 50, opțional, cu validare REGEXP pentru a permite doar adrese de email valide.
- **telefon**: Șir de caractere (VARCHAR2), lungime maximă 15, obligatoriu, utilizat pentru a stoca numărul de telefon.
- **adresa**: Șir de caractere (VARCHAR2), lungime maximă 100, opțională, pentru a înregistra adresa clientului.
- **alte_detalii**: Șir de caractere (VARCHAR2), lungime maximă 200, opțională, pentru informații adiționale despre client.

2. **Colaboratori**
   - Structura similară cu tabelul **Clienti**, dar utilizată pentru a înregistra informațiile colaboratorilor implicați în evenimente.

3. **Evenimente**
   - **eveniment_id**: Cheie primară de tip numeric, autoincrement, utilizată pentru identificarea unică a fiecărui eveniment.
   - **client_id**: Cheie externă către tabelul **Clienti**, asigurând legătura dintre eveniment și client.
   - **data_eveniment**: Tip DATETIME, obligatoriu, pentru a înregistra data și ora evenimentului.
   - **tip_eveniment**: Șir de caractere (VARCHAR2), lungime maximă 40, obligatoriu, care indică tipul evenimentului (ex.: nuntă, botez).
   - **pret_eveniment**: Valoare numerică (NUMBER), obligatoriu, cu CHECK (>=0) pentru a evita valori negative.
   - **avans**: Valoare numerică (NUMBER), obligatoriu, cu CHECK (>=0), reprezentând suma plătită în avans.
   - **durata**: Valoare numerică (NUMBER), obligatoriu, cu CHECK (>0), pentru durata estimată a evenimentului.
   - **stare**: Șir de caractere (VARCHAR2), lungime maximă 20, opțional, pentru a indica starea evenimentului (ex.: confirmat, în așteptare).
   - **alte_detalii**: Șir de caractere (VARCHAR2), lungime maximă 200, opțional, pentru informații adiționale despre eveniment.

4. **Locatii**
   - **locatie_id**: Cheie primară de tip numeric, autoincrement.
   - **nume_locatie**: Șir de caractere (VARCHAR2), lungime maximă 50, obligatoriu, care stochează denumirea locației.
   - **adresa**: Șir de caractere (VARCHAR2), lungime maximă 100, obligatoriu.
   - **telefon**: Șir de caractere (VARCHAR2), lungime maximă 15, obligatoriu.
   - **alte_detalii**: Șir de caractere (VARCHAR2), lungime maximă 200, opțională.

5. **Colaboratori_Eveniment**
   - **eveniment_id**: Cheie externă către tabelul **Evenimente**.
   - **colaborator_id**: Cheie externă către tabelul **Colaboratori**.

- **pret_colaborator**: Valoare numerică (NUMBER), obligatoriu, cu CHECK (>=0), care indică costul asociat colaboratorului pentru evenimentul respectiv.
- Cheie primară compusă: (eveniment_id, colaborator_id).

6. **Locatii_Eveniment**
   - Structură similară cu tabelul **Colaboratori_Eveniment**, dar utilizată pentru a asocia evenimentele cu locațiile în care se desfășoară.

7. **Materiale**
   - **eveniment_id**: Cheie externă către tabelul **Evenimente**.
   - **nr_fotografii**: Valoare numerică (NUMBER), obligatoriu, cu CHECK (>=0), indicând numărul de fotografii realizate.
   - **link_foto**: Șir de caractere (VARCHAR2), lungime maximă 100, opțional, pentru a stoca un link către fotografiile evenimentului.
   - **durata_video**: Valoare numerică (NUMBER), obligatoriu, cu CHECK (>0), care indică durata totală a materialului video realizat.
   - **link_video**: Șir de caractere (VARCHAR2), lungime maximă 100, opțional, pentru a stoca un link către videoclipuri.
   - **alte_detalii**: Șir de caractere (VARCHAR2), lungime maximă 200, opțional.
   - Cheie primară: eveniment_id.

Descrierea constrângerilor utilizate:

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.

1. **Chei Primare (Primary Key):**
   - Au fost utilizate pentru identificarea unică a fiecărei înregistrări din tabele. De exemplu, *client_id* din tabelul **Clienti** sau *eveniment_id* din tabelul **Evenimente** permit identificarea precisă a unui client sau eveniment. Aceste chei previn existența unor rânduri duplicate.

2. **Chei Externe (Foreign Key):**
   - Asigură legături logice între tabele, menținând integritatea referențială. De exemplu, câmpul *client_id* din tabelul **Evenimente** face referire la *client_id* din tabelul **Clienti**, garantând că un eveniment este asociat unui client valid.

3. **UNIQUE:**
   - Această constrângere este folosită pentru câmpuri unde unicitatea este esențială, cum ar fi *email* în tabelele **Clienti** și **Colaboratori**. Aceasta previne introducerea de adrese de email duplicate, asigurând astfel coerența datelor de contact.

4. **CHECK:**
    - Validează datele introduse conform unor reguli specifice. Exemple includ verificarea ca *pret_eveniment* să fie ≥0 sau ca adresa de email să respecte un format valid (prin REGEXP). Aceste verificări reduc erorile umane și mențin standardele de calitate a datelor.

5. **NOT NULL:**
    - Aplicație obligatorie pentru câmpurile esențiale, cum ar fi *nume* și *prenume* din tabelul **Clienti**, care nu pot fi lăsate necompletate. Acest lucru asigură că informațiile critice sunt întotdeauna disponibile.

6. **Autoincrement:**
    - Generarea automată a valorilor pentru cheile primare, cum ar fi *client_id*, elimină necesitatea atribuirii manuale a identificatorilor unici, reducând riscul erorilor și garantând un flux de date consistent.

Impactul utilizării acestor constrângeri este semnificativ:

- Previn erorile de introducere a datelor, cum ar fi duplicatele sau valorile invalide.
- Mențin relațiile logice între entități, asigurând coerența bazei de date.
- Simplifică procesul de administrare a datelor, deoarece constrângerile gestionează regulile de validare automat.

Concluzii

Proiectul pune la dispoziție un sistem simplu de gestionare a evenimentelor foto-video. Structura bazei de date permite administrarea datelor într-un mod sigur și eficient, fiind ușor de extins pentru a adăuga funcționalități suplimentare pe viitor.