



Universitatea Tehnică "Gheorghe Asachi" Iași
Facultatea de Automatică și Calculatoare
Specializarea Calculatoare și Tehnologia Informației

Program de gestiune a unui sistem de rezervare a biletelor la spectacole

-temă de casă-



**Cadru didactic coordonator,
Cristian Buțincu**

**Student,
Savin Damian
Grupa 1307A**

➤ Descrierea proiectului

În cadrul acestui proiect am urmărit crearea unei aplicații care să ofere utilizatorilor posibilitatea de a rezerva locuri la diferite spectacole culturale și evenimente artistice. Această aplicație web va oferi o interfață ușor de utilizat, facilitând atât clienților cât și administratorilor accesul la informațiile despre spectacolele existente.

Aplicația permite:

- Utilizatorii pot căuta spectacolele disponibile, vizualiza detalii despre acestea și să-și rezerve bilete online.
- Administratorul poate manipula datele (adăugarea, ștergerea, modificarea tipurilor de spectacole).
- Vizualizarea coșului de cumpărături și editarea lui, dacă este cazul.

➤ Tehnologiile folosite

În scopul realizării acestei aplicații, pentru partea de front-end s-a folosit **HTML**, **CSS** și **JAVASCRIPT**, iar pentru partea de back-end s-a folosit **PHP**.

➤ Structura și inter-relaționarea tabelelor

În urma analizei și implementării proiectului, s-au realizat relații de tip *one-to-one*, *one-to-many*, *many-to-many*.

Între tabelele **Comenzi** și **Plata** există o relație de tip *one-to-one* întrucât nu pot exista două comenzi cu același set de detalii (în special, cu același id).

Ținând cont de calitatea serviciilor, utilizatorii pot plasa mai multe comenzi, dar fiecare comandă trebuie să aparțină în mod unic unui client, motiv pentru care, tabelele **Client** și **Comenzi** sunt legate printr-o relație *one-to-many*.

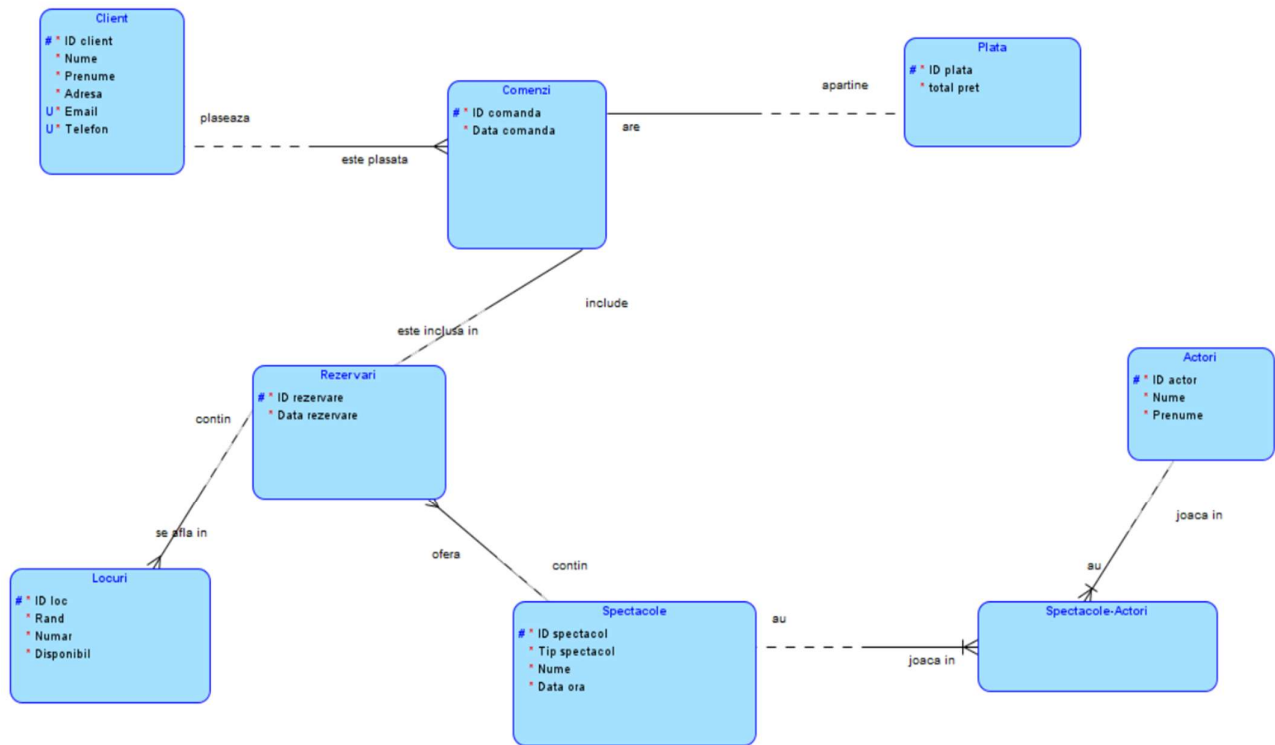
Având în vedere modul de operare a interfeței, efectuarea unei comenzi poate implica o singură rezervare la un anumit spectacol, de unde rezultă relația de tip *one-to-one* între tabelele **Rezervari** și **Comenzi**.

Întrucât la un anumit spectacol pot exista mai multe locuri rezervate de aceeași persoană, rezultă că între tabelele **Locuri** și **Rezervari** există o relație de tip *one-to-many*.

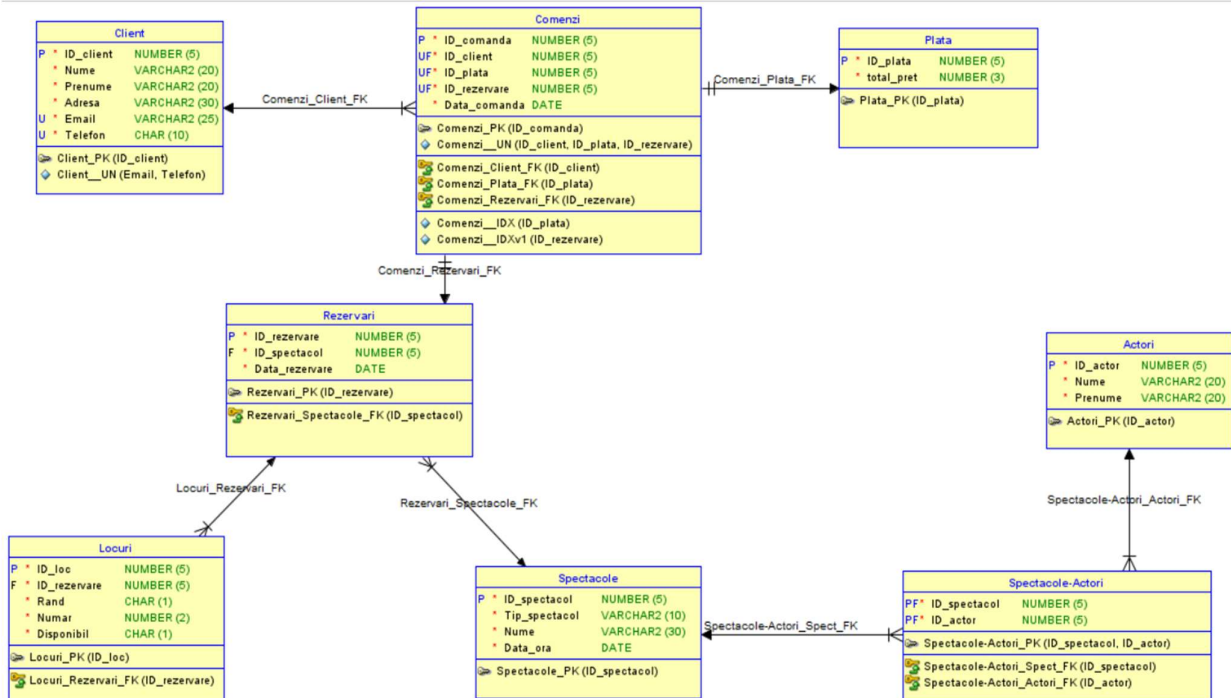
Având în vedere că la același spectacol pot exista mai multe rezervări, tabelele **Spectacole** și **Rezervari** vor avea o relație de tip *one-to-many*.

În mod evident, un spectacol poate conține mai mulți actori, și totodată, un actor poate juca în mai multe spectacole. Acest lucru este evidențiat cel mai bine printr-o legătură de tip *many-to-many* între tabelele **Spectacole** și **Actori**.

➤ Model logic



➤ Model Relațional



➤ Descrierea constrângerilor

La nivelul fiecărei tabelă se pot regăsi constrângeri de patru tipuri: primary key (PK), unique (UN), not null (NN) și check(CK), dintre care ultimele două sunt cele mai utilizate.

Constrângerile de tip primary key sunt generate de baza de date printr-un mecanism de autoincrement, oferind o modalitate automată de a atribui valori unice fiecărei înregistrări. Tabelele *Client*, *Comenzi*, *Plata*, *Rezervari*, *Spectacole*, *Locuri*, *Actori* au ca și **primary key** câmpul *ID*. Necesitatea acestor chei primare se evidențiază mai ales la formarea legăturilor între tabele, acolo unde se creează **foreign keys** care referă acestei chei primare. Mai mult, cheile primare identifică în mod unic câmpurile din tabelă, ceea ce facilitează stocarea datelor în baza de date.

Constrângerile de tip unique sunt necesare pentru a nu avea clienți cu aceeași adresă de email sau cu același număr de telefon și pentru a asigura unicitatea și integritatea datelor caracteristice fiecărei comenzi.

Constrângerile de tip not null arată faptul că anumite coloane dintr-o tabelă nu pot rămâne necomplete, implicit anumite date trebuie știute (în lipsa lor vor fi generate erori la introducerea datelor). Atributele cu aceste constrângeri sunt adresa, email, telefon din tabelă **Client**, total_pret din tabelă **Plata**, rand și numar din tabelă **Locuri** și de asemenea, atributele care constituie datele calendaristice și atributele care conțin numele și prenumele.

Constrângerile de tip check sunt necesare pentru a impune condiții specifice asupra datelor care sunt introduse într-un tabel. Aceste constrângeri asigură că datele respectă anumite reguli sau condiții definite de utilizator. Datorită acestor constrângeri, în cazul aplicației noastre putem verifica dacă adresa de email respectă un anumit format, dacă numărul de telefon este alcătuit din zece cifre, dacă prețul total este un număr pozitiv, dacă tipul unui spectacol este Opera sau Balet, dacă numărul de locuri pe rând este cuprins între 1 și 18 sau dacă numărul de rânduri este cuprins între caracterele 'A' și 'N'.

➤ Descrierea modalității de conectare la baza de date

- Conectarea la baza de date

```
1  <?php
2
3  $host = 'localhost';
4  $user = 'root';
5  $password = '';
6  $db = 'test';
7
8  //Conexiune la server (localhost)
9  $conn = mysqli_connect(
10     $host,
11     $user,
12     $password);
13
14  //Selectare baza de date
15  $database = mysqli_select_db(
16     $conn,
17     $db
18  );
19
20
21  if($conn->connect_error){
22     die("Conexiune esuata: ".$conn->connect_error);
23  }
24  echo "Conexiune la baza de date reușită!";
25  ?>
```

- Selectare elemente din tabela Spectacole

```
function getSpectacole() {  
    global $conn;  
    $sql = "SELECT * FROM Spectacole";  
    $result = $conn->query($sql);  
    $spectacole = [];  
    while ($row = $result->fetch_assoc()) {  
        $spectacole[] = $row;  
    }  
    return $spectacole;  
}
```

- Selectare elemente din tabela Client

```
function getClienti() {  
    global $conn;  
    $sql = "SELECT * FROM client";  
    $result = $conn->query($sql);  
    $clienti = [];  
    while ($row = $result->fetch_assoc()) {  
        $clienti[] = $row;  
    }  
    return $clienti;  
}
```

- Adăugare și afișare coș de cumpărături

```
function loadCart()
{
    cart = JSON.parse(localStorage.getItem("shoppingCart"));
}
```

```
function displayCart()
{
    var cartArray = listCart();
    var output = "";
    for(var i in cartArray)
    {
        output += "<p>"
        +cartArray[i].name
        +" "+cartArray[i].count
        +" x "+ cartArray[i].price
        +" = "+ cartArray[i].total
        + "</p>";
    }
}
```

- Capturi de ecran din interfața aplicației



SPECTACOLE

OPERA

BALET

LOCURI

SCENA



Coș

Produs	Pret
SPĂRGĂTORUL DE NUCI	130.00

Cantitate	Total
x 1	130.00

Nume

Prenume

Email

Locuri Rand-Numar

Golire Cos

Comanda