

Facultatea de Automatică și Calculatoare

Specializare: Ingineria Sistemelor

Aplicație de feedback și clasificare a filmelor

secțiune teoretică



Student:

Nozohor Aida

Coordonator:

Conferențiar dr.ing. Olteanu Adriana

2023 – 2024

Cuprins

Ce este Cinemania?	3
Modul de funcționare și facilitățile aplicației	3
Relațiile între tabele.....	5
Schema relațională.....	7
Constrângeri de integritate	8
Etapă de proiectare	8
Interogările simple.....	9
Interogările complexe	11

Cinemaniania

“Aplicație de feedback și clasificare a filmelor.”

Ce este Cinemaniania?

Cinemaniania se distinge ca o aplicație sofisticată și esențială pentru pasionații de cinematografie, oferind o platformă robustă pentru evaluarea, recenzarea și partajarea experiențelor vizionării acestora. Această aplicație conectează pasionații de film la nivel global, facilitând interacțiunea în jurul lumii cinematografilei.

Cu toate acestea, succesul rapid a generat un volum semnificativ de utilizatori și recenzii, ceea ce a pus presiune asupra bazei de date existente a aplicației. Astfel, fondatorii Cinemaniania au ajuns la concluzia că este absolut necesară o bază de date mai performantă și scalabilă.

De curând, m-am alăturat echipei de dezvoltatori ai acestui proiect, iar scopul nostru curent este de a îmbunătăți experiența utilizatorilor, crescând performanța aplicației în timp ce minimizăm costurile și provocările tehnice ce vor apărea în viitor, toate acestea printr-o bază de date îmbunătățită.

Modul de funcționare și facilitățile aplicației

1. Fiecare utilizator al aplicației are un profil. Un astfel de profil conține numele și prenumele sau, o poză de profil și una de fundal, cât și data la care și-a creat contul.
2. Un utilizator poate furniza evaluări(feedback) unui sau mai multor filme. Fiecare astfel de evaluare va fi afișată sub forma momentului postării (data și ora) alături de conținutul acesteia.
3. Accesând pagina de prezentare a unui film vom fi întâmpinați de informații descriptive precum titlul acestuia, anul apariției, rating IMBD, durata, buget, țara apariției și limba vorbită.
4. În cadrul aplicației, se regăsesc recenziile utilizatorilor. Pot exista una sau mai multe recenzii de la unul sau mai mulți utilizatori.
5. Fiecare film este încadrat în una sau mai multe categorii, acestea distingându-se prin denumire, popularitate și o iconiță sugestivă. O categorie poate include mai multe filme.
6. Distribuția unui film poate fi formată dintr-unul sau mai mulți actori, fiecare fiind caracterizat de un nume, un prenume, data și locul nașterii, statutul carierei(activ/pensionat) cât și o poză de profil. Un actor poate juca în mai multe filme.
7. Fiecare film este produs de un singur studio definit de un nume, anul fondării, sediul central și o poză sugestivă. Un studio poate produce mai multe filme.
8. Orice film este dirijat de un regizor, caracterizat de un nume, un prenume, data și locul nașterii, statutul carierei, cât și o poză de profil. Un regizor poate regiza mai multe filme.

Identificam 6 entitati: **UTILIZATORI, FILME, CATEGORII, ACTORI, STUDIOURI SI REGIZORI.**

Pentru fiecare dintre aceste 6 entitati vom crea un tabel.

Utilizatori
ID_Utilizator (CP)
Nume
Prenume
Data_Deschidere_Cont
Poza_Profil
Poza_Fundal

Filme
ID_Film (CP)
Titlu
Anul_Lansarii
Rating_IMBD
Durata
Buget
Tara_Productie
Limba

Categorii
ID_Categorie (CP)
Denumire
Popularitate
Iconita

Actori
ID_Actor (CP)
Nume
Prenume
Data_Nasterii
Locul_Nasterii
Statutul_Carierei
Imagine

Studiouri
ID_Studio (CP)
Nume
Anul_Fondarii
Sediul_Central
Imagine

Regizori
ID_Regizor (CP)
Nume
Prenume
Data_Nasterii
Locul_Nasterii
Statutul_Carierei
Imagine

Relațiile între tabele

Nu avem relatii cu autoreferire.

Utilizatori:

Utilizatori – Filme: Un utilizator poate recenzia unul sau mai multe filme. **1:N**

Utilizatori – Categori: -

Utilizatori – Actori: -

Utilizatori – Studiouri: -

Utilizatori – Regizori: -

Filme:

Filme – Utilizatori: Un film poate fi recenzat de unul sau mai multi utilizatori. **1:N**

Filme – Categori: Un film poate apartine uneia sau mai multor categorii. **1:N**

Filme – Actori: Un film poate avea unul sau mai multi actori. **1:N**

Filme – Studiouri: Un film poate apartine unui singur studio. **1:1**

Filme – Regizori: Un film poate avea un singur regizor. **1:1**

Categori:

Categori – Utilizatori: -

Categori – Filme: O categorie poate ingloba unul sau mai multe filme. **1:N**

Categori – Actori: -

Categori – Studiouri: -

Categori – Regizori: -

Actori:

Actori – Utilizatori: -

Actori – Filme: Un actor poate juca intr-unul sau mai multe filme. **1:N**

Actori – Categori: -

Actori – Studiouri: -

Actori – Regizori: -

Studiouri:

Studiouri – Utilizatori: -

Studiouri – Filme: Un studio poate produce unul sau mai multe filme. **1:N**

Studiouri – Categori: -

Studiouri – Actori: -

Studiouri – Regizori: -

Regizori:

Regizori – Utilizatori: -

Regizori – Filme: Un regizor poate dirija unul sau mai multe filme. **1:N**

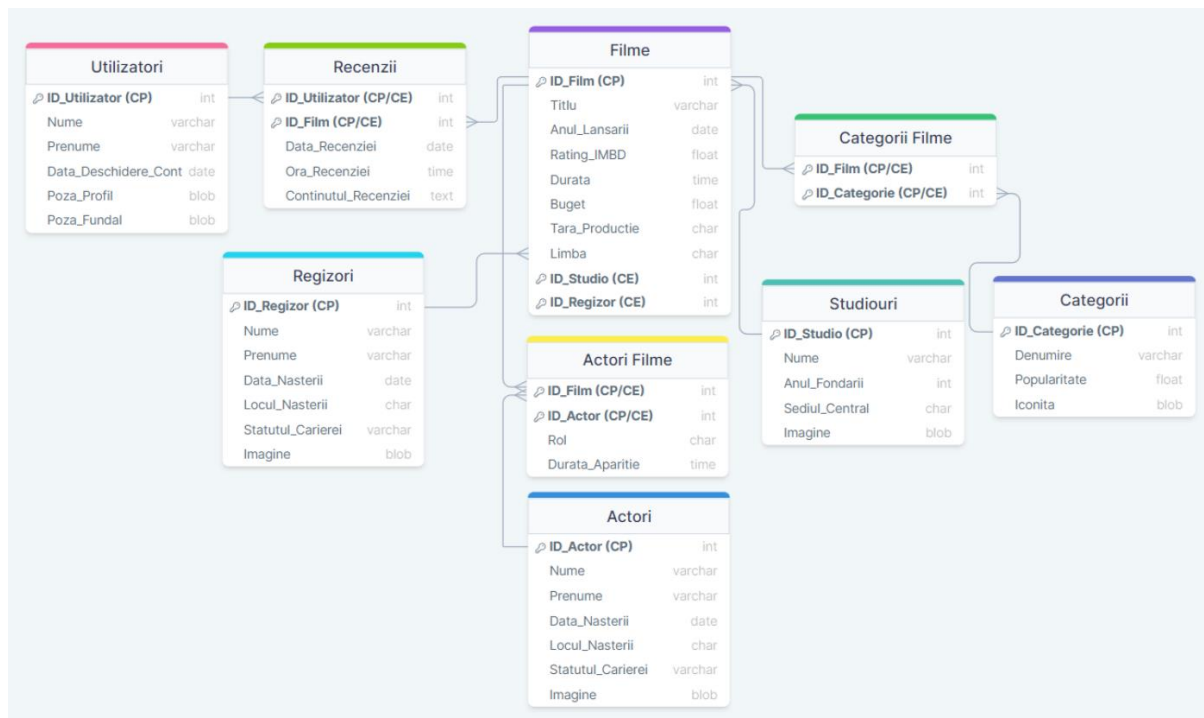
Regizori – Categori: -

Regizori – Actori: -

Regizori – Studiouri: -

	Utilizatori	Filme	Categori	Actori	Studiouri	Regizori
Utilizatori	-	1:N	-	-	-	-
Filme	1:N	-	1:N	1:N	1:1	1:1
Categori	-	1:N	-	-	-	-
Actori	-	1:N	-	-	-	-
Studiouri	-	1:N	-	-	-	-
Regizori	-	1:N	-	-	-	-

Schema relațională



Constrângeri de integritate

1. Constrângeri de cheie primară
 - Tabela Utilizatori are o cheie primara: ID_Utilizator
 - Tabela Regizori are o cheie primara: ID_Regizor
 - Tabela Filme are cheia primara: ID_Film
 - Tabela Actori are cheia primara: ID_Actor
 - Tabela Studiouri are cheia primara: ID_Studio
 - Tabela Categorii are cheia primara: ID_Categorie
2. Constrângeri de cheie externă
 - Tabela Recenzii are cheile externe: ID_Utilizator si ID_Film
 - Tabela Filme are cheile externe: ID_Studio si ID_Regizor
 - Tabela Actori Filme are cheile externe: ID_Film si ID_Actor
 - Tabela Categorii Filme are cheile externe: ID_Film si ID_Categorii

Etapă de proiectare

În vederea implementării acestui proiect, am optat pentru utilizarea mediului de dezvoltare Visual Studio și limbajul de programare C#. Am dezvoltat o aplicație Windows Forms conform cerințelor impuse de proiect.

Am integrat funcționalități diverse, inclusiv afișarea tabelelor din baza de date și implementarea operațiilor de Insert, Update și Delete. De asemenea, am adăugat suport pentru interogări simple și complexe, toate aceste aspecte fiind fundamentate pe principii ale programării orientate pe obiect și cunoștințe în limbajul SQL.

Interogările simple

1. Această interogare utilizează un parametru variabil preluat dintr-un element de interfață grafică de tip casetă de text, având ca scop afișarea tuturor titlurilor de filme aflate în categoria specificată de către utilizator prin intermediul casetei.

```
private void button1_Click(object sender, EventArgs e)
{
    using (SqlConnection conn = new SqlConnection(@"Data Source=LAPTOP-AIDA\SQLEXPRESS;Initial Catalog=Cinemania;Integrated Security=True"))
    {
        conn.Open();

        string Categoria;
        Categoria = textBox5.Text;

        // Creează o comandă SQL pentru inserarea unui nou rând
        string insertQuery = "SELECT Filme.Titlu FROM Filme INNER JOIN \\'Categorii Filme\' ON Filme.ID_Film = \\'Categorii Filme\' ID_Film INNER JOIN Categorii ON \\'Categorii Filme\' ID_Categorie = Categorii.ID_Categorie WHERE Categorii.Denumire = @Categoria";
        using (SqlCommand insertCmd = new SqlCommand(insertQuery, conn))
        {
            // detineți parametrii pentru valorile de inserat
            insertCmd.Parameters.AddWithValue(@"Categoria", Categoria); // Poți înlocui cu datele reale

            // Execută comanda de inserare
            insertCmd.ExecuteNonQuery();

            using (SqlDataAdapter sda = new SqlDataAdapter(insertCmd))
            {
                using (DataTable dt = new DataTable())
                {
                    sda.Fill(dt);
                    dataGridView2.DataSource = dt;
                }
            }
        }
    }
}
```

2. Această interogare prezintă titlul și categoria de care aparțin "cele mai noi filme", sortând filmele în ordine descrescătoare în funcție de anul apariției și afișând primele trei rezultate.

```
private void button6_Click(object sender, EventArgs e)
{
    using (SqlConnection conn = new SqlConnection(@"Data Source=LAPTOP-AIDA\SQLEXPRESS;Initial Catalog=Cinemania;Integrated Security=True"))
    {
        conn.Open();

        string selectQuery = @"SELECT TOP 3 f.Titlu, c.Denumire
FROM Categorii c

INNER JOIN \\'Categorii Filme\' cp
ON c.ID_Categorie = cp.ID_Categorie

INNER JOIN Filme f
ON cp.ID_Film = f.ID_Film

ORDER BY f.Anul_Lansarii DESC ";

        using (SqlCommand cmd = new SqlCommand(selectQuery, conn))
        {
            using (SqlDataAdapter sda = new SqlDataAdapter(cmd))
            {
                DataTable dt = new DataTable();
                sda.Fill(dt);
                dataGridView2.DataSource = dt;
            }
        }
    }
}
```

3. Această interogare expune toate recenziile asociate filmelor.

```
private void button5_Click(object sender, EventArgs e)
{
    using (SqlConnection conn = new SqlConnection(@"Data Source=LAPTOP-AIDA\SQLEXPRESS;Initial Catalog=Cinemania;Integrated Security=True"))
    {
        conn.Open();

        string selectQuery = @"SELECT f.Titlu, r.Continutul_Recenziei
FROM Recenzii r

INNER JOIN Filme f
ON r.ID_Film = f.ID_Film";

        using (SqlCommand cmd = new SqlCommand(selectQuery, conn))
        {
            using (SqlDataAdapter sda = new SqlDataAdapter(cmd))
            {
                DataTable dt = new DataTable();
                sda.Fill(dt);
                dataGridView2.DataSource = dt;
            }
        }
    }
}
```

4. Această interogare prezintă numele și prenumele tuturor regizorilor, împreună cu numărul de filme pe care aceștia le-au regizat.

```
private void button8_Click(object sender, EventArgs e)
{
    using (SqlConnection conn = new SqlConnection(@"Data Source=LAPTOP-AIDA\SQLEXPRESS;Initial Catalog=Cinemania;Integrated Security=True"))
    {
        conn.Open();

        string selectQuery = @"SELECT
            Regizori.Nume,
            Regizori.Prenume,
            COUNT(Filme.ID_Film) AS NumarFilme
        FROM
            Regizori
        LEFT JOIN
            Filme ON Regizori.ID_Regizor = Filme.ID_Regizor
        GROUP BY
            Regizori.Nume, Regizori.Prenume";

        using (SqlCommand cmd = new SqlCommand(selectQuery, conn))
        {
            using (SqlDataAdapter sda = new SqlDataAdapter(cmd))
            {
                DataTable dt = new DataTable();
                sda.Fill(dt);
                dataGridView2.DataSource = dt;
            }
        }
    }
}
```

5. Această interogare prezintă cel mai popular film, evaluând popularitatea pe baza categoriilor.

```
private void button7_Click(object sender, EventArgs e)
{
    using (SqlConnection conn = new SqlConnection(@"Data Source=LAPTOP-AIDA\SQLEXPRESS;Initial Catalog=Cinemania;Integrated Security=True"))
    {
        conn.Open();

        string selectQuery = @"SELECT TOP 1 f.Titlu
        FROM Categori c

        INNER JOIN ""Categori Filme"" cp
        ON c.ID_Categorie = cp.ID_Categorie

        INNER JOIN Filme f
        ON cp.ID_Film = f.ID_Film

        ORDER BY c.Popularitate DESC ";

        using (SqlCommand cmd = new SqlCommand(selectQuery, conn))
        {
            using (SqlDataAdapter sda = new SqlDataAdapter(cmd))
            {
                DataTable dt = new DataTable();
                sda.Fill(dt);
                dataGridView2.DataSource = dt;
            }
        }
    }
}
```

6. Această interogare prezintă informații despre toți actorii.

```
private void button9_Click(object sender, EventArgs e)
{
    using (SqlConnection conn = new SqlConnection(@"Data Source=LAPTOP-AIDA\SQLEXPRESS;Initial Catalog=Cinemania;Integrated Security=True"))
    {
        conn.Open();

        string selectQuery = @"SELECT a.Nume, a.Prenume, af.Rol, f.Titlu
        FROM Actori a

        INNER JOIN ""Actori Filme"" af
        ON a.ID_Actor = af.ID_Actor

        INNER JOIN Filme f
        ON af.ID_Film = f.ID_Film";

        using (SqlCommand cmd = new SqlCommand(selectQuery, conn))
        {
            using (SqlDataAdapter sda = new SqlDataAdapter(cmd))
            {
                DataTable dt = new DataTable();
                sda.Fill(dt);
                dataGridView2.DataSource = dt;
            }
        }
    }
}
```

Interogările complexe

1. Această interogare utilizează un câmp variabil pentru a prelua un an din textbox și a-l utiliza pentru a afișa toate filmele apărute după acel an.

```
private void button10_Click(object sender, EventArgs e)
{
    using (SqlConnection conn = new SqlConnection(@"Data Source=LAPTOP-AIDA\SQLEXPRESS;Initial Catalog=Cinemania;Integrated Security=True"))
    {
        conn.Open();

        string Anul_Lansarii;
        Anul_Lansarii = textBox4.Text;

        // Crearea comenzii SQL pentru interogarea unui an
        string selectQuery = "SELECT f.Titlu, f.Anul_Lansarii, f.Denumire FROM Filme f JOIN " + Anul_Lansarii + " ON f.ID_Film = " + Anul_Lansarii + " WHERE f.ID_Film > " + Anul_Lansarii + " ORDER BY f.ID_Film";
        using (SqlCommand cmd = new SqlCommand(selectQuery, conn))
        {
            // Setarea parametrilor pentru valoarea de inserat
            cmd.Parameters.AddWithValue("@Anul_Lansarii", Anul_Lansarii); // Poziția indicată cu dubla linie

            // Executarea comenzii de interogare
            SqlDataReader reader = cmd.ExecuteReader();

            using (SqlDataAdapter sda = new SqlDataAdapter(cmd))
            {
                DataTable dt = new DataTable();
                sda.Fill(dt);
                dataGridView1.DataSource = dt;
            }
        }
    }
}
```

2. Această interogare afișează titlul, categoria și rating-ul filmului care are valoarea recenziei cea mai apropiată de valoarea medie.

```
private void button11_Click(object sender, EventArgs e)
{
    using (SqlConnection conn = new SqlConnection(@"Data Source=LAPTOP-AIDA\SQLEXPRESS;Initial Catalog=Cinemania;Integrated Security=True"))
    {
        conn.Open();

        string selectQuery = @"SELECT TOP 1 f.Titlu, c.Denumire AS Categoria, f.Rating_IMBD AS ""Rating IMBD""
FROM Filme f
JOIN ""Categorii Filme"" cf
ON f.ID_Film = cf.ID_Film
JOIN Categorii c
ON cf.ID_Categorie = c.ID_Categorie
ORDER BY ABS(f.Rating_IMBD - (SELECT AVG(Rating_IMBD) FROM Filme)) ASC;";

        using (SqlCommand cmd = new SqlCommand(selectQuery, conn))
        {
            using (SqlDataAdapter sda = new SqlDataAdapter(cmd))
            {
                DataTable dt = new DataTable();
                sda.Fill(dt);
                dataGridView3.DataSource = dt;
            }
        }
    }
}
```

3. Această interogare selectează informații despre studiourile care au alocat cel mai mare buget în crearea filmelor.

```
private void button12_Click(object sender, EventArgs e)
{
    using (SqlConnection conn = new SqlConnection(@"Data Source=LAPTOP-AIDA\SQLEXPRESS;Initial Catalog=Cinemania;Integrated Security=True"))
    {
        conn.Open();

        string selectQuery = @"SELECT s.Nume, s.Anul_Fondarii AS ""Anul Fondarii"", s.Sediul_Central AS ""Sediul Central""
FROM Studiouri s
JOIN Filme f
ON s.ID_Studio = f.ID_Studio
WHERE ID_Film
IN (SELECT ID_Film FROM Filme WHERE Buget > 150000000);";

        using (SqlCommand cmd = new SqlCommand(selectQuery, conn))
        {
            using (SqlDataAdapter sda = new SqlDataAdapter(cmd))
            {
                DataTable dt = new DataTable();
                sda.Fill(dt);
                dataGridView3.DataSource = dt;
            }
        }
    }
}
```

4. Această interogare numără actorii cu vârsta sub 24 de ani.

```
private void button13_Click(object sender, EventArgs e)
{
    using (SqlConnection conn = new SqlConnection(@"Data Source=LAPTOP-AIDA\SQLEXPRESS;Initial Catalog=Cinemania;Integrated Security=True"))
    {
        conn.Open();

        string selectQuery = @"SELECT COUNT(DISTINCT a.ID_Actor) AS Numar_Actori
        FROM Actori a
        LEFT JOIN ""Actori Filme"" af ON a.ID_Actor = af.ID_Actor
        LEFT JOIN Filme f ON af.ID_Film = f.ID_Film
        JOIN (SELECT ID_Actor FROM Actori WHERE YEAR(Data_Nasterii) > 2000) subq
        ON a.ID_Actor = subq.ID_Actor;

        using (SqlCommand cmd = new SqlCommand(selectQuery, conn))
        {
            using (SqlDataAdapter sda = new SqlDataAdapter(cmd))
            {
                DataTable dt = new DataTable();
                sda.Fill(dt);
                dataGridView3.DataSource = dt;
            }
        }
    }
}
```