



Aplicație de feedback și clasificare a filmelor

● Proiect ●

Păunoiu Ionuț-Eduard
Grupa 333AA



Definirea cerințelor pentru aplicația aleasă

Aplicația ce urmează a fi dezvoltată își propune să se axeze pe cele două mari componente ale temei: sistemul de clasificare a filmelor și sistemul de feedback. Din punct de vedere al clasificării, aplicația va putea să clasifice o serie de filme după numeroase criterii precum gen, an, actori, categorie de audiență, durată, regizor și altele. Pe partea de client, aplicația va permite utilizatorului să ofere ca feedback un review pentru un anumit film sau pentru mai multe, și un număr de stele ca rating. Mai multe detalii vor fi menționate la finalul proiectării bazei de date.

Aplicația va fi reprezentată de un site proiectat folosind tehnologii web precum HTML, CSS, PHP și JavaScript. Baza de date va fi construită cu ajutorul MySQL și phpMyAdmin. O altă idee de implementare ar fi folosind limbajul Python și framework-ul Django. Prin intermediul site-ului voi asigura interfața vizuală a aplicației, iar prin intermediul bazei de date voi asigura funcționalitățile celor două componente.

Proiectarea bazei de date

Identificarea tabelelor

Pentru a realiza baza de date, voi folosi următoarele obiecte de tip tabelă:

- Film
- Actor
- Gen
- Utilizator
- Review

Stabilirea câmpurilor și a tipurilor de date

Denumire tabelă	Atribute	Tip de dată	Restricții
Film	id_film	INTEGER	PK; Auto Increment
	nume_film	VARCHAR(20)	Unique
	descriere_actiune	VARCHAR(255)	Allow Nulls
	categorie_audienta	CHAR(3)	
	durata	TIME	
	data_lansarii	DATETIME	
	tara_origine	VARCHAR(15)	
	limba_originala	VARCHAR(15)	
	trivia_film	VARCHAR(255)	Allow Nulls
	nume_director	VARCHAR(20)	
	prenume_director	VARCHAR(30)	
	nume_producator	VARCHAR(20)	
	prenume_producator	VARCHAR(30)	
	id_actor	INTEGER	PK; Auto Increment
Actor	nume_actor	VARCHAR(20)	
	prenume_actor	VARCHAR(30)	
	data_nasterii	DATETIME	
	bio_actor	VARCHAR(255)	Allow Nulls
	id_gen_film	INTEGER	PK; Auto Increment
	nume_gen	CHAR(10)	Unique



Gen	descriere_gen	VARCHAR(255)	Allow Nulls
	id_utilizator	INTEGER	PK; Auto Increment
	nume_utilizator	VARCHAR(20)	
Utilizator	prenume_utilizator	VARCHAR(30)	
	data_inregistrarii	DATETIME	
	bio_utilizator	VARCHAR(255)	Allow Nulls
	parola	VARCHAR(100)	
	id_review	INTEGER	PK; Auto Increment
	descriere_review	VARCHAR(255)	Unique
Review	numar_stele	INTEGER	
	id_film	INTEGER	FK
	id_utilizator	INTEGER	FK

Identificarea relațiilor dintre tabele

	Film	Actor	Gen	Utilizator	Review
Film	-	1 : N	1 : N	-	1 : N
Actor	1 : N	-	-	-	-
Gen	1 : N	-	-	-	-
Utilizator	-	-	-	-	1 : N
Review	1 : 1	-	-	1 : 1	-

Pe baza tabelului de mai sus și a formulelor referitoare la relațiile între tabele, am determinat relațiile oficiale între tabelele bazei mele de date:

1. **Film – Actor:** $(1 : N) + (1 : N) = N : N$
2. **Film – Gen:** $(1 : N) + (1 : N) = N : N$
3. **Film – Review:** $(1 : N) + (1 : 1) = 1 : N$
4. **Utilizator – Review:** $(1 : N) + (1 : 1) = 1 : N$

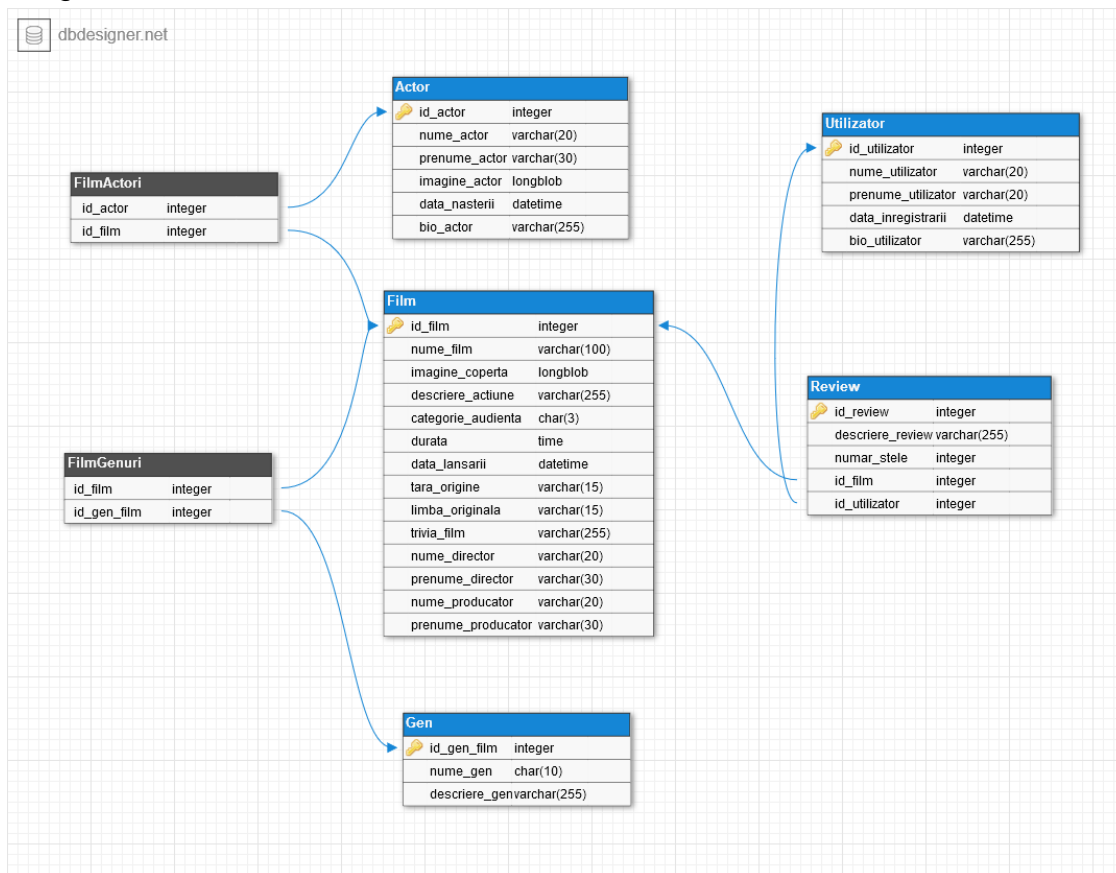
Având în vedere că am identificat 2 relații de tipul many – to – many, voi crea două tabele suplimentare care vor face legătura între:

1. Film și Actor: tabela FilmActori ce va avea două chei străine: id_film și id_actor care vor crea cheia primară compusă a tabelului.
2. Film și Gen: tabela FilmGenuri ce va avea două chei străine: id_film și id_gen care vor crea cheia primară compusă a tabelului.

Pentru realizarea relațiilor one – to – many voi introduce cheia externă id_film în tabela Review și cheia externă id_utilizator în tabela Review.



Pentru o reprezentare mai clară, am realizat următoarea schemă folosind site-ul dbdesigner.net:



Impunerea unor constrângeri de integritate

Constrângerile au fost menționate în tabelul cu câmpurile și tipurile de date, în coloana Restricții. Totuși, pentru a detalia:

1. Cheile primare (PK) ale fiecărui tabel se auto-incrementează.
2. Câmpurile imagine_copertă din Film, imagine_actor din Actor, nume_gen din Gen și descriere_review din Review sunt UNIQUE.
3. Câmpurile descriere_actiune și trivia_film din Film, bio_actor din Actor, descriere_gen din Gen, bio_utilizator din Utilizator pot fi nule (ALLOW NULL).
4. Cheile străine (FK) sunt menționate.

Definirea cerințelor

Baza de date “Feedback și Clasificare Filme” are următoarele caracteristici:

1. În baza de date pot exista n filme cu attributele menționate mai sus.
2. Pentru simplitate, am considerat că filmul are un singur regizor (director) și un singur producător, câmpuri adăugate în tabela Film. Inițial făcusem tabele separate, însă cu acordul doamnei profesoare am făcut această modificare.
3. În fiecare film joacă n actori cu attributele stabilite.
4. Fiecare film poate avea n genuri.
5. Un film poate avea n review-uri.
6. Un utilizator poate acorda n review-uri.



INSERT, UPDATE, DELETE la minimum 2 tabele

- Pentru tabela **utilizator**

- **INSERT** – registration_insert.php

```
INSERT INTO `utilizator` (id_utilizator, username, email,
parola, nume_utilizator, prenume_utilizator,
data_inregistrarii)
VALUES (DEFAULT, '$username', '$email', '$hashed_password',
'$nume', '$prenume', '$today')
```

- **UPDATE** – profile_update2.php

```
UPDATE `utilizator` SET `username`='$usr',
`email`='$email',
`nume_utilizator`='$nume',
`prenume_utilizator` = '$prenume',
`bio_utilizator`='$bio'
WHERE `id_utilizator` = '$id'
```

- **DELETE** – delete_profile.php

```
DELETE FROM `utilizator` WHERE `id_utilizator`='$id'
DELETE FROM `review` WHERE `id_utilizator` = '$id'
```

- Pentru tabela **review**:

- **INSERT** – add_review2.php

```
INSERT INTO `review` (`descriere_review`, `numar_stele`,
`id_film`, `id_utilizator`)
VALUES
('$descriere_review', '$nr_stele', '$id_film', '$id_user')
```

- **UPDATE** – update_review3.php

```
UPDATE `review`
SET `descriere_review`='$descr_review', `numar_stele` =
'$stele'
WHERE id_utilizator = '$id_utilizator' AND id_film = (SELECT
id_film FROM film WHERE nume_film = '$movie')
```

- **DELETE** – delete_review2.php

```
DELETE FROM `review` WHERE id_utilizator = '$id_utilizator'
AND id_film = (SELECT id_film FROM
film WHERE nume_film = '$movie')
```



Interogări simple

1. Vizualizarea detaliilor tuturor review-urilor date de utilizatorul care este conectat. ([see review.php](#))

```
SELECT F.num_film, R.descriere_review, R.numar_stele  
FROM film AS F  
JOIN review AS R ON F.id_film = R.id_film  
WHERE R.id_utilizator = '$id'
```

2. Vizualizarea tuturor filmelor care aparțin unui gen selectat de utilizator. ([filter_genres2.php](#))

```
SELECT DISTINCT num_film FROM film F  
JOIN filmgenuri FG ON F.id_film = FG.id_film  
JOIN gen G ON G.id_gen_film = FG.id_gen_film  
WHERE G.num_gen = '$genre'
```

3. Vizualizarea rating-ului mediu al unui film selectat de utilizator. ([average_stars2.php](#))

```
SELECT F.num_film, AVG(R.numar_stele) AS 'avg' FROM film F  
JOIN review R ON F.id_film = R.id_film  
WHERE F.num_film = '$film'
```

4. Genurile unui film selectat de utilizator. ([filter_genresII2.php](#))

```
SELECT num_gen FROM gen G  
JOIN filmgenuri FG ON G.id_gen_film = FG.id_gen_film  
JOIN film F ON FG.id_film = f.id_film  
WHERE F.num_film = '$film'
```

5. Actorii care joacă într-un film ales de utilizator. ([movie_actors2.php](#))

```
SELECT prenume_actor, num_actor FROM actor A  
JOIN filmactori FA ON A.id_actor = FA.id_actor  
JOIN film F ON FA.id_film = F.id_film  
WHERE F.num_film = '$film'
```

6. Toate review-urile unui film ales de utilizator. ([movie_reviews2.php](#))

```
SELECT R.descriere_review, R.numar_stele, U.username FROM review R  
JOIN film F on R.id_film = F.id_film  
JOIN utilizator U on R.id_utilizator = U.id_utilizator  
WHERE F.num_film = '$film'
```

7. Actorii care joacă în filme cu o anumită audiență selectată de utilizator. ([actors_audience2.php](#))

```
SELECT A.prenume_actor, A.num_actor, F.num_film FROM actor A  
JOIN filmactori FA ON A.id_actor = FA.id_actor  
JOIN film F ON F.id_film = FA.id_film  
WHERE F.categorie_audienta = '$audience'  
ORDER BY F.num_film
```



Interogări complexe

1. Numărul de review-uri și media pentru toate filmele din baza de date.
([review_no.php](#))

```
SELECT F.num_film AS 'MovieTitle', COUNT(R.id_film) AS 'NoRatings',  
      (SELECT AVG(R.numar_stele) FROM film F1  
       JOIN review R ON F1.id_film = R.id_film  
       WHERE F1.num_film = F.num_film ) AS 'AvgRating'  
FROM film F, review R  
WHERE R.id_film = F.id_film  
GROUP BY F.num_film";
```

2. Genurile filmelor în care joacă actori cu anul nașterii mai mare decât o valoare aleasă de utilizator. ([actors_genres2.php](#))

```
SELECT DISTINCT G.num_gen, F.num_film FROM gen G  
      JOIN filmgenuri FG ON G.id_gen_film = FG.id_gen_film  
      JOIN film F ON F.id_film = FG.id_film  
      JOIN filmactori FA ON F.id_film = FA.id_film  
      JOIN actor A ON A.id_actor = FA.id_actor  
WHERE A.id_actor IN (SELECT id_actor FROM actor  
                     HAVING YEAR(data_nasterii) > '$anul')  
ORDER BY F.num_film
```

3. Numele filmelor care au rating-ul mediu mai mare decât o valoare introdusă de utilizator. ([review_stars2.php](#))

```
SELECT F.num_film FROM film F  
      HAVING (SELECT AVG(R.numar_stele) FROM film F1  
              JOIN review R ON F1.id_film = R.id_film  
              WHERE F1.num_film = F.num_film ) >= '$stele'  
ORDER BY F.num_film
```

4. Numele filmelor în care unul sau mai mulți actori joacă în două sau mai multe filme. ([common_actors.php](#))

```
SELECT num_film FROM film  
      WHERE id_film IN (SELECT id_film FROM filmactori  
                        WHERE id_actor IN (SELECT id_actor FROM actor  
                                           WHERE id_actor IN (SELECT id_actor  
                                                             FROM filmactori  
                                                             GROUP BY id_actor HAVING COUNT(id_actor)>1)))
```

5. Filmele care nu au review-uri. ([without_reviews.php](#))

```
SELECT DISTINCT num_film  
      FROM film  
      WHERE id_film IN (SELECT id_film FROM film  
                        WHERE id_film NOT IN (SELECT id_film FROM review));
```

6. Actorii care joacă în filme lansate în alte țări exceptând o țară selectată de utilizator. ([actors_countries2.php](#))

```
SELECT A.num_actor, A.prename_actor, F.num_film  
      FROM ACTOR A  
      JOIN filmactori FA ON FA.id_actor = A.id_actor  
      JOIN film F ON FA.id_film = F.id_film  
      WHERE F.tara_origine IN (SELECT DISTINCT  
                               tara_origine FROM film  
                               WHERE tara_origine != '$tara' )  
ORDER BY F.num_film
```



Funcționarea aplicației

Aplicația implementată este accesibilă utilizatorilor care doresc să ofere review-uri unor filme și să clasifice filme după mai multe criterii pe care le pot alege.

Pentru a face acest lucru, utilizatorul trebuie să își creeze un cont pe site, prin intermediul unui formular în care își introduce datele. În cazul în care datele introduse sunt complete, contul se creează cu succes iar utilizatorul poate să se conecteze.

În momentul în care utilizatorul se conectează pe site, acesta poate să clasifice filmele, genurile sau actorii folosind diverse criterii, prin intermediul unor butoane. Filmele din baza de date se pot clasifica în funcție de gen, de actori, de review-uri, de media-review-urilor, de categoria de audiență și altele. Butoanele reprezintă atât interogări simple, cât și complexe, iar ele sunt explicate în paginile anterioare.

Pe partea de INSERT, UPDATE și DELETE, aceste operații se pot face atât pe tabela utilizator, cât și pe tabela review. Utilizator se poate înregistra pe site, își poate modifica detaliile și își poate șterge contul (și implicit review-urile date de el). Utilizatorul, de asemenea, poate să adauge un singur review unui film și îl poate modifica sau șterge ulterior.