ВЕЛИКОТЪРНОВСКИ УНИВЕРСИТЕТ "СВ. СВ. КИРИЛ И МЕТОДИЙ"



ФАКУЛТЕТ ,,МАТЕМАТИКА И ИНФОРМАТИКА”

**КУРСОВА РАБОТА**

по

учебната дисциплина

БАЗА ОТ ДАННИ

**Тема:** База от данни за филми

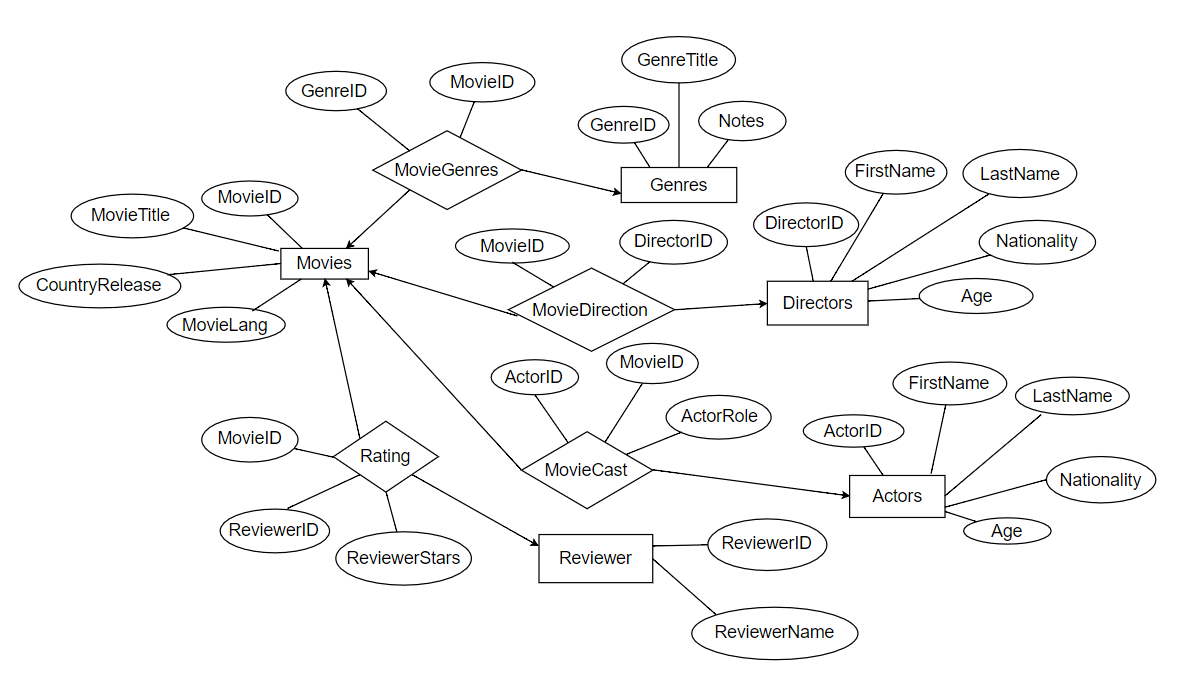
**Изготвил:** Кристина Василева Николова, специалност: Софтуерно инженерство, факултетен №:1809010559

**Проверил:** доц. д-р Цветанка Любомирова Георгиева-Трифонова

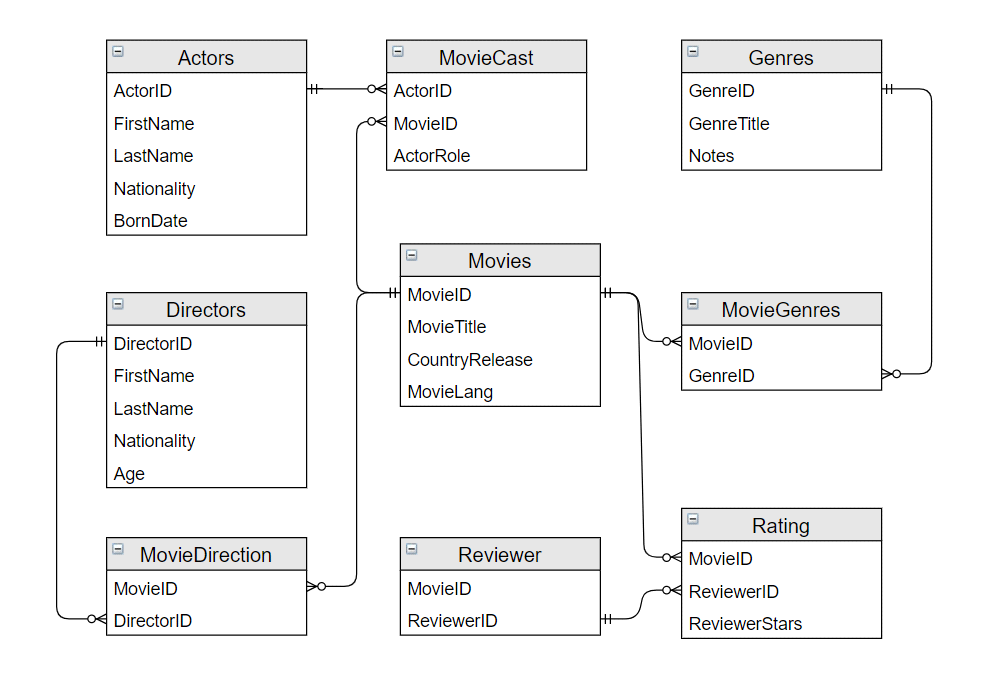
1. Описание на предметната област

Предметната област на базата от данни обхваща работата с информация за богата колекция от филми. За всеки филм са въведени подробни данни, свързани с актьори, които участват, режисьори, държава на излъчване, жанр. Може да се използва за по-лесно търсене и намиране на филми по предпочитание на зрителя. Търсенето може да се извършва по жанр, актьор, режисьор и рейтинг. Базата от данни е достъпна за всички потребители, като по този ги улеснява да намерят точно това, което търсят.

2. ER - диаграма

Обектите на базата от данни са основната таблица Movies, таблиците Genres, Directors, Actors и Reviewer, които са свързани с основната таблица Movies посредством свързващите таблици MovieGenres, MovieDirection, MoviesCast и Rating. Използвани са свързващи таблици, защото например за един филм има повече от един актьор, както и един актьор може да участва в повече от един филм. Тези релации са "много към много" и за да ги осъществим трябва да разглеждаме двете страни на релацията. За тази цел се създава свързваща таблица, която разделя релацията "много към много" на две релации "едно към много". Вмъкваме първичния ключ от двете главни таблици в свързващата трета таблица. В резултат на това третата таблица записва всяко срещане на релацията. 

3. Релационен модел на базата от данни



4. Реализация на релационния модел на базата от данни

-- Създаване на таблици

-- Таблицата Movies съдържа данни за филми: име, година на излизане, държава и език на филма

CREATE TABLE Movies

(

MovieID int NOT NULL PRIMARY KEY IDENTITY,

MovieTitle varchar(50) NOT NULL,

YearRelease int NOT NULL,

CountryRelease varchar(50) NOT NULL,

MovieLang varchar(50) NOT NULL,

)

-- Таблицата Actors съдържа данни за актьори: име, фамилия, националност, дата на раждане

CREATE TABLE Actors

(

ActorID int NOT NULL PRIMARY KEY IDENTITY,

FirstName varchar(50) NOT NULL,

LastName varchar(50) NOT NULL,

Nationality varchar(20) NOT NULL,

BornDate date NULL,

)

-- Таблицата Directors съдържа данни за режисьори: име, фамилия, националност, възраст

CREATE TABLE Directors

(

DirectorID int NOT NULL PRIMARY KEY IDENTITY,

FirstName varchar(50) NOT NULL,

LastName varchar(50) NOT NULL,

Nationality varchar(20) NOT NULL,

Age int NOT NULL,

)

-- Таблицата Genres съдържа данни за жанрове: име на жанр, бележки/обяснения

CREATE TABLE Genres

(

GenreID int NOT NULL PRIMARY KEY IDENTITY,

GenreTitle varchar(50) NOT NULL,

Notes varchar(max) NULL,

)

-- Таблицата Reviewer съдържаа данни за зрители: име

CREATE TABLE Reviewer

(

ReviewerID int NOT NULL PRIMARY KEY IDENTITY,

ReviewerName varchar(50) NOT NULL,

)

-- Таблицата MovieCast е свързваща таблица на таблиците Movies и Actors, като стойностите за ActorID и MovieID са референти по стойност

CREATE TABLE MovieCast

(

ActorID int NOT NULL REFERENCES Actors(ActorID),

MovieID int NOT NULL REFERENCES Movies(MovieID),

ActorRole varchar(max) NULL,

)

-- Таблицата MovieDirection е свързваща таблица на таблиците Movies и Directors, като стойностите за DirectorID и MovieID са референти по стойност

CREATE TABLE MovieDirection

(

DirectorID int NOT NULL REFERENCES Directors(DirectorID),

MovieID int NOT NULL REFERENCES Movies(MovieID),

)

-- Таблицата MovieGenres е свързваща таблица на таблиците Movies и Genres, като стойностите за GenreID и MovieID са референти по стойност

CREATE TABLE MovieGenres

(

MovieID int NOT NULL REFERENCES Movies(MovieID),

GenreID int NOT NULL REFERENCES Genres(GenreID),

)

-- Таблицата Rating е свързваща таблица на таблиците Movies и Reviewer, като стойностите за ReviewerID и MovieID са референти по стойност

CREATE TABLE Rating

(

MovieID int NOT NULL REFERENCES Movies(MovieID),

ReviewerID int NOT NULL REFERENCES Reviewer(ReviewerID),

ReviewerStars int NOT NULL,

)

-- Добавяме ограничение за стойностите на ReviewerStars

ALTER TABLE Rating

ADD CHECK (ReviewerStars >=0)

ALTER TABLE Rating

ADD CHECK (ReviewerStars <= 10)

-- Въвеждане на данни в таблицата Movies чрез командата INSERT INTO

INSERT INTO Movies(MovieTitle, YearRelease, CountryRelease, MovieLang)

VALUES ('Bohemian Rhapsody', 2018, 'United Kingdom', 'English')

INSERT INTO Movies(MovieTitle, YearRelease, CountryRelease, MovieLang)

VALUES ('SAW', 2004, 'USA', 'English')

INSERT INTO Movies(MovieTitle, YearRelease, CountryRelease, MovieLang)

VALUES ('Divergent‘, 2017, 'USA', 'English')

INSERT INTO Movies(MovieTitle, YearRelease, CountryRelease, MovieLang)

VALUES ('Finding Dory', 2016, 'USA', 'English')

INSERT INTO Movies(MovieTitle, YearRelease, CountryRelease, MovieLang)

VALUES ('The imitation game', 2014, 'USA', 'English')

INSERT INTO Movies(MovieTitle, YearRelease, CountryRelease, MovieLang)

VALUES ('Lucy', 2014, 'France', 'English')

-- Редактиране на стойността на MovieLang в таблицата Movies, където MovieID = 18

UPDATE Movies

SET MovieLang = 'French'

WHERE MovieID = 18

-- Изтриване на записи от таблицата Movies

DELETE FROM Movies

WHERE MovieTitle ='Divergent'

DELETE FROM Movies

WHERE MovieTitle ='It'

-- Въвеждане на данни в таблицата Genres чрез командата INSERT INTO

INSERT INTO Genres (GenreTitle, Notes)

VALUES ('Horror', 'A horror story is told to

deliberately scare or frighten the audience,

through suspense, violence or shock.')

INSERT INTO Genres (GenreTitle, Notes)

VALUES ('Comedy', 'Comedy is a story that

tells about a series of funny, or comical

events, intended to make the audience laugh.

It is a very open genre, and thus crosses over

with many other genres on a frequent basis.')

INSERT INTO Genres (GenreTitle, Notes)

VALUES ('Animation', NULL)

INSERT INTO Genres (GenreTitle, Notes)

VALUES ('Biographical', NULL)

INSERT INTO Genres (GenreTitle, Notes)

VALUES ('Sci-Fi', NULL)

INSERT INTO Genres (GenreTitle, Notes)

VALUES ('Action', NULL)

INSERT INTO Genres (GenreTitle, Notes)

VALUES ('Drama', NULL)

INSERT INTO Genres (GenreTitle, Notes)

VALUES ('Adventure', NULL)

INSERT INTO Genres (GenreTitle, Notes)

VALUES ('Fantasy', NULL)

INSERT INTO Genres (GenreTitle, Notes)

VALUES ('Romantic', NULL)

-- Актуализиране на данните в таблица Genres, където GenreID = 5

UPDATE Genres

SET Notes = 'Science fiction is typically dealing

with imaginative concepts such as advanced science

and technology, spaceflight, time travel,

and extraterrestrial life. '

WHERE GenreID = 5

-- Актуализиране на данните в таблица Genres, където GenreID = 6

UPDATE Genres

SET Notes = 'An action story is similar to

adventure, and the protagonist usually takes

a risky turn, which leads to desperate situations'

WHERE GenreID = 6

-- Актуализиране на данните в таблица Genres, където GenreID = 8

UPDATE Genres

SET Notes = 'An adventure story is about

a protagonist who journeys to epic or distant

places to accomplish something.'

WHERE GenreID = 8

-- Актуализиране на данните в таблица Genres, където GenreID = 9

UPDATE Genres

SET Notes = 'A fantasy story is about magic

or supernatural forces'

WHERE GenreID = 9

-- Въвеждане на данни в таблицата Directors чрез командата INSERT INTO

INSERT INTO Directors (FirstName, LastName, Nationality, Age)

VALUES ('Bryan', 'Singer', 'American', 53)

INSERT INTO Directors (FirstName, LastName, Nationality, Age)

VALUES ('James', 'Wan', 'Australian', 41)

INSERT INTO Directors (FirstName, LastName, Nationality, Age)

VALUES ('David', 'Ayer', 'American', 50)

INSERT INTO Directors (FirstName, LastName, Nationality, Age)

VALUES ('Luc', 'Besson', 'French', 59)

INSERT INTO Directors (FirstName, LastName, Nationality, Age)

VALUES ('John', 'Krasinski', 'American', 39)

-- Актуализиране на данните в таблица Directors, където DirectorID = 4

UPDATE Directors

SET FirstName = 'Nick'

WHERE DirectorID = 4

-- Въвеждане на данни в таблицата Actors чрез командата INSERT INTO

INSERT INTO Actors(FirstName, LastName, Nationality, BornDate)

VALUES ('Rami', 'Malek', 'Egyptian-American', '1981-05-12')

INSERT INTO Actors(FirstName, LastName, Nationality, BornDate)

VALUES ('Monica', 'Potter', 'American', '1971-06-30')

INSERT INTO Actors(FirstName, LastName, Nationality, BornDate)

VALUES ('Tobin', 'Bell', 'American', '1942-08-07')

INSERT INTO Actors(FirstName, LastName, Nationality, BornDate)

VALUES ('Rachel', 'McAdams', 'Canadian', '1978-11-17')

INSERT INTO Actors(FirstName, LastName, Nationality, BornDate)

VALUES ('Amanda', 'Seyfried', 'American', '1985-12-03')

INSERT INTO Actors(FirstName, LastName, Nationality, BornDate)

VALUES ('Benedict', 'Cumberbatch', 'English', '1976-07-19')

INSERT INTO Actors(FirstName, LastName, Nationality, BornDate)

VALUES ('Morgan', 'Freeman', 'American', '1937-06-01')

-- Въвеждане на данни в таблицата Reviewer чрез командата INSERT INTO

INSERT INTO Reviewer(ReviewerName)

VALUES ('Floriana Quickley')

INSERT INTO Reviewer(ReviewerName)

VALUES ('Clara Morison')

INSERT INTO Reviewer(ReviewerName)

INSERT INTO Reviewer(ReviewerName)

VALUES ('Evie Hanson')

-- Актуализиране на данните в таблицата Reviewer, където ReviewerID = 1

UPDATE Reviewer

SET ReviewerName = 'Janey Becket'

WHERE ReviewerID = 1

-- Въвеждане на данни в таблицата MovieCast чрез командата INSERT INTO

INSERT INTO MovieCast(ActorID, MovieID, ActorRole)

VALUES (1, 1, NULL)

INSERT INTO MovieCast(ActorID, MovieID, ActorRole)

VALUES (2, 2, NULL)

INSERT INTO MovieCast(ActorID, MovieID, ActorRole)

VALUES (3, 2, NULL)

INSERT INTO MovieCast(ActorID, MovieID, ActorRole)

VALUES (4, 3, NULL)

INSERT INTO MovieCast(ActorID, MovieID, ActorRole)

VALUES (10, 7, NULL)

INSERT INTO MovieCast(ActorID, MovieID, ActorRole)

VALUES (10, 8, NULL)

-- Въвеждане на данни в таблицата MovieDirection чрез командата INSERT INTO

INSERT INTO MovieDirection(DirectorID, MovieID)

VALUES (1, 1)

INSERT INTO MovieDirection(DirectorID, MovieID)

VALUES (2, 2)

INSERT INTO MovieDirection(DirectorID, MovieID)

VALUES (5, 6)

INSERT INTO MovieDirection(DirectorID, MovieID)

VALUES (6, 7)

-- Въвеждане на данни в таблицата MovieGenres чрез командата INSERT INTO

INSERT INTO MovieGenres(MovieID, GenreID)

VALUES(1, 4)

INSERT INTO MovieGenres(MovieID, GenreID)

VALUES(1, 7)

INSERT INTO MovieGenres(MovieID, GenreID)

VALUES(2, 1)

INSERT INTO MovieGenres(MovieID, GenreID)

VALUES(3, 5)

INSERT INTO MovieGenres(MovieID, GenreID)

VALUES(3, 8)

INSERT INTO MovieGenres(MovieID, GenreID)

VALUES(4, 2)

-- Въвеждане на данни в таблицата Rating чрез командата INSERT INTO

INSERT INTO Rating(MovieID, ReviewerID, ReviewerStars)

VALUES(1, 1, 10)

INSERT INTO Rating(MovieID, ReviewerID, ReviewerStars)

VALUES(1, 2, 8)

INSERT INTO Rating(MovieID, ReviewerID, ReviewerStars)

VALUES(2, 3, 5)

INSERT INTO Rating(MovieID, ReviewerID, ReviewerStars)

VALUES(2, 4, 7)

INSERT INTO Rating(MovieID, ReviewerID, ReviewerStars)

VALUES(3, 5, 7)

5. Създаване на заявки

-- Заявката извежда всички филми, които за почват с буквата "Т"

SELECT \* FROM Movies

WHERE MovieTitle LIKE 'T%'

-- Заявката извежда всички данни за актьорите, родени между 01.01.1970г. и 31.12.1980г.

SELECT \* FROM Actors

WHERE BornDate BETWEEN '1970-01-01' AND '1980-12-31'

--Заявката извежда данни за актьорите, подредени по дата на раждане във възходящ ред

SELECT \* FROM Actors

ORDER BY BornDate ASC

-- Заявката извежда броя на режисьорите от всяка националност, от която има двама или повече режисьори, и ги подрежда във възходящ ред

SELECT COUNT(DirectorID), Nationality

FROM Directors

GROUP BY Nationality

HAVING COUNT(DirectorID) >= 2

ORDER BY COUNT(DirectorID) ASC;

-- Заявката извежда броя на актьорите, които са американци

SELECT COUNT(ActorID)

FROM Actors

WHERE Nationality LIKE 'american';

-- Заявката извежда данните за всички жанрове, които за почват с буквата "А", подредени по низходящ ред

SELECT \* FROM Genres

WHERE GenreTitle LIKE 'A%'

ORDER BY GenreID DESC;

-- Заявката извежда номера на филмите, които са жанр с ID номер = 1, като е използвано вътрешно съединение между таблиците, за да се изведе и името на съответния филм

SELECT Movies.MovieID, MovieTitle FROM MovieGenres AS Horrors

INNER JOIN Movies ON Horrors.MovieID = Movies.MovieID

WHERE GenreID = 1

6. Създаване на изгледи, съхранени процедури, потребителски дефинирани функции, тригери

-- Изглед, който извлича името на филмите, чиято година на издаване е 2016

CREATE VIEW [2016 Movies] AS

SELECT MovieTitle

FROM Movies

WHERE YearRelease = 2016;

-- Заявка, която прави обръщение към изгледа

SELECT \* FROM [2016 Movies];

-- Създаване на съхранена процедура, която съдържа структура за извикване на всички американски актьори, осигурява възможност за многократно извикване

CREATE PROCEDURE SelectAmericanActors

AS

SELECT \*

FROM Actors

WHERE Nationality LIKE 'american';

-- Извикване на съхранената процедура

EXEC SelectAmericanActors;

--Тригера забранява изтриване на 2, 3 и 5ти ред. След опит за изтриване, чрез заявка изписна: "Не е разрешено изтриване на редове в таблицата"

CREATE TRIGGER No\_DeleteGenres

ON Genres

INSTEAD OF DELETE

AS

IF @@ROWCOUNT = 0 RETURN

RAISERROR ('Не е разрешено изтриване на

редове в таблицата.', 2, 3, 5 )

7. Бъдеща работа

Възможна посока на бъдещо развитие на разработеното приложение е базата от данни да се използва в сайт, който предлага богат каталог от филми. Могат да се добавят допълнителни характеристики за филмите, като например дали са с българско или различно аудио, дали базата данни съхранява субтитри за съответен филм и други. Могат да се разширят възможностите за търсене, например да се търси по ключова дума в описание. Би било подходящо приложението да се използва в сайт в съвместност с други приложения, които предлагат каталози от сериали, музика, игри и други.