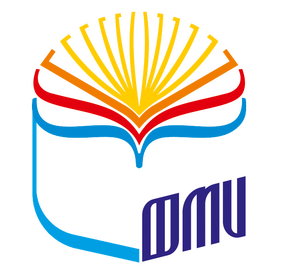
****

База от данни за филми

Курсова работа

Дисциплина: Бази от данни

**Изготвил**

Име: Иван Добринов Пеев

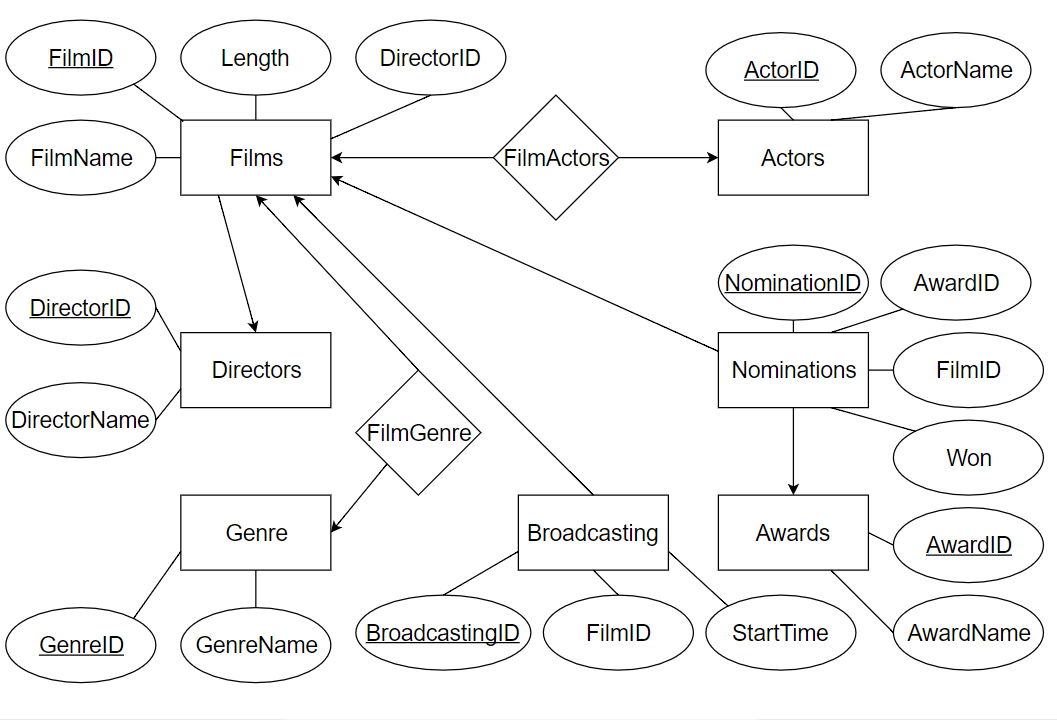
Специалност: Софтуерно Инженерство, I курс

Факултетен номер: 20048

**1.Определяне на предметната област:**

Базата от данни може да бъде използвана в киносалон за попълване и извличане на информация за филмите, актьорите, участващи в тях, жанровете им, техните режисьори, номинации, награди и прожекциите в дадения киносалон.

**2.ER-модел на базата от данни**



**3.Релационен модел на базата от данни**

След преобразуване в релационен модел таблиците имат следната структура:

**Films**(FilmID,FilmName,Length,DirectorID)

**Actors**(ActorID,ActorName)

**Directors**(DirectorID,DirectorName)

**Genre**(GenreID,GenreName)

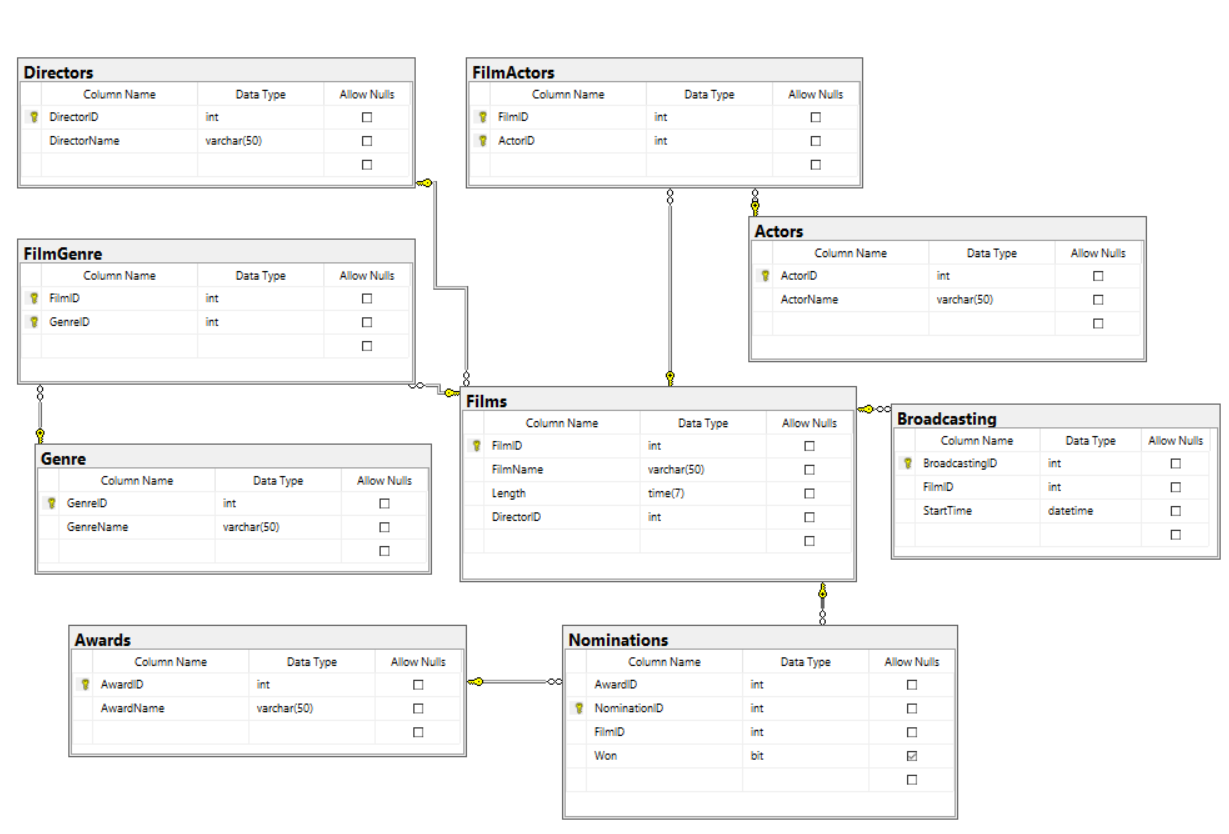
**Nominations**(NominationID,AwardID,FilmID,Won)

**Awards**(AwardID,AwardName)

**Broadcasting**(BroadcastingID,FilmID,StartTime)

-две от връзките са от тип много към много и са направени с две допълнителни таблици, защото един актьор може да участва в много филми и в един филм могат да участват много актьори, както и един филм може да принадлежи към няколко жанра и към един жанр могат да принадлежат много филми. За целта са използвани таблиците **FilmActors**(FilmID,ActorID) и **FilmGenre**(FilmID,GenreID).

-другите връзки между таблиците са от типа едно към много: един режисьор може да режисира много филми, за една награда може да има много номинации, както и един филм може да бъде номиниран за много награди и един филм може да бъде излъчен много пъти.



**4.Реализация на релационния модел на базата от данни**

**-Създаване на таблицата Films:**

CREATE TABLE dbo.Films(

FilmID int NOT NULL,

FilmName varchar(50) NOT NULL,

Length time(7) NOT NULL,

DirectorID int NOT NULL,

CONSTRAINT PK\_Films PRIMARY KEY

(

FilmID ASC

),

CONSTRAINT UK\_Films UNIQUE

(

FilmName ASC

)

);

**-Създаване на таблицата Actors:**

CREATE TABLE dbo.Actors(

ActorID int NOT NULL,

ActorName varchar(50) NOT NULL,

CONSTRAINT PK\_Actors PRIMARY KEY

(

ActorID ASC

)

);

**-Създаване на таблицата Directors:**

CREATE TABLE dbo.Directors(

DirectorID int NOT NULL,

DirectorName varchar(50) NOT NULL,

CONSTRAINT PK\_Directors PRIMARY KEY

(

DirectorID ASC

)

);

**-Създаване на таблицата Genre:**

CREATE TABLE dbo.Genre(

GenreID int NOT NULL,

GenreName varchar(50) NOT NULL,

CONSTRAINT PK\_Genre PRIMARY KEY

(

GenreID ASC

)

);

**-Създаване на таблицата Nominations:**

CREATE TABLE dbo.Nominations(

AwardID int NOT NULL,

NominationID int NOT NULL,

FilmID int NOT NULL,

Won bit NULL,

CONSTRAINT PK\_Nominations PRIMARY KEY

(

NominationID ASC

)

);

**-Създаване на таблицата Awards:**

CREATE TABLE dbo.Awards(

AwardID int NOT NULL,

AwardName varchar(50) NOT NULL,

CONSTRAINT PK\_Awards PRIMARY KEY CLUSTERED

(

AwardID ASC

)

);

**-Създаване на таблицата Broadcasting:**

CREATE TABLE dbo.Broadcasting(

BroadcastingID int NOT NULL,

FilmID int NOT NULL,

StartTime datetime NOT NULL,

CONSTRAINT PK\_Broadcasting PRIMARY KEY

(

BroadcastingID ASC

)

);

**-Създаване на таблицата FilmActors:**

CREATE TABLE dbo.FilmActors(

FilmID int NOT NULL,

ActorID int NOT NULL,

CONSTRAINT PK\_FilmActors PRIMARY KEY

(

FilmID ASC,

ActorID ASC

)

);

**-Създаване на таблицата FilmGenre:**

CREATE TABLE dbo.FilmGenre(

FilmID int NOT NULL,

[GenreID] int NOT NULL,

CONSTRAINT PK\_FilmGenre PRIMARY KEY

(

FilmID ASC,

GenreID ASC

)

);

**-Добавяне на нужните външни ключове, on update и on delete:**

ALTER TABLE dbo.Broadcasting

ADD CONSTRAINT FK\_Broadcasting\_Films

FOREIGN KEY(FilmID)

REFERENCES dbo.Films (FilmID)

ON UPDATE CASCADE

ON DELETE CASCADE

ALTER TABLE dbo.FilmActors

ADD CONSTRAINT FK\_FilmActors\_Actors

FOREIGN KEY(ActorID)

REFERENCES dbo.Actors (ActorID)

ON UPDATE CASCADE

ON DELETE CASCADE

ALTER TABLE dbo.FilmActors

ADD CONSTRAINT FK\_FilmActors\_Films

FOREIGN KEY(FilmID)

REFERENCES dbo.Films (FilmID)

ON UPDATE CASCADE

ON DELETE CASCADE

ALTER TABLE dbo.FilmGenre

ADD CONSTRAINT FK\_FilmGenre\_Films]

FOREIGN KEY(FilmID)

REFERENCES dbo.Films (FilmID)

ON UPDATE CASCADE

ON DELETE CASCADE

ALTER TABLE dbo.FilmGenre

ADD CONSTRAINT FK\_FilmGenre\_Genre

FOREIGN KEY(GenreID)

REFERENCES dbo.Genre (GenreID)

ON UPDATE CASCADE

ON DELETE CASCADE

ALTER TABLE dbo.Films

ADD CONSTRAINT FK\_Films\_Directors

FOREIGN KEY(DirectorID)

REFERENCES dbo.Directors (DirectorID)

ON UPDATE CASCADE

ALTER TABLE dbo.Nominations

ADD CONSTRAINT FK\_Nominations\_Awards

FOREIGN KEY(AwardID)

REFERENCES dbo.Awards (AwardID)

ON UPDATE CASCADE

ON DELETE CASCADE

ALTER TABLE dbo.Nominations

ADD CONSTRAINT FK\_Nominations\_Films

FOREIGN KEY(FilmID)

REFERENCES dbo.Films (FilmID)

ON UPDATE CASCADE

ON DELETE CASCADE

**-Попълване на таблицата Actors:**

INSERT INTO Actors

VALUES(1,’Vin Diesel’)

INSERT INTO Actors

VALUES(2,’Paul Walker’)

INSERT INTO Actors

VALUES(3,’Jesica Bart’)

INSERT INTO Actors

VALUES(4,’Mark Wahlberg’)

INSERT INTO Actors

VALUES(5,’Will Smith’)

UPDATE Actors

SET ActorName=’Mila Kunis’

WHERE ActorID=3;

DELETE FROM Actors

WHERE ActorName=’Will Smith’;

**-Попълване на таблицата Directors:**

INSERT INTO Directors

VALUES(1,’Rob Coen’)

INSERT INTO Actors

VALUES(2,’Andy Tenant’)

INSERT INTO Actors

VALUES(3,’Will Gluck’)

UPDATE Directors

SET DirectorName=’Seth Macfarlane’

WHERE DirectorID=2;

DELETE FROM Directors

WHERE DirectorName=’Will Gluck’;

**-Попълване на таблицата Genre:**

INSERT INTO Genre

VALUES(1,’Action’)

INSERT INTO Genre

VALUES(2,’Drama’)

INSERT INTO Genre

VALUES(3,’Triler’)

UPDATE Genre

SET GenreName=’Comedy’

WHERE GenreID=2;

DELETE FROM Genre

WHERE GenreID=3;

**-Попълване на таблицата Films:**

INSERT INTO Films

VALUES(1,’Fast and furious’, 01:46:00,1)

INSERT INTO Films

VALUES(2,’Hitch’, 01:46:00,2)

INSERT INTO Films

VALUES(3,’Friends with benefits’, 01:49:00,2)

UPDATE Films

SET FilmName=’Ted’

WHERE FilmID=2;

DELETE FROM Films

WHERE FilmID=3;

**-Попълване на таблицата FilmActors:**

INSERT INTO FilmActors

VALUES(1,1)

INSERT INTO FilmActors

VALUES(1,2)

INSERT INTO FilmActors

VALUES(2,3)

INSERT INTO FilmActors

VALUES(2,4)

**-Попълване на таблицата FilmGenre:**

INSERT INTO FilmGenre

VALUES(1,2)

INSERT INTO FilmGenre

VALUES(2,1)

UPDATE FilmGenre

SET GenreID=1

WHERE FilmID=1;

UPDATE FilmGenre

SET GenreID=2

WHERE FilmID=2;

**-Попълване на таблицата Broadcasting:**

INSERT INTO Broadcasting

VALUES(1,1, 2018-02-02 15:30:00)

INSERT INTO Broadcasting

VALUES(2,2, 2018-02-02 18:30:00)

INSERT INTO Broadcasting

VALUES(3,2, 2018-02-03 15:30:00)

INSERT INTO Broadcasting

VALUES(4,2, 2018-02-03 17:30:00)

UPDATE Broadcasting

SET FilmID=1

WHERE BroadcastingID=2;

DELETE FROM Broadcasting

WHERE BroadcastingID=4;

**-Попълване на таблицата Awards:**

INSERT INTO Awards

VALUES(1,’Black Reel’)

INSERT INTO Awards

VALUES(2,’Tony’)

INSERT INTO Awards

VALUES(3,’Eddie’)

UPDATE Awards

SET AwardName=’Oscar’

WHERE AwardID=2;

DELETE FROM Awards

WHERE AwardID=3;

**-Попълване на таблицата Nominations:**

INSERT INTO Nominations

VALUES(1,1,1,0)

INSERT INTO Nominations

VALUES(2,2,2,1)

INSERT INTO Nominations

VALUES(2,3,1,0)

UPDATE Nominations

SET Won=0

WHERE NominationID=2;

DELETE FROM Nominations

WHERE NominationID=3;

**5.Създаване на заявки**

**-Извеждане на филмите заедно с всички часове на тяхното излъчване:**

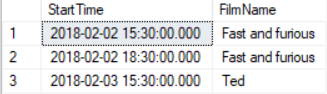
SELECT

b.StartTime

,f.FilmName

FROM dbo.Broadcasting b

INNER JOIN Films f ON f.FilmID=b.FilmID

****

**-Извеждане на броя предстоящи прожекции на филмите:**

SELECT

COUNT(b.StartTime) AS NumberOfBroadcasts

,f.FilmName

FROM dbo.Broadcasting b

INNER JOIN Films f ON f.FilmID=b.FilmID

GROUP BY f.FilmName

****

**-Извеждане на броя на филмите режисирани от Rob Coen**

SELECT

COUNT(f.FilmName) AS DirectedFilms

,d.DirectorName

FROM dbo.Directors d

INNER JOIN Films f ON f.DirectorID=d.DirectorID

GROUP BY d.DirectorName

HAVING d.DirectorName='Rob Coen'



**-Извеждане на най-скорошната прожекция заедно с филма, на който ще е:**

SELECT

b.StartTime

,f.FilmName

FROM dbo.Broadcasting b

INNER JOIN Films f ON f.FilmID=b.FilmID

WHERE b.StartTime=(SELECT MIN(b.StartTime) FROM Broadcasting b) AND b.StartTime>GETDATE()

****

**-Извеждане на филмите заедно с наградите, за които са били номинирани, но не са спечелили:**

SELECT

a.AwardName

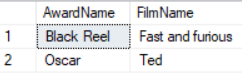
,f.FilmName

FROM dbo.Awards a

INNER JOIN Nominations n ON a.AwardID= n.AwardID

INNER JOIN Films f ON f.DirectorID=n.FilmID

WHERE n.Won=0

****

**6.Създаване на необходимите изгледи, съхранени процедури или тригери**

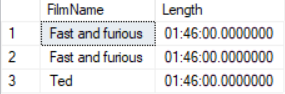
**-Създаване на изглед за филмите, които все още се излъчват заедно с времето на тяхното излъчване:**

CREATE VIEW ActiveFilms AS

SELECT f.FilmName,f.Length FROM Films f

INNER JOIN Broadcasting b ON f.FilmID=b.FilmID

WHERE b.StartTime>GETDATE()

****

**-Създаване на процедура за извеждане на всички филми с дължина по-малка от зададената:**

CREATE PROCEDURE GetLength @Length varchar(50)

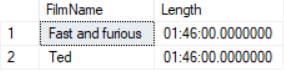
AS

SELECT f.FilmName, f.Length FROM Films f

WHERE f.Length<@Length;

**-Изпълнение на процедурата:**

EXEC GetLength @Length='02:00:00';

****

**-Създаване на тригер, който се стартира след изтриване от таблицата Films и запаметява в таблицата OldFilms:**

**За целта създаваме таблица OldFilms:**

CREATE TABLE OldFilms(

FilmID int NOT NULL,

FilmName varchar(50) NOT NULL,

Length time(7) NOT NULL,

DeletedDate datetime NULL

);

**Създаваме тригера AFDEL\_OldFilms:**

CREATE TRIGGER AFDEL\_OldFilms

ON Films

AFTER DELETE AS

BEGIN

INSERT INTO OldFilms

SELECT d.FilmID,d.FilmName, d.Length, GETDATE() FROM DELETED d

END

**Стартиране на тригера:**

DELETE FROM Films

WHERE FilmID=3;

**Чрез SELECT заявка извличаме данните от таблицата OldFilms:**

****

**7. Бъдеща работа**

Възможности за бъдещо развитие на базата данни са да се включат продажбите на билети и запазване на места в кинозалата, възможно е да се включи и продажбата на храни и напитки в киносалона.