

СОФИЙСКИ УНИВЕРСИТЕТ “СВ. КЛИМЕНТ ОХРИДСКИ”

ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА

Документация

Дисциплина:

**Системи за управление на бази от данни –
Практикум**

Тема: Филм – Хари Потър

Преподавател:

Доц. Радослава Христова

Изготвено от:

Александрина Ковачка, ФН:71890

Част 1: Описание на множествата същности

1.Обхват на модела. Дефиниране на задачата.

Проектът ще съхранява информация за всички части на филма Хари Потър. В базата от данни ще се съхранява информация за трите филма, както и книгите по които са създадени, трейлърите за всеки един филм, актьорите, участвали в трилогията и директорите на всеки филм. Всеки филм е направен по точно една книга. Всеки трейлър и директор е за точно един филм. Актьорите имат своите реални и игрални имена, като даден актьор може да участва само веднъж или в повече от един филм.

2. Множества от същности и техните атрибути

- Филми –име на филм, дата на излизане, времетраене, рейтинг
- Трейлъри – код на филм, дата на излизане, времетраене
- Актьори – игрално име, реално име, брой филми
- Директори – код на директор, име
- Книги – код на книга, заглавие, година на издаване, име на автор

3. Домейн на атрибутите

Филм

- име на филм - низ
- дата на излизане - дата
- времетраене - време(time)
- рейтинг - реално число

Трейлъри (слабо множество същности)

- име на филм - низ
- дата на излизане - дата
- времетраене - време(time)

Актьори

- игрално име - низ
- реално име - низ
- брой филми – цяло положително число

Директори

- код на директор – низ, точно 3 символа
- име – низ

Книги

- код на книга – низ, точно 4 символа
- заглавие – низ
- година на издаване – цяло положително число
- име на автора – низ

4. Връзки

Филми - Трейлъри

Филми - Актьори

Филми – Директори

Филми – Книги

5. Ограничения по единствена стойност, референтна цялостност и друг тип ограничения

Филми

- име на филм – низ, **ключ**
- рейтинг - реално число, между 0 и 10
- дата на излъчване - дата
- времетраене - време(time), повече от 60мин

Трейлъри

- име на филм - низ, **ключ**
- дата на излъчване - дата, **ключ**
- времетраене - време(time)

Актьори

- игрално име - низ, **ключ**
- реално име - низ
- брой филми – цяло положително число ≤ 3

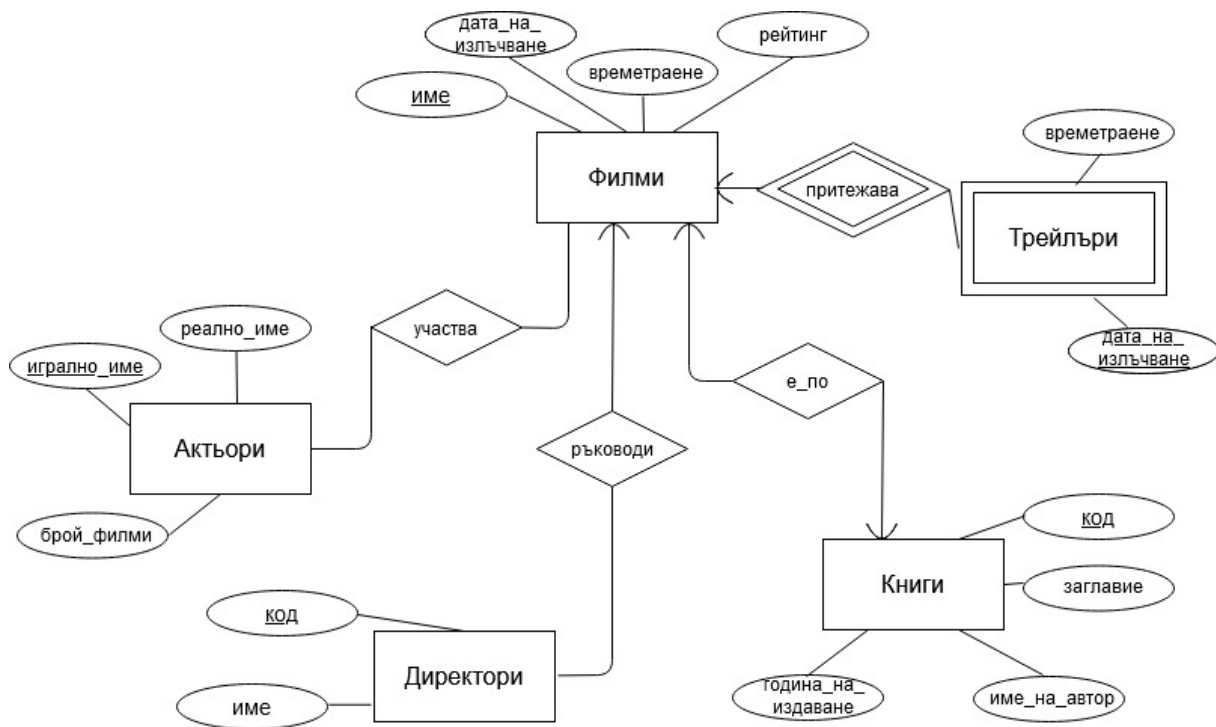
Директори

- код на директор - низ, точно 3 символа, **ключ**
- име - низ

Книги

- код на книга - низ, точно 4 символа, **ключ**
- заглавие - низ
- година на издаване – цяло положително число
- име на автора – низ

Част 2: Картинка на E/R диаграма на модела на БД



Част 3: Преобразуване от E/R модел към релационен модел

Филми (име, датаНаИзлизане, времетраене, рейтинг)

Трейлъри (датаНаИзлизане, имеНаФилм, времетраене)

Актьори (игралноИме, реалноИме, бройФилми)

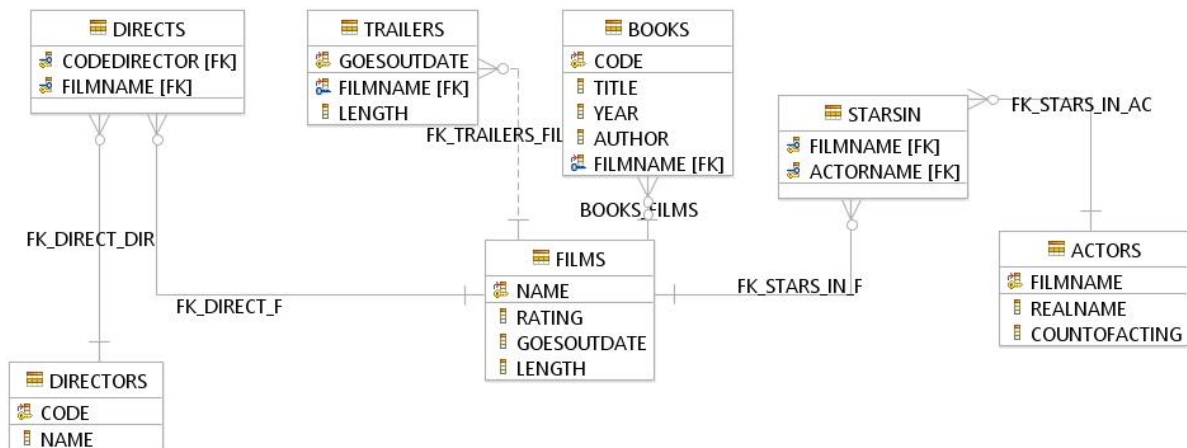
Участва (имеНаФилм, игралноИмеНаАктьор)

Директори (код, име)

Ръководи (кодНаДиректор, имеНаФилм)

Книги (код, заглавие, годинаНаИздаване, имеНаАвтор, имеНаФилм)

Част 4: Картинка на релационния модел



Част 5: Описание на функциите

Функция 1:

```
--FUNCTION_1
CREATE FUNCTION GET_FIMNAME(V_BOOKNAME VARCHAR(50))
RETURNS VARCHAR(50)
RETURN
    SELECT FILMNAME
    FROM BOOKS
    WHERE TITLE = V_BOOKNAME;
```

По подадено име на книга, функцията връща филма направен по книгата.

```
--CALL FUNCTION_1
VALUES GET_FIMNAME('Goblet of Fire');
```

При извикване на функцията резултата е:

Properties	SQL Results
1	----- Harry Potter and the Goblet of Fire

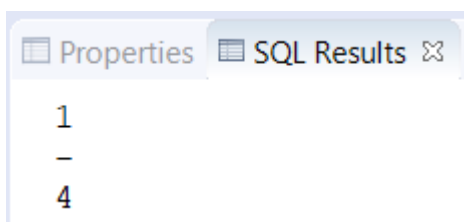
Функция 2:

```
--FUNCTION_2
CREATE FUNCTION GET_COUNT_DIRECTOR(V_CODEDIRECTOR CHAR(4))
RETURNS INT
RETURN
    SELECT COUNT(*)
    FROM DIRECTS
    WHERE CODEDIRECTOR = V_CODEDIRECTOR;
```

По подаден номер на директор, функцията връща броя на филмите, които е режисирал.

```
--CALL FUNCTION_2
VALUES GET_COUNT_DIRECTOR('004');
```

При извикване на функцията резултата е:



1
-
4

Част 6: Описание на тригерите

Тригер 1:

```
--TRIGGER_1
CREATE TRIGGER INSERT_RATING
    NO CASCADE BEFORE INSERT ON FILMS
    REFERENCING NEW AS N
    FOR EACH ROW
    WHEN ( N.RATING IS NULL)
        SET N.RATING = 10;
```

Тригер, който при въвеждане на рейтинг, в таблицата FILMS, при стойност NULL задава стойност 10.

Част 7: Описание на изгледите

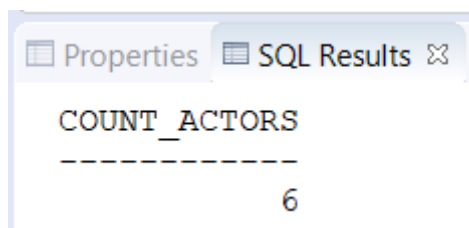
Изглед 1:

```
--VIEW_1
CREATE VIEW ACTORS_IN_FILM(COUNT_ACTORS)
AS
    SELECT COUNT(*)
    FROM STARSIN
    WHERE FILMNAME = 'Harry Potter and the Order of the Phoenix';
```

Изглед, който връща броя на актьорите участвали във филма Harry Potter and the Order of the Phoenix.

```
--CALL VIEW
SELECT * FROM ACTORS_IN_FILM;
```

Резултат от извикването на изгледа:



COUNT_ACTORS
6

Изглед 2:

```
--VIEW_2
CREATE VIEW ACTORS_DIRECTORS(FILMNAME, ACTORNAME, CODEDIRECTOR)
AS
    SELECT S.FILMNAME, S.ACTORNAME, D.CODEDIRECTOR
    FROM STARSIN S, DIRECTS D
    WHERE S.FILMNAME = D.FILMNAME AND S.ACTORNAME <> 'Percy Weasley';
```

Изглед, който връща филмите, актьорите и номера на директора на филма, без актьора Percy Weasley.

```
--CALL VIEW
SELECT * FROM ACTORS_DIRECTORS;
```


Резултат от извикването на изгледа:

FILMNAME	ACTORNAME	CODEDIRECTOR
Harry Potter and the Chamber of Secrets	Harry Potter	001
Harry Potter and the Chamber of Secrets	Hermione Granger	001
Harry Potter and the Chamber of Secrets	James Potter	001
Harry Potter and the Chamber of Secrets	Lily Potter	001
Harry Potter and the Chamber of Secrets	Ron Weasley	001
Harry Potter and the Deathly Hallows - Part 1	Harry Potter	004
Harry Potter and the Deathly Hallows - Part 1	Hermione Granger	004
Harry Potter and the Deathly Hallows - Part 1	James Potter	004
Harry Potter and the Deathly Hallows - Part 1	Lily Potter	004
Harry Potter and the Deathly Hallows - Part 1	Ron Weasley	004
Harry Potter and the Deathly Hallows - Part 2	Harry Potter	004
Harry Potter and the Deathly Hallows - Part 2	Hermione Granger	004
Harry Potter and the Deathly Hallows - Part 2	James Potter	004
Harry Potter and the Deathly Hallows - Part 2	Lily Potter	004
Harry Potter and the Deathly Hallows - Part 2	Quirinus Quirrell	004
Harry Potter and the Deathly Hallows - Part 2	Ron Weasley	004
Harry Potter and the Goblet of Fire	Harry Potter	003
Harry Potter and the Goblet of Fire	Hermione Granger	003
Harry Potter and the Goblet of Fire	James Potter	003
Harry Potter and the Goblet of Fire	Lily Potter	003
Harry Potter and the Goblet of Fire	Ron Weasley	003
Harry Potter and the Half-Blood Prince	Harry Potter	004
Harry Potter and the Half-Blood Prince	Hermione Granger	004
Harry Potter and the Half-Blood Prince	Lily Potter	004
Harry Potter and the Half-Blood Prince	Ron Weasley	004
Harry Potter and the Order of the Phoenix	Harry Potter	004
Harry Potter and the Order of the Phoenix	Hermione Granger	004
Harry Potter and the Order of the Phoenix	James Potter	004
Harry Potter and the Order of the Phoenix	Lily Potter	004
Harry Potter and the Order of the Phoenix	Ron Weasley	004
Harry Potter and the Philosophers Stone	Adrian Pucey	001
Harry Potter and the Philosophers Stone	Harry Potter	001

Част 8: Описание на процедурите

Процедура 1:

```
--PROCEDURE_1 - процедура с курсор и входни и изходни параметри
CREATE PROCEDURE FN71890.film_director(IN film_actor_name VARCHAR(50), OUT result VARCHAR(50))
RESULT SETS 1
LANGUAGE SQL
SPECIFIC film_director
BEGIN
    DECLARE cursor1 CURSOR WITH RETURN FOR SELECT CODEDIRECTOR
        FROM DIRECTS WHERE FILMNAME = result AND (film_actor_name) IN (SELECT ACTORNAME FROM STARSIN WHERE FILMNAME = result);

    SET result = 'Harry Potter and the Philosophers Stone';

    OPEN cursor1;
END@
```

Процедура, която по въведено име на актьор, задава изходен параметър филмът Harry Potter and the Philosophers Stone и извежда кода на директора на този филм само ако въведения актьор участва в този филм.

```
--CALL PROCEDURE_1
CALL FN71890.film_director('Harry Potter', ?)@
```

Резултат от процедурата:

Properties	SQL Results
CODEDIRECTOR	

001	

Properties

SQL Results

Name	Type	Data type	Value	Value (OUT)
FILM_ACTOR_NAME	INPUT	VARCHAR	Harry Potter	
RESULT	OUTPUT	VARCHAR		Harry Potter and the Philosophers Stone

Процедура 2:

```
--PROCEDURE_2 - процедура с прихващане на изключение
CREATE PROCEDURE FN71890.film_name_error(IN book_name VARCHAR(50), OUT result VARCHAR(50))
LANGUAGE SQL
BEGIN
    DECLARE exit_state INTEGER DEFAULT 0;
    DECLARE v_name_book VARCHAR(50) DEFAULT ' ';
    DECLARE v_name_film VARCHAR(50) DEFAULT ' ';
    DECLARE not_found CONDITION FOR SQLSTATE '02000';

    DECLARE cursor1 CURSOR FOR SELECT TITLE, FILMNAME FROM BOOKS;

    DECLARE CONTINUE HANDLER FOR not_found SET exit_state = 1;

    OPEN cursor1;
    filmname_loop: LOOP
        FETCH cursor1 INTO v_name_book, v_name_film;
        IF exit_state = 1 THEN LEAVE filmname_loop;
        ELSEIF v_name_book = book_name THEN SET result = v_name_film;
        END IF;
    END LOOP;

    CLOSE cursor1;
END@
```

Процедурата приема име на книга, създава параметри за име на книга и име на филм като взима информация от таблицата BOOKS и ги поставя в курсора. С цикъл се обхожда тази информация и ако подаденото име на книга се съдържа то в изходния параметър се запазва името на филма, направен по книгата. Ако не го открие то процедурата ще излезе с грешка 02000.

```
--CALL PROCEDURE_2
CALL FN71890.film_name_error('Philosophers Stone', ?)@
```

Резултат от процедурата:

Properties SQL Results				
Name	Type	Data type	Value	Value (OUT)
BOOK_NAME	INPUT	VARCHAR	Philosophers Stone	
RESULT	OUTPUT	VARCHAR		Harry Potter and the Philosophers Stone

Процедура 3:

```
--PROCEDURE_3 - процедура с курсор и while цикъл
CREATE PROCEDURE FN71890.real_name(IN real_actor_name VARCHAR(50), OUT result VARCHAR(50))
LANGUAGE SQL
BEGIN
    DECLARE exit_state INTEGER DEFAULT 0;
    DECLARE v_real_name VARCHAR(50) DEFAULT ' ';
    DECLARE v_film_name VARCHAR(50) DEFAULT ' ';

    DECLARE cursor1 CURSOR FOR SELECT FILMNAME, REALNAME FROM ACTORS;

    OPEN cursor1;

    WHILE exit_state = 0 DO
        FETCH cursor1 INTO v_film_name, v_real_name;
        IF v_real_name = real_actor_name THEN SET exit_state = 1;
        END IF;
    END WHILE;

    SET result = v_film_name;

    CLOSE cursor1;
END@
```

Процедура, която приема истинското име на актьора и създава курсор с филмите и имената на актьорите от таблица ACTORS. Чрез цикъл се обхожда тази информация и ако бъде открито въведеното име това ще доведе до прекратяване на цикъла и като изходен параметър ще бъде записан филма, в който е участвал този актьор.

```
--CALL PROCEDURE_3
CALL FN71890.real_name('Daniel Radcliffe', ?)@
```

Резултат от процедурата:

Properties SQL Results ✕				
Name	Type	Data type	Value	Value (OUT)
REAL_ACTOR_NAME	INPUT	VARCHAR	Daniel Radcliffe	
RESULT	OUTPUT	VARCHAR		Harry Potter