

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ **ФАКУЛЬТЕТ**
ПРИКЛАДНОЙ МАТЕМАТИКИ – ПРОЦЕССОВ УПРАВЛЕНИЯ

РУСАКОВ АЛЕКСЕЙ ВЛАДИМИРОВИЧ

Курсовая работа

ФИЛЬМЫ
(FILMS)

Направление 010302
Прикладная математика и информатика

Преподаватель:

Филиппов Р.О.

Санкт-Петербург

2017

Содержание

Глава 1 Схема.....	3
Глава 2 Описание базы данных.....	5
Глава 3 Легкие запросы.....	10
Глава 4 Средние запросы.....	13
Глава 5 Сложные запросы.....	15
Ссылка на репозиторий:	https://github.com/kr0nverk/a.rusakov

Глава 1 Схема

Здесь представлена структура базы данных «Фильмы».

Movies

- id
- name
- genre_id
- country_id
- release_date (дата релиза)
- directors_id
- rating_id
- box_office (сборы в мире)

Directors

- id
- name
- born_date
- death_date
- country_id
- information

Actors

- id
- name
- born_date
- death_date
- country_id
- height
- complexion
- information

Country

- id
- name

Genre

- id
- name

Rating

- id
- IMDb

Movies_actors_relations

- movies_id
- actors_id

Глава 2 Описание базы данных

База данных «Films» описывает фильмы, режиссёров и актёров. Также она содержит таблицы стран и жанров.

«Films» предназначена в основном для киноманов, которые хотят посмотреть много хороших фильмов. В базе представлены также статистические данные, которые помогут отыскать самые лучшие фильмы.

Описание взаимоотношений объектов

У каждого фильма обязательно должен быть режиссёр и в нём должны сниматься актёры. Фильмы снимают в какой-то стране, они выходят в релиз определённой датой. У каждого фильма есть множество рейтингов и кассовый сбор по миру. Режиссёры и актёры рождаются и умирают, в определённой стране. Так же у актёров есть рост и цвет кожи, для удобства подбора актёрского состава по базе данных. Ещё у актёров и режиссёров есть личная информация. Каждый фильм имеет определённый жанр.

Таблица актёров и таблица фильмов соединяется отношением m:m так как в каждом фильме снимается много актёров и они сами снимаются в нескольких фильмах. Таблица фильмов и таблица режиссёров соединяется отношением m:1 так как у фильма один режиссёр, а снять он может много фильмов.

Movies

В этой таблице представлены наиболее интересные характеристики фильмов.

Id PRIMARY KEY

Использован тип integer.

Name

(Название фильма). Название фильма не может повторяться и отсутствовать в таблице. Оно может содержать слова, начинающиеся как с заглавных, так и со строчных букв (ограничения на написание отсутствуют, кроме размера: не более 50 символов).

Genre_id

В данном поле записано id жанра фильма и прикреплена ссылка (внешний ключ) к отдельной таблице с типами жанров.

Country_id

В данном поле записано id страны в которой был выпущен фильм и прикреплен ссылка (внешний ключ) к отдельной таблице со странами.

Release_date

Это поле показывает когда вышел фильм (дату релиза). На дату наложено ограничение больше чем 1800-01-01.

Directors_id

В данном поле записано id режиссёра который режиссировал фильм и прикреплен ссылка (внешний ключ) к отдельной таблице с ними.

Rating_id

В данном поле записано id рейтинга этого фильм и прикреплен ссылка (внешний ключ) к отдельной таблице с ними.

Box_office

Здесь записана прибыль от проката фильма по миру. На прибыль наложено ограничение ≥ 0 .

Directors

Эта таблица описывает режиссёров. В ней вы можете найти интересную информацию о них.

***Id* PRIMARY KEY**

Использован тип integer.

Name

(Имя режиссёра) Имя режиссёра не может повторяться и отсутствовать в таблице. Оно может содержать слова, начинающиеся как с заглавных, так и со строчных букв (ограничения на написание отсутствуют, кроме размера: не более 50 символов).

Born_date

Дата рождения режиссёра. На дату наложено ограничение больше чем 1600-01-01.

Death_date

Дата смерти режиссёра. На дату наложено ограничение больше чем 1600-01-01.

Country_id

В данном поле записано id страны в которой родился режиссёр и прикреплен ссылка (внешний ключ) к отдельной таблице со странами.

Information

Некоторая информация о каждом режиссёре, не поддающаяся систематическому распределению. Ограничение на размер в 1000 символов.

Country

Эта таблица содержит все возможные страны в которых был выпущен фильм, родился режиссёр или актёр.

***Id* PRIMARY KEY**

Использован тип integer.

Name

(Название страны) Название страны не может повторяться и отсутствовать в таблице. Оно может содержать слова, начинающиеся как с заглавных, так и со строчных букв (ограничения на написание отсутствуют, кроме размера: не более 20 символов).

Actors

Эта таблица описывает актеров. В ней вы можете найти интересную информацию о них.

***Id* PRIMARY KEY**

Использован тип integer.

Name

(Имя актёра) Имя актёра не может повторяться и отсутствовать в таблице. Оно может содержать слова, начинающиеся как с заглавных, так и со строчных букв (ограничения на написание отсутствуют, кроме размера: не более 50 символов).

Born_date

Дата рождения актёра. На дату наложено ограничение больше чем 1800-01-01.

Death_date

Дата смерти актёра. На дату наложено ограничение больше чем 1800-01-01.

Country_id

В данном поле записано id страны в которой родился актёр и прикреплена ссылка (внешний ключ) к отдельной таблице со странами.

Height

Здесь указан рост каждого актёра. Он нужен для подбора актерского состава. На рост наложено ограничение на не отрицательность.

Complexion

Здесь указан цвет кожи актёра. Он нужен для подбора актерского состава.

Information

Некоторая информация о каждом актёре, не поддающаяся систематическому распределению. Ограничение на размер в 1000 символов.

Genre

Эта таблица с различными жанрами.

***Id* PRIMARY KEY**

Использован тип integer.

Name

(Название жанра) Название жанра не может повторяться и отсутствовать в таблице. Оно может содержать слова, начинающиеся как с заглавных, так и со строчных букв (ограничения на написание отсутствуют, кроме размера: не более 20 символов).

Rating

Таблица с рейтингами.

***Id* PRIMARY KEY**

Использован тип integer.

IMDb

Рейтинг IMDb, взят с сайта кинопоиск. На рейтинг наложено ограничение на не отрицательность.

Movies actors relations

Эта таблица отношения актеров и фильмов.

Movies_id

В данном поле записано id фильма и прикреплена ссылка (внешний ключ) к отдельной таблице со странами.

Actors_id

В данном поле записано id актёра и прикреплена ссылка (внешний ключ) к отдельной таблице со странами.

Глава 3 Легкие запросы

1 запрос

Выбрать режиссёра, дату рождения и дату смерти, имя которого равно заданному значению.

```
SELECT d.name, d.born_date, d.death_date
FROM
    directors d
WHERE
    d.name = 'Sidney Lumet';
```

Необходимость:

Мы собираемся изучить биографию одного режиссёра, и для этого нам нужны его годы жизни.

Пояснения:

Результат запроса выведет три столбца. В первом будет имя режиссёра, во втором его дату рождения, а в третьем дату смерти если она есть.

Оптимизация

Был добавлен индекс:

```
CREATE INDEX ON directors(name text_pattern_ops);
```

2 запрос

Выбрать название фильма, дату релиза и кассовый сбор выпущенный после заданной даты и сбор которого больше заданного, вывести первый 10.

```
SELECT m.name, m.release_date, m.box_office
FROM
    movies m
WHERE
    m.release_date > '2000-01-01'
    AND
    m.box_office > 100
ORDER BY
    m.release_date DESC NULLS LAST
LIMIT 10;
```

Необходимость:

Мы ищем фильмы вышедшие после определённой даты и хотим посмотреть кассовый сбор этих фильмов.

Пояснения:

По запросу будут выведены три столбца. В первом будет название фильма, во втором его дату выхода, а в третьем кассовый сбор.

Оптимизация

Был добавлен индекс:

```
CREATE INDEX ON movies(release_date, box_office);
```

3 запрос

Выбрать имя актера его рост, который задан, и комплекцию и отсортировать по имени.

```
SELECT a.name, a.height, a.complexion
FROM
    actors a
WHERE
    a.height>=1.70
    AND
    a.height<1.90
ORDER BY
    a.name;
```

Необходимость:

Вы хотите снять фильм, и вам нужны актеры. А этот запрос поможет вам выбрать среди всех актеров, выбрав по росту и по комплекции.

Пояснения:

По запросу будут выведены три столбца. В первом будет имя актера, во втором его рост, а в третьем цвет его кожи.

Оптимизация

Был добавлен индекс:

```
CREATE INDEX ON actors(height);
```

4 запрос

Выбрать фильмы и сборы, где сборы больше заданного, отсортировать по длине названия и показать первые 10.

```
SELECT m.name, m.box_office
FROM
    movies m
WHERE
    m.box_office > 100
ORDER BY
    char_length(m.name) DESC NULLS LAST
LIMIT 10;
```

Необходимость:

Вам нужны фильмы с самым длинным названием и чтоб сбор был больше заданного.

Пояснения:

По запросу будут выведены два столбца. В первом будет название фильма, а во втором его сбор.

Оптимизации

Был добавлен индекс:

```
CREATE INDEX ON movies(box_office);
```

Глава 4 Средние запросы

5 запрос

Выбрать фильм, режисера и страну, страна которой задана.

```
SELECT movies.name AS movie, directors.name AS directors,  
country.name AS country  
FROM movies INNER JOIN country  
ON (movies.country_id = country.id) AND  
(country.name = 'New Zealand')  
INNER JOIN directors  
ON (movies.directors_id = directors.id);
```

Необходимость:

Вам нужны фильмы и их режиссёры, вышедшие в определённой стране.

Пояснения:

По запросу будут выведены три столбца. В первом будет название фильма, во втором его режиссёр, а в третьем страна.

6 запрос

Выбрать фильм, жанр, дату релиза и имя режиссёра, жанр которого задан и дата релиза больше заданной.

```
SELECT movies.name AS movie, genre.name AS genre,  
movies.release_date AS release_date, directors.name AS  
directors  
FROM movies INNER JOIN genre  
ON (movies.genre_id = genre.id) AND (genre.name  
= 'Thriller')  
INNER JOIN directors  
ON (movies.directors_id = directors.id)  
WHERE movies.release_date >= '1994-01-01';
```

Необходимость:

Нам нужны фильмы с определённым жанром вышедшие не позднее какого-то числа.

Пояснения:

По запросу будут выведены четыре столбца. В первом будет название фильма, во втором заданный жанр, в третьем дата релиза, а в четвёртом имя режиссёра.

7 запрос

Выбрать фильм, режиссёра, возраст режиссера, когда он выпустил фильм, страну. Сортировать по возрасту.

```
SELECT movies.name AS movie, directors.name AS directors,  
age(movies.release_date, directors.born_date) AS age,  
country.name AS country  
FROM movies INNER JOIN directors  
ON (movies.directors_id = directors.id)  
INNER JOIN country  
ON (movies.country_id = country.id) AND  
(country.name = 'New Zealand')  
ORDER BY  
age(movies.release_date, directors.born_date) DESC  
NULLS LAST;
```

Необходимость:

Мы проводим исследование на тему, в каком возрасте режиссёры больше всего выпускают фильмы.

Пояснения:

По запросу будут выведены четыре столбца. В первом будет название фильма, во втором его режиссёр, в третьем возраст режиссёра во время выпуска фильма, а в четвертом страну.

Глава 5 Сложные запросы

8 запрос

Вывести все страны, количество фильмов снятых в них и эти фильмы, через запятую, сортировать по имени страны.

```
SELECT cou.name AS country, count(mov.name) AS count_movies,
string_agg(mov.name, ', ' ORDER BY mov.name) AS movies
FROM (SELECT id, name
      FROM country) AS cou
INNER JOIN
(SELECT name, country_id
 FROM movies) AS mov
ON mov.country_id = cou.id
GROUP BY cou.name;
```

Необходимость:

Мы проводим исследование в какой стране больше всего выпускают фильмы.

Пояснения:

По запросу будут выведены три столбца. В первом будет название страны, во втором количество фильмов выпущенных в стране, а в третьем столбце все фильмы перечисленные через запятую.

9 запрос

Вывести всех режисеров и фильмы которые они сняли, с суммой прибыли этих фильмов, сортировать по имени режисёра.

```
SELECT dir.name AS directors, string_agg(mov.name, ', '
ORDER BY mov.name) AS movies, sum(mov.box_office) AS
sum_box_office
FROM (SELECT id, name
      FROM directors) AS dir
INNER JOIN
(SELECT name, directors_id, box_office
 FROM movies) AS mov ON mov.directors_id =
dir.id
GROUP BY dir.name;
```

Необходимость:

Мы пытаемся найти самых богатых режиссеров мира.

Пояснения:

По запросу будут выведены три столбца. В первом будет имя режиссёра, во втором какие фильмы он снял, а в третьем вся прибыль этих фильмов.

10 запрос

Вывести все фильмы и актеров снимающихся в них, перечислив через запятую, сортировать по имени фильма.

```
SELECT mov.name AS movies, string_agg(act.name, ', ' ORDER
BY act.name) AS actors
  FROM (SELECT movies_id, actors_id
        FROM movies_actors_relations) AS movact
 INNER JOIN
  (SELECT id, name
   FROM movies) AS mov ON movact.movies_id =
mov.id
 INNER JOIN
  (SELECT id, name
   FROM actors) AS act ON movact.actors_id =
act.id
GROUP BY mov.name;
```

Необходимость:

Мы смотрим актерский состав фильмов.

Пояснения:

По запросу будут выведены два столбца. В первом будет название фильма, а во втором все актеры снимающиеся в нём.