

Санкт-Петербургский государственный университет
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ – ПРОЦЕССОВ УПРАВЛЕНИЯ

Никитина Анастасия Александровна

Курсовая работа

Список
(LIST)

Направление 010302
Прикладная математика, фундаментальная информатика и
программирование

Преподаватель

Филиппов Р.О.

Санкт-Петербург
2017

Содержание

Глава 1 Схема.....	3
Глава 2 Описание базы данных.....	6
Глава 3 Легкие запросы.....	13
Глава 4 Средние запросы.....	14
Глава 5 Сложные запросы.....	15
Приложения.....	16

Глава 1 Схема

Структура базы данных «Список».

products

- id
- typename
- name
- genre
- year

type

- id
- typename

genres

- id
- genre

countries

- id
- country

directors

- id
- name
- surname
- nationality
- year
- sex
- alive_or_dead

director_product

- id
- product_id
- director_id

actors

- id
- name
- surname
- nationality
- year
- sex
- alive_or_dead

actor_product

- id
- actor_id
- product_id

advisers

- id
- name
- surname

adviser_product

- id
- product_id
- adviser_id

Глава 2

База данных «Список» содержит список произведений кинематографа, включающих в себя сериалы и фильмы. Также она хранит таблицы, описывающие основных актеров и режиссеров, связанных с соответствующими им произведениями, и таблицы, являющиеся вспомогательными для основных таблиц.

Эта база данных проектировалась таким образом, что имеет некую ценность в основном только для ее создателя и, может быть, для людей, чьи интересы в искусстве неким образом совпадают с интересами создателя.

Описание взаимоотношений объектов

Все произведения кинематографа можно классифицировать по жанрам, как, например, характеризуют картины по их принадлежности к различным направлениям (стилям) в искусстве (например, знаменитую «Звездную ночь» В. Ван Гога относят к постимпрессионизму, а «Чтение» П. Пикассо – к сюрреализму).

У фильмов также существует отличительная черта, которая позволяет понять, чем он будет выделяться в общих чертах только при упоминании фамилии его режиссера (правда, это правило работает в основном с более-менее широко известными режиссерами), как, например, при звучании фамилии Тарантино зритель уже предполагает, что его ожидают нелинейная структура повествования и черты постмодернизма.

Также можно учитывать, что для некоторых людей (например, создателя данной базы данных) имеет значение, какие актеры снимаются в главных ролях. Например, существует вероятность, что фильм, где в главной роли снимается один актер, будет в приоритете в списке на просмотр, чем фильм, где в главных ролях неизвестные или, что еще меньше играет в пользу фильма, неудачно известные (в плане оценки актерской игры) актеры.

Не стоит забывать, что страна производства (что обычно имеет прямую зависимость с национальностью режиссера) имеет свое влияние на атмосферу произведения и расставленные в нем акценты, впрочем, ту же роль играет и год выпуска.

products

В данной таблице представлена основная информация о произведении.

id PRIMARY KEY

Использован тип integer. В дальнейшем это будет подразумеваться по умолчанию.

typename

(Тип произведения). Прикреплена ссылка (внешний ключ) к отдельной таблице *type*. Произведение относится либо к фильмам, либо к сериалам (ограничение NOT NULL).

name

(Название фильма/сериала). Название должно быть уникально, ограничение NOT NULL (должно обязательно присутствовать в таблице). Нет ограничений на написание, кроме размера: максимальное количество символов 255.

genre

Основной жанр произведения, прикреплена ссылка (внешний ключ) к отдельной вспомогательной таблице *genres*.

year

Год выпуска/премьеры.

directors

В данной таблице представлена основная информация о режиссерах, чьи фильмы присутствуют в таблице products.

id PRIMARY KEY

name

Имя режиссера, строка без ограничений (разве что на присутствие (NOT NULL) и размер – 255 символов).

surname

Фамилия, по сути полностью повторяет структуру поля name.

nationality

Указывает национальную принадлежность, прикреплена ссылка (внешний ключ) к отдельной вспомогательной таблице *countries*.

year

Год рождения.

sex

Биологический пол (или гендерное самоопределение, если режиссер выразил свою позицию).

alive_or_dead

Статус в мире живых, предполагается, что хранит два значения, что непосредственно указаны в названии поля – “alive”, “dead”, посему ограничение на 6 символов (еще 1 на всякий случай).

actors

В данной таблице представлена основная информация об актерах, кто снимался в фильмах или сериалах, присутствующих в таблице products.

id PRIMARY KEY

name

Имя актера, строка без ограничений (разве что на присутствие (NOT NULL) и размер – 255 символов).

surname

Фамилия, по сути полностью повторяет структуру поля name.

nationality

Указывает национальную принадлежность, прикреплена ссылка (внешний ключ) к отдельной вспомогательной таблице *countries*.

year

Год рождения.

sex

Биологический пол (или гендерное самоопределение, если была выражена позиция актера).

alive_or_dead

Статус в мире живых, предполагается, что хранит два значения, что непосредственно указаны в названии поля – “alive”, “dead”, посему ограничение на 6 символов.

advisers

Знакомые проектировщику данной базы данных личности, приложившие недюжинные старания по ознакомлению уже упомянутого создателя с основными шедеврами кинематографа или недавно вышедшими, но уже оцененными новинками.

id PRIMARY KEY

name

Имя знакомого/друга, строка без ограничений (разве что на присутствие (NOT NULL) и размер – 255 символов).

surname

Фамилия, по сути полностью повторяет структуру поля name.

countries

Вспомогательная таблица, предоставляющая значения на ссылающиеся на нее таблицы actors и directors.

id PRIMARY KEY

country

Название страны, строка без ограничений (разве что на присутствие (NOT NULL) и размер – 255 символов).

genres

Список жанров, относящихся и к фильмам, и к сериалам.

id PRIMARY KEY

genre

Собственно, название жанра, строка без ограничений (разве что на присутствие (NOT NULL) и размер – 255 символов).

type

Производит классификацию произведений по двум типам – фильм или сериал.

id PRIMARY KEY

typename

Название типа, строка без ограничений (разве что на присутствие (NOT NULL) и размер – 255 символов).

director_product

Соотносит режиссера со снятым им фильмом. Реализует отношение 1:m, так как у фильма только один главный режиссер.

id PRIMARY KEY

product_id

Ссылается на таблицу *products* (внешний ключ).

director_id

Ссылается на таблицу *directors* (внешний ключ).

actor_product

Указывает, в каком фильме снимался какой актер, реализует таким образом отношение m:m, так как в одном фильме может быть несколько интересующих заполняющего базу данных актеров, и один актер может сниматься в нескольких фильмах.

id PRIMARY KEY

product_id

Ссылается на таблицу *products* (внешний ключ).

actor_id

Ссылается на таблицу *actors* (внешний ключ).

adviser_product

Распределяет произведения по советовавшим их людям, чтобы потом воздать должное (в смысле, не забыть, кого надо благодарить) после просмотра.

id PRIMARY KEY

product_id

Ссылается на таблицу *products* (внешний ключ).

adviser_id

Ссылается на таблицу *advisers* (внешний ключ).

Глава 3 Легкие запросы

1. Вывести имена и фамилии актеров (мужского пола), чей год рождения позже 1970го года, отсортировать по году рождения.

```
SELECT a.name, a.surname, a.year FROM actors AS a WHERE  
a.year > 1970 AND a.sex = 'male' ORDER BY a.year DESC NULLS  
LAST;
```

2. Вывести всех режиссеров, то есть их имена и фамилии, кто на данный момент еще присутствует в мире живых. Отсортировать в алфавитном порядке.

```
SELECT d.name, d.surname FROM directors AS d WHERE  
d.alive_or_dead = 'alive' ORDER BY d.surname, d.name;
```

3. Вывести все фильмы, чей год премьеры позднее 2005 года, сортировка по году выпуска.

```
SELECT p.name, p.year FROM products AS p WHERE year > 2005  
AND p.typename = 1 ORDER BY p.year DESC NULLS LAST;
```

4. Вывести всех актрис в алфавитном порядке.

```
SELECT a.name, a.surname, a.year FROM actors AS a WHERE  
a.sex = 'female' ORDER BY a.surname, a.name;
```

Глава 4 Средние запросы

1. Вывести полные имена американских актеров.

```
SELECT actors.name || ' ' || actors.surname AS FullName FROM  
actors INNER JOIN countries ON actors.nationality = countries.id  
WHERE countries.country = 'USA';
```

2. Вывести названия сериалов и фильмов, жанр которых относится к комедии, отсортировав в алфавитном порядке.

```
SELECT products.name, products.year FROM products INNER  
JOIN genres ON products.genre = genres.id WHERE genres.genre  
= 'Comedy' ORDER BY products.name;
```

3. Вывести названия сериалов, присутствующих в таблице products, отсортировав их по году выпуска.

```
SELECT products.name, products.year FROM products INNER  
JOIN type ON products.typename = type.id WHERE  
type.typename = 'Serial' ORDER BY products.year;
```

Глава 5 Сложные запросы

1. Вывести имена и фамилии актеров, которые появлялись в фильмах или сериалах из списка более одного раза.

```
SELECT name || ' ' || surname AS "Full Name" FROM (SELECT
actors.name, actors.surname, count(*) AS "times" FROM
actor_product JOIN actors ON actors.id = actor_product.actor_id
JOIN products ON products.id = actor_product.product_id
GROUP BY actors.name, actors.surname) AS actors WHERE times
> 1;
```

2. Вывести названия комедий, присутствующих в списке и чей год выпуска позднее 2000 года, и полные имена актеров, снимавшихся в них.

```
SELECT products.name, products.year, actors.name || ' ' ||
actors.surname AS "Full Name" FROM products JOIN genres ON
products.genre = genres.id JOIN actor_product ON
actor_product.product_id = products.id JOIN actors ON actors.id
= actor_product.actor_id WHERE (genres.genre = 'Comedy' AND
products.year > 2000);
```

3. Узнать, какой жанр повторяется более 4 раз, и вывести названия фильмов/сериалов с этим жанром.

```
SELECT genres.genre, name FROM genres JOIN products ON
products.genre = genres.id WHERE genres.genre = (SELECT genre
FROM (SELECT genres.genre, count(*) FROM genres RIGHT
JOIN products ON genres.id = products.genre GROUP BY
genres.genre) AS new WHERE count > 4);
```

Приложения

➤ <https://github.com/NeKyon/database.git>

