**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное образовательное учреждение**

**высшего профессионального образования**

**Национальный исследовательский университет**

**«Высшая школа экономики»**

**Московский институт электроники и математики**

**Департамент прикладной математики**

**Образовательная программа «Компьютерная безопасность»**

**Отчёт по практической работе по дисциплине «СУБД»**

**Домашняя практическая работа № 2**

**«Проектирование баз данных»**

**Вариант «Фильмотека»**

**Выполнил:**

**Астраханцев Роман Геннадьевич**

**СКБ-171**

**Проверил:**

**Профессор, Белов А.В.**

**Москва-2020**

**Постановка задачи**

Модель «Фильмотека» должна содержать информацию о кинофильмах, актерах, киностудиях и контрактах между актерами и киностудиями

Требуется построить E-R диаграмму средствами AllFusion Data Modeler (ERwin), определить нормальную форму и сформулировать запросы на языке реляционной алгебры следующие запросы:

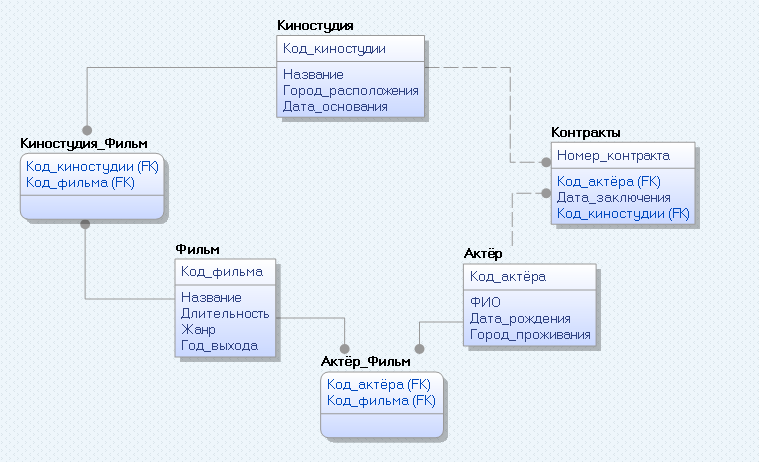
* получить список всех актеров, снимающихся на заданной киностудии;
* получить список кинофильмов, в которых снимаются те же актеры, что и в заданном кинофильме;
* получить пары (Ф.И.О. актера, № контракта), занятых в фильмах, выпущенных на заданной киностудии;
* получить список киностудий, в которых были сняты фильмы по заданной тематике и в заданном году;
* получить список актеров, у которых имеется контракт с киностудией, расположенной в том же городе, в каком проживают эти актеры;
* получить список кинофильмов, в которых заданный актер не снялся ни разу.

*Выбранная среда реализации – Microsoft SQL Server*

**Создание БД**

С помощью мастера была создана база данных в MS SQL сервере с простой моделью восстановления.

С помощью Erwin была спроектирована логическая модель базы данных (см. рис 1), а также сгенерирован код запроса на языке T-SQL (рис. 2), который позже был отредактирован и выполнен на SQL сервере.

  
Рис.1 Логическая модель БД «Фильмотека»

Итоговый код создания базы данных и результат его выполнения приведён в приложении 1. Его функция заключается только создании таблиц данных. Необходимы ограничения и триггеры приведены в дальнейшем пункте.

**Ограничения и триггеры**

Для сохранения целостности данных были созданы ограничения на данные. Так, например, год выхода фильма быть больше 1800 (то есть после появления технологии записи видеозаписей) и быть не сильно больше текущего года (это сделано для анонсов фильмов). Подобные ограничения были сделаны для всех полей.

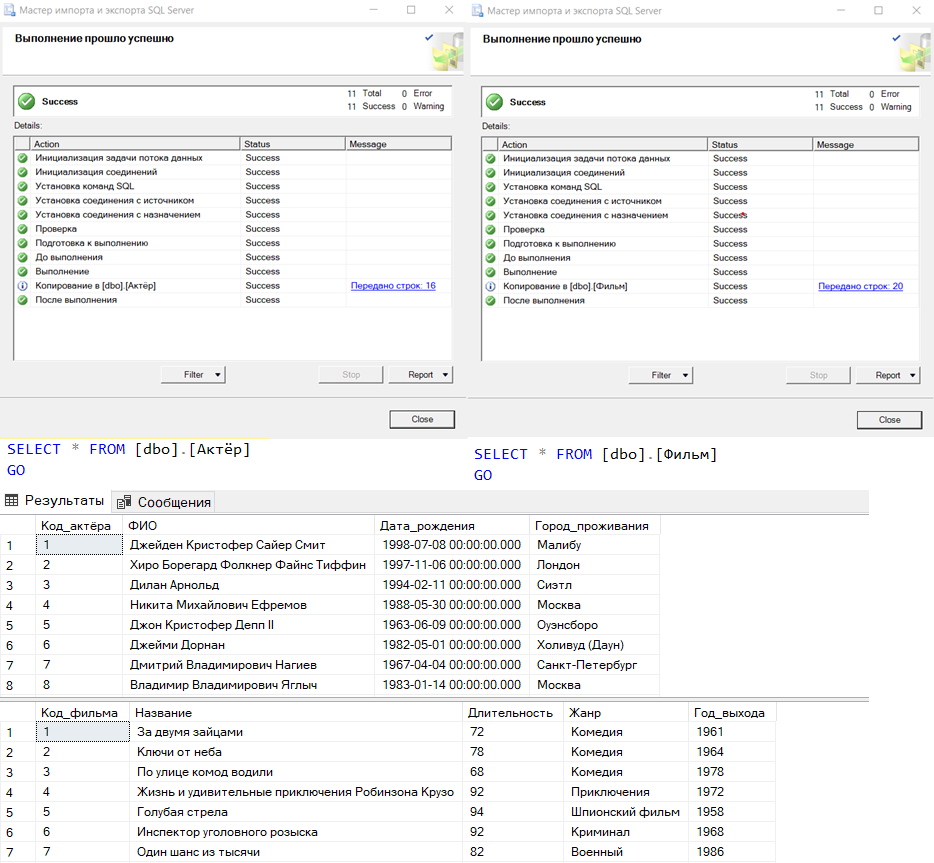
Так же были предусмотрены случаи удаления данных и их изменения. Так, например, в таблицах выражающие отношения многие ко многим при удаления родительских сущностей данные выставляются в NULL. Это сделано для того, чтобы сохранить информацию о, например, количестве актёров в том или ином фильме.

В связи с этим, были созданы триггеры, которые удаляют записи, в котором не осталось информации. Например, если в таблице Актёр\_Фильм оба поля Код\_актёра и Код\_фильма стали NULL, то такая запись удаляется. Код ограничений и триггеров находится в приложении 2.

**Импорт и экспорт данных.**

1. **Используя мастер импорта и экспорта.**

В связи с этим, были созданы триггеры, которые удаляют записи, в котором не осталось информации. Например, если в таблице Актёр\_Фильм оба поля Код\_актёра и Код\_фильма стали NULL, то такая запись удаляется. Код ограничений и триггеров находится в приложении 2.

  
Рис.2 Импорт данных через мастера.

1. **Используя язык SQL.**

В связи с этим, были созданы триггеры, которые удаляют записи, в котором не осталось информации. Например, если в таблице Актёр\_Фильм оба поля Код\_актёра и Код\_фильма стали NULL, то такая запись удаляется. Код ограничений и триггеров находится в приложении 2.

**Запросы к базе на языке SQL.**

1. получить список всех актеров, снимающихся на заданной киностудии;

(«Контракты» JOIN («Киностудия» WHERE Название=<заданная киностудия>){Код\_киностудии}){Код\_актёра} JOIN «Актёр»

WITH [Коды\_киностудий] AS (

SELECT [Код\_киностудии] FROM [Киностудия] WHERE [Название]=<заданная киностудия>

)

WITH [Коды\_актёров] AS (

SELECT [Код\_актёра] FROM [Коды\_киностудий] INNER JOIN [Контракты] ON [Коды\_киностудий].[Код\_киностудии]=[Контракты].[Код\_киностудии]

)

SELECT \* FROM [Коды\_актёров] INNER JOIN [Актёр] ON [Коды\_актёров].[Код\_актёра]=[Актёр].[Код\_актёра]

1. получить список кинофильмов, в которых снимаются те же актеры, что и в заданном кинофильме;

(«Актёр\_Фильм» JOIN («Фильм» WHERE Название=<заданный кинофильм>){Код\_актёра}){Код\_фильма} JOIN «Фильм»

WITH [Заданный\_фильм] AS (

SELECT [Код\_фильма] FROM [Фильм] WHERE [Название]=<заданный кинофильм>

)

WITH [Все\_актёры\_фильма] AS (

SELECT [Код\_актёра] FROM [Заданный\_фильм] AS [Зф] INNER JOIN [Актёр\_Фильм] AS [АФ] ON [Зф].[Код\_актёра]=[АФ].[Код\_актёра]

)

WITH [Коды\_искомых\_фильмов] AS (

SELECT [Код\_фильма] FROM [Заданный\_фильм] AS [Зф] INNER JOIN [Все\_актёры\_фильма] AS [А] INNER JOIN [Актёр\_Фильм] AS [АФ] ON [А].[Код\_актёра]=[АФ].[Код\_актёра]

SELECT \* FROM [Фильм] AS [Ф] INNER JOIN [Коды\_искомых\_фильмов] AS [КИФ] ON [КИФ].[Код\_фильма] = [Ф].[Код\_фильма]

1. получить пары (Ф.И.О. актера, № контракта), занятых в фильмах, выпущенных на заданной киностудии;

(((«Киностудия\_Фильм» JOIN («Киностудия» WHERE Название=<заданная киностудия>){Код\_киностудии}){Код\_фильма} JOIN «Актёр\_Фильм»){Код\_актёра} JOIN «Актёр» JOIN «Контракты»){ФИО, Номер контракта}

1. получить список киностудий, в которых были сняты фильмы по заданной тематике и в заданном году;

((«Фильм» WHERE Жанр=<заданная тематика> AND Год\_выхода=<заданный год>){Код фильма} JOIN «Киностудия\_Фильм»){Код\_киностудии} JOIN «Киностудия»

1. получить список актеров, у которых имеется контракт с киностудией, расположенной в том же городе, в каком проживают эти актеры;

(((«Актёр» JOIN «Контракты»{ALL BUT Дата заключения}) JOIN «Киностудия»{Код\_киностудии, Город расположения}) WHERE Город проживания = Город расположения){ALL BUT Код киностудии, Город расположения}

1. получить список кинофильмов, в которых заданный актер не снялся ни разу.

(«Фильм»{Код\_фильма} MINUS («Фильм»{Код\_фильма} DEVIDED BY («Актёры» WHERE ФИО=<заданный актёр>){Код\_актёра} PER «Актёр\_Фильм»)) JOIN «Фильм»

**Настройка ограничений целостности**

В фильмотеке необходимо хранить следующие данные:

* информация об актёрах;
* информация о фильмах;
* информация о киностудиях;
* список всех контрактов;

**Описание выходной информации**

В соответствии с предметной областью были созданы следующие сущности:

* «Актёр» - хранится информация обо всех актёрах;
* «Фильм» - хранится информация обо всех фильмах;
* «Киностудия» - хранится информация обо всех киностудиях;
* «Контракты» - хранится информация о всех заключённых контрактах между актёрами и киностудиями;

**Описание сущностей и атрибутов**

1. Сущность «Актёр» содержит:

* Код актёра – уникальный номер актёра;
* ФИО – фамилия, имя и отчество актёра;
* Дата рождения – дата рождения актёра;
* Город проживания – город проживания актёра;

2. Сущность «Фильм» содержит:

* Код фильма – уникальный номер фильма;
* Название – название фильма;
* Жанр – жанр фильма;
* Год выхода – год выхода фильма;
* Код киностудии – идентификатор киностудии, которая выпустила данный фильм

3. Сущность «Киностудия» содержит:

* Код киностудии – уникальный номер киностудии;
* Название – название киностудии;
* Город расположения – город расположения киностудии;
* Дата основания – дата основания киностудии;

4. Сущность «Контракты» содержит:

* Номер контракта – уникальный юридический номер контракта;
* Код актёра – идентификатор актёра, который участвует в контракте;
* Дата заключения – дата заключения контракта;
* Код киностудии – идентификатор киностудии, которая участвует в контракте;

**Описание связей**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Родительская сущность | Дочерняя сущность | Тип связи |
| 1 | Киностудия | Фильм | M:N |
| 2 | Фильм | Актёр | M:N |
| 3 | Киностудия | Контракты | 1:М |
| 4 | Актёр | Контракты | 1:M |

У сущностей «Киностудия» и «Фильм» отношение «многие ко многим», так как над одинм фильмом может работать несколько киностудий, а, в свою очередь, одна киностудия может принимать участие в создании нескольких фильмов.

У сущностей «Фильм» и «Актёр» отношение «многие ко многим», так как у одного фильма может быть несколько актёров, а, в свою очередь, один актёр может принимать участие в нескольких фильмах за свою карьеру.

У сущностей «Киностудия» и «Контракты» отношение «один ко многим», так как у одной киностудии может быть несколько контрактов, а, в свою очередь, конкретный контракт заключатся только с одной киностудией.

У сущностей «Актёр» и «Контракты» отношение «один ко многим», так как у одного актёра может быть несколько контрактов, а, в свою очередь, конкретный контракт заключатся только с одним актёром.

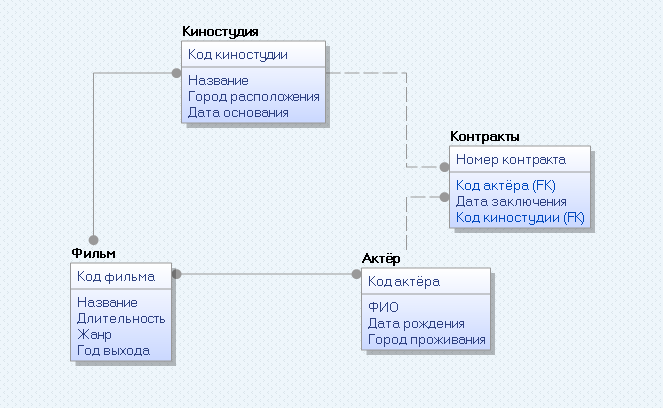


Рис.1 Семантическая модель БД «Фильмотека»

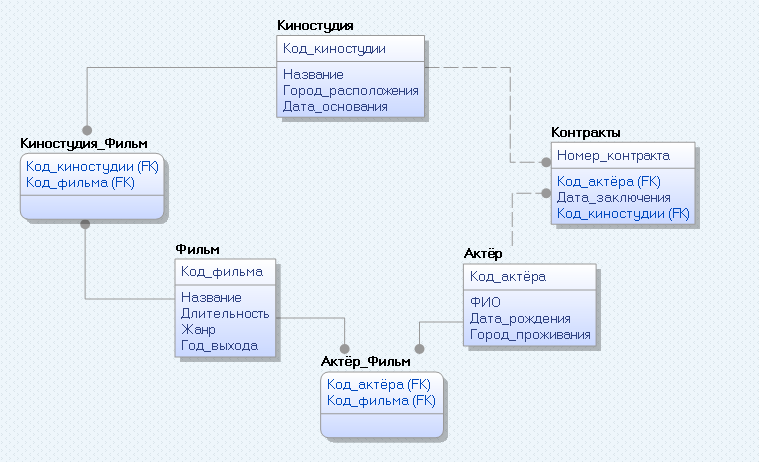


Рис.2 Логическая модель БД «Фильмотека»

Как видно из рисунков выше, данная модель находится в нормальной форме Бойса-Кодда.

**На языке реляционной алгебры построить запросы:**

1. получить список всех актеров, снимающихся на заданной киностудии;

(«Контракты» JOIN («Киностудия» WHERE Название=<заданная киностудия>){Код\_киностудии}){Код\_актёра} JOIN «Актёр»

1. получить список кинофильмов, в которых снимаются те же актеры, что и в заданном кинофильме;

(«Актёр\_Фильм» JOIN («Фильм» WHERE Название=<заданный кинофильм>){Код\_актёра}){Код\_фильма} JOIN «Фильм»

1. получить пары (Ф.И.О. актера, № контракта), занятых в фильмах, выпущенных на заданной киностудии;

(((«Киностудия\_Фильм» JOIN («Киностудия» WHERE Название=<заданная киностудия>){Код\_киностудии}){Код\_фильма} JOIN «Актёр\_Фильм»){Код\_актёра} JOIN «Актёр» JOIN «Контракты»){ФИО, Номер контракта}

1. получить список киностудий, в которых были сняты фильмы по заданной тематике и в заданном году;

((«Фильм» WHERE Жанр=<заданная тематика> AND Год\_выхода=<заданный год>){Код фильма} JOIN «Киностудия\_Фильм»){Код\_киностудии} JOIN «Киностудия»

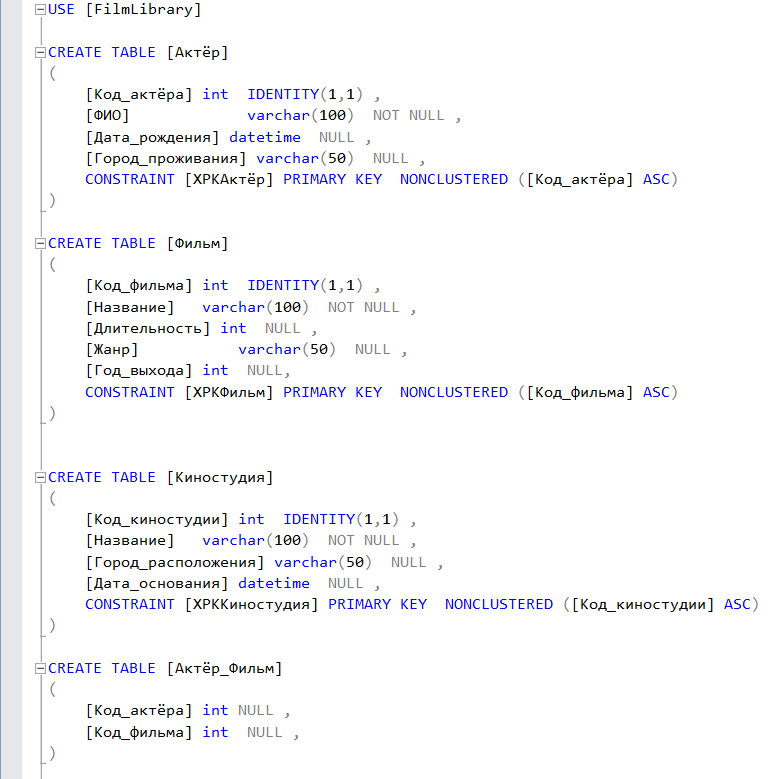
1. получить список актеров, у которых имеется контракт с киностудией, расположенной в том же городе, в каком проживают эти актеры;

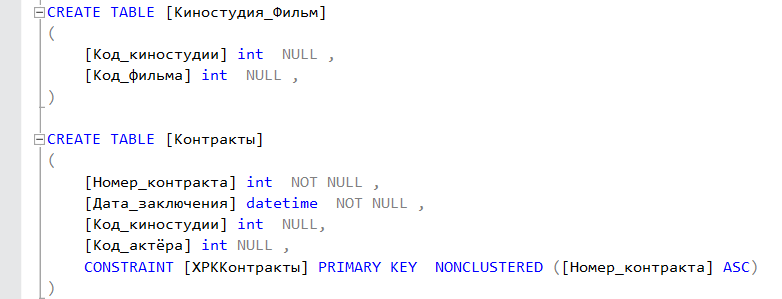
(((«Актёр» JOIN «Контракты»{ALL BUT Дата заключения}) JOIN «Киностудия»{Код\_киностудии, Город расположения}) WHERE Город проживания = Город расположения){ALL BUT Код киностудии, Город расположения}

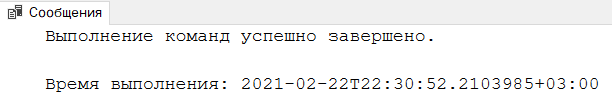
1. получить список кинофильмов, в которых заданный актер не снялся ни разу.

(«Фильм»{Код\_фильма} MINUS («Фильм»{Код\_фильма} DEVIDED BY («Актёры» WHERE ФИО=<заданный актёр>){Код\_актёра} PER «Актёр\_Фильм»)) JOIN «Фильм»

**Приложение 1**







**Приложение 2**

