**Образовательная экосистема**

**GeekBrains**

**КУРСОВАЯ РАБОТА (КР)**

|  |  |  |
| --- | --- | --- |
| **Направление:** | *Geek University Python-разработка* | |
|  | | (шифр и название направления подготовки/специальности) | |
| **Курс:** | *Основы реляционных баз данных. MySQL* | | |
|  | (наименование образовательной программы) | | |

|  |  |
| --- | --- |
| **Тема КР:** | ***БД «Кинопоиск». Superlite version*** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Выполнил:** |  |  | ***Колокольчиков В.Д.*** |
|  | (подпись) |  | (ФИО выпускника) |

**Москва, 2021 г.**

# **Оглавление**

[Оглавление 2](#_Toc67252385)

[Ведение 3](#_Toc67252386)

[Работы с сервером. Подключение к БД 4](#_Toc67252387)

[Сбор данных для БД 5](#_Toc67252388)

[Собственно БД 7](#_Toc67252389)

[Представления. Процедуры и запросы 9](#_Toc67252390)

[Представления 9](#_Toc67252391)

[Процедуры 9](#_Toc67252392)

[Запросы 11](#_Toc67252393)

[Заключение 13](#_Toc67252394)

[Список литературы 14](#_Toc67252395)

# Ведение

Темой курсовой работы в рамках курса основ по изучению реляционных баз данных (MySQL) была выбрана эко-система кино-портала «Кинопоиск». Курсовая писалась не правах рекламы, а для учебных целей. Данные о фильмах брались из открытых источников.

Для написание данной КР использовалось следующее ПО:

1. MySQL – version 8.0
2. Dbeaver
3. Python – version 3.9.0
4. PyCharm
5. Oracle VM
6. Ubuntu Desktop (для сервера)

Библиотеки, задействованные в ходе работы:

1. Для парсинга сайтов – Requests и BS4 (BeautifulSoup)
2. Для работы с таблицами Excel – xlrd, xlwt
3. Для удаленной работы с БД посредством Python – pymysql и sshtunnel

Шаблоны кодов, а также основные запросы, триггеры и т.п. документы связанные с MySQL будут находится в репозитории.

# Работы с сервером. Подключение к БД

Подключение к базе данных производилось из двух программ DBeaver, как основа для редакции структуры БД и создания запросов, и PyCharm для парсинга данных с последующей их загрузкой на сервер (Рис. 1). Подключался через SSH-подключение. Предварительно пришлось отредактировать конфигурационный файл – .my.cnf .



Рисунок 1. Подключение к БД с помощью python

# Сбор данных для БД

Данные собирались с двух ресурсов: генератора данных (<http://www.generatedata.com/#generator>) и сайта Википедия (<https://ru.wikipedia.org/wiki/250_лучших_фильмов_по_версии_IMDb>) для основной информации по кино и его создателям. Для основы БД взял таблицу топ-250 фильмов. Таблицу из википедии забрал с помощью специальных функций в Excel (Рис. 2). Оставил для работы только 100 строк.

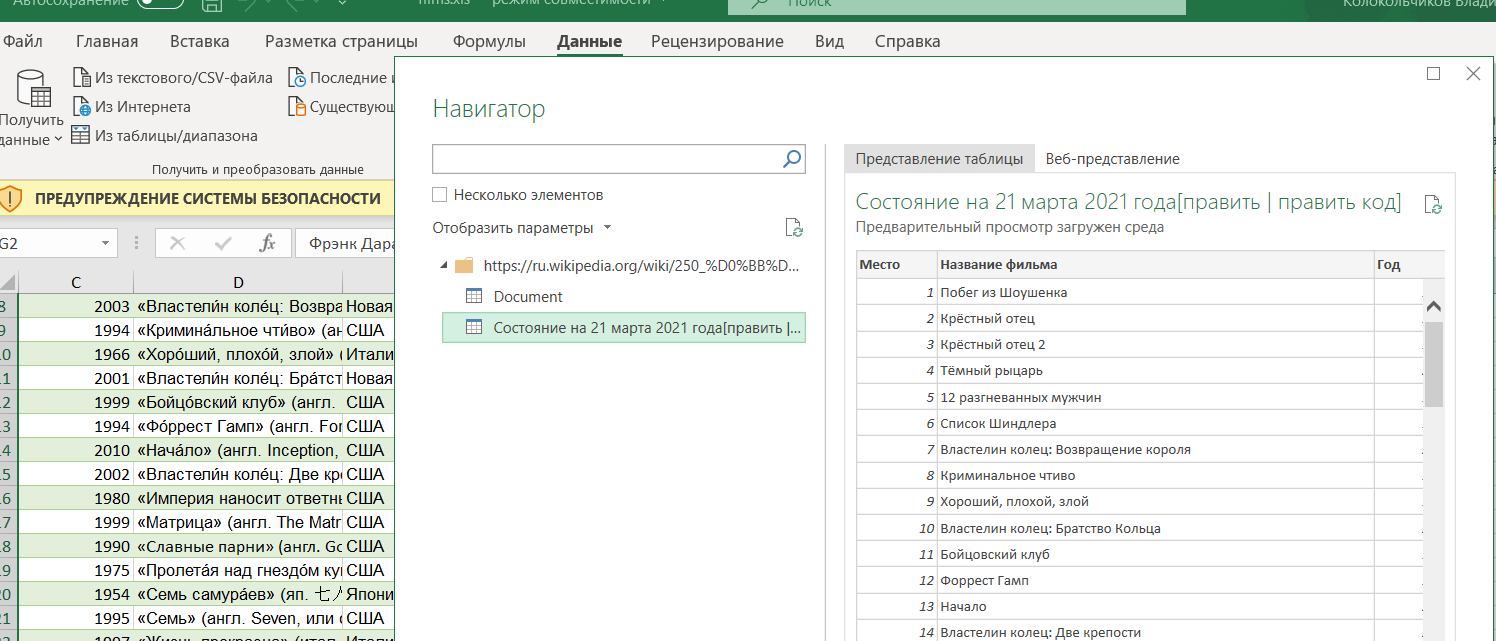


Рисунок 2. Основа для личной БД

Те данные, которых не было в полученной таблице, пришлось собирать с помощью парсинга страниц в Википедии. Настроил программу так, чтобы она переходила самостоятельно по нужным мне ссылкам и забирала нужную для меня информацию (имена режиссеров, сценаристов, актеров, страна выпуска, описание к фильмам). Пример кода для сбора описаний будет доступен в репозитории.

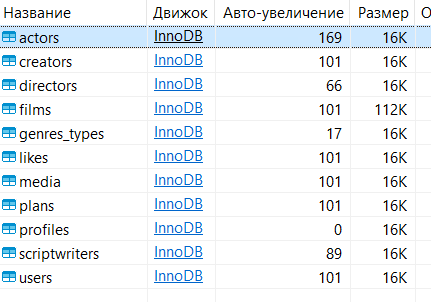
Для увязки данных была написана также программа, которая сначала преобразовала массив данных во множество, а после в словарь (ключ – значение). На основе последнего выстраивалась связь таблиц между собой.

После данных подгружались также с помощью python и библиотеки pymysql (Рис. 3).



Рисунок 3. Закомментированная часть добавляла данные для таблицы films. Часть вне комментария исправляла дату релиза (точнее изначальное его отсутствие)

# Собственно БД

В ходе работы над БД kinopoisk было создано 11 таблиц (Рис. 4). Все таблицы созданы на движке InnoDB. БД в настоящее время имеет следующую структуру:

После добавления всех данных таблицу была выстроена связь между ними, а также проиндексированные столбцы, которые не попали в число уникальных или не стали ключами. Для таких таблиц как profiles, directors, actors, scriptwriter и media были созданы триггеры либо редакции данных (как в первых 4), либо для автозаполнения (как это было с таблицей медиа). Все триггеры и индексы будут представлены в отдельном файле с соответствующими им названиями.

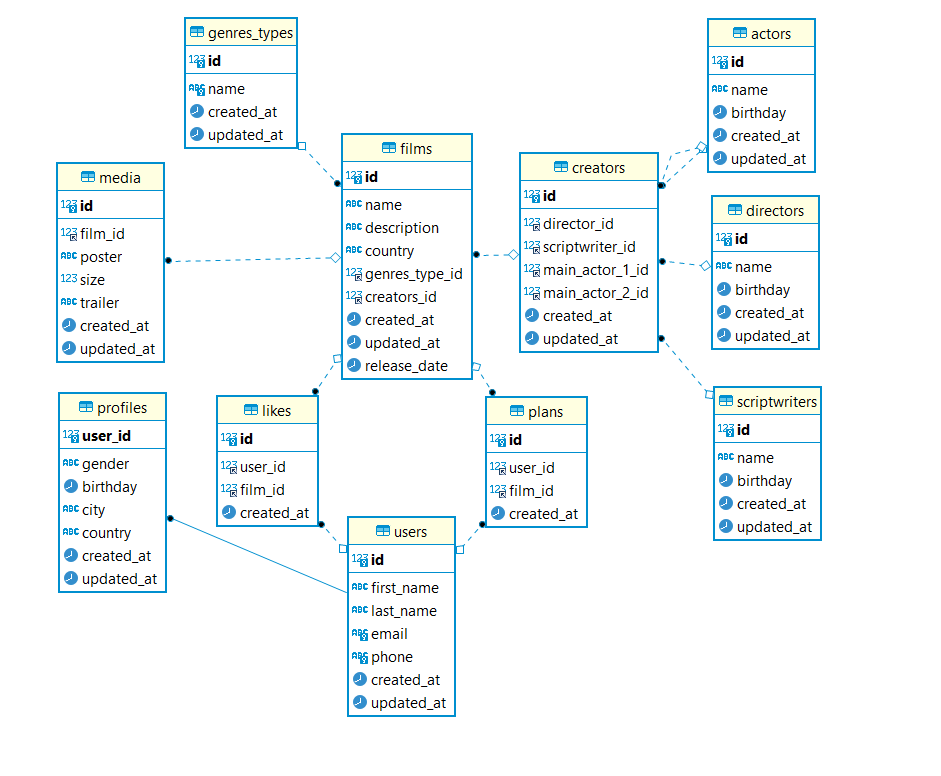
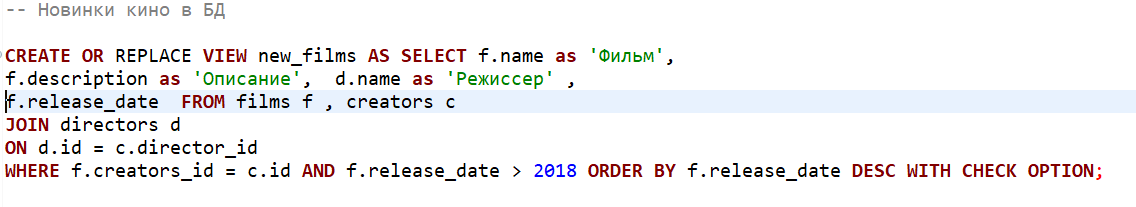


Рисунок 4. ERDiagrama для БД

# Представления. Процедуры и запросы

## Представления

Для БД kinopoisk были созданы два представления, которые предоставляли пользователям информацию о последних фильмах (новинках кино) и информацию о создателях. Ознакомиться с ними можно в отдельном файле с соответствующим названием. Пример такого запроса с новинками кино представлен на рисунке 5.

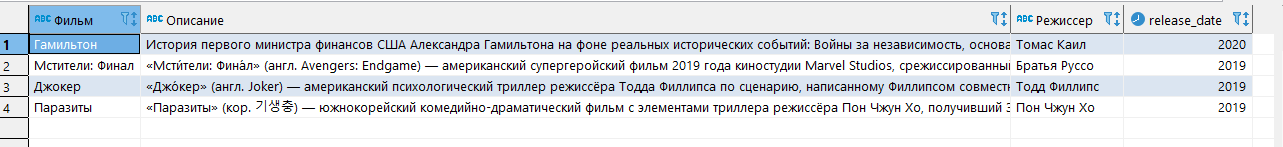


Рисунок 5. Новинки кино (2021 год пока еще не в топе =) )

## Процедуры

В процедурах были реализованы рекомендации по фильмам для пользователей. Первая была основана на жанрах тех, фильмов, которые были «лайкнуты», исключая посмотренные из списка вывода. Вторая была построена аналогичным образом, но уже по режиссерам. Ознакомиться с ними подробно можно в отдельном файле. Примеры хранимой процедуры представлен на рисунке 6.

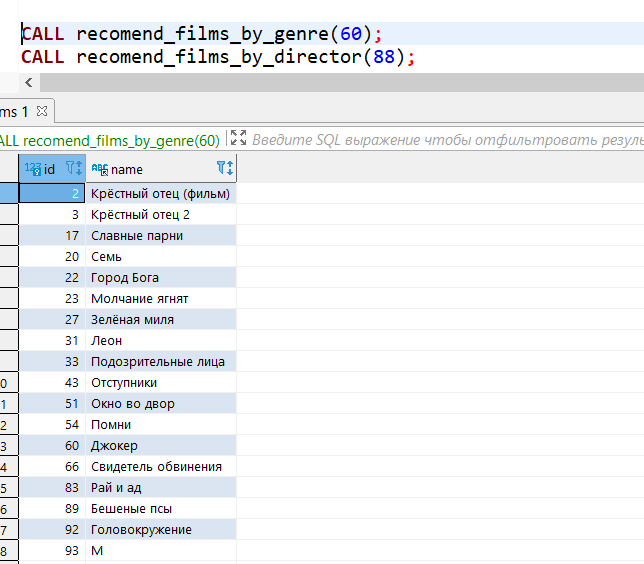
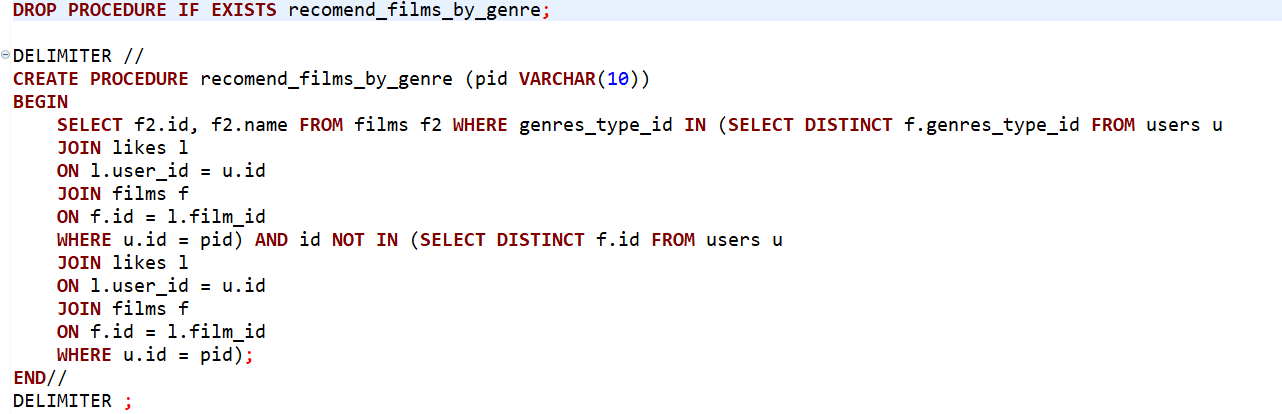


Рисунок 6. Вывод хранимой процедуры. Рекомендации для пользователя с ид 60.

## Запросы

Были сформированы наиболее интересные запросы (на мой взгляд и что пришло в голову) разных типов: вложенные, с JOIN и оконные. Со всеми запросами можно ознакомиться в соответствующем файле. Примеры некоторых будут представлены на рисунках 7 и 8.

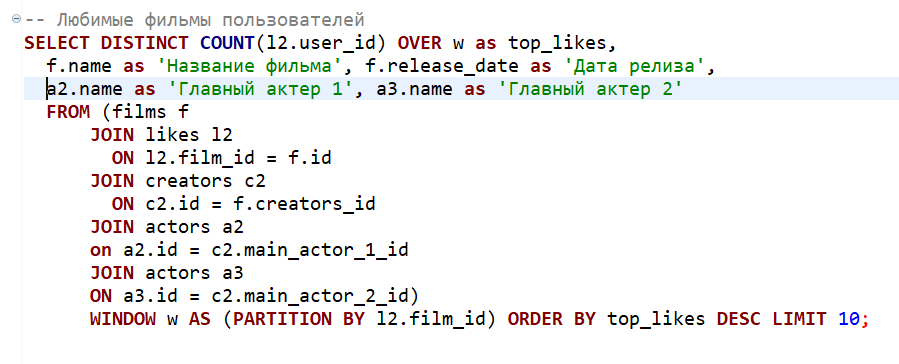
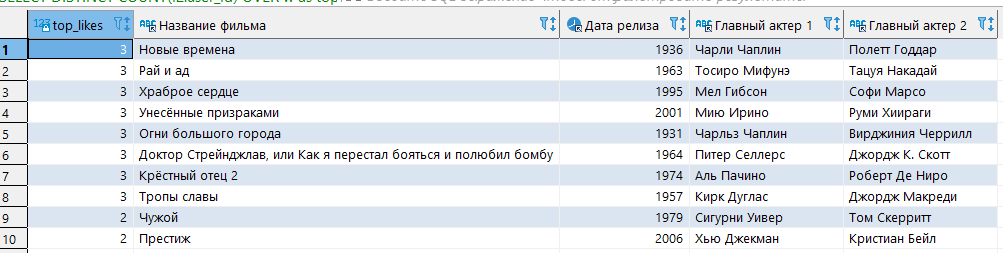


Рисунок 7. Любимые фильмы пользователей

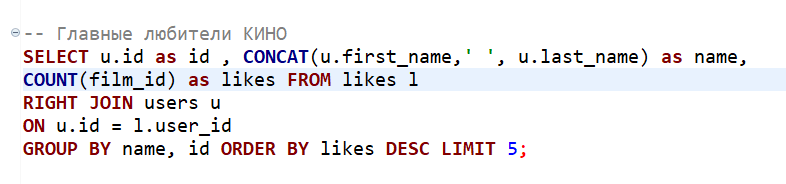
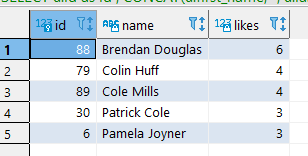


Рисунок 8. Главные любители кино

# Заключение

Была проделана большая работа по сбору данных и по её дальнейшей обработке. Лично для меня данная работа принесла огромное количество опыта, который, как мне кажется, может пригодиться для дальнейшей работы. Возможно, таблица получилось не такой хорошей и корректной, однако во всяком случае я доволен ей. Прежде всего тем, что смог применить знания Python для обработки данных что заняло у меня много дней и ночей. В общем сделал все возможное, чтобы сделать курсовую работу более интересное и менее шаблонной.

Ждем оценки, результатов и мнений. Спасибо за знания!

# Список литературы

1. Портал GeekBrains
2. Википедия
3. CodeRoad
4. Proglib
5. Habr
6. Tproger