Оглавление

[Описание предметной области. Назначение базы данных, предполагаемые запросы 2](#_Toc480629167)

[Допущения, ограничения. 3](#_Toc480629168)

[Ограничения: 3](#_Toc480629169)

[Допущения: 3](#_Toc480629170)

[Логическая модель данных 4](#_Toc480629171)

[Построение E-R диаграммы 4](#_Toc480629172)

[Выделение сущностей. Свойства сущностей и первичные ключи 4](#_Toc480629173)

[Связи между сущностями. 4](#_Toc480629174)

[Переход к физической модели. Внешние ключи, таблицы-проекции 4](#_Toc480629175)

[E-R диаграмма 4](#_Toc480629176)

[Нормализация базы данных 4](#_Toc480629177)

[Универсальное отношение. 4](#_Toc480629178)

[Функциональные зависимости. Первичный ключ универсального отношения 5](#_Toc480629179)

[Переход к 2НФ 5](#_Toc480629180)

[Переход к 3 НФ 6](#_Toc480629181)

[Физическая модель данных. 8](#_Toc480629182)

[Создание структуры базы данных. Таблицы, ключи, ограничения, индексы 8](#_Toc480629183)

[Заполнение таблиц. Скрипты запросов 10](#_Toc480629184)

[Дамп базы данных 20](#_Toc480629185)

[Запросы к базе данных 21](#_Toc480629186)

Описание предметной области. Назначение базы данных, предполагаемые запросы

Данная база данных предназначена для удобного хранения и использования информации о фильмах: поиска нужной информации, добавления, исправления. Также в ней представлена информация об актерах и создателях фильмов, о жанрах, кроме того – рецензии (отзывы) к фильмам.

Представленная информация:

* об актёрах, режиссёрах, продюсерах и сценаристах: фамилия, имя, год рождения и родная страна.
* Также представлена информация о фильмах: название, год, рейтинг, страна, которая выпустила фильм, жанр, описание жанра, актёры, снимавшиеся в этом фильме, их роли, ведущий режиссёр, ведущий сценарист, ведущий продюсер (при этом указывается только один ведущий специалист на каждую должность (а актёров может и должно бы быть несколько), а также фильм соответствует только одному жанру).
* О награждениях Оскар для актёров: год награждения, номинация, кто награждён, за какой фильм.
* О рецензиях о фильме: название, автор, текст, оценка автора
* О жанрах: название жанра и его описание

По данной базе данных, зная название фильма, можно найти всю информацию о нем: о создателях этого фильма (режиссёре, продюсере и сценаристе) со всеми их доступными личными данными (год рождения, страна, возраст), об актёрах, игравших в этом фильме (с аналогичными личными сведениями), о стране, в которой снимался фильм. О жанре и о годе выпуска фильма, а также о том, получил ли кто-нибудь из актёров, игравших в этом фильме, награду (здесь – Оскар) за игру в этом фильме.

Запросы также предполагаются направленными на получение все или части перечисленной выше информации.

Допущения, ограничения.

Ограничения:

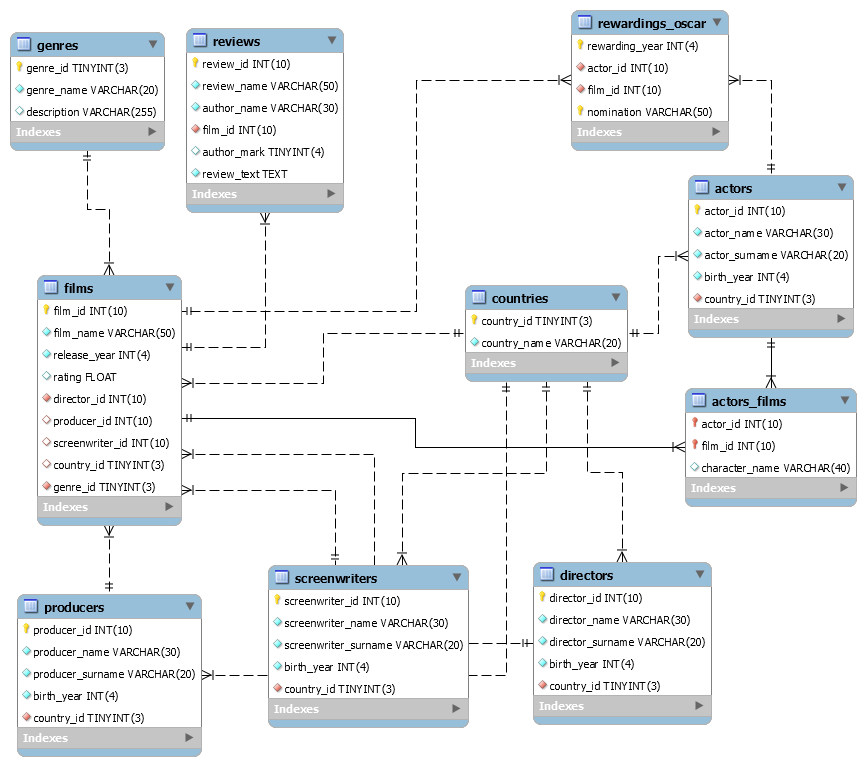
* В одном фильме только один ведущий продюсер, один ведущий режиссёр и один ведущий сценарист;
* Один фильм соответствует только одному жанру;
* Оскаром награждаются только актёры;
* Одно награждение определяется годом и номинацией;
* В одном фильме может сниматься множество актёров;
* Об одном фильме у одного автора один отзыв;
* Стран не больше, чем 255;
* Жанров не больше, чем 255;
* Оценка от автора рецензии - целое число от 0 до 10;
* Рейтинг фильма – число с плавающей точкой от 0 до 10;

Допущения:

* Описание жанра не обязано присутствовать
* Указываются только 3-5 актеров на фильм (считаемые главными в этом фильме)

Логическая модель данных

Построение E-R диаграммы



Нормализация базы данных

Универсальное отношение.

Ru(actor\_surname, actor\_name, actor\_birth\_year, actor\_country, actor\_role director\_surname, director\_name, director\_birth\_year, director\_country, producer\_surname, producer\_name, producer\_birth\_year, producer\_country, screenwriter\_surname, screenwriter\_name, screenwriter\_birth\_year, screenwriter\_country, film\_name, release\_year, rating, film\_country, genre\_name, genre\_description, Oscar\_rewarding\_year, Oscar\_nomination, review\_name, review\_author, review\_autor\_mark, review\_text).

В данном отношении довольно трудно определить некоторые первичные ключи, возможны неоднозначности.

Введём идентификаторы:   
actor\_id, director\_id, producer\_id, screenwriter\_id, genre\_id, film\_id, review\_id для большей точности и удобства, в этом случае

Получим следующее универсальное отношение:   
Ru2(actor\_surname, actor\_name, actor\_birth\_year, actor\_country, actor\_role director\_surname, director\_name, director\_birth\_year, director\_country, producer\_surname, producer\_name, producer\_birth\_year, producer\_country, screenwriter\_surname, screenwriter\_name, screenwriter\_birth\_year, screenwriter\_country, film\_name, release\_year, rating, film\_country, genre\_name, genre\_description, Oscar\_rewarding\_year, Oscar\_nomination, review\_name, review\_author, review\_autor\_mark, review\_text, actor\_id, director\_id, producer\_id, screenwriter\_id, genre\_id, film\_id, review\_id)

Функциональные зависимости. Первичный ключ универсального отношения

Первичный ключ:   
{ actor\_id, film\_id, review\_id, Oscar\_rewarding\_year, Oscar\_nomination }

Распишем зависимости:

* actor\_id => actor\_name, actor\_surname, actor\_birth\_year, actor\_country
* film\_id => film\_name, release\_year, rating, film\_country, director\_id, producer\_id, producer\_country, screenwriter\_id, genre\_id, director\_surname, director\_name, director\_birth\_year, director\_country, producer\_surname, producer\_name, producer\_birth\_year, producer\_country, creenwriter\_surname, screenwriter\_name, screenwriter\_birth\_year, screenwriter\_country, genre\_name, genre\_discription
* { Oscar\_rewarding\_year, Oscar\_nomination } => film\_id, actor\_id
* { film\_id, actor\_id } => actor\_role
* review\_id => review\_author, review\_name, review\_author\_mark, review\_text, film\_id

Переход к 2НФ

Тогда можно перейти ко второй нормальной , выделив неключевые атрибуты, зависящие от части ключа в отдельное отношение:

R0(actor\_id, film\_id, review\_id, Oscar\_rewarding\_year, Oscar\_nomination)

R1(actor\_id, actor\_name, actor\_surname, actor\_birth\_year, actor\_country)

R2(film\_id, film\_name, release\_year, rating, film\_country, director\_id, director\_surname, director\_name, director\_birth\_year, director\_country, producer\_id, producer\_surname, producer\_name, producer\_birth\_year, producer\_country, screenwriter\_id, screenwriter\_surname, screenwriter\_name, screenwriter\_birth\_year, screenwriter\_country, genre\_id, genre\_name, genre\_discription)

R3(Oscar\_rewarding\_year, Oscar\_nomination, film\_id, actor\_id)  
(Здесь отдельное отношение, так как не все актёры и не за все фильмы получат Оскар)

R4(film\_id, actor\_id, actor\_role)

R5(review\_id, review\_author, review\_name, review\_author\_mark, review\_text, film\_id)  
(Здесь отдельное отношение, так как не на каждый фильм будет написана рецензия)

Данные отношения находятся во второй нормальной форме.

Отношение R0 сводится к отношениям R3 и R5.

Переход к 3 НФ

Перейдём к третьей нормальной форме, выделив в отдельные отношения зависимости неключевых атрибтов от неключевых.

Выделяем следующие транзитивные зависимости:

* director\_id => director\_surname, director\_name, director\_birth\_year, director\_country
* producer\_id => producer\_surname, producer\_name, producer\_birth\_year, producer\_country
* screenwriter\_id => screenwriter\_surname, screenwriter\_name, screenwriter\_birth\_year, screenwriter\_country
* genre\_id => genre\_name, genre\_discription

Тогда отношени R2 декомпозируется на 5 отношения: R2, R6, R7, R8, R9.

Получим:

R1(actor\_id, actor\_name, actor\_surname, actor\_birth\_year, actor\_country)

R2(film\_id, film\_name, release\_year, rating, film\_country, director\_id, producer\_id, screenwriter\_id, genre\_id)

R3(Oscar\_rewarding\_year, Oscar\_nomination, film\_id, actor\_id)

R4(film\_id, actor\_id, actor\_role)

R5(review\_id, review\_author, review\_name, review\_author\_mark, review\_text, film\_id)

R6(director\_id, director\_surname, director\_name, director\_birth\_year, director\_country)

R7(producer\_id, producer\_surname, producer\_name, producer\_birth\_year, producer\_country)

R8(screenwriter\_id, screenwriter\_surname, screenwriter\_name, screenwriter\_birth\_year, screenwriter\_country)

R9(genre\_id, genre\_name, genre\_discription)

Однако, почти в каждом отношении есть атрибут, указывающий на страну, но набор возможных стран постоянный и общий для всех этих отношений. Страну можно вынести в отдельное отношение (так экономится память и появляется возможность добавить какую-то информацию, касающуюся только стран), тогда конечный набор отношений имеет вид:

R1(actor\_id, actor\_name, actor\_surname, actor\_birth\_year, country\_id)

R2(film\_id, film\_name, release\_year, rating, film\_country, director\_id, producer\_id, screenwriter\_id, genre\_id)

R3(Oscar\_rewarding\_year, Oscar\_nomination, film\_id, actor\_id)

R4(film\_id, actor\_id, actor\_role)

R5(review\_id, review\_author, review\_name, review\_author\_mark, review\_text, film\_id)

R6(director\_id, director\_surname, director\_name, director\_birth\_year, country\_id)

R7(producer\_id, producer\_surname, producer\_name, producer\_birth\_year, country\_id)

R8(screenwriter\_id, screenwriter\_surname, screenwriter\_name, screenwriter\_birth\_year, country\_id)

R9(genre\_id, genre\_name, genre\_discription)

R10(country\_id, country\_name).

Физическая модель данных.

Создание структуры базы данных. Таблицы, ключи, ограничения, индексы

create table if not exists countries(

country\_id tinyint unsigned primary key auto\_increment,

country\_name varchar(20) not null unique

)CHARACTER SET=UTF8;

create table if not exists actors(

actor\_id int unsigned primary key,

actor\_name varchar(30) not null,

actor\_surname varchar(20) not null,

birth\_year int(4) not null,

country\_id tinyint unsigned not null,

foreign key (country\_id) references countries(country\_id)

)CHARSET=UTF8;

create table if not exists genres(

genre\_id tinyint unsigned primary key,

genre\_name varchar(20) not null,

description varchar(255)

)CHARSET=UTF8;

create table if not exists directors(

director\_id int unsigned primary key,

director\_name varchar(30) not null,

director\_surname varchar(20) not null,

birth\_year int(4) not null,

country\_id tinyint unsigned not null,

foreign key (country\_id) references countries(country\_id)

)CHARSET=UTF8;

create table if not exists producers(

producer\_id int unsigned primary key,

producer\_name varchar(30) not null,

producer\_surname varchar(20) not null,

birth\_year int(4) not null,

country\_id tinyint unsigned not null,

foreign key (country\_id) references countries(country\_id)

)CHARSET=UTF8;

create table if not exists screenwriters(

screenwriter\_id int unsigned primary key,

screenwriter\_name varchar(30) not null,

screenwriter\_surname varchar(20) not null,

birth\_year int(4) not null,

country\_id tinyint unsigned not null,

foreign key (country\_id) references countries(country\_id)

)CHARSET=UTF8;

create table if not exists films(

film\_id int unsigned primary key,

film\_name varchar(50) not null,

release\_year int(4) not null,

rating float,

director\_id int unsigned not null,

producer\_id int unsigned,

screenwriter\_id int unsigned,

country\_id tinyint unsigned,

genre\_id tinyint unsigned not null,

foreign key (director\_id) references directors (director\_id),

foreign key (producer\_id) references producers (producer\_id),

foreign key (screenwriter\_id) references screenwriters (screenwriter\_id),

foreign key (country\_id) references countries (country\_id),

foreign key (genre\_id) references genres (genre\_id)

)CHARSET=UTF8;

create table if not exists reviews(

review\_id int unsigned primary key auto\_increment,

review\_name varchar(50) not null,

author\_name varchar(30) not null,

film\_id int unsigned not null,

author\_mark tinyint,

review\_text text not null,

foreign key (film\_id) references films (film\_id)

)CHARSET=UTF8;

create table if not exists actors\_films(

actor\_id int unsigned,

film\_id int unsigned,

character\_name varchar(40),

foreign key (film\_id) references films (film\_id),

foreign key (actor\_id) references actors (actor\_id),

primary key (actor\_id, film\_id)

)CHARSET=UTF8;

create table if not exists rewardings\_Oscar(

rewarding\_year int(4),

actor\_id int unsigned not null,

film\_id int unsigned not null,

nomination varchar(50),

foreign key (actor\_id) references actors (actor\_id),

foreign key (film\_id) references films (film\_id),

primary key (rewarding\_year, nomination)

)CHARSET=UTF8;

Заполнение таблиц. Скрипты запросов

insert ignore into countries

(country\_name) values

("США"),

("Новая Зеландия"),

("Великобритания"),

("Австралия"),

("Франция"),

("Канада"),

("Мексика"),

("Ирландия"),

("Неизвестно"),

("Уганда"),

("Германия"),

("Ливан"),

("Нигерия");

insert ignore into directors

(director\_id, director\_name, director\_surname, birth\_year, country\_id) values

(32383, "Питер", "Джексон", 1961, 2),

(41477, "Кристофер", "Нолан", 1970, 3),

(24817, "Крис", "Коламбус", 1958, 1),

(27977, "Джеймс", "Кэмерон", 1954, 6),

(563585, "Алехандро", "Иньярриту", 1963, 7),

(8442, "Уэс", "Крэйвен", 1939, 1),

(30870, "Гор", "Вербински", 1964, 1),

(29361, "Скотт", "Хикс", 1953, 10),

(30966, "Роберт", "Родригес", 1968, 1),

(23330, "Лана", "Вачовски", 1965, 1),

(22260, "Стивен", "Спилберг", 1946, 1),

(1190481, "Филлида", "Ллойд", 1957, 3),

(10346, "Стивен", "Долдри", 1961, 3),

(31799, "Роберт", "Бентон", 1932, 1),

(104876, "Стюарт", "Орм", 1954, 3),

(462603, "Зак", "Снайдер", 1966, 1),

(64418, "Дэвид", "Карсон", 0, 3),

(30189, "Кэтрин", "Хардвик", 1955, 1);

insert ignore into producers

(producer\_id, producer\_name, producer\_surname, birth\_year, country\_id) values

(32383, "Питер", "Джексон", 1961, 2),

(41477, "Кристофер", "Нолан", 1970, 3),

(23449, "Дэвид", "Хейман", 1961, 3),

(27977, "Джеймс", "Кэмерон", 1954, 6),

(6784, "Стив", "Голин", 1955 , 1),

(3023, "Роберт", "Шей", 1939, 1),

(10207, "Джерри", "Брукхаймер", 1943, 1),

(90143, "Джейн", "Скотт", 1945, 3),

(30966, "Роберт", "Родригес", 1968, 1),

(1790, "Джоэл", "Силвер", 1952, 1),

(20397, "Фрэнк", "Маршалл", 1946, 1),

(15755, "Дэмиэн", "Джонс", 1964, 3),

(65747, "Донна", "Джильотти", 1995, 9),

(14197, "Стэнли Р.", "Джаффе", 1940, 1),

(8288, "Ральф", "Уинтер", 1952, 1),

(8523, "Марк", "Кантон", 1949, 1),

(78017, "Брайан", "Истмэн", 1949, 9),

(37859, "Леонардо", "ДиКаприо", 1974, 1),

(539928, "Жан", "Бленкин", 0, 9);

insert ignore into screenwriters

(screenwriter\_id, screenwriter\_name, screenwriter\_surname, birth\_year, country\_id) values

(32385, "Фрэнсис", "Уолш", 1959, 2),

(41477, "Кристофер", "Нолан", 1970, 3),

(10093, "Стивен", "Кловз", 1960, 1),

(27977, "Джеймс", "Кэмерон", 1954, 6),

(734897, "Марк", "Смит", 0, 9),

(8442, "Уэс", "Крэйвен", 1939, 1),

(30871, "Тед", "Эллиот", 1961, 1),

(90096, "Ян", "Сарди", 0, 4),

(30966, "Роберт", "Родригес", 1968, 1),

(23329, "Лилли", "Вачовски", 1967, 1),

(26537, "Лоуренс", "Кэздан", 1949, 1),

(1190793, "Эби", "Морган", 1968, 3),

(36297, "Дэвид", "Хэа", 1947, 3),

(226493, "Джастин", "Хэйс", 0, 9),

(31799, "Роберт", "Бентон", 1932, 1),

(462603, "Зак", "Снайдер", 1966, 1),

(55143, "Саймон", "Мур", 1958, 9),

(1047677, "Дэвид Лесли", "Джонсон", 0, 9);

insert ignore into genres

(genre\_id, genre\_name) values

(5, "Фентези"),

(2, "Фантастика"),

(8, "Драма"),

(4, "Триллер"),

(1, "Ужасы"),

(10, "Приключения"),

(13, "Вестерн"),

(22, "Биография");

insert ignore into films

(film\_id, film\_name, release\_year, rating, director\_id, producer\_id, screenwriter\_id, country\_id, genre\_id) values

(3498, "Властелин колец: Возвращение Короля", 2003, 8.611, 32383, 32383, 32385, 1, 5),

(111543, "Темный рыцарь", 2008, 8.502, 41477, 41477, 41477, 1, 2),

(689, "Гарри Поттер и философский камень", 2001, 8.193, 24817, 23449, 10093, 3, 5),

(2213, "Титаник", 1997, 8.371, 27977, 27977, 27977, 1, 8),

(522941, "Выживший", 2015, 7.814, 563585, 6784, 734897, 1, 4),

(5198, "Кошмар на улице Вязов", 1984, 7.703, 8442, 3023, 8442, 1, 1),

(4374, "Пираты Карибского моря:Проклятие Черной жемчужины", 2003, 8.342, 30870, 10207, 30871, 1, 5),

(16686, "Блеск", 1996, 7.870, 29361, 90143, 90096, 4, 8),

(477764, "Дети шпионов 4D", 2011, 3.695, 30966, 30966, 30966, 1, 2),

(301, "Матрица", 1999, 8.492, 23330, 1790, 23329, 1, 2),

(339, "Индиана Джонс: В поисках утраченного ковчега", 1981, 8.007, 22260, 20397, 26537, 1, 10),

(463647, "Железная леди", 2011, 7.013, 1190481, 15755, 1190793, 3, 22),

(63991, "Пираты Карибского моря: Сундук мертвеца", 2006, 8.098, 30870, 10207, 30871, 1, 5),

(688, "Гарри Поттер и Тайная комната", 2002, 8.030, 24817, 23449, 10093, 3, 5),

(325439, "Чтец", 2008, 7.935, 10346, 65747, 36297, 1, 8),

(461981, "Одинокий рейнджер", 2013, 6.791, 30870, 10207, 226493, 1, 13),

(1902, "Крамер против Крамера", 1979, 8.011, 31799, 14197, 31799, 1, 8),

(6226, "Кукловоды", 1994, 6.438, 104876, 8288, 30871, 1, 1),

(81924, "300 спартанцев", 2007, 7.629, 462603, 8523, 462603, 1, 5),

(77217, "Десятое королевство (сериал)", 1999, 8.457, 64418, 78017, 55143, 3, 5),

(470179, "Красная Шапочка", 2011, 6.126, 30189, 37859, 1047677, 1, 1),

(5908, "Кинг Конг", 2005, 7.573, 32383, 539928, 32385, 2, 8),

(437410, "Темный рыцарь: Возрождение легенды", 2012, 8.144, 41477, 41477, 41477, 1, 2);

insert ignore into actors

(actor\_id, actor\_name, actor\_surname, birth\_year, country\_id) values

(20287, "Элайджа", "Вуд", 1981, 1),

(10779, "Вигго", "Мортенсен", 1958, 1),

(28426, "Шон", "Эстин", 1971, 1),

(21495, "Кристиан", "Бэйл", 1974, 3),

(1183, "Хит", "Леджер", 1979, 4),

(6752, "Аарон", "Экхарт", 1968, 1),

(40778, "Дэниэл", "Рэдклифф", 1989, 3),

(40780, "Руперт", "Гринт", 1988, 3),

(40779, "Эмма", "Уотсон", 1990, 5),

(37859, "Леонардо", "ДиКаприо", 1974, 1),

(16653, "Роберт", "Инглунд", 1947, 1),

(21709, "Кейт", "Уинслет", 1975, 3),

(45019, "Билли", "Зейн", 1966, 1),

(39984, "Том", "Харди", 1977 , 3),

(671252, "Донал", "Глисон", 1983, 8),

(94877, "Хэзер", "Лэнгенкэмп", 1964, 1),

(6245, "Джонни", "Депп", 1963, 1),

(24683, "Джеффри", "Раш", 1951, 4),

(30875, "Орландо", "Блум", 1977, 3),

(7227, "Ноа", "Тейлор", 1969, 3),

(8012, "Армин", "Мюллер-Шталь", 1930, 11),

(21931, "Джессика", "Альба", 1981, 1),

(9278, "Дэнни", "Трехо", 1944, 1),

(1967244, "Роуэн", "Бланчард", 2001, 1),

(7836, "Киану", "Ривз", 1964, 12),

(9838, "Лоренс", "Фишбёрн", 1961, 1),

(6226, "Кэрри-Энн", "Мосс", 1967, 6),

(1491, "Хьюго", "Уивинг", 1960, 13),

(5679, "Харрисон", "Форд", 1942, 1),

(2674, "Карен", "Аллен", 1951, 1),

(141431, "Рональд", "Лейси", 1935, 3),

(23100, "Мэрил", "Стрип", 1949, 1),

(38704, "Джим", "Бродбент", 1949, 3),

(147467, "Сьюзэн", "Браун", 1946, 3),

(24302, "Кира", "Найтли", 1985, 3),

(22670, "Рэйф", "Файнс", 1962, 3),

(459452, "Давид", "Кросс", 1990, 11),

(1121926, "Арми", "Хаммер", 1986, 1),

(13096, "Уильям", "Фихтнер", 1956, 1),

(22234, "Дастин", "Хоффман", 1937, 1),

(22308, "Джейн", "Александр", 1939, 1),

(11362, "Дональд", "Сазерленд", 1935, 6),

(29367, "Эрик", "Тал", 1965, 1),

(26583, "Джули", "Уорнер", 1965, 1),

(12086, "Джерард", "Батлер", 1969, 3),

(659495, "Майкл", "Фассбендер", 1977, 11),

(4810, "Винсент", "Риган", 1965, 3),

(47163, "Дайэнн", "Уист", 1948, 1),

(2412, "Дэниэл", "Лапэйн", 1970, 4),

(21571, "Кимберли", "Уильямс-Пэйсли", 1971, 1),

(477028, "Аманда", "Сайфред", 1985, 1),

(6650, "Гари", "Олдман", 1958, 3),

(973659, "Шайло", "Фернандес", 1985, 1),

(27221, "Наоми", "Уоттс", 1968, 3),

(2523, "Эдриан", "Броуди", 1973, 1),

(23724, "Джек", "Блэк", 1969, 1),

(32383, "Питер", "Джексон", 1961, 2),

(38703, "Энн", "Хэтэуэй", 1982, 1);

insert ignore into actors\_films

(actor\_id, film\_id, character\_name) values

(20287, 3498, "Frodo"),

(10779, 3498, "Aragorn"),

(28426, 3498, "Sam"),

(30875, 3498, "Legolas"),

(21495, 111543, "Bruce Wayne"),

(1183, 111543, "Joker"),

(6752, 111543, "Harvey Dent"),

(40778, 689, "Harry Potter"),

(40780, 689, "Ron Weasley"),

(40779, 689, "Hermione Granger"),

(37859, 2213, "Jack Dawson"),

(21709, 2213, "Rose DeWitt Bukater"),

(45019, 2213, "Cal Hockley"),

(37859, 522941, "Hugh Glass"),

(39984, 522941, "John Fitzgerald"),

(671252, 522941, "Captain Andrew Henry"),

(16653, 5198, "Fred Krueger"),

(94877, 5198, "Nancy Thompson"),

(6245, 5198, "Glen Lantz"),

(6245, 4374, "Jack Sparrow"),

(24683, 4374, "Barbossa"),

(30875, 4374, "Will Turner"),

(24683, 16686, "David Helfgott - Adult"),

(7227, 16686, "David Helfgott - Adolescent"),

(8012, 16686, "Peter"),

(21931, 477764, "Marissa Wilson"),

(9278, 477764, "Uncle Machete"),

(1967244, 477764, "Rebecca Wilson"),

(7836, 301, "Neo"),

(9838, 301, "Morpheus"),

(6226, 301, "Trinity"),

(1491, 301, "Agent Smith"),

(5679, 339, "Indy"),

(2674, 339, "Marion Ravenwood"),

(141431, 339, "Major Arnold Toht"),

(23100, 463647, "Margaret Thatcher"),

(38704, 463647, "Denis Thatcher"),

(147467, 463647, "June"),

(6245, 63991, "Jack Sparrow"),

(30875, 63991, "Will Turner"),

(24302, 63991, "Elizabeth Swann"),

(40778, 688, "Harry Potter"),

(40780, 688, "Ron Weasley"),

(40779, 688, "Hermione Granger"),

(22670, 325439, "Michael Berg"),

(21709, 325439, "Hanna Schmitz"),

(459452, 325439, "Young Michael Berg"),

(6245, 461981, "Tonto"),

(1121926, 461981, "John Reid (Lone Ranger)"),

(13096, 461981, "Butch Cavendish"),

(22234, 1902, "Ted Kramer"),

(23100, 1902, "Joanna Kramer"),

(22308, 1902, "Margaret Phelps"),

(11362, 6226, "Andrew Nivens"),

(29367, 6226, "Sam Nivens"),

(26583, 6226, "Mary Sefton"),

(12086, 81924, "King Leonidas"),

(659495, 81924, "Stelios"),

(4810, 81924, "Captain"),

(47163, 77217, "Evil Queen"),

(2412, 77217, "Prince Wendell"),

(21571, 77217, "Virginia Lewis"),

(477028, 470179, "Valerie"),

(6650, 470179, "Solomon"),

(973659, 470179, "Peter"),

(27221, 5908, "Ann Darrow"),

(2523, 5908, "Jack Driscol"),

(23724, 5908, "Carl Denham"),

(32383, 5908, "Gunner"),

(39984, 437410, "Bane"),

(21495, 437410, "Bruce Wayne"),

(38703, 437410, "Selina");

insert ignore into rewardings\_Oscar

(rewarding\_year, actor\_id, film\_id, nomination) values

(2009, 1183, 111543, "Лучшая мужская роль второго плана"),

(2016, 37859, 522941, "Лучшая мужская роль"),

(1997, 24683, 16686, "Лучшая мужская роль"),

(2012, 23100, 463647, "Лучшая женская роль"),

(2009, 21709, 325439, "Лучшая женская роль"),

(1980, 23100, 1902, "Лучшая женская роль второго плана"),

(2013, 38703, 437410, "Лучшая женская роль второго плана");

insert ignore into reviews

(review\_name, author\_name, film\_id, author\_mark, review\_text) values

("Грандиозная битва за Средиземье", "Jesse Pinkman", 3498, 10,

"«Возвращение короля» — эпичное завершение трилогии Властелин колец. Питер Джексон снял нечто прекрасное, красивое, и захватывающее дух кино.

Актерский состав поразил, настолько актеры вжились в роль, что веришь всему, что происходит на экране. Сразу хочу отметить игру Энди Серкиса, просто отлично передал образ Голлума, его внутренний мир, поведение, его жалкое существование под влиянием кольца. Элайджа Вуд в этой части открыл персонажа по полной, здесь уже и переживания видны, и хорошая игра, Фродо стал более сильным, храбрым и смелым хоббитом. Вигго Мортенсен выложился на все сто процентов, мне в его игре понравилось все, он уже необычный странник из севера,а уже правитель который поведет свое войско на битву со злом. Все остальные актеры просто великолепны.

Битва на Пеленнорских полях показана очень эпично, захватывающе, особенно когда армия Рохана вступила в бой с харадримцами. Красивейшие пейзажи, горы, просторы средиземья все выглядит идеально.

«Возвращение Короля» — шедевр киноиндустрии, получивший множество Оскаров. Питер Джексон просто гениален! Браво!

Ждем с нетерпением «Хоббита»."),

(" ", "chopin", 3498, 9,

"В этих строках хочу выразить свое мнение. Фильм на мой взгляд получился лучше книги. Кто-нибудь когда читал книгу не задавал себе вопрос, почему описание идет так занудисто? Честно признаюсь, несколько страниц я перелистнул, бегло просмотрев текст.

Фильм же начинается, и сразу же начинают происходить события. Гендальф уехал, Гендальф приехал. Мы не знаем, что прошло много лет. А в книге? Все немного скучнее. Смотря фильм не спрашиваешь себя сколько проходит времени. Все три часа (а в общем больше девяти) находишься в напряжении.

Конечно есть ляпы. Когда Фродо убегает от призраков на плоту, непонятно почему лошадь, причем очень здоровая лошадь,

не смогла одолеть расстояние которое маленький хоббит перепрыгнул шутя? Из братства кольца мои симпатии на стороне

Леголаса и Арагорна."),

("Разочарование во всем фронтам", "Лорелея", 461981, 7,

"Во-первых, затянуто и неинтересно. Герои бродят туда-сюда, что-то делают, а в итоге из 150 минут запоминаются только последние двадцать, благодаря хоть какому-то экшену.

Во-вторых, штамп на штампе и штампом погоняет. Я уж молчу о том, что всё, что только можно было, стырено из фильмов-предшественников (фильмы про супер-героев, обе части новой Маски Зорро, вестерны — выбирай-не хочу), но даже лейтмотив «правильный герой встречает эксцентричного персонажа с относительными моральными устоями и в итоге отказывается от своих принципов» создатели умудрились утащить из собственных же «Пиратов Карибского моря» (притом, что оригинал гораздо увлекательнее, а метаморфоза Уилла Смита куда убедительнее).

Что характерно, эксцентричного персонажа с относительными моральными устоями снова играет Джонни Депп. Который мог бы стать единственным светлым пятном фильма, если бы не являлся по сути индейским воплощением Джека Воробья. А раз так, то первоисточник посмотреть, опять же, приятнее.

Нет, я бы и рада не сравнивать фильм с «Пиратами», но создатели сами не оставляют нам ни единого шанса избежать параллелей. В первую очередь, как уже было сказано, лейтмотив. Далее, один из второстепенных персонажей своей любовью к дамским нарядам и зонтикам уж больно напоминает нам одного старого одноглазого приятеля… Да что уж там, некоторые сцены и вовсе идентичны «Проклятью Чёрной жемчужины», разве что на заднем фоне вместо моря простирается прерия.

А выносить на афишу Хелену Бонэм-Картер и вовсе читерство: из 2,5 (!) часов, что идёт картина, Хелене отведено на экране от силы минут 15. Причём персонаж её до того бесполезен, что создаётся впечатление, будто вписывали его в сценарий исключительно ради «а давайте, у нас ещё и Бонэм-Картер будет, круто же, ну».

На самом деле, будь фильм на час короче, то как сухопутная версия «Пиратов Карибского моря» вполне подошёл бы для

ненапряжного просмотра. Но он утомил меня настолько, что больше 7 из 10 я ему не могу поставить, и то балл сверху за

джековоробеистого индейца. Соскучилась я что-то по старине Капитану."),

("Школа, тайны прошлого и мифические животные", "1mira1", 688, 8,

"Все мы, в большинстве своем, ходили в школу не только ради любимого предмета, но и ради встречи с друзьями или учителями, чтобы весело провести время, впервые влюбится и прочее. Но была ли для Вас школа домом? А для главного героя Гарри Поттера Хогвартс стал самым настоящим домом, который он не хочет покидать или потерять. А я в свое время хотела, чтобы моя школа сгорела…

И именно вокруг этого крутится весь сюжет. Хогвартс и его тайны, сочетание древности, а от нее страха, страх неизвестности и непознанности. «Тайная комната» оказалась фильмом не для слабонервных, однако. Были такие моменты и сцены, которые или вызывали рвотный инстинкт, или заставляли закрыть глаза и заткнуть уши. Но не все так страшно, мини-ужастик это лишь часть фильма, в остальное время все прекрасно и весело. Комедия с Роном, эльф Доби вообще умора. Виды школы и маленькие приключения известной троицы наоборот разжигают интерес к просмотру.

Всегда думала, что в этой части о Гарри Поттере Гермиону мало показывают. Она моя любимая героиня, и может даже стать на одном уровне с Гарри. Может она и не избранная и в ее семье все живы, но так же борется со злом, как и ее друг. Рон как всегда был клоуном, над которым все смеялись. На этот раз в клоунаде участвовала сломанная волшебная палочка. В «Тайной комнате» образ Гарри Поттера уже больше раскрывается. Наверно, это такая задумка, чтобы подогревать интерес к самой важной особе. Гарри проявляет смекалку и находчивость в решении страшной проблемы. В тоже самое время его постоянно сравнивают с Воландемортом, что сильно напрягает. Постепенно с каждым кадром появляются, как общие черты этих двух волшебников, так и разные. Последних больше.

«Гарри Поттер и Тайная комната» — это хорошее продолжение хорошее начала. Может по сравнению с первым фильмом мрачноватая,

но думаю, что задачей этой части являлась подготовка зрителя к еще более мрачному, а временами и беспросветному

продолжению истории о Гарри и его друзьях."),

("Адская смесь эпичности и брутальности!", "Liron", 81924, 8,

"Подобные фильмы закладываются в памяти на долгие годы. Во многом из за необычной стилистики и красоты происходящего. Зак в этом деле профессионал, он может наполнить фильм красотой картинки, необычной музыкой (ведь в фильме местами играет елктро-рок), живыми диалогами и интересными персонажами. И если все это убрать останется довольно скудный сюжет, но в этом и есть профессионализм режиссера, ведь после просмотра вы вряд ли прорисуете у себя в голове весь сюжет.

«Леонид желал что бы ты стоял прямо, с меня хватит и того что ты стоишь на коленях.»

Актеры с двух фронтов отыграли ну просто бесподобно, это воинственный и непоколебимый царь Леонид и бессмертный бог Ксеркс. Безусловно более харизматичен был Леонид в исполнении Джерарда Батлера, его призывы оставляли мурашки не то что у войнов Спарты а у обычного любителя кино, а вся его брутальность переходит все рамки адекватности.

«Спартанцы! Готовьте завтрак и наедайтесь! А ужин нас ждет в АДУ!»

Актеров второго плана обсуждать не стоит, ничего особенного там и нет, все отыгранные роли смотрелись не плохо. Про бои скажу пару слов: кровь, расчлененка, кровь, жестокость. Так же как и в любом другом фильме здесь есть и недочеты, но он один. В некоторых моментах мне показалось что спец эффекты выполнены довольно скудно, не смею отрицать тот факт что это было сделано специально для создания своей, «Снайдеровской» атмосферы.

Фильмы Зака Снайдера я за то и люблю что сюжет можно отодвинуть на второй план и наслаждаться пробивающей атмосферой.

Не знаю как вы а я пропитываюсь ей до мозга костей… Спасибо Зак!"),

(" ", "1mira1", 63991, 7,

"Фильм хороший, снято всё действительно очень красиво, но незаконченность просто бесит. Понятно, что диснеевцы решили заработать, но когда после просмотра вопросов на порядок больше, чем ответов на них, это начинает напрягать.

Первая часть всё-таки больше понравилась, тогда не было уверенности в успехе фильма и всё было сделано качественно в плане сюжета и прочее. А теперь, вследствие ажиотажа, фильм в любом случае прокатит, как прокатит и третья часть, на которую я также пойду и которую уже жду не дождусь.

Разумеется это ИМХО"),

("Не детская сказка.", "Cas\_000", 77217, 8,

"«Десятое королевство» мне советовали посмотреть многие люди. Все смотрели его в детстве, и уже будучи взрослыми питали к данному сериалу нежные чувства. А я всё откладывала просмотр, казалось, что не моё это. Но вот оно, пришло моё вдохновение для просмотра. Посмотрела на одном дыхании, теперь жалею, что не увидела его раньше.

Сценарий интересен, не затянут, вызывает желание посмотреть новую серию и узнать что же будет дальше. Мне нравится идея того, как сказка пересекается с реальным миром. Не смотря на сказочный сюжет, присутствует в сериале много «взрослого» юмора, совсем не детских историй и переживаний. Чего стоит сюжетная ветка главной героини Вирджинией и её матерью.

Актерский состав, не побоюсь этого слова, идеален. Каждый на своем месте, персонажи раскрыты и переданы очень хорошо. Особенно хотелось бы отметить Волка. Харизматичный, притягательный, опасный, добрый. Герой сумевший обуздать свою суть ради любви.

Режиссура, спец. эффекты, музыка находятся на должном уровне не смотря на то, что прошло много лет и сделать такую конфетку, для тех времен, стоило больших сил.

Итог Не смотря не то, что «Десятое королевство» я посмотрела уже в сознательном возрасте он останется для меня сериалом детства. И я обязательно пересмотрю его еще несколько раз уже со своей дочерью.

Надеюсь он станет для неё сериалом детства."),

("Смех да и только…", "Cas\_000", 5198, 5,

"Скажу сразу, посмотрел этот фильм после просмотра ремейка 2010ого года, прям в этот же день. Так как Кошмар.. 2010ого года меня очень впечатлил, а я и читал, и мне говорили, что оригинал намного лучше чем ремейк, с удовольствием сел за компьютер смотреть (даже свет выключил), и вот что я увидел: совершенно глупый, уж прошу прощения за выражение, идиотский фильм. То ли это комедия, то ли ужасы, то ли триллер, то ли ещё что-то.

Пишут, что Кошмар на улице Вязов 1984 года — это классика ужасов, что это тот фильм, после которого ты не можешь заснуть, от которого волосы встают дыбом, и сердце в пятки уходит — абсолютно ничего подобного я не испытал, хоть выключил в квартире свет и оставил дверь чуть открытой, чтоб поскрипывала. Настоящим страшным фильмом является тот фильм, в котором создаётся атмосфера ужаса, который давит на тебя со всех сторон, и тебе реально страшно…, а в этом фильме не то что это, а даже ни одного резкого момента нету. В начале фильма (второе появление Фредди) мы видим его тень вдоль забора, его длинные металлические лезвия на пальцах, а что после — он бежит как клоун и смеётся… ну что это такое? серьёзный хоррор? классика ужаса?

Все эпизоды сводятся к шутке… Естественно фильму нужно делать скидку, так как он был снят в 1984 году…, но снимать фильм без точного определения жанра — как то глупо.

В общем досмотрел я его с трудом, наверно это единственный фильм, который мне был противен.

Только одно есть, за что можно зацепиться — маска (или грим) Фредди.");

Дамп базы данных

Запросы к базе данных

1. *Отобрать всех актеров, играющих в фильме "Гарри Поттер и философский камень"*

select actor\_name, actor\_surname from actors natural join actors\_films

where film\_id = (select film\_id from films where film\_name = "Гарри Поттер и философский камень");

*#или другой варинат*

select actor\_name, actor\_surname from actors inner join actors\_films on(actors.actor\_id = actors\_films.actor\_id) inner join films on(actors\_films.film\_id = films.film\_id)

where film\_name = "Гарри Поттер и философский камень";

Результат:

|  |  |
| --- | --- |
| actor\_name | actor\_surname |
| Дэниэл | Рэдклифф |
| Эмма | Уотсон |
| Руперт | Гринт |

1. *Отобрать все фильмы актера Джонни Деппа*

select film\_name from actors natural join actors\_films natural join films

where actor\_name = "Джонни" and actor\_surname = "Депп";

*#или другой варинат*

select film\_name from films where film\_id in

(select film\_id from actors\_films where actor\_id =

(select actor\_id from actors where actor\_name = "Джонни" and actor\_surname = "Депп"));

Результат:

|  |
| --- |
| film\_name |
| Пираты Карибского моря:Проклятие Черной жемчужины |
| Кошмар на улице Вязов |
| Пираты Карибского моря: Сундук мертвеца |
| Одинокий рейнджер |

1. *Отобрать все фильмы, где режиссер Крис Коламбус и продюсер Дэвид Хейман*

select film\_name from films where director\_id =

(select director\_id from directors where director\_name = "Крис" and director\_surname = "Коламбус")

and producer\_id =

(select producer\_id from producers where producer\_name = "Дэвид" and producer\_surname = "Хейман");

Результат:

|  |
| --- |
| film\_name |
| Гарри Поттер и Тайная комната |
| Гарри Поттер и философский камень |

1. *Отобрать все фильмы, выпущенные раньше 1990 года*

select film\_name from films where release\_year < 1990 order by release\_year;

Результат:

|  |
| --- |
| film\_name |
| Крамер против Крамера |
| Индиана Джонс: В поисках утраченного ковчега |
| Кошмар на улице Вязов |

1. *Отобрать все фильмы, выпущенные режиссером Гор Вербински до 2010 года*

select film\_name from films where release\_year < 2010 and director\_id =

(select director\_id from directors

where director\_name = "Гор" and director\_surname = "Вербински");

Результат:

|  |
| --- |
| film\_name |
| Пираты Карибского моря:Проклятие Черной жемчужины |
| Пираты Карибского моря: Сундук мертвеца |

1. *Отобрать все фильмы, фигурирующие в номинациях премии Оскар в 2009 году*

select film\_name from films where film\_id in

(select film\_id from rewardings\_Oscar where rewarding\_year = 2009);

Результат:

|  |
| --- |
| film\_name |
| Темный рыцарь |
| Чтец |

1. *Отобрать все фильмы с актером Джонни Депп, где режиссер Гор Вербински*

select film\_name from films where director\_id =

(select director\_id from directors where director\_name = "Гор" and director\_surname = "Вербински")

and film\_id = any (select film\_id from actors\_films where actor\_id =

(select actor\_id from actors where actor\_name = "Джонни" and actor\_surname = "Депп"));

Результат:

|  |
| --- |
| film\_name |
| Пираты Карибского моря:Проклятие Черной жемчужины |
| Пираты Карибского моря: Сундук мертвеца |
| Одинокий рейнджер |

1. *Отобрать все фильмы режиссера Крис Коламбус, снятые в Англии*

select film\_name from films where country\_id =

(select country\_id from countries where country\_name = "Великобритания")

and director\_id =

(select director\_id from directors where director\_name = "Крис" and director\_surname = "Коламбус");

Результат:

|  |
| --- |
| film\_name |
| Гарри Поттер и Тайная комната |
| Гарри Поттер и философский камень |

1. *Отобрать количество фильмов, снятых на территории США в период 1980-2017 год*

select count(film\_id) from films where (release\_year between 1995 and 2000) and country\_id in

(select country\_id from countries where country\_name = "США");

Результат:

|  |
| --- |
| count(film\_id) |
| 2 |

1. *Отобрать все рецензии о фильме Властелин колец: Возвращение Короля*

select \* from reviews

where film\_id = (select film\_id from films where film\_name = "Властелин колец: Возвращение Короля");

Результат:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| review\_id | review\_name | author\_name | film\_id | author\_mark | review\_text |
| 1 | Грандиозная битва за Средиземье | Jesse Pinkman | 3498 | 10 | «Возвращение короля» — эпичное завершение трилогии Властелин колец. Питер Джексон снял нечто прекрасное, красивое, и захватывающее дух кино. Актерский состав поразил, настолько актеры вжились в роль, что веришь всему, что происходит на экране. Сразу хочу отметить игру Энди Серкиса, просто отлично передал образ Голлума, его внутренний мир, поведение, его жалкое существование под влиянием кольца. Элайджа Вуд в этой части открыл персонажа по полной, здесь уже и переживания видны, и хорошая игра, Фродо стал более сильным, храбрым и смелым хоббитом. Вигго Мортенсен выложился на все сто процентов, мне в его игре понравилось все, он уже необычный странник из севера,а уже правитель который поведет свое войско на битву со злом. Все остальные актеры просто великолепны. Битва на Пеленнорских полях показана очень эпично, захватывающе, особенно когда армия Рохана вступила в бой с харадримцами. Красивейшие пейзажи, горы, просторы средиземья все выглядит идеально. «Возвращение Короля» — шедевр киноиндустрии, получивший множество Оскаров. Питер Джексон просто гениален! Браво! Ждем с нетерпением «Хоббита». |
| 2 |  | chopin | 3498 | 9 | В этих строках хочу выразить свое мнение. Фильм на мой взгляд получился лучше книги. Кто-нибудь когда читал книгу не задавал себе вопрос, почему описание идет так занудисто? Честно признаюсь, несколько страниц я перелистнул, бегло просмотрев текст. Фильм же начинается, и сразу же начинают происходить события. Гендальф уехал, Гендальф приехал. Мы не знаем, что прошло много лет. А в книге? Все немного скучнее. Смотря фильм не спрашиваешь себя сколько проходит времени. Все три часа (а в общем больше девяти) находишься в напряжении. Конечно есть ляпы. Когда Фродо убегает от призраков на плоту, непонятно почему лошадь, причем очень здоровая лошадь, не смогла одолеть расстояние которое маленький хоббит перепрыгнул шутя? Из братства кольца мои симпатии на стороне Леголаса и Арагорна. |

1. *Отобрать все фильмы жанра Фентези от режиссера Горa Вербински в период 2002-2010*

select film\_name from films where genre\_id =

(select genre\_id from genres where genre\_name = "Фентези")

and director\_id =

(select director\_id from directors where director\_name = "Гор"

and director\_surname = "Вербински")

and release\_year between 2002 and 2010;

Результат:

|  |
| --- |
| film\_name |
| Пираты Карибского моря:Проклятие Черной жемчужины |
| Пираты Карибского моря: Сундук мертвеца |

1. *Отобрать всех актеров, получивших Оскар в 2009 году*

select actor\_name, actor\_surname from actors where actor\_id in (select actor\_id from rewardings\_Oscar where rewarding\_year = 2009);

Результат:

|  |  |
| --- | --- |
| actor\_name | actor\_surname |
| Хит | Леджер |
| Кейт | Уинслет |

1. *Выбрать всех актеров, игравших в фильмах, сценаристом которых является Тед Эллиот*

select actor\_name, actor\_surname from actors where actor\_id in

(select actor\_id from actors\_films where film\_id in

(select film\_id from films where screenwriter\_id =

(select screenwriter\_id from screenwriters where screenwriter\_name = "Тед" and screenwriter\_surname = "Эллиот")));

Результат:

|  |  |
| --- | --- |
| actor\_name | actor\_surname |
| Джонни | Депп |
| Джеффри | Раш |
| Орландо | Блум |
| Дональд | Сазерленд |
| Джули | Уорнер |
| Эрик | Тал |
| Кира | Найтли |

1. *Отобрать всех сценаристов, продюсеров и режиссеров "оскороносных" фильмов*

select concat(director\_name, " ", director\_surname, ", режиссер") as разработчик from directors where director\_id in

(select director\_id from films where film\_id in

(select film\_id from rewardings\_Oscar))

union select concat(producer\_name, " ", producer\_surname, ", продюсер") from producers where producer\_id in

(select producer\_id from films where film\_id in

(select film\_id from rewardings\_Oscar))

union select concat(screenwriter\_name, " ", screenwriter\_surname, ", сценарист") from screenwriters where screenwriter\_id in

(select screenwriter\_id from films where film\_id in

(select film\_id from rewardings\_Oscar));

Результат:

|  |
| --- |
| разработчик |
| Роберт Бентон, режиссер |
| Скотт Хикс, режиссер |
| Кристофер Нолан, режиссер |
| Стивен Долдри, режиссер |
| Филлида Ллойд, режиссер |
| Алехандро Иньярриту, режиссер |
| Стэнли Р. Джаффе, продюсер |
| Джейн Скотт, продюсер |
| Кристофер Нолан, продюсер |
| Донна Джильотти, продюсер |
| Дэмиэн Джонс, продюсер |
| Стив Голин, продюсер |
| Роберт Бентон, сценарист |
| Ян Сарди, сценарист |
| Кристофер Нолан, сценарист |
| Дэвид Хэа, сценарист |
| Эби Морган, сценарист |
| Марк Смит, сценарист |

1. *Показать режиссера и сценариста первого фильма, обеспечившего победу в номинации "Лучшая мужская роль"*

select concat(director\_name, " ", director\_surname) as режиссер, concat(screenwriter\_name, " ", screenwriter\_surname) as сценарист from directors, screenwriters where director\_id =

(select director\_id from films where film\_id =

(select film\_id from rewardings\_Oscar where nomination = "Лучшая мужская роль" order by rewarding\_year limit 1)) and screenwriter\_id =

(select screenwriter\_id from films where film\_id =

(select film\_id from rewardings\_Oscar where

nomination = "Лучшая мужская роль" order by rewarding\_year limit 1));

*#или другой вариант*

select concat(director\_name, " ", director\_surname) as режиссер, concat(screenwriter\_name, " ", screenwriter\_surname) as сценарист from directors, screenwriters, rewardings\_Oscar where

director\_id = (select director\_id from films where film\_id = rewardings\_Oscar.film\_id)

and screenwriter\_id = (select screenwriter\_id from films where film\_id = rewardings\_Oscar.film\_id)

group by film\_id having rewardings\_Oscar.film\_id =

(select film\_id from rewardings\_Oscar where nomination = "Лучшая мужская роль" order by rewarding\_year limit 1);

Результат:

|  |  |
| --- | --- |
| режиссер | сценарист |
| Скотт Хикс | Ян Сарди |

1. *Отобрать первые 2 фильмов, снятых на территории США, за участие в которых были получены Оскары*

select film\_name from films where film\_id in

(select film\_id from rewardings\_Oscar) and country\_id =

(select country\_id from countries where country\_name = "США") order by release\_year limit 2;

Результат:

|  |
| --- |
| film\_name |
| Крамер против Крамера |
| Темный рыцарь |

1. *Вывести страну, на территории которой снято больше всего фильмов*

select country\_name, max((select count(film\_id) from films as t2 where t2.country\_id = t1.country\_id )) as количество

from countries as t1 group by country\_id limit 1;

Результат:

|  |  |
| --- | --- |
| country\_name | количество |
| США | 17 |

1. *Вывести "провальные" фильмы актера Джонни Деппа (фильмы, рейтинг которых ниже среднего)*

select film\_name from films where film\_id in

(select film\_id from actors\_films where actor\_id =

(select actor\_id from actors where actor\_name = "Джонни" and actor\_surname = "Депп")) group by rating

having rating < (select avg(rating) from films);

Результат:

|  |
| --- |
| film\_name |
| Одинокий рейнджер |

1. *Показать режиссера, продюсера и название первого фильма, обеспечившего победу в номинации "Лучшая женская роль" актрисе Мэрил Стрип*

select concat(director\_name, " ", director\_surname) as режиссер, concat(producer\_name, " ", producer\_surname) as продюсер, film\_name

from directors, producers, (select \* from films where exists(select film\_id from rewardings\_Oscar where rewardings\_Oscar.film\_id = films.film\_id)) as t2

where directors.director\_id = t2.director\_id and producers.producer\_id = t2.producer\_id and exists(select actor\_id from actors as t1 where actor\_name = "Мэрил" and actor\_surname = "Стрип" and

exists(select film\_id from rewardings\_Oscar where nomination = "Лучшая женская роль" and rewardings\_Oscar.actor\_id=t1.actor\_id and rewardings\_Oscar.film\_id = t2.film\_id group by rewarding\_year limit 1));

Результат:

|  |  |  |
| --- | --- | --- |
| режиссер | продюсер | film\_name |
| Филлида Ллойд | Дэмиэн Джонс | Железная леди |

1. *Вывести все награды, полученные актером Мэрил Стрип*

select nomination, rewarding\_year, film\_name from rewardings\_Oscar as t1 natural join films where

exists(select \* from actors\_films where actors\_films.film\_id = t1.film\_id and

exists(select \* from actors where concat(actor\_name, " ", actor\_surname) = "Мэрил Стрип" and actors.actor\_id = actors\_films.actor\_id));

Результат:

|  |  |  |
| --- | --- | --- |
| nomination | rewarding\_year | film\_name |
| Лучшая женская роль второго плана | 1980 | Крамер против Крамера |
| Лучшая женская роль | 2012 | Железная леди |

1. *Вывести всех актеров в порядке возрастания успешности (имеется в виду количество полученных наград)*

1. *Вывести 4 наиболее востребованных актеров за 2000-2005 годы (тех, кто снимался в большем количестве фильмов)*

select concat(actor\_name, " ", actor\_surname) as актёр, count(film\_id) as количество\_фильмов from actors natural join actors\_films as t1

where exists(select \* from films as t2 where release\_year between 2000 and 2005 and t2.film\_id = t1.film\_id)

group by actor\_id order by количество\_фильмов desc limit 4;

Результат:

|  |  |
| --- | --- |
| актёр | количество\_фильмов |
| Руперт Гринт | 2 |
| Орландо Блум | 2 |
| Эмма Уотсон | 2 |
| Дэниэл Рэдклифф | 2 |

1. *Вывести список фильмов, снятых на территории Великобритании, имеющих рейтинг больше 8*

select film\_name from films where rating > 8 and exists(select \* from countries where country\_name = "Великобритания" and countries.country\_id = films.country\_id);

Результат:

|  |
| --- |
| film\_name |
| Гарри Поттер и Тайная комната |
| Гарри Поттер и философский камень |
| Десятое королевство (сериал) |

1. *Вывести Топ-7 фильмов*

select film\_name, rating from films order by rating desc limit 7;

Результат:

|  |  |
| --- | --- |
| film\_name | rating |
| Властелин колец: Возвращение Короля | 8.611 |
| Темный рыцарь | 8.502 |
| Матрица | 8.492 |
| Десятое королевство (сериал) | 8.457 |
| Титаник | 8.371 |
| Пираты Карибского моря:Проклятие Черной жемчужины | 8.342 |
| Гарри Поттер и философский камень | 8.193 |

1. *Посчитать средний рейтинг фильмов США, снятых после 2000 года*

select avg(rating) from films where release\_year > 2000 and exists(select \* from countries where country\_name = "США" and countries.country\_id = films.country\_id);

Результат:

|  |
| --- |
| avg(rating) |
| 7.426090869036588 |

1. *Вывести все фильмы (название, год, страна), рейтинг которых ниже среднего*

select film\_name, release\_year, rating, country\_name from films natural join countries where rating < (select avg(rating) from films);

Результат:

|  |  |  |  |
| --- | --- | --- | --- |
| film\_name | release\_year | rating | country\_name |
| Кинг Конг | 2005 | 7.573 | Новая Зеландия |
| Кукловоды | 1994 | 6.438 | США |
| 300 спартанцев | 2007 | 7.629 | США |
| Одинокий рейнджер | 2013 | 6.791 | США |
| Железная леди | 2011 | 7.013 | Великобритания |
| Красная Шапочка | 2011 | 6.126 | США |
| Дети шпионов 4D | 2011 | 3.695 | США |

1. *Вывести всех актеров из Германии, игравших в фильмах, снятых в США*

select concat(actor\_name, " ", actor\_surname) as актёр from actors natural join countries where country\_name = "Германия" and

exists(select actor\_id from actors\_films where film\_id in (select film\_id from films natural join countries where country\_name = "США") and actors\_films.actor\_id = actors.actor\_id);

Результат:

|  |
| --- |
| актёр |
| Давид Кросс |
| Майкл Фассбендер |

1. *Отобрать всех актеров, младше 30 лет*

select concat(actor\_name, " ", actor\_surname) as актёр, (year(now())-birth\_year) as возраст from actors having возраст < 30;

Результат:

|  |  |
| --- | --- |
| актёр | возраст |
| Дэниэл Рэдклифф | 28 |
| Эмма Уотсон | 27 |
| Руперт Гринт | 29 |
| Давид Кросс | 27 |
| Роуэн Бланчард | 16 |

1. *Отобрать всех актеров, игравших в фильмах с рейтингом, который выше среднего*

select concat(actor\_name, " ", actor\_surname) from actors as t1 where

exists(select \* from actors\_films as t2 where t1.actor\_id = t2.actor\_id and

exists(select \* from films as t3 where t2.film\_id = t3.film\_id and rating > (select avg(rating) from films)));

Результат:

|  |
| --- |
| concat(actor\_name, " ", actor\_surname) |
| Хит Леджер |
| Хьюго Уивинг |
| Дэниэл Лапэйн |
| Карен Аллен |
| Харрисон Форд |
| Кэрри-Энн Мосс |
| Джонни Депп |
| Аарон Экхарт |
| Ноа Тейлор |
| Киану Ривз |
| Армин Мюллер-Шталь |
| Лоренс Фишбёрн |
| Вигго Мортенсен |
| Роберт Инглунд |
| Элайджа Вуд |
| Кристиан Бэйл |
| Кимберли Уильямс-Пэйсли |
| Кейт Уинслет |
| Дастин Хоффман |
| Джейн Александр |
| Рэйф Файнс |
| Мэрил Стрип |
| Кира Найтли |
| Джеффри Раш |
| Шон Эстин |
| Орландо Блум |
| Леонардо ДиКаприо |
| Энн Хэтэуэй |
| Том Харди |
| Дэниэл Рэдклифф |
| Эмма Уотсон |
| Руперт Гринт |
| Билли Зейн |
| Дайэнн Уист |
| Хэзер Лэнгенкэмп |
| Рональд Лейси |
| Давид Кросс |
| Донал Глисон |

1. *Вывести все жанры в порядке убывания их популярности у режиссеров (по частоте использования)*

select genre\_name as жанр, (select count(genre\_id) from films as t1 group by genre\_id having t1.genre\_id = t2.genre\_id) as частота

from genres as t2 group by genre\_id order by частота desc;

*#или другой вариант*

select genre\_name as жанр, count(genre\_id) as частота from genres natural join films group by genre\_id order by частота desc;

Результат:

|  |  |
| --- | --- |
| жанр | частота |
| Фентези | 7 |
| Драма | 5 |
| Фантастика | 4 |
| Ужасы | 3 |
| Приключения | 1 |
| Триллер | 1 |
| Вестерн | 1 |
| Биография | 1 |

1. *Выбрать все фильмы, в которых играли Джонни Депп и \_Орландо Блум, и жанр – фэнтези*

select film\_name from films as t1 where exists(

select \* from actors\_films as t2 where exists(

select \* from actors as t3 where concat(actor\_name, " ", actor\_surname) = "Джонни Депп" and t3.actor\_id = t2.actor\_id

)

and t2.film\_id = t1.film\_id and film\_id in (

select film\_id from actors\_films as t4 where exists(

select \* from actors as t5 where concat(actor\_name, " ", actor\_surname) = "Орландо Блум" and t5.actor\_id = t4.actor\_id

) group by film\_id

)

) and exists(select \* from genres as t6 where t6.genre\_id = t1.genre\_id);

Результат:

|  |
| --- |
| film\_name |
| Пираты Карибского моря:Проклятие Черной жемчужины |
| Пираты Карибского моря: Сундук мертвеца |

1. *Вывести все страны, в которых снимались "оскороносные" фильмы*

select country\_name from countries as t1 where exists(

select \* from films as t2 where exists(

select \* from rewardings\_oscar as t3 where t3.film\_id = t2.film\_id

) and t2.country\_id = t1.country\_id

);

*#или другой вариант*

select distinct country\_name from countries natural join films as t2 where exists(

select \* from rewardings\_oscar as t3 where t3.film\_id = t2.film\_id

);

Результат:

|  |
| --- |
| country\_name |
| Австралия |
| Великобритания |
| США |

1. *Вывести названия всех рецензий, оценка фильма от автора у которых выше рейтинга самого фильма*

select review\_id, review\_name, author\_name, author\_mark, film\_id from reviews as t1 where author\_mark > (

select rating from films as t2 where t1.film\_id = t2.film\_id

);

Результат:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| review\_id | review\_name | author\_name | author\_mark | film\_id |
| 1 | Грандиозная битва за Средиземье | Jesse Pinkman | 10 | 3498 |
| 2 |  | chopin | 9 | 3498 |
| 3 | Разочарование во всем фронтам | Лорелея | 7 | 461981 |
| 5 | Адская смесь эпичности и брутальности! | Liron | 8 | 81924 |

1. *Вывести режиссёров, которые писали сценарий к своему фильму*

select concat(director\_name, " ", director\_surname) as режиссер from directors as t1 where

exists(select \* from screenwriters as t2 where concat(t1.director\_name, " ", t1.director\_surname) = concat(t2.screenwriter\_name, " ", t2.screenwriter\_surname))

and exists(select \* from films as t3 where director\_id = screenwriter\_id and t3.director\_id = t1.director\_id);

Результат:

|  |
| --- |
| режиссер |
| Уэс Крэйвен |
| Джеймс Кэмерон |
| Роберт Родригес |
| Роберт Бентон |
| Кристофер Нолан |
| Зак Снайдер |

1. *Вывести все фильмы, которые снимались в той же стране, что и Гарри Поттер, и они того же жанра*

select distinct if(country\_id = (select country\_id from films where film\_name = "Гарри Поттер и Тайная комната")

and genre\_id = (select genre\_id from films where film\_name = "Гарри Поттер и Тайная комната"), film\_name, null) as фильмы from films;

Результат:

|  |
| --- |
| фильмы |
| NULL |
| Гарри Поттер и Тайная комната |
| Гарри Поттер и философский камень |
| Десятое королевство (сериал) |

1. *Вывести для каждого актера из списка его фильмов тот, у которого наибольший рейтинг*

select concat(actor\_name, " ", actor\_surname) as актёр, film\_name, max(rating), character\_name

from actors inner join actors\_films as t1 on(actors.actor\_id = t1.actor\_id)

inner join films on(t1.film\_id = films.film\_id)

group by t1.actor\_id;

Результат:

|  |  |  |  |
| --- | --- | --- | --- |
| актёр | film\_name | max(rating) | character\_name |
| Хит Леджер | Темный рыцарь | 8.502 | Joker |
| Хьюго Уивинг | Матрица | 8.492 | Agent Smith |
| Дэниэл Лапэйн | Десятое королевство (сериал) | 8.457 | Prince Wendell |
| Эдриан Броуди | Кинг Конг | 7.573 | Jack Driscol |
| Карен Аллен | Индиана Джонс: В поисках утраченного ковчега | 8.007 | Marion Ravenwood |
| Винсент Риган | 300 спартанцев | 7.629 | Captain |
| Харрисон Форд | Индиана Джонс: В поисках утраченного ковчега | 8.007 | Indy |
| Кэрри-Энн Мосс | Матрица | 8.492 | Trinity |
| Джонни Депп | Пираты Карибского моря:Проклятие Черной жемчужины | 8.342 | Jack Sparrow |
| Гари Олдман | Красная Шапочка | 6.126 | Solomon |
| Аарон Экхарт | Темный рыцарь | 8.502 | Harvey Dent |
| Ноа Тейлор | Блеск | 7.87 | David Helfgott - Adolescent |
| Киану Ривз | Матрица | 8.492 | Neo |
| Армин Мюллер-Шталь | Блеск | 7.87 | Peter |
| Дэнни Трехо | Дети шпионов 4D | 3.695 | Uncle Machete |
| Лоренс Фишбёрн | Матрица | 8.492 | Morpheus |
| Вигго Мортенсен | Властелин колец: Возвращение Короля | 8.611 | Aragorn |
| Дональд Сазерленд | Кукловоды | 6.438 | Andrew Nivens |
| Джерард Батлер | 300 спартанцев | 7.629 | King Leonidas |
| Уильям Фихтнер | Одинокий рейнджер | 6.791 | Butch Cavendish |
| Роберт Инглунд | Кошмар на улице Вязов | 7.703 | Fred Krueger |
| Элайджа Вуд | Властелин колец: Возвращение Короля | 8.611 | Frodo |
| Кристиан Бэйл | Темный рыцарь | 8.502 | Bruce Wayne |
| Кимберли Уильямс-Пэйсли | Десятое королевство (сериал) | 8.457 | Virginia Lewis |
| Кейт Уинслет | Титаник | 8.371 | Rose DeWitt Bukater |
| Джессика Альба | Дети шпионов 4D | 3.695 | Marissa Wilson |
| Дастин Хоффман | Крамер против Крамера | 8.011 | Ted Kramer |
| Джейн Александр | Крамер против Крамера | 8.011 | Margaret Phelps |
| Рэйф Файнс | Чтец | 7.935 | Michael Berg |
| Мэрил Стрип | Крамер против Крамера | 8.011 | Joanna Kramer |
| Джек Блэк | Кинг Конг | 7.573 | Carl Denham |
| Кира Найтли | Пираты Карибского моря: Сундук мертвеца | 8.098 | Elizabeth Swann |
| Джеффри Раш | Пираты Карибского моря:Проклятие Черной жемчужины | 8.342 | Barbossa |
| Джули Уорнер | Кукловоды | 6.438 | Mary Sefton |
| Наоми Уоттс | Кинг Конг | 7.573 | Ann Darrow |
| Шон Эстин | Властелин колец: Возвращение Короля | 8.611 | Sam |
| Эрик Тал | Кукловоды | 6.438 | Sam Nivens |
| Орландо Блум | Властелин колец: Возвращение Короля | 8.611 | Legolas |
| Питер Джексон | Кинг Конг | 7.573 | Gunner |
| Леонардо ДиКаприо | Титаник | 8.371 | Jack Dawson |
| Энн Хэтэуэй | Темный рыцарь: Возрождение легенды | 8.144 | Selina |
| Джим Бродбент | Железная леди | 7.013 | Denis Thatcher |
| Том Харди | Темный рыцарь: Возрождение легенды | 8.144 | Bane |
| Дэниэл Рэдклифф | Гарри Поттер и Тайная комната | 8.193 | Harry Potter |
| Эмма Уотсон | Гарри Поттер и Тайная комната | 8.193 | Hermione Granger |
| Руперт Гринт | Гарри Поттер и Тайная комната | 8.193 | Ron Weasley |
| Билли Зейн | Титаник | 8.371 | Cal Hockley |
| Дайэнн Уист | Десятое королевство (сериал) | 8.457 | Evil Queen |
| Хэзер Лэнгенкэмп | Кошмар на улице Вязов | 7.703 | Nancy Thompson |
| Рональд Лейси | Индиана Джонс: В поисках утраченного ковчега | 8.007 | Major Arnold Toht |
| Сьюзэн Браун | Железная леди | 7.013 | June |
| Давид Кросс | Чтец | 7.935 | Young Michael Berg |
| Аманда Сайфред | Красная Шапочка | 6.126 | Valerie |
| Майкл Фассбендер | 300 спартанцев | 7.629 | Stelios |
| Донал Глисон | Выживший | 7.814 | Captain Andrew Henry |
| Шайло Фернандес | Красная Шапочка | 6.126 | Peter |
| Арми Хаммер | Одинокий рейнджер | 6.791 | John Reid (Lone Ranger) |
| Роуэн Бланчард | Дети шпионов 4D | 3.695 | Rebecca Wilson |

1. *Вывести авторов статей, которые всегда ставят оценки ниже рейтинга фильма*

select distinct author\_name from reviews as t1 where not exists(

select author\_mark from reviews as t3 where author\_mark > (

select rating from films as t2 where t3.film\_id = t2.film\_id

) and t1.author\_name = t3.author\_name

);

Результат:

|  |
| --- |
| author\_name |
| 1mira1 |
| Cas\_000 |

1. *Вывести все страны, в которых снимались фильмы в порядке убывания количества снятых в них фильмов*

select country\_name, (select count(film\_id) from films as t3 where t1.country\_id = t3.country\_id) as количество\_фильмов from countries as t1 where exists(select \* from films as t2 where t1.country\_id = t2.country\_id) order by количество\_фильмов desc;

*#или*

select country\_name, count(film\_id) as количество\_фильмов from countries as t1 natural join films where if(country\_id in (select country\_id from films as t2 where t1.country\_id = t2.country\_id), true, false) group by country\_id order by количество\_фильмов desc;

Результат:

|  |  |
| --- | --- |
| country\_name | количество\_фильмов |
| США | 17 |
| Великобритания | 4 |
| Новая Зеландия | 1 |
| Австралия | 1 |

1. *Вывести режиссеров в порядке убывания количества наград, полученных актерами за съемки в их фильмах*

select concat(director\_name, " ", director\_surname) as режиссер,

(select count(nomination) from rewardings\_oscar as t1 where

exists(select \* from films as t2 where t1.film\_id = t2.film\_id and t3.director\_id = t2.director\_id)) as количество\_наград

from directors as t3 order by количество\_наград desc;

Результат:

|  |  |
| --- | --- |
| режиссер | количество\_наград |
| Кристофер Нолан | 2 |
| Алехандро Иньярриту | 1 |
| Роберт Бентон | 1 |
| Стивен Долдри | 1 |
| Скотт Хикс | 1 |
| Филлида Ллойд | 1 |
| Стюарт Орм | 0 |
| Питер Джексон | 0 |
| Гор Вербински | 0 |
| Джеймс Кэмерон | 0 |
| Стивен Спилберг | 0 |
| Дэвид Карсон | 0 |
| Кэтрин Хардвик | 0 |
| Крис Коламбус | 0 |
| Зак Снайдер | 0 |
| Роберт Родригес | 0 |
| Лана Вачовски | 0 |
| Уэс Крэйвен | 0 |

1. *Вывести всех актеров, которые являются ещё и режиссерами или продюсерами (возможно в других фильмах)*

select concat(actor\_name, " ", actor\_surname, ", режиссёр") as актёр from actors as t1 where

if(exists(select \* from directors as t2 where concat(actor\_name, " ", actor\_surname) = concat(director\_name, " ", director\_surname)), true,false)

union select concat(actor\_name, " ", actor\_surname, ", продюсер") as актёр from actors as t1 where

if(exists(select \* from producers as t2 where concat(actor\_name, " ", actor\_surname) = concat(producer\_name, " ", producer\_surname)), true,false);

Результат:

|  |
| --- |
| актёр |
| Питер Джексон, режиссёр |
| Питер Джексон, продюсер |
| Леонардо ДиКаприо, продюсер |