Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

Рязанский государственный радиотехнический университет имени В.Ф. Уткина

Кафедра ЭВМ

К защите

Руководитель работы:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

дата, подпись

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**К КУРСОВОЙ РАБОТЕ**

по дисциплине

**«Базы данных»**

Тема:

«Разработка базы данных для предметной области:

Онлайн-кинотеатр»

|  |  |  |
| --- | --- | --- |
| Выполнил студент группы 945 |  |  |
| Горбачева Д.Д. |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  дата сдачи на проверку, подпись |
|  |  |  |
| Руководитель работы |  |  |
| ассистент кафедры ЭВМ |  |  |
| Тарасова В.Ю. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  оценка | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  дата защиты, подпись |

Рязань 2020

**Содержание**

[Введение 2](#_Toc59969089)

[1 Семантическое описание предметной области 3](#_Toc59969090)

[2 Разработка базы данных 5](#_Toc59969091)

[2.1 Разработка логической модели данных 5](#_Toc59969092)

[2.2 Определение типов, хранимых данных 9](#_Toc59969093)

[2.3 Задание ограничений целостности данных 10](#_Toc59969094)

[2.4 Построение схемы БД 12](#_Toc59969095)

[2.5 Разработка сценария для создания БД и основных объектов структуры БД 12](#_Toc59969096)

[2.6 Разработка сценария для заполнения таблиц БД 13](#_Toc59969097)

[3 Разработка основных операций реляционной алгебры 16](#_Toc59969098)

[4 Разработка запросов к БД для реализации информационных потребностей пользователя 19](#_Toc59969099)

[Заключение 40](#_Toc59969100)

[Список литературы 41](#_Toc59969101)

# Введение

База данных (БД) — это организованная структура, предназначенная для хранения, изменения и обработки взаимосвязанной информации, преимущественно больших объемов.

Для изменения и управления данными существуют системы управления базами данных (СУБД). Это комплекс программно-языковых средств, позволяющих создать БД и управлять данными.

Целью данной курсовой работы является создание реляционной базы данных для хранения информации и работы с ней. Предметная область курсовой работы – «Онлайн-кинотеатр». В базу должны входить различные данные о пользователях, тарифах, фильмах и информация о подписках. Необходимо предусмотреть то, чтобы пользователь мог удобно пользоваться этой базой данных и получать необходимую ему информацию.

# 1 Семантическое описание предметной области

Онлайн-кинотеатр – это сервис для просмотров фильмов с любого электронного устройства и в любой точке мира.

В каждом онлайн-кинотеатре существуют тарифы для пользователей. Один тариф включает в себя один фильм. О тарифах требуется хранить следующую информацию: название, количество фильмов, срок действия тарифа, стоимость.

Пользователи покупают любой тариф на свой вкус, но пользователь не может приобрести несколько тарифов одновременно. О пользователях должна храниться информация: ФИО, дата рождения, наличие льгот (да/нет), логин, пароль, адрес электронной почты, тариф.

О фильмах требуется хранить следующую информацию: название, возрастная категория, страна производителя, год выпуска, жанр. Один фильм может находиться в одном тарифе.

Пользователь, приобретая тариф, получает подписку в онлайн-кинотеатре на определенный срок. О подписках должна храниться следующая информация: время и дата подписки пользователя на фильм.

# 2 Разработка базы данных

# 2.1 Разработка логической модели данных

***Описание основных сущностей ПО.***

На основании проведенного анализа предметной области «Онлайн-кинотеатр», можно выделить следующие сущности:

Тарифы, Пользователь, Фильмы, Подписка.

Представим список сущностей предметной области (таблица 1).

Таблица 1. Список сущностей

|  |  |  |
| --- | --- | --- |
| № | Название | Назначение |
| 1 | Тарифы | Перечень тарифов, которые существуют в онлайн-кинотеатре |
| 2 | Пользователь | Описание пользовательской базы в онлайн-кинотеатре |
| 3 | Фильмы | Список фильмов, показывающих в онлайн-кинотеатре |
| 4 | Подписка | Информация о пользователях и тарифе, который они приобрели |

Для каждой сущности приведем описание ее атрибутов в виде таблицы. На данном этапе выделяются ключевые (это первичные, внешние и альтернативные ключи) и неключевые атрибуты. Результаты приведены в таблицах 2-5.

Следует отметить, что для сущностей Тарифы, в отличие от других сущностей, нет атрибута, с помощью которого можно идентифицировать конкретный тариф. Поэтому требуется ввести дополнительный атрибут, например, id тарифа, который и будет являться первичным ключом.

Атрибут id тарифа является дополнительным служебным полем, добавленным к уже имеющимся информационным полям таблицы. Такие атрибуты называются суррогатными, их единственное предназначение - служить первичным ключом.

Также мы добавляем в сущности Пользователь, Фильмы и Подписка дополнительные атрибуты, которые после будут являться первичными ключами.

1. В сущность Пользователь – id пользователя;
2. В сущность Фильмы – id фильма;
3. В сущность Подписка – Номер пользователя и Номер фильма.

Таблица 2. Список атрибутов таблицы «Тарифы»

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Название | Назначение |
| ПК  (Первичный ключ) | id тарифа | Ключевое поле.  Представляет собой первичный ключ.  Это уникальное значение, соответствующее каждому тарифу.  Значения автоматически генерируется СУБД при вставке новой записи в таблицу. |
|  | Название |  |
|  | Количество фильмов |  |
|  | Срок действия тарифа |  |
|  | Стоимость |  |

Таблица 3. Список атрибутов таблицы «Пользователь»

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Название | Назначение |
| ПК  (Первичный ключ) | id пользователя | Ключевое поле.  Представляет собой первичный ключ.  Это уникальное значение, соответствующее каждому пользователю.  Значения автоматически генерируется СУБД при вставке новой записи в таблицу. |
|  | ФИО |  |
|  | Дата рождения |  |
|  | Наличие льгот |  |
|  | Логин |  |
|  | Пароль |  |

Продолжение таблицы 3

|  |  |  |
| --- | --- | --- |
|  | Адрес электронной почты |  |
|  | Номер тарифа |  |

Таблица 4. Список атрибутов таблицы «Фильмы»

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Название | Назначение |
| ПК  (Первичный ключ) | id фильма | Ключевое поле.  Представляет собой первичный ключ.  Это уникальное значение, соответствующее каждому фильму.  Значения автоматически генерируется СУБД при вставке новой записи в таблицу. |
|  | Название |  |
|  | Возрастная категория |  |
|  | Страна производитель |  |
|  | Год выпуска |  |
|  | Жанр |  |
| ВК  (Внешний ключ) | id тарифа | С помощью данного ключа, можно узнать, какой фильм входит в определенный тариф. |

Таблица 5. Список атрибутов таблицы «Подписка»

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Название | Назначение |
| ПК  (Первичный ключ) | Номер пользователя | Ключевое поле.  Представляет собой первичный ключ.  Это уникальное значение, соответствующее каждому пользователю.  Значения автоматически генерируется СУБД при вставке новой записи в таблицу. |

Продолжение таблицы 5

|  |  |  |
| --- | --- | --- |
| ПК  (Первичный ключ) | Номер фильма | Ключевое поле.  Представляет собой первичный ключ.  Это уникальное значение, соответствующее каждому фильму.  Значения автоматически генерируется СУБД при вставке новой записи в таблицу. |
|  | Дата и время подписки |  |

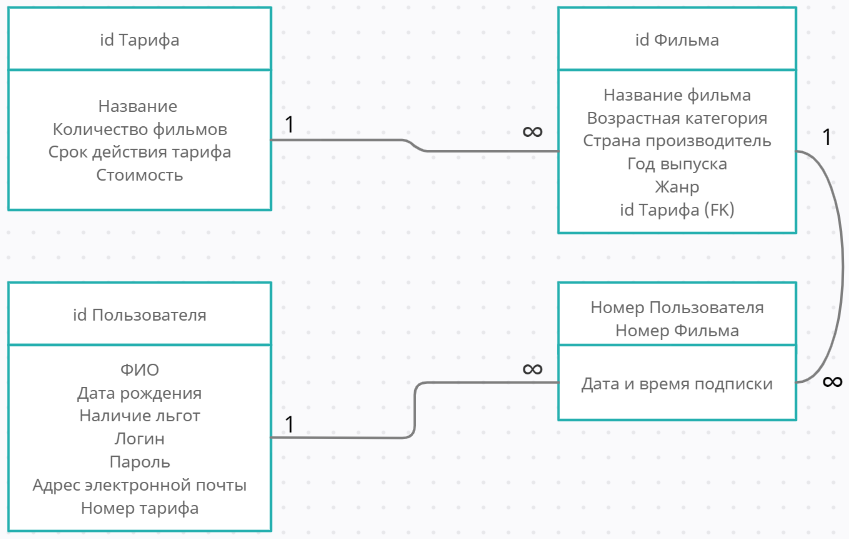
***Построение логической модели***

В таблице 6 представлен список связей в предметной области «Онлайн-кинотеатр».

Таблица 6. Список связей

|  |  |  |  |
| --- | --- | --- | --- |
| № | Сущности, участвующие в связи | Тип связи | Обоснование |
| 1 | Тарифы – Фильмы | 1:N | В одном тарифе может быть несколько фильмов, но один фильм может быть в оном тарифе. |
| 2 | Фильмы – Подписка | 1:N | Один фильм может входить в несколько подписок, но подписка должна быть у одного фильма. |
| 3 | Пользователь – Подписка | 1:N | У пользователя может быть несколько подписок на фильмы, но одна подписка может быть только у одного пользователя. |

Логическая модель предметной области «Онлайн-кинотеатр» представлена на рисунке 1.

  
Рисунок 1 – Логическая модель предметной области «Онлайн-кинотеатр»

# 2.2 Определение типов, хранимых данных

Для каждой таблицы приведем список всех атрибутов с указанием типа данных. Результаты приведены в таблицах 7-10.

Таблица 7 – Список атрибутов таблицы «Тарифы»

|  |  |
| --- | --- |
| Название | Типы данных |
| id тарифа | int |
| Название | nchar(20) |
| Количество фильмов | int |
| Срок действия тарифа | int |
| Стоимость | money |

Таблица 8. Список атрибутов таблицы «Пользователи»

|  |  |
| --- | --- |
| Название | Типы данных |
| id пользователя | int |
| ФИО | nchar(35) |
| Дата рождения | date |
| Наличие льгот | nchar(10) |
| Логин | nchar(15) |
| Пароль | nchar(15) |
| Адрес электронной почты | nchar(30) |
| Номер тарифа | int |

Таблица 9. Список атрибутов таблицы «Фильмы»

|  |  |
| --- | --- |
| Название | Типы данных |
| id фильма | int |
| Название | nchar(35) |
| Возрастная категория | nchar(10) |
| Страна производитель | nchar(20) |
| Год выпуска | int |
| Жанр | nchar(20) |
| id тарифа | int |

Таблица 10. Список атрибутов таблицы «Подписки»

|  |  |
| --- | --- |
| Название | Типы данных |
| Номер пользователя | int |
| Номер фильма | int |
| Время и дата подписки | datetime |

# 2.3 Задание ограничений целостности данных

Для каждой таблицы необходимо выделим ограничения, накладываемые на значения атрибутов. Результаты представлены в таблицах 11-14.

Таблица 11. Ограничения таблицы «Тарифы»

|  |  |
| --- | --- |
| Название | Ограничение |
| id тарифа | Первичный ключ. Это уникальное значение, не может принимать NULL-значения.  Значения автоматически генерируются СУБД, начиная от 1 с шагом 1. |
| Название | Начинается с букв русского алфавита: А-Я. |
| Количество фильмов | Значение больше 0. |
| Срок действия тарифа | Значения записываются в днях, в диапазоне от 1 до 365. Значение больше 0. |
| Стоимость | 000,0000, NOT NULL |

Таблица 12. Ограничения таблицы «Пользователи»

|  |  |
| --- | --- |
| Название | Ограничения |
| id пользователя | Первичный ключ. Это уникальное значение, не может принимать NULL-значения.  Может принимать следующие значения: 2\*\*, где \*\*-порядковый номер, начиная с 01. |
| ФИО | Начинается с букв русского алфавита: А-Я. |
| Дата рождения | ДД-ММ-ГГГГ |
| Наличие льгот | Да/Нет. Начинается с букв русского алфавита: А-Я. |
| Логин | NOT NULL |
| Пароль | NOT NULL |
| Адрес электронной почты | NOT NULL |
| Номер Тарифа | Id тарифа из существующих |

Таблица 13. Ограничения таблицы «Фильмы»

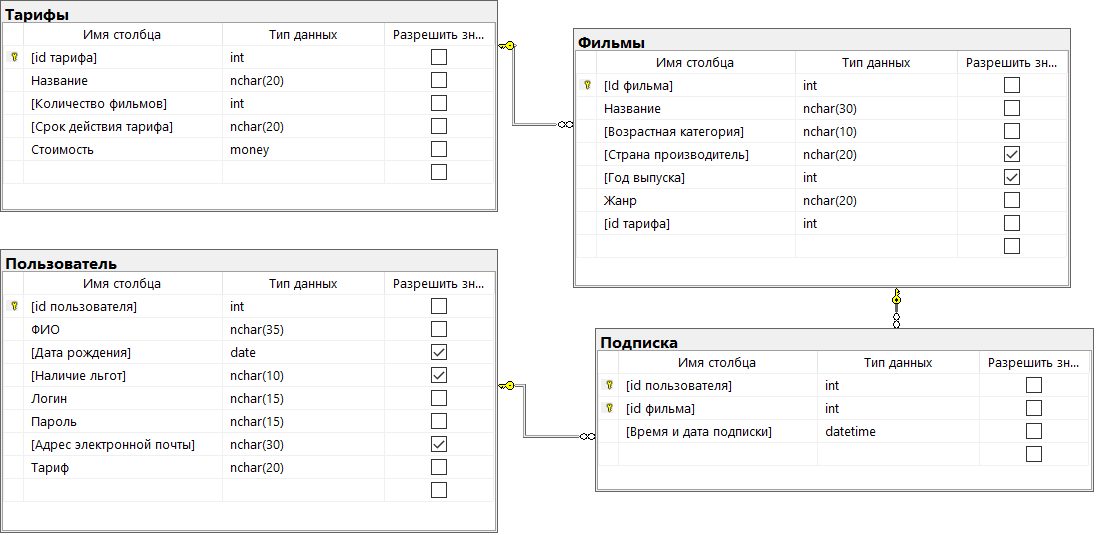
|  |  |
| --- | --- |
| Название | Ограничения |
| id фильма | Первичный ключ. Это уникальное значение, не может принимать NULL-значения.  Может принимать следующие значения: 3\*\*, где \*\*-порядковый номер, начиная с 01. |
| Название | Начинается с букв русского алфавита: А-Я. |
| Возрастная категория | От 0+ до 18+ |
| Страна производитель | Начинается с букв русского алфавита: А-Я. |
| Год выпуска | Значения принимаются от 1900 до 2099. |
| Жанр | Комедия, аниме, мультфильм, драма, боевик, ужасы и тд. |
| id тарифа | Внешний ключ. |

Таблица 14. Ограничения таблицы «Подписки»

|  |  |
| --- | --- |
| Название | Ограничения |
| Номер пользователя | Атрибут, входящий в составной первичный ключ. Атрибут, входящий во внешний ключ |
| Номер фильма | Атрибут, входящий в составной первичный ключ. Атрибут, входящий во внешний ключ. |
| Время и дата подписки | ГГГГ-ММ-ДД  чч:мм:сс |

# 2.4 Построение схемы БД

Схема БД представляет собой набор заголовков таблиц. На рисунке 2 представлена схема базы данных «Онлайн-кинотеатр».

  
Рисунок 2 – Схема базы данных «Онлайн-кинотеатр»

# 2.5 Разработка сценария для создания БД и основных объектов структуры БД

use master

CREATE DATABASE Gorbacheva\_945 ON

(NAME = 'Online\_cinema',

FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\Online\_cinema.mdf',

SIZE=1,

MAXSIZE=10,

FILEGROWTH=1)

LOG ON

(NAME = 'Gorbacheva\_945\_log',

FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\Online\_cinema\_log.ldf',

SIZE = 1,

MAXSIZE = 5,

FILEGROWTH = 1)

GO

use Gorbacheva\_945

CREATE TABLE Тариф

([id тарифа] int NOT NULL identity(1,1),

Название nchar(20) NOT NULL check (Название like '[А-Я]%'),

[Количество фильмов] int NOT NULL check ([Количество фильмов]>0) ,

[Срок действия тарифа] int NOT NULL check ([Срок действия тарифа]>0),

Стоимость money NOT NULL check (Стоимость >0),

CONSTRAINT PK\_Tarif

PRIMARY KEY ([id тарифа]))

GO

CREATE TABLE Пользователь

([id пользователя] int PRIMARY KEY,

ФИО nchar(35) NOT NULL check (ФИО like '[А-Я]%'),

[Дата рождения] date,

[Наличие льгот] nchar(10) NOT NULL check ([Наличие льгот] like '[А-Я]%'),

Логин nchar(15) NOT NULL UNIQUE,

Пароль nchar(15) NOT NULL,

[Адрес электронной почты] nchar(30),

[Номер Тарифа] int NOT NULL check ([Номер Тарифа] >0))

GO

CREATE TABLE Фильмы

([id фильма] int PRIMARY KEY,

[Название фильма] nchar(30) NOT NULL UNIQUE,

[Возрастная категория] nchar(10) NOT NULL,

[Страна производитель] nchar(20) check ([Страна производитель] like '[А-Я]%'),

[Год выпуска] int check ([Год выпуска] like '[0-9][0-9][0-9][0-9]'),

Жанр nchar(20) NOT NULL check (Жанр like '[А-Я]%'),

[ID тарифа] int NOT NULL)

GO

ALTER TABLE Фильмы

ADD CONSTRAINT FK\_тариф\_фильмы FOREIGN KEY ([id тарифа]) REFERENCES [Тариф]([id тарифа]) ON DELETE CASCADE

GO

use Gorbacheva\_945

CREATE TABLE Подписка

([Номер пользователя] int,

[Номер фильма] int,

[Время и дата подписки] datetime NOT NULL,

CONSTRAINT PK\_Podpiska

PRIMARY KEY ([Номер пользователя], [Номер фильма]))

GO

# 2.6 Разработка сценария для заполнения таблиц БД

use Gorbacheva\_945

go

insert into Тариф (Название,[Количество фильмов],[Срок действия тарифа],Стоимость)

values

('Детский', 3, 20,200),

('Семейный', 10, 30,500),

('Один дома', 5, 10,250),

('Новогодний', 6, 10,250),

('Оптимум', 12, 30,700),

('Премиум', 14, 60,1000),

('Люкс', 15, 365,3000),

('Три месяца', 15, 90,1500),

('Полгода', 15, 180,2500),

('Для друзей',10, 5,200)

use Gorbacheva\_945

go

insert into Пользователь

values

(201,'Киркоров Филипп Бедросович', '1967-04-30', 'нет', 'filkir', '1nfuo222', 'filkir@yandex.ru', 7),

(202,'Горбачева Дарья Дмитриевна', '2000-12-29', 'да', 'grbchvd', 'fhqij888', 'grbchvd@yandex.ru', 10),

(203,'Пертов Александр Сергеевич', '1997-05-09', 'нет', 'perser', 'pa5777cs', 'alex@yandex.ru', 3),

(204,'Цифрова Диана Вячеславовна','2002-09-08', 'нет', 'yuipr', 'l11248xcc', 'tsifra@gmail.com',2),

(205,'Диалова Екатерина Алексеевна', '1959-01-30', 'да', 'katuxa', 'pdytn333', NULL, 9),

(206,'Сафронов Алексей Сергеевич', '1990-12-23', 'нет', 'safron', 'uadlдда78', 'safrooo1990@mail.ru', 8),

(207,'Гагарина Полина Сергеевна', '1987-03-27', 'нет', 'gagar', 'papppadg567', 'gagar@yandex.ru', 1),

(208,'Афошкина Юлия Сергеевна','2001-03-23', 'да', 'fshkn', 'sdo9239fc', 'fshkn@yandex.ru', 2),

(209,'Горбачев Дмитрий Николаевич','1977-06-01', 'нет', 'gorbachev', 'dlls[dfn782', 'gorbachev@yandex.ru',5),

(210,'Митина Зоя Станиславовна', NULL, 'нет', 'miti', 'erjwpsnv395', 'motti@yandex.ru', 6)

use Gorbacheva\_945

go

insert into Фильмы

values

(301,'Форрест Гамп','12+','США',1994,'Драма',2),

(302,'1+1','16+','Франция',2011,'Комедия',8),

(303, 'Тайна Коко', '12+', 'США', 2017, 'Мультфильм',1),

(304, 'Достать ножи', '16+', 'США', 2019, 'Детектив',2),

(305, 'Шрек', '6+', 'США', 2001, 'Фентези',7),

(306, 'Престиж', '12+', 'США', 2006, 'Триллер',9),

(307, 'Поймай меня, если сможешь', '12+', 'Канада', 2002, 'Комедия',7),

(308, 'Остров проклятых', '16+', 'США', 2009, 'Триллер',10),

(309, 'Джентельмены удачи', '6+', 'СССР', 1971, 'Комедия',5),

(310, 'Дьявол носит Prada', '16+', 'США', 2006, 'Драма',3),

(311, 'Один дома', '12+', 'США', 1990, 'Комедия',4),

(312, 'Дневник Бриджит Джонс', '16+', 'Франция', 2001, 'Драма',6),

(313, 'Джанго освобожденный', '18+', 'США', 2012, 'Принкоючение',10),

(314, 'Ford против Ferrari', '16+', 'США', 2019, 'Биография',2),

(315, 'Бриллиантовая рука', '6+', 'СССР', 1968, 'Комедия',7),

(316, 'Джентельмены', '18+', 'США', 2020, 'Боевик',8)

use Gorbacheva\_945

go

insert into Подписка

values

(201,307,'2020-12-06T15:07:33'),

(202,308,'2020-12-01T15:29:45'),

(203,310,'2020-11-06T20:54:29'),

(204,301,'2020-11-26T21:45:19'),

(205,306,'2020-11-23T16:17:17'),

(206,316,'2020-12-05T14:56:09'),

(207,303,'2020-12-05T17:38:52'),

(208,314,'2020-12-07T16:58:23'),

(209,311,'2020-11-15T11:00:25'),

(210,306,'2020-12-10T22:37:30'),

(203,311,'2020-10-15T00:00:25'),

(201,306,'2020-01-10T03:09:48'),

(204,311,'2020-04-26T12:48:23'),

(205,315,'2020-10-23T14:12:10'),

(209,316,'2020-09-05T19:50:09'),

(201,303,'2020-08-05T10:01:53'),

(208,310,'2020-04-07T16:49:23'),

(209,308,'2020-05-15T19:01:25'),

(202,304,'2020-11-10T23:30:30'),

(205,302,'2020-01-09T23:30:30')

# 3 Разработка основных операций реляционной алгебры

1. Вывести полную информацию о фильмах, произведенных в СССР.

R = Фильмы WHERE [Страна производителя] = ‘СССР’

1. Вывести ФИО, логин и электронную почту пользователей, у которых есть льготы.

R= (Пользователь WHERE [Наличие льгот] =’да’) [ФИО, Логин, Электронная почта]

1. Вывести список пользователей, у которых не известна дата рождения.

R= (Пользователь WHERE [Дата рождения] = NULL) [ФИО]

1. Вывести информацию о дате и времени подписки и ФИО пользователя.  
   R1 = Пользователь RENAME [id пользователя] AS ФИО  
   R2 = (R1 JOIN Подписка) [Время и дата подписки, ФИО]
2. Вывести название тарифа, на который никто из пользователей не подписан.

R1 = Тариф [id тарифа]

R2 = Пользователь [Номер тарифа]

R3 = R1 EXCEPT R2  
R4 = (R3 JOIN Тариф) [Название]

1. Вывести информацию о пользователях, у которых не указан адрес электронной почты и не указана дата рождения.

R1=(Пользователь WHERE [Адрес электронной почты]=NULL)  
R2=(Пользователь WHERE [Дата рождения]=NULL)

R3=R1 UNION R2

1. Вывести информацию о стоимости тарифа 200 и 1000.

R1= (Тариф WHERE Стоимость = 200)

R2= (Тариф WHERE Стоимость = 1000)

R3= R1 UNION R2

1. Вывести название тарифа, у которого ID тарифа равно 102 и в то же время срок действия тарифа был равен 30.  
   R1= (Тариф WHERE [ID тарифа]=102) [Название]  
   R2= (Тариф WHERE [Срок действия тарифа]=30) [Название]

R3= R1 INTERSECT R2

1. Вывести информацию о дате рождения пользователей, у которых есть льготы и вычесть из них информацию о пользователе Афошкина Юлия Сергеевна.

R1 = (Пользователь WHERE [Наличие льгот] = ‘Да’)[Дата рождения]  
R2 = (Пользователь WHERE ФИО = ‘Афошкина Юлия Сергеевна’) [Дата рождения]  
R3 = R1 EXCEPT R2

1. Вывести названия фильмов, входящие в тарифы, у которых срок его действия больше 60.  
   R1 = ([Тариф] WHERE [Срок действия тарифа] > 60)[ID тарифа]  
   R2 = (R1 JOIN Фильм)[Название фильма]
2. Получить информацию о пользователях в следующем виде:

|  |  |  |  |
| --- | --- | --- | --- |
| ФИО | Логин | Id тарифа | Номер фильма |

R = (Пользователь JOIN Подписка) [ФИО, Логин, [id тарифа], [Номер фильма]]

1. Вывести подписки пользователей, которые есть у тех, кто имеет льготы.  
   R1= (Пользователь WHERE [Наличие льгот] = ‘да’)[ID пользователя]  
   R2 = (R1 JOIN Подписки)[ ID фильма, [Дата и время подписки]]
2. Вывести страны производителей фильмов, которые выпущены позднее 2011 года.  
   R=(Фильмы WHERE [Год выпуска] >2011) [[Название фильма], [Страна производитель]]
3. Вывести даты и время подписки на фильмы, пользователи которых не имеют льгот, а также даты и время подписок на фильмы, которые были выпущены до 1999 года.

R1= (Пользователи WHERE [Наличие льгот] = ‘Нет’) [ID пользователя]

R2 = (Фильмы WHERE [Год выпуска] <1999) [ID фильма]

R3 = R1 JOIN Подписки

R4 = R2 JOIN Подписки

R5 = R3 UNION R4

1. Вывести полную информацию о фильмах, на которые подписаны пользователи и жанр которых – комедия.

R1= (Подписка)[ ID фильма]

R2=(Фильмы WHERE жанр=’Комедия’)[ID фильма]

R3=R1 EXCEPT R2

R4 = (R3 JOIN Фильмы) [[ ID фильма],[Название фильма],[Год выпуска],[Страна производитель],Жанр,[id тарифа]]

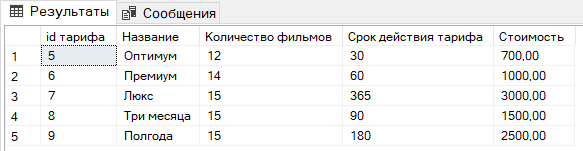
# 4 Разработка запросов к БД для реализации информационных потребностей пользователя

1. Вывести полную информацию обо всех тарифах, цена которых больше 500.

SELECT \*

FROM Тариф

WHERE Стоимость > 500

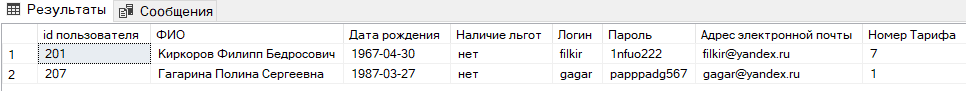
  
Рисунок 3 – Запрос с использованием операторов сравнения

1. Вывести информацию о пользователях Киркорове и Гагариной.

SELECT \*

FROM Пользователь

WHERE ФИО = 'Киркоров Филипп Бедросович' or ФИО='Гагарина Полина Сергеевна'

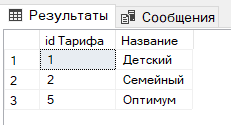
  
Рисунок 4 – Запрос с использованием OR

1. Вывести id тарифа, у длительность которых больше 10 дней и меньше 50.

SELECT [id Тарифа], [Название]

FROM Тариф

WHERE [Срок действия тарифа]>10 and [Срок действия тарифа]<50

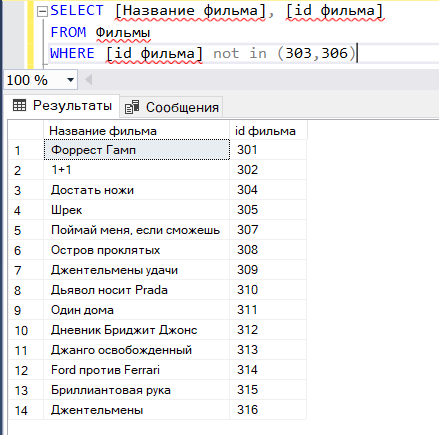
  
Рисунок 5 – Запрос с использованием AND

1. Вывести название фильма, id которых не равны 306 и 303.

SELECT [Название фильма], [id фильма]

FROM Фильмы

WHERE [id фильма] not in (303,306)

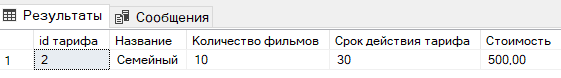
  
Рисунок 6 – Запрос с использованием NOT

1. Вывести информацию о тарифах, стоимость подписки на которые равна 500 и 1000, а длительность равна 30.

SELECT \*

FROM Тариф

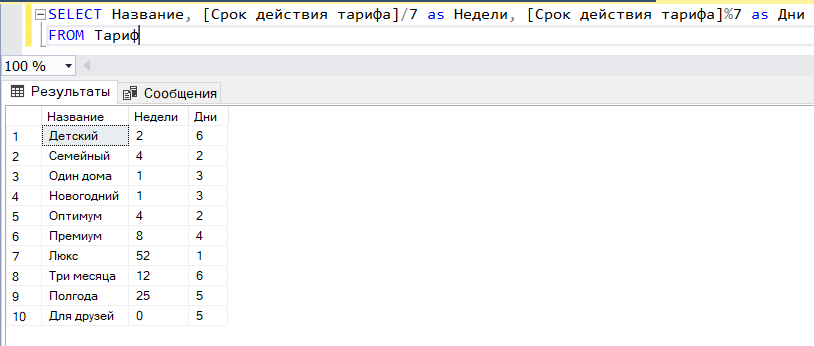
WHERE (Стоимость=500 or Стоимость=1000) and [Срок действия тарифа]=30

  
Рисунок 7 – Запрос на использование комбинации логических операторов

1. Вывести название всех тарифов с указанием их сроков действия в сутках и неделях.

SELECT Название, [Срок действия тарифа]/7 as Недели, [Срок действия тарифа]%7 as Дни

FROM Тариф

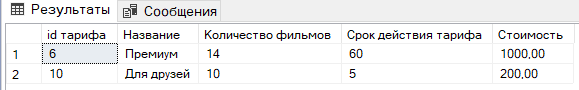
  
Рисунок 8 – Запрос на использование выражений над столбцами

1. Получить информацию о тарифах с id равным 110 и 106.

SELECT \*

FROM Тариф

WHERE [id тарифа] in (6,10)

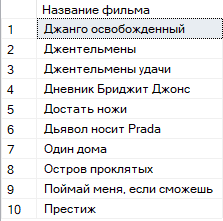
  
Рисунок 9 – Запрос с проверкой на принадлежность множеству

1. Вывести названия фильмов, начинающихся с букв от 'Г' по 'Т'.

SELECT [Название фильма]

FROM Фильмы

WHERE [Название фильма] BETWEEN 'Г' AND 'Т'

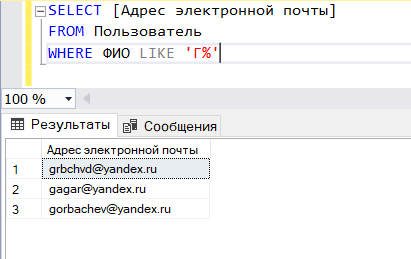
  
Рисунок 10 – Запрос с проверкой на принадлежность диапазону значений

1. Вывести адреса электронной почты пользователей, ФИО которых начинается на букву ‘Г’.

SELECT [Адрес электронной почты]

FROM Пользователь

WHERE ФИО LIKE 'Г%'

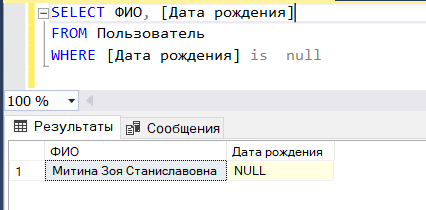
  
Рисунок 11 – Запрос с проверкой на соответствие шаблону

1. Вывести ФИО пользователей, у которых не указана дата рождения.

SELECT ФИО, [Дата рождения]

FROM Пользователь

WHERE [Дата рождения] is null

  
Рисунок 12 – запрос с проверкой на неопределенное значение

1. Вывести ФИО, адрес электронной почты и логин всех пользователей. Если адрес электронной почты не указан, то в соответствующем столбце вывести строку: «отсутствует».

SELECT ФИО, [Адрес электронной почты], Логин

FROM Пользователь

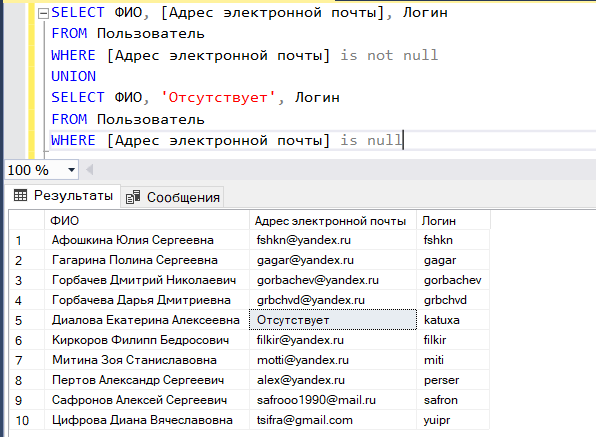
WHERE [Адрес электронной почты] is not null

UNION

SELECT ФИО, 'Отсутствует', Логин

FROM Пользователь

WHERE [Адрес электронной почты] is null

  
Рисунок 13 – Запрос на объединение таблиц

1. Вывести коды фильмов, у которых год выпуска больше 1999 года, и меньше 2019 года.

SELECT [id фильма]

FROM Фильмы

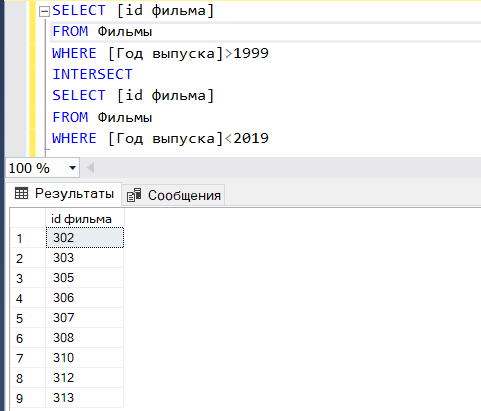
WHERE [Год выпуска]>1999

INTERSECT

SELECT [id фильма]

FROM Фильмы

WHERE [Год выпуска] <2019

  
Рисунок 14 – Запрос на пересечение таблиц

1. Вывести названия тарифов, которые никто из пользователей не приобрел.

SELECT [id тарифа]

FROM Тариф

EXCEPT

SELECT [Номер тарифа]

FROM Пользователь

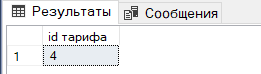


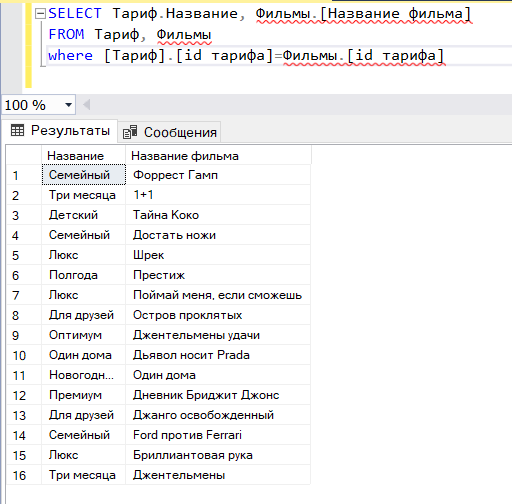
Рисунок 15 – Запрос на вычитание таблиц

1. Вывести названия тарифов для фильмов.

SELECT Тариф.Название, Фильмы.[Название фильма]

FROM Тариф, Фильмы

where [Тариф].[id тарифа]=Фильмы.[id тарифа]

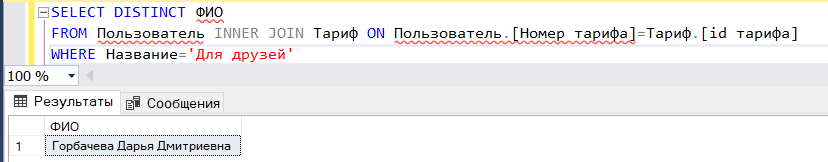
  
Рисунок 16 – Декартово произведение таблиц

1. Вывести ФИО пользователя, который подписался на тариф с названием «Для друзей».

SELECT DISTINCT ФИО

FROM Пользователь INNER JOIN Тариф ON Пользователь.[Номер тарифа]=Тариф.[id тарифа]

WHERE Название='Для друзей'

  
Рисунок 17 – Запрос с использованием соединения двух таблиц по равенству и условием отбора

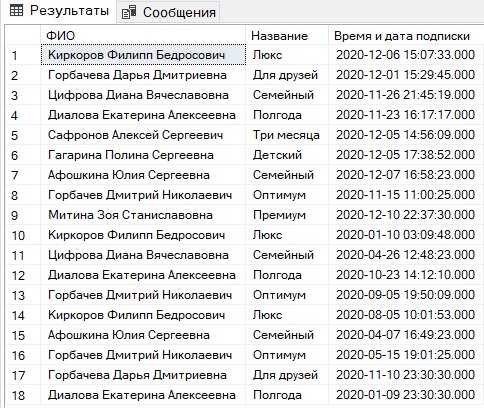
1. Вывести ФИО пользователей, названия тарифов, а также дату и время подписки, исключив информацию о тарифе «Один дома».

SELECT ФИО, Название, [Время и дата подписки]

FROM Пользователь INNER JOIN Тариф ON Пользователь.[Номер тарифа]=Тариф.[id тарифа]

INNER JOIN Подписка ON Пользователь.[ID пользователя]=Подписка.[Номер пользователя]

WHERE Название <> 'Один дома'

  
Рисунок 18 – Запрос с использованием соединения трех таблиц по равенству и условием отбора

1. Вывести ФИО пользователя, id тарифа, Название фильма и Время и дату подписки пользователя на фильмы «Тайна Коко», «Престиж», «Ford против Ferrari».

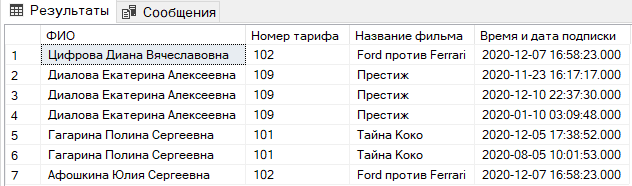
SELECT ФИО, [Номер тарифа], [Название фильма], [Время и дата подписки]

FROM Тариф INNER JOIN Пользователь ON Тариф.[id тарифа]=Пользователь.[Номер тарифа]

INNER JOIN Фильмы ON Тариф.[id тарифа]=Фильмы.[ID тарифа]

INNER JOIN Подписка ON Фильмы.[id фильма]=Подписка.[Номер фильма]

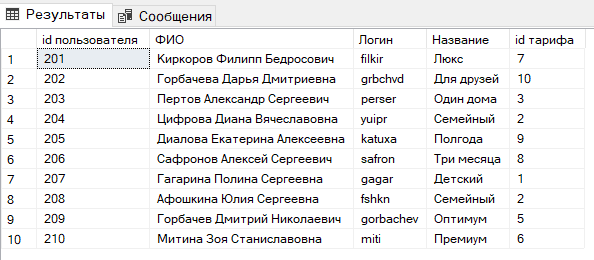
WHERE [Название фильма] = 'Тайна Коко' OR [Название фильма] = 'Престиж' OR [Название фильма] = 'Ford против Ferrari'

  
Рисунок 19 – Запрос с использованием соединения по четырем таблицам

1. Вывести информацию обо всех существующих пользователях, указав их тарифы.

SELECT [id пользователя], ФИО, Логин, Название, [id тарифа]

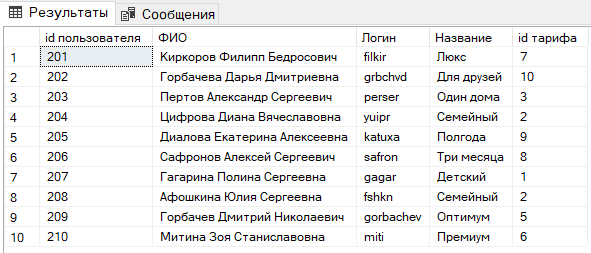
FROM Пользователь left join Тариф on Пользователь.[Номер тарифа]=Тариф.[id тарифа]

  
Рисунок 20 – Запрос с использованием левого внешнего соединения

1. Вывести информацию обо всех существующих пользователях, указав их тарифы.

SELECT [id пользователя], ФИО, Логин, Название, [id тарифа]

FROM Тариф right join Пользователь on Тариф.[id тарифа]=Пользователь.[Номер тарифа]

  
Рисунок 21 – Запрос с использованием правого внешнего соединения

1. Вывести ФИО пользователей и их количество подписок.

SELECT ФИО, COUNT (Подписка.[Номер пользователя]) as [Кол-во фильмов]

FROM Пользователь, Подписка

WHERE Пользователь.[id пользователя]=Подписка.[Номер пользователя]

GROUP BY Пользователь.ФИО

  
Рисунок 22 – Запрос с использованием функции COUNT

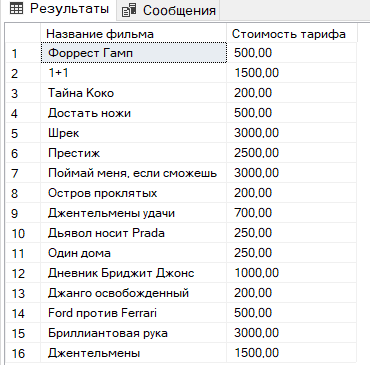
1. Найти сколько заплатит пользователь, если купит все фильмы.

SELECT [Название фильма], SUM (Тариф.Стоимость) as [Стоимость тарифа]

FROM Фильмы, Тариф

WHERE Фильмы.[id тарифа]=Тариф.[id тарифа]

GROUP BY Фильмы.[Название фильма]

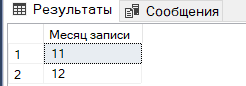
  
Рисунок 23 – Запрос с использованием функции SUM

1. Вывести месяц, когда подписалась на онлайн-кинотеатр Горбачева Дарья Дмитриевна.

SELECT DISTINCT MONTH([Время и дата подписки]) AS [Месяц записи]

FROM Пользователь INNER JOIN Подписка ON Пользователь.[ID пользователя]=Подписка.[Номер пользователя]

WHERE ФИО='Горбачева Дарья Дмитриевна'

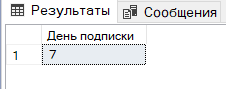
  
Рисунок 24 – Запрос с использованием временных функций

1. Вывести день, когда подписалась на онлайн-кинотеатр Афошкина Юлия Сергеевна.

SELECT DISTINCT DAY([Время и дата подписки]) AS [День подписки ]

FROM Пользователь INNER JOIN Подписка ON Пользователь.[ID пользователя]=Подписка.[Номер пользователя]

WHERE ФИО='Афошкина Юлия Сергеевна'

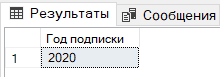
  
Рисунок 25 – Запрос с использованием временных функций

1. Вывести год, когда подписался на онлайн-кинотеатр Киркоров Филипп Бедросович.

SELECT DISTINCT YEAR([Время и дата подписки]) AS [Год подписки ]

FROM Пользователь INNER JOIN Подписка ON Пользователь.[ID пользователя]=Подписка.[Номер пользователя]

WHERE ФИО='Киркоров Филипп Бедросович'

  
Рисунок 26 – Запрос с использованием временных функций

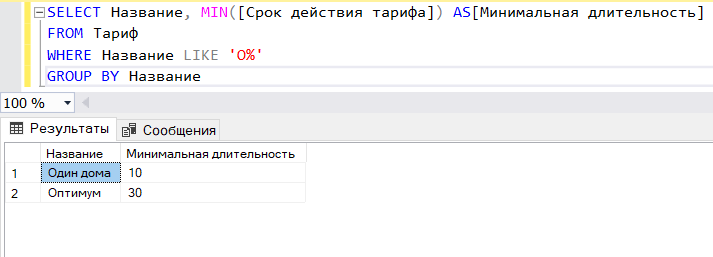
1. Вывести название и минимальную длительность тарифа, названия которых начинаются с буквы О.

SELECT Название, MIN([Срок действия тарифа]) AS[Минимальная длительность]

FROM Тариф

WHERE Название LIKE 'О%'

GROUP BY Название

  
Рисунок 27 – Запрос с использованием группировки по одному столбцу

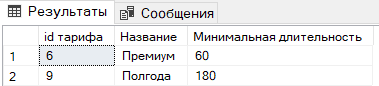
1. Вывести название, код тарифа и минимальную длительность тарифа, названия которых начинаются с буквы П.

SELECT [id тарифа], Название, MIN([Срок действия тарифа]) AS[Минимальная длительность]

FROM Тариф

WHERE Название LIKE 'П%'

GROUP BY [id тарифа], Название

  
Рисунок 28 – Запрос на использование группировки по нескольким столбцам

1. Вывести количество подписок тех пользователей, которые покупали подписку не менее двух раз.

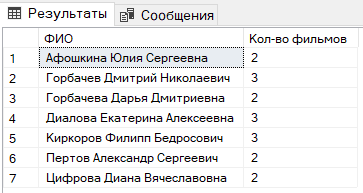
select Пользователь.ФИО, COUNT (Подписка.[Номер фильма]) As [Кол-во фильмов]

from Пользователь, Подписка

where Пользователь.[id пользователя]=Подписка.[Номер пользователя]

group by Пользователь.ФИО

having COUNT(Подписка.[Номер фильма])>=2

  
Рисунок 29 – Запрос с использованием условия отбора групп HAVING

1. Вывести ФИО пользователей в порядке позрастания в зависимости от количества приобретенных подписок.

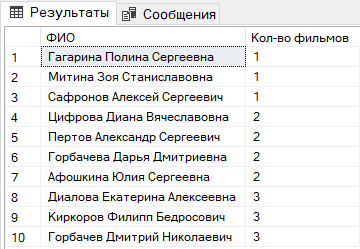
SELECT Пользователь.ФИО, COUNT (Подписка.[Номер фильма]) AS [Кол-во фильмов]

FROM Пользователь, Подписка

WHERE Пользователь.[id пользователя]=Подписка.[Номер пользователя]

GROUP BY Пользователь.ФИО

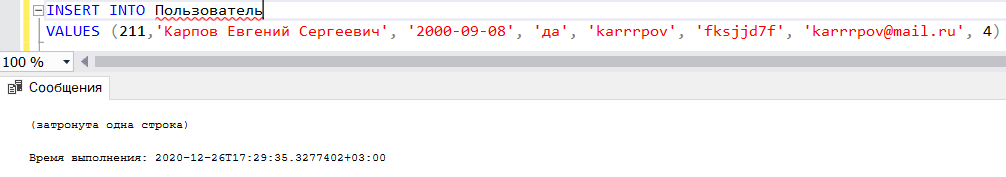
ORDER BY COUNT (Подписка.[Номер фильма]) ASC

  
Рисунок 30 – Запрос с использованием сортировки по столбцу

1. Добавить новую строку в таблицу «Пользователь».

INSERT INTO Пользователь

VALUES (211,'Карпов Евгений Сергеевич', '2000-09-08', 'да', 'karrrpov', 'fksjjd7f', 'karrrpov@mail.ru', 4)

  
Рисунок 31 – Запрос на добавление новых данных в таблицу

1. Создать таблицу «Фильм «Шрек»», поместив в нее информацию о фильме «Шрек».

use [Gorbacheva\_945]

CREATE TABLE [Фильм "Шрек"]

([id фильма] int,

[Название фильма] nchar(30) NOT NULL UNIQUE,

[Возрастная категория] nchar(10) NOT NULL,

[Страна производитель] nchar(20) check ([Страна производитель] like '[А-Я]%'),

[Год выпуска] int check ([Год выпуска]>0),

Жанр nchar(20) NOT NULL check (Жанр like '[А-Я]%'),

[id тарифа] int NOT NULL)

GO

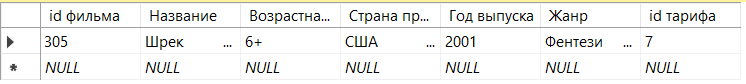
INSERT INTO [Фильм "Шрек"]

SELECT \*

FROM Фильмы

WHERE [Название фильма]='Шрек'

go

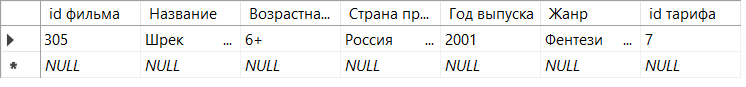
  
Рисунок 32 – Запрос на добавление новых данных по результатам выполненного запроса

1. Изменить название страны производителя «США» на «Россию» в таблице «Фильм «Шрек»».

UPDATE [Фильм "Шрек"]

set [Страна производитель] ='Россия'

WHERE [Страна производитель] ='США'

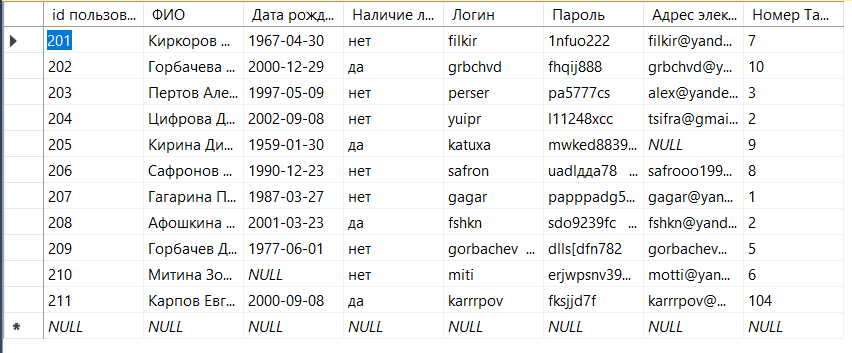
  
Рисунок 33 – Запрос на обновление существующих данных в таблице

1. Изменить ФИО и пароль Диаловой Дианы Вячеславовны, в связи с ее сменой фамилии.

UPDATE Пользователь

set ФИО='Кирина Диана Вячеславовна', Пароль='mwked8839'

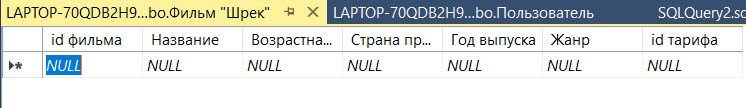
WHERE [id пользователя]=205

  
Рисунок 34 – Запрос на обновление существующих данных по результатам подзапроса во фразе WHERE

1. Удалить все записи из таблицы «Фильм «Шрек»».

DELETE

FROM [Фильм "Шрек"]

  
Рисунок 35 – Запрос на удаление существующих данных

1. Добавить скидку 20% на стоимость тарифа, на который подписана Цифрова Диана Вячеславовна.

UPDATE Тариф

SET Стоимость=Стоимость\*0.8

WHERE [id тарифа] in (SELECT [Номер тарифа]

FROM Пользователь

WHERE ФИО='Цифрова Диана Вячеславовна')

  
Рисунок 36 – Запрос на удаление существующих данных по результатам подзапроса во фразе WHERE

1. Удалить тарифы, пользователи которых родились позже 2002-09-01

DELETE [Тариф]

FROM Пользователь INNER JOIN Тариф ON Пользователь.[Номер тарифа]=Тариф.[ID Тарифа]

WHERE [Дата рождения]> '2002-09-01'

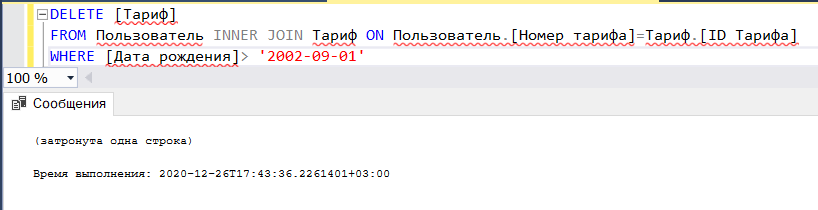


Рисунок 37 – запрос на удаление существующих данных из одной таблицы на основе связанных с ней таблиц

1. Вывести дату и время подписки на тариф пользователя, дата рождения которого равна 1977-06-01.

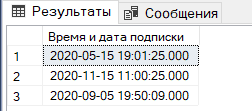
SELECT [Время и дата подписки]

FROM Подписка

WHERE [Номер пользователя] = (SELECT [id пользователя]

FROM Пользователь

WHERE [Дата рождения]='1977-06-01')

  
Рисунок 38 – Запрос с использованием операций сравнения

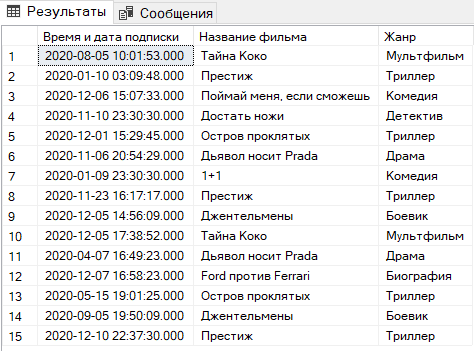
1. Вывести информацию о фильмах, а также о дате и времени подписки, год выпуска которых не ниже, чем средний год выпуска фильма.

SELECT Подписка.[Время и дата подписки], Фильмы.[Название фильма], Фильмы.Жанр

FROM Подписка INNER JOIN Фильмы ON Подписка.[Номер фильма]=Фильмы.[id фильма]

WHERE [Год выпуска] >=(SELECT AVG ([Год выпуска])

FROM Фильмы)

  
Рисунок 39 – Запрос с использованием операций сравнения

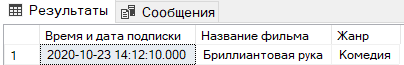
1. Вывести информацию о фильмах, а также о дате и времени подписки, год выпуска которых минимальный.

SELECT Подписка.[Время и дата подписки], Фильмы.[Название фильма], Фильмы.Жанр

FROM Подписка INNER JOIN Фильмы ON Подписка.[Номер фильма]=Фильмы.[id фильма]

WHERE [Год выпуска] =(SELECT MIN ([Год выпуска])

FROM Фильмы)

  
Рисунок 40 – Запрос с использованием операций сравнения

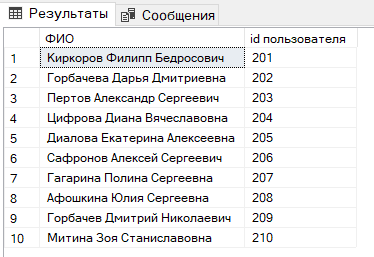
1. Вывести ФИО пользователей, которые подписаны на Онлайн-кинотеатр, судя по подписке.

SELECT ФИО, [id пользователя]

FROM Пользователь

WHERE [id пользователя] = any(SELECT [Номер пользователя]

FROM Подписка)

  
Рисунок 41 – Запрос с использованием операции ANY

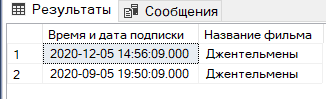
1. Вывести информацию о тарифах, стоимость которых не меньше, чем стоимость каждого из тарифов, существующий в онлайн-кинотеатре.

SELECT Подписка.[Время и дата подписки], Фильмы.[Название фильма]

FROM Подписка INNER JOIN Фильмы ON Подписка.[Номер фильма]=Фильмы.[id фильма]

WHERE [Год выпуска] >= all (SELECT [Год выпуска]

FROM Фильмы)

  
Рисунок 42 – Запрос с использованием операции ALL

1. Выведем список тарифов, на которые не подписан ни один пользователь.

SELECT Название, [id тарифа]

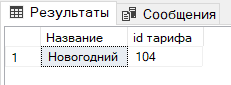
FROM Тариф

WHERE [id тарифа] IN

(SELECT Тариф.[id тарифа]

FROM Пользователь full JOIN Тариф ON Пользователь.[Номер тарифа] = Тариф.[id тарифа]

WHERE [Номер тарифа] IS NULL)

  
Рисунок 43 – Запрос с использованием операции IN

1. Выведем информацию о тех фильмах, на которые в настоящий момент хотя бы один пользователь подписан.

SELECT \*

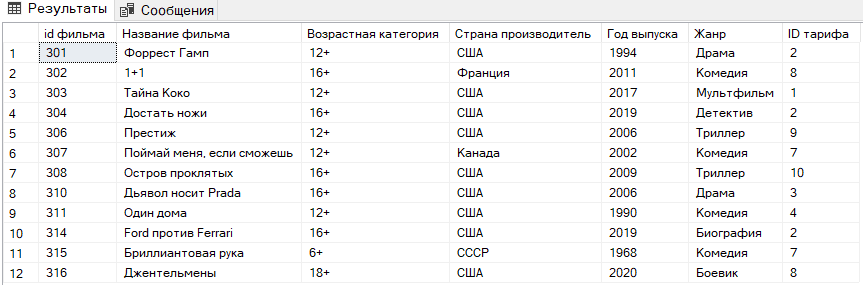
FROM Фильмы

WHERE EXISTS

(SELECT \*

FROM Подписки

WHERE Фильмы.[id фильма] = Подписки.[id фильма])

  
Рисунок 44 – Запрос с использованием операции EXISTS

1. Вывести пользователей, которые подписаны на фильм 2019 года.

SELECT ФИО

FROM Пользователь

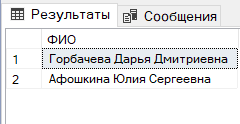
WHERE [id пользователя] IN (SELECT [Номер пользователя]

FROM Подписка

WHERE [Номер фильма] IN (SELECT [id фильма]

FROM Фильмы

WHERE [Год выпуска] = 2019))

  
Рисунок 45 – Запрос с использованием двух вложенных друг в друга подзапросов

1. Вывести пользователей, которые подписаны на тариф «Для друзей»

SELECT DISTINCT ФИО

FROM Пользователь

WHERE [id пользователя]IN(SELECT [Номер пользователя]

FROM Подписка

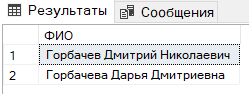
WHERE [Номер фильма]IN(SELECT [id фильма]

FROM Фильмы

WHERE [id тарифа]IN (SELECT [id тарифа]

FROM Тариф

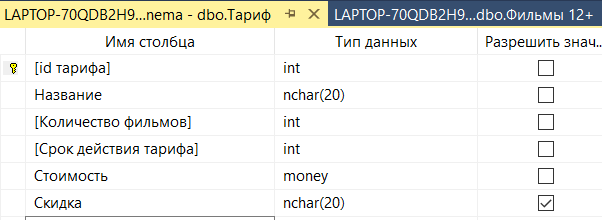
WHERE Название='Для друзей')))

  
Рисунок 46 – Запрос с использованием трех вложенных друг в друга подзапросов

1. Добавить к таблице «Тариф» столбец «Скидка».

ALTER TABLE Тариф

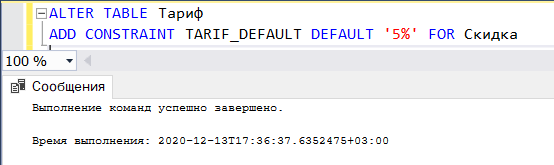
ADD Скидка nchar(20)

  
Рисунок 47 – Запрос на добавление нового столбца к таблице

1. Добавить ограничение по умолчанию в таблицу «Тариф», где скидка тарифа будет равна 5%.

ALTER TABLE Тариф

ADD CONSTRAINT TARIF\_DEFAULT DEFAULT '5%' FOR Скидка

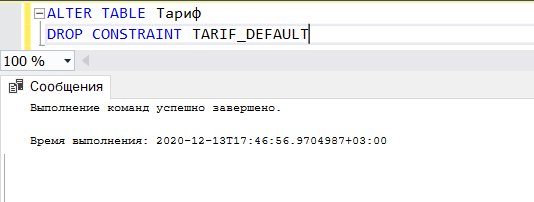
  
Рисунок 48 – Запрос на добавление нового ограничения к таблице

е

1. Удалить ограничение, которое было установлено в прошлом запросе.

ALTER TABLE Тариф

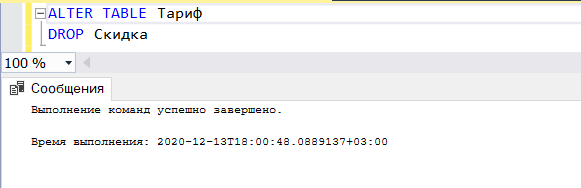
DROP CONSTRAINT TARIF\_DEFAULT  
GO

  
Рисунок 49 – Запрос на удаление нового ограничения к таблице

1. Удалить столбец «Скидка» из таблицы «Тариф».

ALTER TABLE Тариф

DROP Скидка  
GO

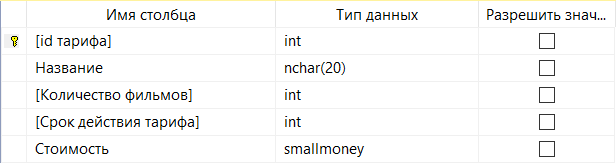
  
Рисунок 50 – Запрос на удаление нового столбца к таблице

1. Изменить тип данных у Стоимости в таблице «Тариф» с money на smallmoney.

ALTER TABLE Тариф

ALTER COLUMN Стоимость smallmoney

GO

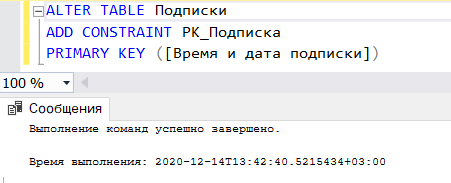
  
Рисунок 51 – Запрос на изменение типа данных в таблице

1. Добавить первичный ключ в таблицу «Пользователь».

ALTER TABLE Подписка

ADD CONSTRAINT PK\_Подписка

PRIMARY KEY ([Время и дата подписки])

  
Рисунок 52 – Запрос на добавление нового первичного ключа к таблице с сохранением целостности таблицы

# Заключение

В результате данной курсовой работы была создана база данных «Онлайн -кинотеатр».

Также были разработаны основных операций реляционной алгебры и запросы для реализации информационных потребностей пользователя, а также для обеспечения целостности данных. В данной работе для выполнения поставленной задачи были использованы возможности СУБД Microsoft SQL Server 2018.

# Список литературы

1. Р. Виейра. Программирование баз данных Microsoft SQL Server 2005. Базовый курс — М.: «Диалектика», 2007. —832 c.
2. Клайн К. SQL. Справочник. 2-е издание / Пер. с англ. – М.: КУ-ДИЦ-ОБРАЗ, 2006 – 832 с.
3. Ицик Бен-Ган Microsoft SQL Server 2008 Основы T-SQL:Пер. с англ. – СПб:БВХ-Петербург, 2009. – 432с. :ил.
4. Глушаков С.В., Ломотько Д.В. Базы данных: Учебный курс / Харьков, Ростов-на-Дону, Киев,2002
5. Токмаков Г.П. Базы данных. Концепция баз данных, реляционная модель данных, языки SQL и XML. – УлГТУ, 2010. - 193 с.