МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ 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:

Кинопробы»

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

Рязань 2020

*СОДЕРЖАНИЕ*

[Введение 3](#_Toc58887885)

[1 Семантическое описание предметной области 5](#_Toc58887888)

[2 Разработка базы данных 7](#_Toc58887889)

[2.1 Разработка логической модели данных 7](#_Toc58887890)

[2.2 Определение типов хранимых данных 8](#_Toc58887891)

[2.3 Задание ограничений целостности данных 14](#_Toc58887892)

[2.4 Построение схемы данных 18](#_Toc58887893)

[2.5 Разработка сценария для создания БД и основных объектов структуры БД 18](#_Toc58887894)

[2.6 Разработка сценария для заполнения таблиц БД 20](#_Toc58887895)

[3 Разработка основных операций реляционной алгебры 22](#_Toc58887896)

[4 Разработка запросов к БД для реализации информационных потребностей пользователя 26](#_Toc58887897)

[Заключение 58](#_Toc58887934)

[Список литературы 59](#_Toc58887935)

# ВВЕДЕНИЕ

Потребность в информации существует всегда и в любой сфере. Еще до изобретения компьютера люди записывали многочисленные таблицы в тетрадях и вели статистику, а поиск нужных данных занимал большое количество времени. С появлением вычислительных машин хранение информации и управление ей перешло на новый уровень - появились компьютерные базы, способные хранить до тысячи терабайт данных. Главная проблема любой организации - создание идеально работающей базы данных, не имеющей повторных строк и способной выдать любую нужную пользователю информацию без ошибок. Эта проблема была и остаётся актуальной с момента появления компьютерных баз данных.

Одним из важнейших условий обеспечения эффективного функционирования любой организации является наличие развитой информационной системы (ИС).

**Информационная система**представляет собой систему, реализующую автоматизированный сбор, хранение, обработку и представление информации. В общем случае информационная система включает в себя одну или несколько БД, программное и аппаратное обеспечение, а также обслуживающий персонал.

**Базой данных**называется поименованная совокупность данных, организованная по определенным правилам, которые включают общие принципы описания, хранения и манипулирования данными. В базе данных хранится информация об объектах предметной области (сущностях) и связях между ними.

Основными компонентами ПО информационной системы являются СУБД и пользовательские прикладные программы или клиентские приложения (КП).  
**СУБД**представляет собой пакет прикладных программ, предназначенных для создания, сопровождения и использования БД.

Клиентские приложения служат для обработки данных и представления информации в удобном для пользователя виде.

Целью данной курсовой работы является создание реляционной базы данных для хранения информации и работы с ней. Предметная область курсовой работы - кинопробы на новый фильм. В базу должны входить различные данные об актерах, пробах, ролях. Необходимо предусмотреть то, чтобы пользователь мог удобно пользоваться этой базой данных и получать необходимую ему информацию.

**1 СЕМАНТИЧЕСКОЕ ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ**

Режиссер в поиске кандидатов для своего нового фильма. Для этого он проводит кинопробы, чтобы понять, какой актер лучше подходит на определенную роль.

Требуется разработать информационную систему для автоматизации учета получения информации об актерах, их ролях и утверждении их на определенную роль. База данных должна предусматривать просмотр информации об актерах, о ролях, также информацию на какую роль актер пробуется и утвержденный список, в котором указывается какому актеру досталась какая роль в новом фильме. Каждая роль может быть сыграна разными актерами, которые могут быть утверждены на эту роль или нет. Могут встретиться актеры с одинаковыми фамилиями, и для этого необходим уникальный идентификатор, который будет различать актеров.

Прежде чем, проводится кинопроба, режиссер выкладывает список ролей, на которые будет проводиться проба. В списке указывается:

1. Номер роли

2. Роли

3.Вид роли (главная роль, второстепенная роль, массовые съемки)

4.Описание роли (положительная или отрицательная)

На кинопробе каждый актер заполняет следующую анкету, чтобы потом его могли идентифицировать:

1. IP

2. ФИО

3. Дата рождения

4. Телефон (Телефон, по которому можно связаться с актером)

5. Образование (Высшее, среднее общее, среднее специальное)

6. Стаж (Актерский стаж)

7. Звание (Заслуженный артист или нет)

Далее составляется список, в котором указывается актер и роль, на которую он пробуется, в какое время пробуется и количество дублей. Чтобы упростить работу сотрудникам пишется не ФИО актера и не название роли, а уникальные идентификаторы, которые помогают оптимизировать работу:

1. Актер (IP актера)

2. Роль (Номер роли)

3. Дата и время (Дата и время пробы)

4. Количество дублей

После кинопробы должен присутствовать результирующий список об утверждении на роли, который сформирован тоже с помощью уникальных идентификаторов:

1. Актер (IP актера)

2. Роль (Номер роли)

3. Гонорар (Размер дохода от сыгранной роли)

4. Дата и время начала съемок

Таким образом, требуется разработать базу данных «Кинопробы», в которой будут храниться 4 таблицы:

1. Актеры (IP, ФИО, Дата рождения, телефон, Образование, Звание, Стаж)

2. Роль (Номер, Название роли, Вид, Описание)

3. Кинопробы (Актер, Роль, Дата и время, Количество дублей)

4. Утверждены (Актер, Роль, Гонорар, Дата и время начала съемок)

**2 РАЗРАБОТКА БАЗЫ ДАННЫХ**

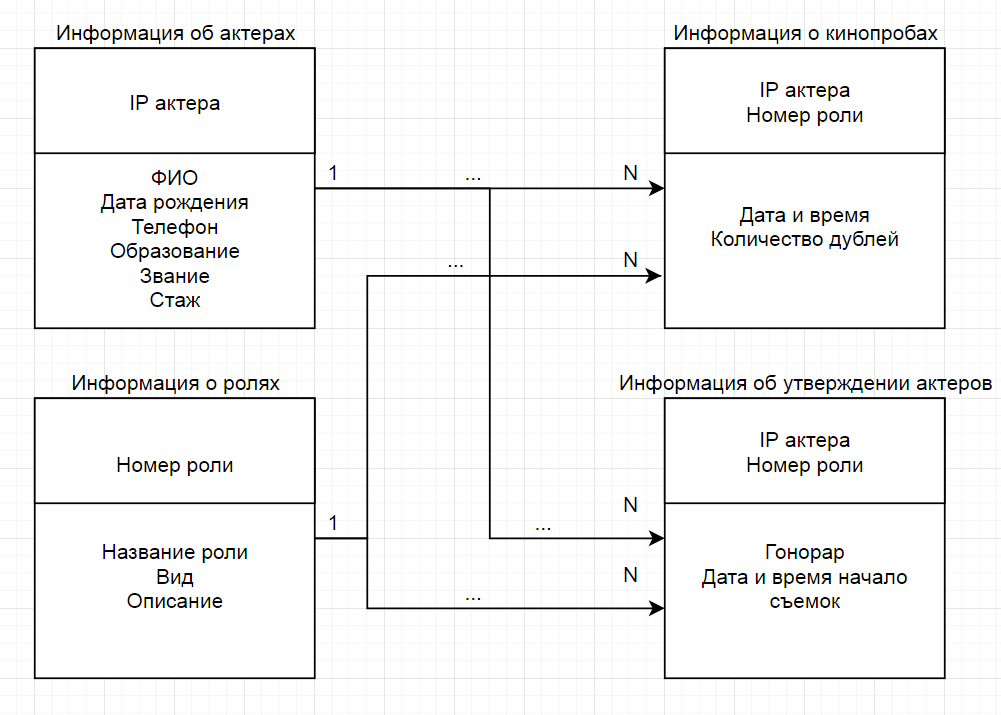
**2.1 Разработка логической модели данных**

Для того чтобы у нас были отсылки на другие таблицы, нужно понять, какие таблицы главные, а какие подчиненные.

Есть таблица «Роли», в которой указаны все доступные для пробы роли. Актер приходит на кинопробу и ему необходимо зарегистрироваться, так, собственно, заполняется таблица «Актеры». Дальше составляется таблица «Кинопроба», в которой указывается информация: о роли, на которую хочет пробоваться актер, время и количество дублей. Так мы уже определили уже первую связь: Таблица «Кинопроба» является подчиненной, а «Актеры» и «Роли» - главные.

В итоге, после кинопробы, появляется еще одна таблица – «Утверждены», в которой указаны те IP актеров и номера ролей, которые утверждены. Вот и появилась вторая связь: Таблица «Утверждены» является подчиненной, а «Актеры» и «Роли» - главные.

А теперь мы создадим логическую модель. Для простого восприятия сделаем следующую диаграмму:

**

*Рисунок 1 – Логическая модель предметной области «Кинопробы».*

**2.2 Определение типов хранимых данных**

Прежде чем начать определять типы данных в таблицах, обратимся немного к теории:

Типы данных SQL Server 2019 можно разбить на следующие группы:

1. целочисленные *–* bigint, int, smallint и tinyint;
2. нецелочисленные *–* decimal, numeric, float и real;
3. денежные *–* money и smallmoney;
4. дата и время *–* datetime и smalldatetime;
5. двоичные *–* binary, varbinary и image*;*
6. строковые *–* char, varchar, nchar и nvarchar;
7. текстовые *–* text и ntext;
8. логические *–* bit;
9. специальные *–* timestamp, uniqueidentifier, cursor, table и sql\_variant.

**1) Целочисленные типы данных**

Встречается во всех данных таблица. Чтобы прописать стаж актера, IP адрес, номер роли и количество дублей.

Для этого мы возьмем int - 4 байта - числа в диапазоне от -231 до 231-1, что соответствует интервалу -2 147 483 648 до 2 147 483 647.

И tinyint - числа в диапазоне от 0 до 255. Используем для количества дублей, так как оно не может принимать отрицательные значения и быть слишком большим. Поэтому нам хватит диапазон до 255.

**2) Денежные типы данных**

Используется для определения размера гонорара актера.

Денежные типы данных с помощью money хранит до 8 байт, что обеспечивает представление значений в диапазоне от -263 (-922 337 203 685 477,5808) до 263 (+922 337 203 685 477,5807).

**3) Типы данных "дата и время"**

Для указания времени кинопробы, даты рождения актера и времени начала съемок.

datetime. Для хранения данных применяются два 4-байтовых блока. Первые 4 байта хранят информацию о количестве дней, прошедших после базовой ' даты – I января 1753 года. Вторые 4 байта предназначены для хранения информации о количестве миллисекунд, прошедших после полуночи. В итоге тип данных datetime позволяет хранить информацию о датах в интервале от 1 января 1753 года до 31 декабря 9999 года с точностью до 3,33 миллисекунды.

date - формат YYYY-MM-DD. Для хранения даты.

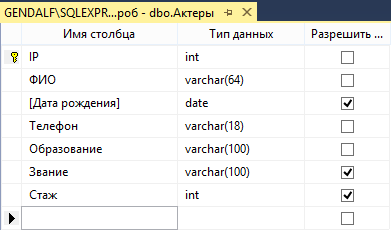
**4)** **Строковые типы данных**

Для ввода ФИО, названия роли, ее вида, описания, номера телефона, образования и звания будем использовать:

varchar [ (n) ]. Как и предыдущий, тип данных varchar обеспечивает хранение блоков данных до 8000 символов в стандарте ASCII.

А теперь рассмотрим данную БД и распределим типы данных:

**1. Таблица «Актеры»**

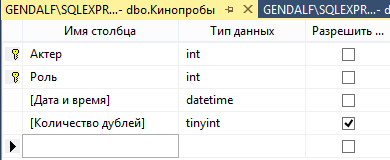
******

*Рисунок 2 – Создание таблицы «Актеры» и распределение типов данных в соответствии с именем столбца.*

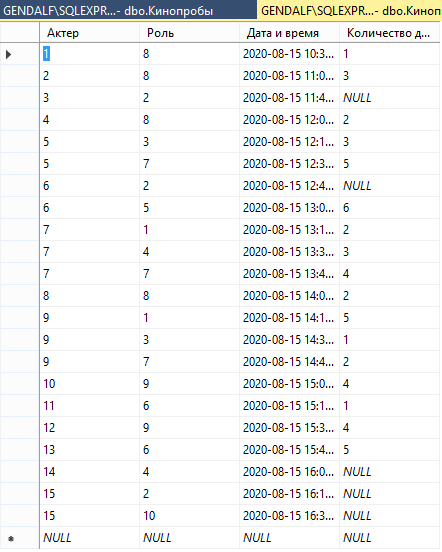
******

*Рисунок 3 – Заполнение таблицы «Актеры»*

**2. Таблица «Кинопробы»**

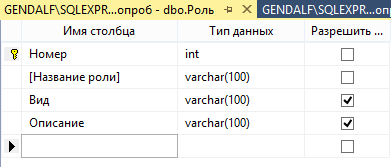
******

*Рисунок 4 – Создание таблицы «Кинопробы» и распределение типов данных в соответствии с именем столбца.*

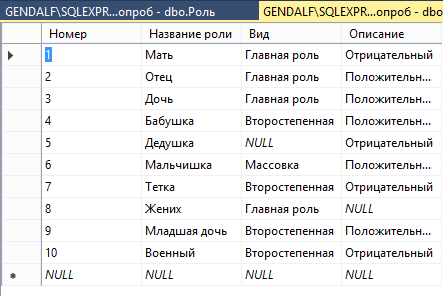
******

*Рисунок 5 – Заполнение таблицы «Кинопробы».*

**3. Таблица «Роли»**

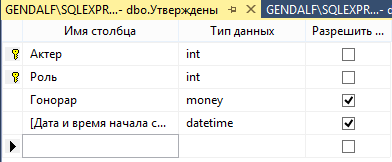
******

*Рисунок 6 – Создание таблицы «Роли» и распределение типов данных в соответствии с именем столбца.*

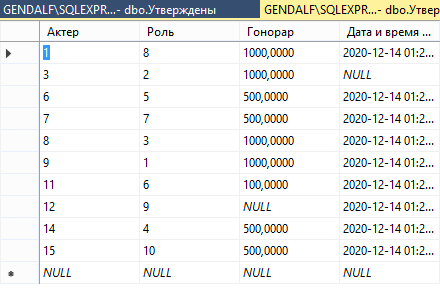
******

*Рисунок 7 – Заполнение таблицы «Роли»*

**4. Таблица «Утверждены»**

******

*Рисунок 8 – Создание таблицы «Утверждены» и распределение типов данных в соответствии с именем столбца.*

******

*Рисунок 9 – Заполнение таблицы «Утверждены»*

**2.3 Задание ограничений целостности данных**

Ограничение — некоторое ограничивающее условие, или правило. В базе данных — общее понятие, охватывающее широкий круг аспектов управления базой данных: ключи, значения, типы и форматы данных и т.д.

Ограничения устанавливают для пользователя некоторые рамки при вводе, изменении или удалении данных приложения. Вся система ограничений при создании приложения базы данных соотносится с целью обеспечения *целостности*данных. Целостность данных представляет собой набор правил, используемых ядром базы данных для поддержания связей между записями в связанных таблицах, а также для защиты от случайного удаления или изменения связанных данных.

Например, ограничение можно использовать, чтобы гарантировать, что каждый служащий в базе данных будет относиться к какому-либо отделу, или гарантировать, что пользователи не смогут случайно ввести отрицательное значение для цены товара.

База данных поддерживает следующие виды ограничений:

1. PRIMARY KEY
2. UNIQUE
3. FOREIGN KEY
4. CHECK
5. NOT NULL
6. DEFAULT

В своей работе я применяла следующие ограничения:

1.**PRIMARY KEY** *–* ограничение первичного ключа, который гарантирует уникальность каждой строки, что дает нам возможность идентифицировать каждую строку в таблице по данному ключу.

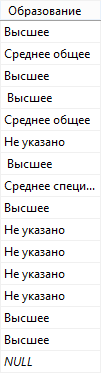
2. **NOT NULL** *–* ограничение, с помощью которого мы можем запретить или наоборот разрешить хранение в столбце значений NULL, т.е. неопределенных значений. Таким образом, мы можем сказать, что если у нас запрещены значения NULL в столбце, то этот столбец является обязательным к заполнению, а если у нас разрешены значения NULL, то столбец можно и не заполнять, т.е. данное ограничение поможет нам контролировать внесение и хранение обязательных характеристик той или иной сущности.

3. **DEFAULT** *–* это значение, которое можно задать для столбца значение по умолчанию, которое будет сохранено, если при вводе данных мы не указали никакого значения. Рассмотрим пример:

Пример кода:

Образование varchar(100) NOT NULL DEFAULT 'Не указано'

Реализация примера:

**

*Рисунок 10 – Реализация ограничения DEFAULT*

То есть, когда мы не вводим информацию об уровне образования, то у нас по умолчанию будет указано 'Не указано'.

4. **FOREIGN KEY**  - используется для ограничения по ссылкам.  
Когда все  значения в одном поле таблицы представлены в поле другой таблицы, говорится, что первое поле ссылается на второе. Это указывает на прямую связь между значениями двух полей.

Структурируем данные в таблицы:

Таблица 1 — Отношение «Актеры» и ограничения.

|  |  |
| --- | --- |
| Актеры | |
| IP | PK |
| ФИО | CHECK LIKE(‘[A-Я]%’) |
| Дата рождения |  |
| Телефон | NOT NULL, CHECK LIKE '+7 ([0-9][0-9][0-9]) [0-9][0-9][0-9]-[0-9][0-9]-[0-9][0-9]' |
| Образование | NOT NULL, DEFAULT 'Не указано' |
| Звание | CHECK IN 'Заслуженный артист' |
| Стаж | CHECK BETWEEN 0 AND 75 |

Таблица 2 — Отношение «Роль» и ограничения.

|  |  |
| --- | --- |
| Роль | |
| Номер | PK |
| Название роли | NOT NULL |
| Вид |  |
| Описание |  |

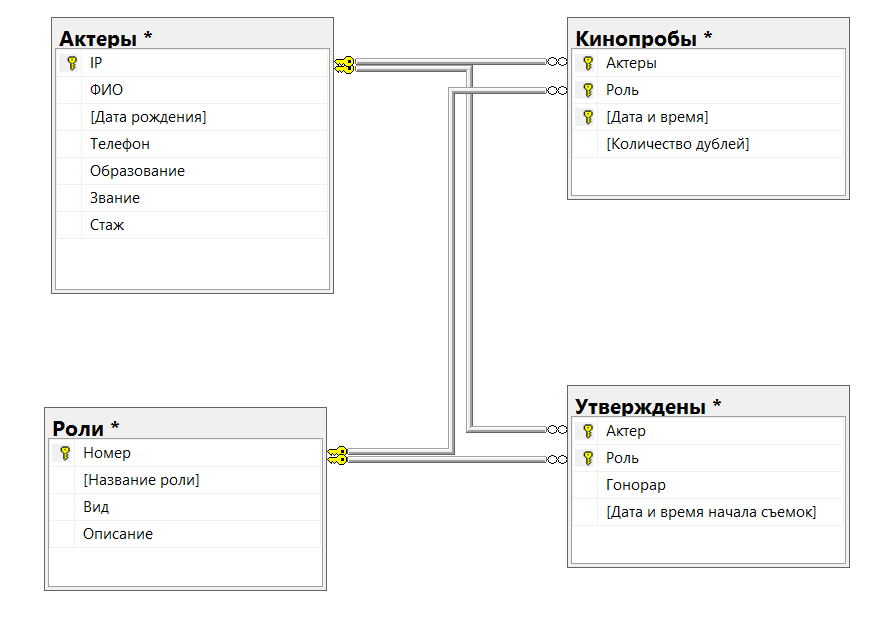
Таблица 3 — Отношение «Кинопробы» и ограничения.

|  |  |
| --- | --- |
| Кинопробы | |
| Актер | FK(Актеры), NOT NULL |
| Роль | FK(Роль), NOT NULL |
| Дата и время | NOT NULL |
| Количество дублей |  |

Таблица 4 — Отношение «Утверждены» и ограничения.

|  |  |
| --- | --- |
| Утверждены | |
| Актер | FK(Актеры), NOT NULL |
| Роль | FK(Роль), NOT NULL |
| Гонорар | CHECK>0, DEFAULT '100' |
| Дата и время начала съемок |  |

**2.4 Построение схемы данных**

*Рисунок 11 – Схема базы данных «Кинопробы».*

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

USE [master]

GO

CREATE DATABASE [Кинопроб]

CONTAINMENT = NONE

ON PRIMARY

( NAME = N'Кинопроб', FILENAME = N'F:\Кинопроб.mdf' ,

SIZE = 5120KB ,

MAXSIZE = UNLIMITED,

FILEGROWTH = 1024KB )

LOG ON

( NAME = N'Кинопроб\_log', FILENAME = N'F:\Кинопроб\_log.ldf' ,

SIZE = 1024KB ,

MAXSIZE = 2048GB ,

FILEGROWTH = 10%)

GO

USE [Кинопроб]

GO

CREATE TABLE Актеры

( IP int NOT NULL PRIMARY KEY ,

ФИО varchar(64) NOT NULL CHECK ([ФИО] LIKE '[А-Я]%'),

[Дата рождения] date NULL,

Телефон varchar(18) NOT NULL CHECK ([Телефон] LIKE '+7 ([0-9][0-9][0-9]) [0-9][0-9][0-9]-[0-9][0-9]-[0-9][0-9]'),

Образование varchar(100) NOT NULL DEFAULT 'Не указано',

Звание varchar(100) NULL CHECK (Звание IN ('Заслуженный артист')),

Стаж int NULL CHECK (Стаж BETWEEN 0 AND 75))

GO

CREATE TABLE Роль

( Номер int NOT NULL PRIMARY KEY,

[Название роли] varchar(100) NOT NULL,

Вид varchar(100) NULL,

Описание varchar(100) NULL)

GO

CREATE TABLE Кинопробы

( [Актер] [int] NOT NULL,

[Роль] [int] NOT NULL,

[Дата и время] [datetime] NOT NULL,

[Количество дублей] [tinyint] NULL)

ALTER TABLE Кинопробы

ADD CONSTRAINT PK\_Кинопробы PRIMARY KEY (Актер, Роль)

GO

ALTER TABLE Кинопробы

ADD CONSTRAINT FK\_Кинопробы\_Актеры FOREIGN KEY (Актер) REFERENCES Актеры(IP)

GO

ALTER TABLE Кинопробы

ADD CONSTRAINT FK\_Кинопробы\_Роль FOREIGN KEY (Роль) REFERENCES Роль(Номер)

GO

CREATE TABLE Утверждены

( [Актер] [int] NOT NULL,

[Роль] [int] NOT NULL,

[Гонорар] [money] NULL CHECK (Гонорар> 0) DEFAULT '100',

[Дата и время начала съемок] [datetime] NULL)

ALTER TABLE Утверждены

ADD CONSTRAINT PK\_Утверждены PRIMARY KEY (Актер, Роль)

GO

ALTER TABLE Утверждены

ADD CONSTRAINT FK\_Утверждены\_Актеры FOREIGN KEY (Актер) REFERENCES Актеры(IP)

GO

ALTER TABLE Утверждены

ADD CONSTRAINT FK\_Утверждены\_Роль FOREIGN KEY (Роль) REFERENCES Роль(Номер)

GO

**2.6 Разработка сценария для заполнения таблиц БД**

INSERT INTO Актеры VALUES

(1, 'Пушкин Е.А.', '1995-03-03', '+7 (900) 910-98-35', 'Высшее', NULL, 2),

(2, 'Тургенев А.П.', '2001-06-15', '+7 (930) 900-24-67', 'Среднее общее', NULL, 1),

(3, 'Толстой П.В.', '1985-10-15', '+7 (920) 876-78-25', 'Высшее', 'Заслуженный артист', 15),

(4, 'Арсентьев С.С.', '2000-02-18', '+7 (920) 876-78-28', ' Высшее', NULL, 3),

(5, 'Попова С.В. ', '1991-12-23', '+7 (920) 876-78-29', 'Среднее общее', NULL, NULL),

(6, 'Логунский А.Ф.', NULL, '+7 (900) 910-98-36', DEFAULT, 'Заслуженный артист', 35),

(7, 'Логунская Т.А.', '1965-12-23', '+7 (900) 910-98-37', ' Высшее', 'Заслуженный артист', NULL),

(8, 'Бодрова Н.А.', '1992-10-12', '+7 (900) 910-98-38', 'Среднее специальное', NULL, 4),

(9, 'Жилищина Т.Р.', '1982-09-29', '+7 (920) 876-78-20', 'Высшее', NULL, 10),

(10,'Анохина Ю.Н.', '2013-05-07', '+7 (930) 900-24-68', DEFAULT , NULL, 2),

(11,'Волохин Ю.С.', '2012-08-15', '+7 (900) 910-98-32', DEFAULT , NULL, 3),

(12,'Мартынова К.С.', '2014-03-09', '+7 (920) 876-78-21', DEFAULT , NULL, 1),

(13,'Бурдинский А.А.', '2010-01-09', '+7 (900) 910-98-31', DEFAULT , NULL, 5),

(14,'Зубова Н.Е.', '1956-02-03', '+7 (930) 900-24-69', 'Высшее', 'Заслуженный артист', 46),

(15,'Рахимов Р.Р.', '1974-09-23', '+7 (900) 910-98-45', 'Высшее', 'Заслуженный артист', 20)

GO

INSERT INTO Роль VALUES

(1, 'Мать', 'Главная роль', 'Отрицательный'),

(2, 'Отец', 'Главная роль', 'Положительный'),

(3, 'Дочь', 'Главная роль', 'Положительный'),

(4, 'Бабушка', 'Второстепенная', 'Положительный'),

(5, 'Дедушка', NULL , 'Отрицательный'),

(6, 'Мальчишка', 'Массовка', 'Положительный'),

(7, 'Тетка', 'Второстепенная', 'Отрицательный'),

(8, 'Жених', 'Главная роль', NULL ),

(9, 'Младшая дочь', 'Второстепенная', 'Положительный'),

(10, 'Военный', 'Второстепенная', 'Отрицательный')

GO

INSERT INTO Кинопробы VALUES

(1, 8, convert(datetime,'2020-08-15T10:30:00'), 1),

(2, 8, convert(datetime,'2020-08-15T11:00:00'), 3),

(3, 2, convert(datetime,'2020-08-15T11:45:00'),NULL),

(4, 8, convert(datetime,'2020-08-15T12:00:00'), 2),

(5, 3, convert(datetime,'2020-08-15T12:15:00'), 3),

(5, 7, convert(datetime,'2020-08-15T12:30:00'), 5),

(6, 2, convert(datetime,'2020-08-15T12:45:00'),NULL),

(6, 5, convert(datetime,'2020-08-15T13:00:00'), 6),

(7, 1, convert(datetime,'2020-08-15T13:15:00'), 2),

(7, 4, convert(datetime,'2020-08-15T13:30:00'), 3),

(7, 7, convert(datetime,'2020-08-15T13:45:00'), 4),

(8, 8, convert(datetime,'2020-08-15T14:00:00'), 2),

(9, 1, convert(datetime,'2020-08-15T14:15:00'), 5),

(9, 3, convert(datetime,'2020-08-15T14:30:00'), 1),

(9, 7, convert(datetime,'2020-08-15T14:45:00'), 2),

(10, 9, convert(datetime,'2020-08-15T15:00:00'), 4),

(11, 6, convert(datetime,'2020-08-15T15:15:00'), 1),

(12, 9, convert(datetime,'2020-08-15T15:30:00'), 4),

(13, 6, convert(datetime,'2020-08-15T15:45:00'), 5),

(14, 4, convert(datetime,'2020-08-15T16:00:00'),NULL),

(15, 2, convert(datetime,'2020-08-15T16:15:00'),NULL),

(15, 10, convert(datetime,'2020-08-15T16:30:00'),NULL)

GO

INSERT INTO Утверждены VALUES

(1, 8, 1000, convert(datetime, GETDATE())),

(3, 2, 1000, NULL ),

(6, 5, 500, convert(datetime, GETDATE())),

(7, 7, 500, convert(datetime, GETDATE())),

(8, 3, 1000, convert(datetime, GETDATE())),

(9, 1, 1000, convert(datetime, GETDATE())),

(11, 6, DEFAULT, convert(datetime, GETDATE())),

(12, 9, NULL, convert(datetime, GETDATE())),

(14, 4, 500, convert(datetime, GETDATE())),

(15, 10, 500, convert(datetime, GETDATE()))

GO

# 3 РАЗРАБОТКА ОСНОВНЫХ ОПЕРАЦИЙ РЕЛЯЦИОННОЙ АЛГЕБРЫ

**1.** Вывести информация об актерах с высшим образованием.

R= (Актеры WHERE Образование = 'Высшее')

**2.** Вывести ФИО, номер телефона и дату рождения актеров со званием заслуженного артиста.

R= (Актеры WHERE Звание = 'Заслуженный артист') [ФИО, Телефон, Звание]

**3.**Вывести список тех, у кого неизвестен актерский стаж.

R = (Актеры WHERE Стаж = NULL) [ФИО]

**4.** Вывести полную информацию об утвержденных ролях с указанием ФИО актера, утвержденного на данную роль.

R1 = Актеры RENAME IP AS Актер

R2 = (R1 JOIN Утверждены) [ФИО, Роль]

**5.** Вывести полную информацию обо всех актерах. Если актер утвержден укажите его гонорар.

R1 = Актеры RENAME ФИО AS Актер

R2 = (Актеры LEFT JOIN Утверждены) [ФИО, [Дата рождения], Телефон, Образование, Звание, Стаж, Гонорар]

**6.** Вывести информацию об утвержденных ролях с номерами 4 и 5.

R1 = (Утверждены WHERE Роль = 4)

R2 = (Утверждены WHERE Роль = 5)

R3 = R1 UNION R2

**7.** Вывести IP и телефон тех актеров, которые не утверждены.

R1 = (Актеры)[IP]

R2 = (Утверждены ) [Актер]

R3 = R1 EXCEPT R2

R4 = (R3 JOIN Актеры) [ФИО, Телефон]

**8.** Вывести информацию об актере, у которого положительно утвержденная роль. R1= (Роли WHERE Описание = ’Положительная’) [Номер]

R2=Утверждены [Роль]

R3= Утверждены EXCEPT R1

R4=R3[Актер]

R5=(R3 JOIN Актеры) [ФИО]

**9.** Вывести ФИО актеров, которые пробовались на роль с номером 1, пробовались на роль с номером 2.

R1 = (Кинопробы) WHERE Роль = '1'

R2 = (Кинопробы) WHERE Роль = '2'

R3 = R1 EXCEPT R2

R4 = (R3 JOIN Актеры)[ФИО]

**10.** Вывести ФИО актеров и стаж, которые не пробовались на роль тетки.

R1 = (( Роли JOIN Кинопробы) WHERE Наименование Роли = ’Тетка’) [Номер]

R2= Кинопробы EXCEPT R1 [Актеры]

R3 =((R2 JOINАктеры) [ФИО]

**11.**Вывести ФИО актеров, пробовавшихся на роль жениха

R1 =(( Роли JOIN Кинопробы) WHERE Наименование Роли = ’Жених’) [Номер]

R2 = ((R1 JOIN Кинопробы) WHERE Роль<>Номер)[Актеры]

R3 = (Актеры JOIN R2) [ФИО]

**12.** Вывести информацию о тех актерах, у которых количество дублей больше 5

R1 = (Кинопробы WHERE Количество дублей>5)[Актер]

R2 =( R1 JOIN Актеры) [ФИО, [Дата рождения], Телефон, Образование, Звание, Стаж]

**13.** Вывести информацию о кинопробе в следующем виде:

ФИО актера, Название роль, Дата и время пробы, Количество дублей

R1 = ((Кинопробы JOIN Актеры)JOIN Роли)[ ФИО, Название роли, [Дата и время ], Количество дублей]

**14.** Вывести информацию об утверждении актеров тех, у кого стаж превышает 5 и звание заслуженного артиста.

R1 = (Утверждены) [Актер]

R2 = (Утверждены JOIN Актеры )WHERE Стаж<5) [Актер]

R3 = R1 EXCEPT R2

R4 = (Утверждены JOIN Актеры )WHERE Звание=’Заслуженный артист’) [Актер]

R5 = (R3 EXCEPT R4)[Актер]

R6 = R5 [ФИО, Телефон, Стаж, Звание]

**15.** Вывести полную информацию об актерах, которые утверждены и гонорар превышает 500.

R1= (Утверждены) [Актер]

R2 = (Утверждены WHERE Гонорар< 600) [Актер]

R3 = R1 EXCEPT R2

R4 = (R3 JOIN Актеры)[ФИО, [Дата рождения],Телефон, Образование, Звание, Стаж]

**4 РАЗРАБОТКА ЗАПРОСОВ К БД ДЛЯ РЕАЛИЗАЦИИ ИНФОРМАЦИОННЫХ ПОТРЕБНОСТЕЙ ПОЛЬЗОВАТЕЛЯ**

**Запросы с выбором строк по условию:**

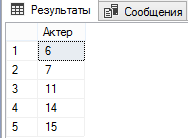
–Запрос с использованием операторов сравнения.

**1.** Вывести информацию об актерах, с гонораром меньше 1 млн. Руб.

SELECT Актер

FROM Утверждены

WHERE [Гонорар]<1000



*Рисунок 12 – Результат запроса 1.*

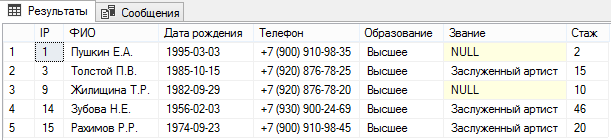
–Запрос с использованием логических операторов AND, OR и NOT.

**2.** Вывести информацию о тех актерах, у которых указано высшее и средне специальное образования.

SELECT\*

FROM Актеры

WHERE Образование='Высшее'OR Образование='Средне специальное'



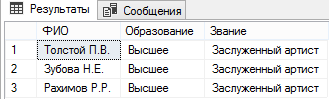
*Рисунок 13 – Результат запроса 2.*

**3.** Вывести ФИО актеров, которые имеют образования среднее общее и высшее, и являются заслуженными артистами.

SELECT ФИО, Образование, Звание

FROM Актеры

WHERE (Образование = 'Среднее общее' OR Образование = 'Высшее') AND Звание = 'Заслуженный артист'



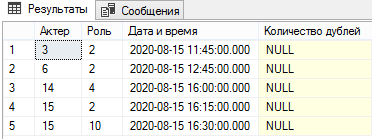
*Рисунок 14 – Результат запроса 3.*

**4.** Вывести информацию об актерах, у которых не указано количество дублей.

SELECT\*

FROM Кинопробы

WHERE [Количество дублей] IS NULL

**

*Рисунок 15 – Результат запроса 4.*

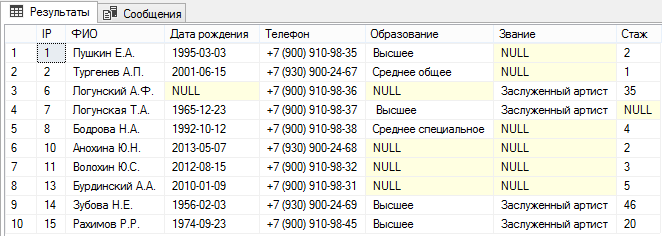
–Запрос на использование комбинации логических операторов.

**5.** Вывести информацию об актерах, у которых код оператора 910 или не 920.

SELECT\*

FROM Актеры

WHERE Телефон LIKE ('%900%') OR NOT Телефон LIKE('%920%')



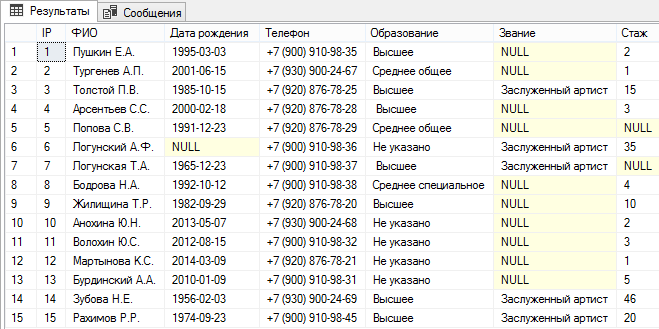
*Рисунок 16 – Результат запроса 5.*

–Запрос на использование выражений над столбцами

**6.** Вывести полную информацию обо всех актерах.

SELECT \*

FROM Актеры

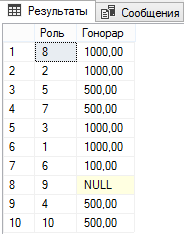


*Рисунок 17 – Результат запроса 6.*

**7.** Вывести информацию обо всех утвержденных ролях с указанием гонорара.

SELECT Роль, Гонорар

FROM Утверждены

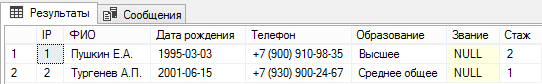


*Рисунок 18 – Результат запроса 7.*

**8.** Вывести информацию о двух первых хранимых записях об актерах.

SELECT TOP 2\*

FROM Актеры

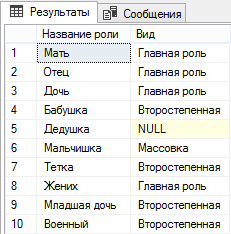


*Рисунок 19 – Результат запроса 8.*

**9.** Вывести информацию о названии и виде роли.

SELECT [Название роли], Вид

FROM Роль

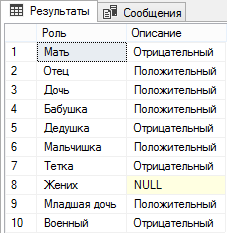


*Рисунок 20 – Результат запроса 9.*

**10.** Вывести описание ролей.

SELECT [Название роли] AS [Роль], Описание

FROM Роль



*Рисунок 21 – Результат запроса 10.*

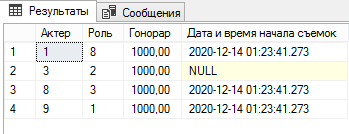
–Запрос с проверкой на принадлежность множеству.

**11.** Вывести всю информацию об утверждении тех актеров, у которых гонорар равен 1000.

SELECT \*

FROM Утверждены

WHERE Гонорар IN (1000)



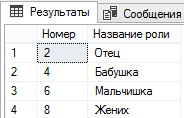
*Рисунок 22 – Результат запроса 11.*

**12.** Вывести название роли с номером равным 2,4,6,8.

SELECT Номер, [Название роли]

FROM Роль

WHERE Номер IN(2,4,6,8)



*Рисунок 23 – Результат запроса 12.*

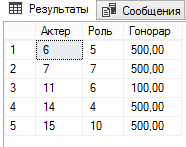
–Запрос с проверкой на принадлежность диапазону значений.

**13.** Вывести роли, у которых гонорар в диапазоне между 100 и 600.

SELECT Актер, Роль, Гонорар

FROM Утверждены

WHERE Гонорар BETWEEN 100 AND 600



*Рисунок 24 – Результат запроса 13.*

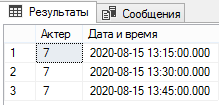
–Запрос с проверкой на соответствие шаблону.

**14.** Вывести дату и время кинопроб актера с IP равным 7.

SELECT Актер, [Дата и время]

FROM Кинопробы

WHERE Актер LIKE ('7%')



*Рисунок 25 – Результат запроса 14.*

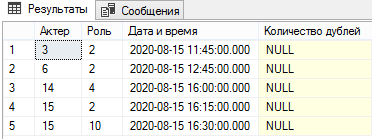
–Запрос с проверкой на неопределенное значение.

**15.** Вывести информацию об актерах, у которых не указано количество дублей.

SELECT\*

FROM Кинопробы

WHERE [Количество дублей] IS NULL

**

*Рисунок 26 – Результат запроса 15.*

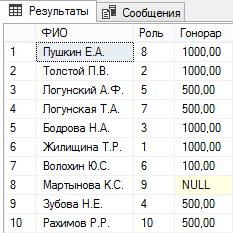
**Запросы на теоретико-множественные реляционные операции:**

–Запрос на объединение таблиц.

**16.** Вывести информацию об утверждении с указанием ФИО актеров.

SELECT [ФИО], [Роль], [Гонорар]

FROM [Актеры] INNER JOIN [Утверждены] ON Актеры.IP = Утверждены.Актер



*Рисунок 27 – Результат запроса 16.*

–Запрос на пересечение таблиц.

**17.** Вывести номера ролей, на которые пробовались и актер с IP= 1, и актер с IP= 4.

SELECT Кинопробы.Роль

FROM Кинопробы

WHERE Кинопробы.Актер='1'

INTERSECT

SELECT Кинопробы.Роль

FROM Кинопробы

WHERE Кинопробы.Актер='4'



*Рисунок 28 – Результат запроса 17.*

–Запрос на вычитание таблиц.

**18.** Вывести номера ролей, на которые не пробовался ни актер с IP=1, и актер с IP=9.

SELECT Роль

FROM Кинопробы

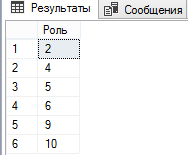
EXCEPT

SELECT Роль

FROM Кинопробы

WHERE Актер ='1'

OR Актер ='9'



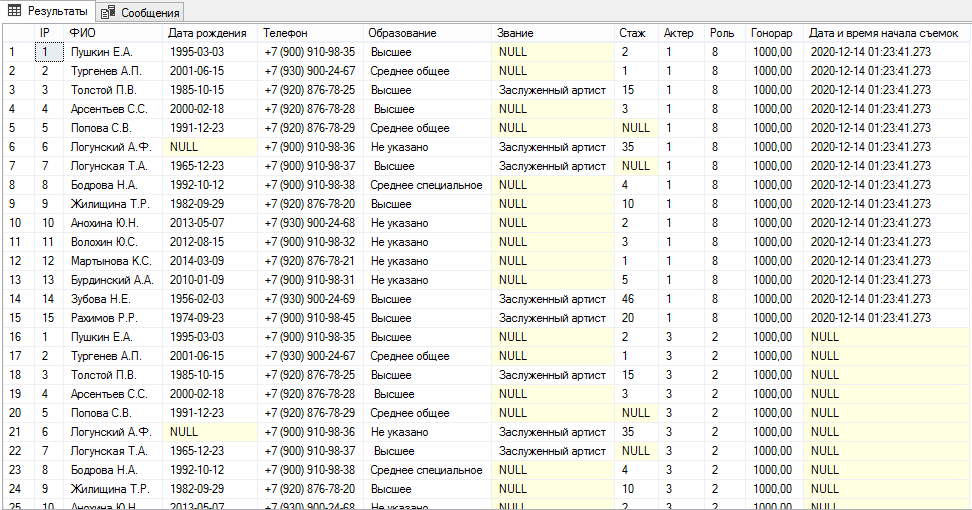
*Рисунок 29 – Результат запроса 18.*

–Запрос с использованием декартового произведения таблиц.

**19.** Вывести декартово произведение таблиц Актеры и Утверждены.

SELECT\*

FROM Актеры CROSS JOIN Утверждены



*Рисунок 30 – Результат запроса 19.*

**Многотабличные запросы:**

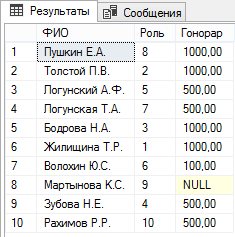
–Запрос с использованием соединения двух таблиц по равенству и условием отбора.

**20.** Вывести информацию об утверждении с указанием ФИО актеров.

SELECT ФИО, Роль, Гонорар

FROM Актеры, Утверждены

WHERE Актеры.IP=Утверждены.Актер



*Рисунок 31 – Результат запроса 20.*

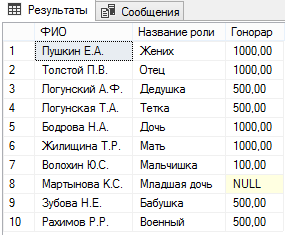
–Запрос с использованием соединения трех таблиц по равенству и условием отбора.

**21.** Вывести информацию об утверждении с указанием ФИО актеров и названием ролей.

SELECT ФИО, [Название роли], Гонорар

FROM Актеры, Утверждены, Роль

WHERE Актеры.IP=Утверждены.Актер AND Роль.Номер=Утверждены.Роль



*Рисунок 32 – Результат запроса 21.*

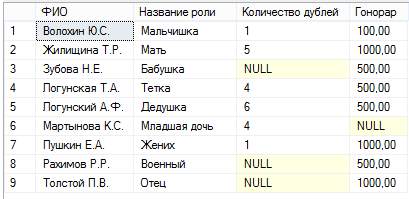
–Запрос с использованием соединения по четырем таблицам.

**22.** Вывести информацию об утверждении с указанием ФИО актеров, названием ролей и количеством дублей.

SELECT DISTINCT ФИО, [Название роли], [Количество дублей], Гонорар

FROM Актеры, Утверждены, Роль, Кинопробы

WHERE Актеры.IP=Утверждены.Актер AND Роль.Номер=Утверждены.Роль AND Актеры.IP=Кинопробы.Актер AND Роль.Номер=Кинопробы.Роль



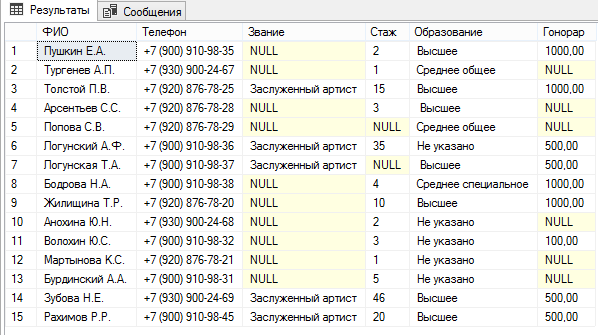
*Рисунок 33 – Результат запроса 22.*

–Запрос с использованием левого внешнего соединения.

**23.** Вывести информацию обо всех актерах, указав их гонорар, если такие имеются.

SELECT Актеры.ФИО,Актеры.Телефон, Актеры.Звание, Актеры.Стаж, Актеры.Образование, Утверждены.Гонорар

FROM Актеры LEFT JOIN Утверждены ON Актеры.IP= Утверждены.Актер



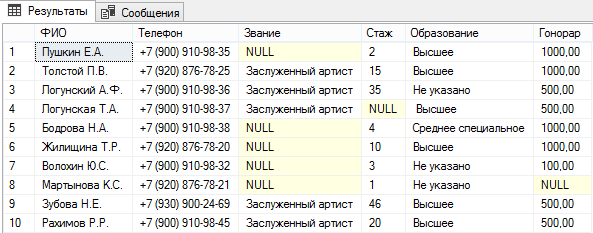
*Рисунок 34 – Результат запроса 23.*

–Запрос на использование правого внешнего соединения.

**24.** Вывести информацию обо всех актерах, указав их гонорар, если такие имеются.

SELECT Актеры.ФИО,Актеры.Телефон, Актеры.Звание, Актеры.Стаж, Актеры.Образование, Утверждены.Гонорар

FROM Актеры RIGHT JOIN Утверждены ON Актеры.IP= Утверждены.Актер



*Рисунок 35 – Результат запроса 24.*

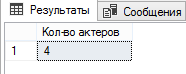
–Запрос с использованием функции COUNT.

**25.** Вывести количество актеров, пробовавшихся на роль «Жених».

SELECT COUNT(Актер)AS [Кол-во актеров]

FROM Кинопробы JOIN Роль ON Кинопробы.Роль = Роль.Номер

WHERE [Название роли] = 'Жених'



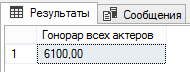
*Рисунок 36 – Результат запроса 25.*

–Запрос с использованием функции SUM.

**26.** Вывести общую сумму гонораров.

SELECT SUM (Гонорар) As [Гонорар всех актеров]

FROM Утверждены



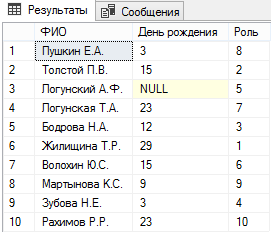
*Рисунок 37 – Результат запроса 26.*

–3 запроса с использованием временных функций.

**27.** Вывести ФИО актеров, их дни рождения и номер роли, на какую некоторые актеры утвердились.

SELECT ФИО,DAY([Дата рождения])AS [День рождения], Роль

FROM Актеры JOIN Утверждены ON Актеры.IP = Утверждены.Актер

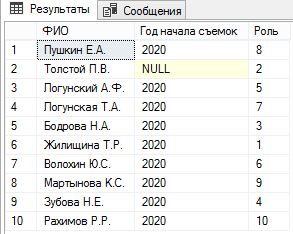


*Рисунок 38 – Результат запроса 27.*

**28.** Вывести ФИО утвержденного актера, год начала съемок, название роли, на которую его утвердили.

SELECT ФИО,YEAR([Дата и время начала съемок])AS [Год начала съемок], Роль

FROM Актеры JOIN Утверждены ON Актеры.IP=Утверждены.Актер JOIN Роль ON Утверждены.Роль=Роль.Номер

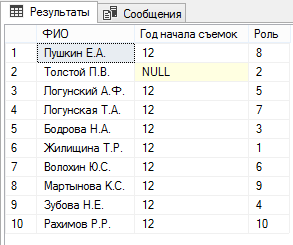


*Рисунок 39 – Результат запроса 28.*

**29.** Вывести ФИО утвержденного актера, месяц начала съемок, название роли, на которую его утвердили.

SELECT ФИО,month ([Дата и время начала съемок])AS [Год начала съемок], Роль

FROM Актеры JOIN Утверждены ON Актеры.IP=Утверждены.Актер JOIN Роль ON Утверждены.Роль=Роль.Номер



*Рисунок 40 – Результат запроса 29.*

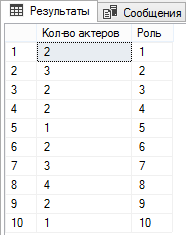
–Запрос с использованием группировки по одному столбцу.

**30.** Вывести количество актеров, которые пробовались на каждую роль.

SELECT COUNT(Актер) AS [Кол-во актеров], Роль

FROM Кинопробы

GROUP BY Роль



*Рисунок 41 – Результат запроса 30.*

–Запрос на использование группировки по нескольким столбцам.

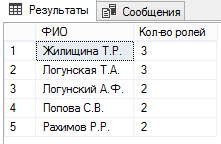
**31.** Вывести IP актера и кол-во ролей, на которые он пробовался, если их больше 1.

SELECT ФИО,COUNT(Роль)AS[Кол-во ролей]

FROM Актеры JOIN Кинопробы ON Актеры.IP =Кинопробы.Актер

GROUP BY ФИО

HAVING COUNT(Роль)>1



*Рисунок 42 – Результат запроса 31.*

–Запрос с использованием условия отбора групп HAVING.

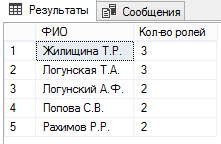
**32.** Вывести IP актера и кол-во ролей, на которые он пробовался, если их больше 1.

SELECT ФИО,COUNT(Роль)AS[Кол-во ролей]

FROM Актеры JOIN Кинопробы ON Актеры.IP =Кинопробы.Актер

GROUP BY ФИО

HAVING COUNT(Роль)>1



*Рисунок 43 – Результат запроса 32.*

–Запрос с использованием сортировки по столбцу.

**33.** Вывести фио актеров в порядке убывания количества дублей

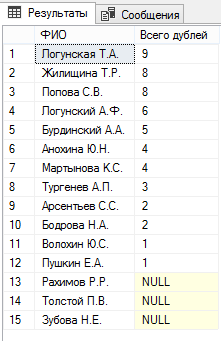
SELECT Актеры.ФИО, SUM(Количество дублей) AS [Всего дублей]

FROM Кинопробы, Актеры

WHERE Актеры.IP=Кинопробы.Актер

GROUP BY Актеры.ФИО

ORDER BY [Всего дублей] DESC



*Рисунок 44 – Результат запроса 33.*

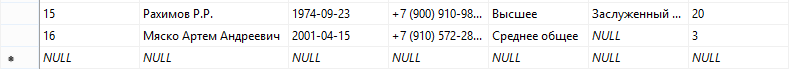
**Запросы на модификацию таблиц:**

–Запрос на добавление новых данных в таблицу.

**34.** Добавить информацию о новом актере.

INSERT INTO Актеры

VALUES (16, 'Мяско Артем Андреевич','2001.04.15','+7 (910) 572-28-90', 'Среднее общее', NULL, 3)

**

*Рисунок 45 – Результат запроса 34.*

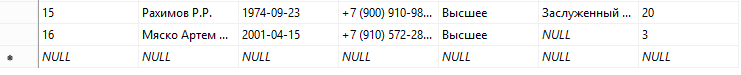
–Запрос на добавление новых данных по результатам выполненного запроса.

**35.** Мяско Артем получил высшее образование, обновить данные в таблице Актеры.

UPDATE Актеры

SET Образование='Высшее'

WHERE ФИО='Мяско Артем Андреевич'



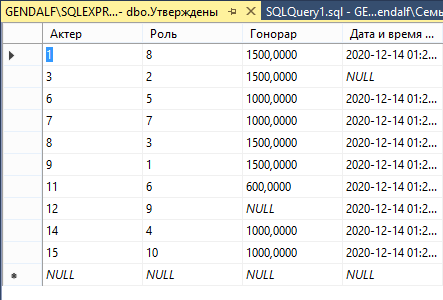
*Рисунок 46 – Результат запроса 35.*

–Запрос на обновление существующих данных в таблице.

**36.** Увеличить все гонорары на 500.

UPDATE Утверждены

SET Гонорар = Гонорар + 500



*Рисунок 47 – Результат запроса 36.*

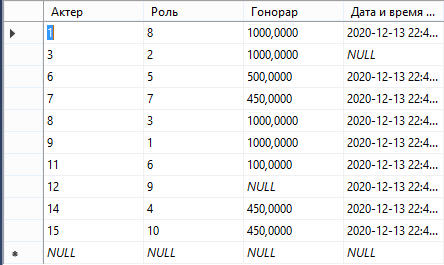
–Запрос на обновление существующих данных по результатам подзапроса во фразе WHERE.

**37.** Уменьшить гонорар на 10% у актеров, у которых второстепенные роли.

UPDATE Утверждены

SET Гонорар=Гонорар\*0.9

WHERE Роль IN(SELECT Номер FROM Роль WHERE Вид ='Второстепенная')



*Рисунок 48 – Результат запроса 37.*

–Запрос на удаление существующих данных.

**38.** Удалить информацию о Мяско Артеме Андреевиче.

DELETE

FROM Актеры

WHERE ФИО='Мяско Артем Андреевич'



*Рисунок 49 – Результат запроса 38.*

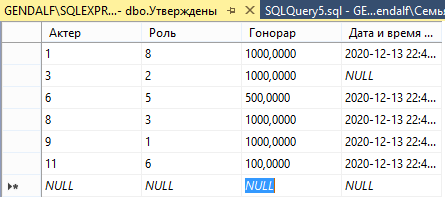
–Запрос на удаление существующих данных из одной таблицы на основе связанных с ней таблиц.

**39.** Удалить информацию о второстепенных ролях.

DELETE Утверждены

FROM Роль JOIN Утверждены ON Роль.Номер = Утверждены.Роль

WHERE Вид ='Второстепенная'



*Рисунок 50 – Результат запроса 39.*

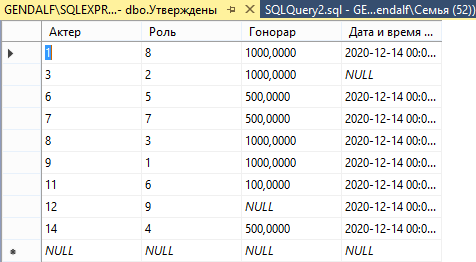
–Запрос на удаление существующих данных по результатам подзапроса во фразе WHERE.

**40.** Удалить из утвержденных актеров Рахимова Р.Р.

DELETE

FROM Утверждены

WHERE Актер =(SELECT IP FROM Актеры WHERE ФИО = 'Рахимов Р.Р.')



*Рисунок 51 – Результат запроса 40.*

**Многотабличные запросы с использованием подзапросов:**

–3 запроса с использованием операций сравнения.

**41.** Вывести самого взрослого актера.

SELECT\*

FROM Актеры

WHERE [Дата рождения] =(SELECT MIN([Дата рождения]) FROM Актеры )



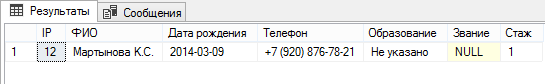
*Рисунок 52 – Результат запроса 41.*

**42.** Вывести самого юного актера.

SELECT\*

FROM Актеры

WHERE [Дата рождения] =(SELECT MAX([Дата рождения]) FROM Актеры )



*Рисунок 53 – Результат запроса 42.*

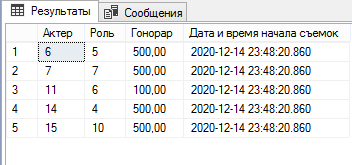
**43.** Вывести информацию об актерах, у которых гонорар ниже среднего.

SELECT \*

FROM Утверждены

WHERE Гонорар < (SELECT AVG(Гонорар)

FROM Утверждены )



*Рисунок 54 – Результат запроса 43.*

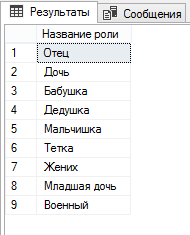
–Запрос с использованием операции ANY;

**44.** Вывести название роли, номер которой превышает номера хотя бы одной другой роли.

SELECT Роль.[Название роли]

FROM Роль

WHERE Роль.Номер>ANY(SELECT Роль.Номер FROM Роль)



*Рисунок 55 – Результат запроса 44.*

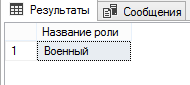
–Запрос с использованием операции ALL.

**45.** Вывести название роли, номер которой не меньше номера всех остальных роле.

SELECT Роль.[Название роли]

FROM Роль

WHERE Роль.Номер>=ALL(SELECT Роль.Номер FROM Роль)



*Рисунок 56 – Результат запроса 45.*

–Запрос с использованием операции IN.

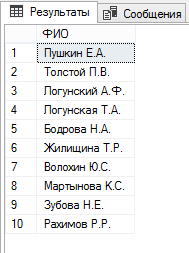
**46.** Вывести список актеров, которые не утверждены на роль.

SELECT Актеры.ФИО

FROM Актеры

WHERE Актеры.IP IN (SELECT Утверждены.Актер

FROM Утверждены)



*Рисунок 56 – Результат запроса 46.*

–Запрос с использованием операции EXISTS.

**47.** Вывести IP утвержденных актеров.

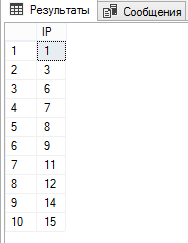
SELECT IP

FROM Актеры

WHERE EXISTS(SELECT Актер

FROM Утверждены

WHERE Актеры.IP=Утверждены.Актер)



*Рисунок 57 – Результат запроса 47.*

–Запрос с использованием двух вложенных друг в друга подзапросов.

**48.** Вывести информацию о тех актерах, которые утверждены и играю главную роль.

SELECT \*

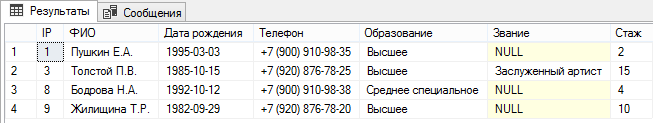
FROM Актеры

WHERE IP IN (SELECT Актер

FROM Утверждены

WHERE Роль IN (SELECT Номер

FROM Роль

WHERE Вид='Главная роль'))

*Рисунок 58 – Результат запроса 48.*

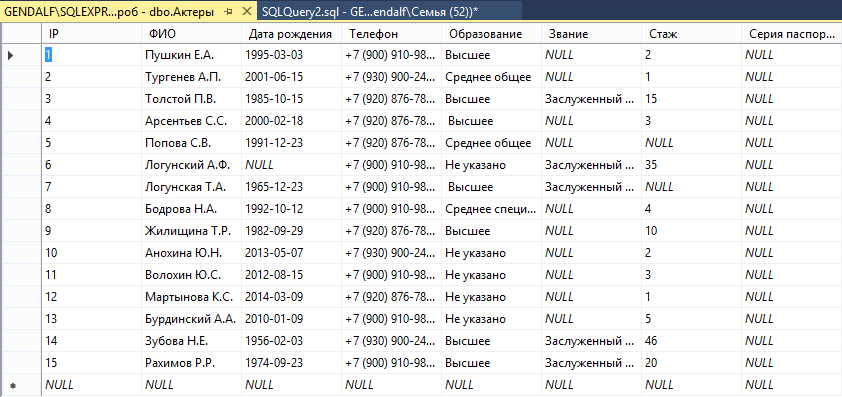
**Запросы на изменение структуры существующей таблицы:**

–Запрос на добавление нового столбца к таблице.

**49.** Добавить в таблицу Актеры столбец Серия паспорта.

ALTER TABLE Актеры

ADD [Серия паспорта] int



*Рисунок 59 – Результат запроса 49.*

–Запрос на добавление нового ограничения к таблице.

**50.** Сделать номер телефона в таблице Актеры уникальным значением.

ALTER TABLE Актеры

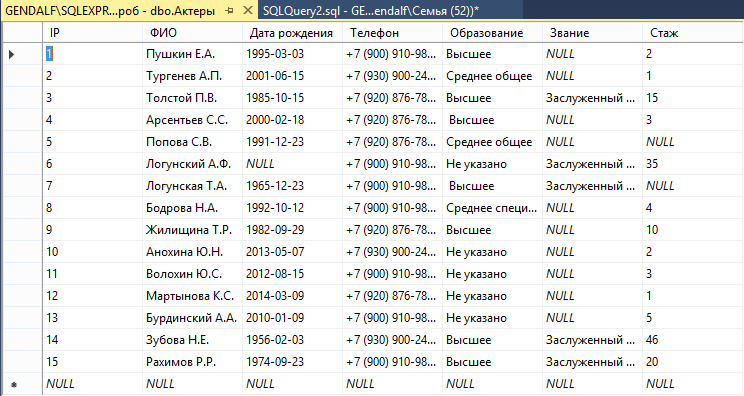
ADD CONSTRAINT tel UNIQUE (Телефон)

–Запрос на удаление нового столбца к таблице.

**51.** Удалить столбец Серия паспорта из таблицы Актеры.

ALTER TABLE Актеры

DROP COLUMN [Серия паспорта]



*Рисунок 60 – Результат запроса 51.*

–Запрос на удаление нового ограничения к таблице.

**52.** Удалить ограничение к номеру телефона в таблице Актеры.

ALTER TABLE Актеры

DROP CONSTRAINT tel

–Запрос на изменение типа данных в таблице.

**53.** Изменить тип данных у столбца Гонорар на int.

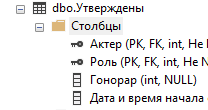
ALTER TABLE Утверждены

DROP CONSTRAINT CK\_\_Утвержден\_\_Гонор\_\_1BFD2C07

GO

ALTER TABLE Утверждены

ALTER COLUMN Гонорар int NULL



*Рисунок 61 – Результат запроса 53.*

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

**54.** Сделать первичным ключом, в таблице Актеры, ФИО.

ALTER TABLE Актеры

DROP CONSTRAINT PK\_IP

GO

ALTER TABLE Актеры

ADD CONSTRAINT PK PRIMARY KEY (ФИО)

**Прочие запросы**

**55.** Добавить в таблицу Кинопробы дополнительный столбец «Вероятность утверждения на роль». Разработать запросы, которые позволяют занести в столбец значение в соответствии со следующей таблицей:

|  |  |
| --- | --- |
| **Количество дублей** | **Вероятность утверждения на роль** |
| NULL | NULL |
| <3 | Средняя вероятность |
| 3 и выше | Низкая вероятность |

ALTER TABLE Кинопробы

ADD [Вероятность утверждения на роль] varchar (30)

GO

UPDATE Кинопробы

SET [Вероятность утверждения на роль] = 'NULL'

WHERE [Количество дублей]=NULL

GO

UPDATE Кинопробы

SET [Вероятность утверждения на роль] = 'Средняя вероятность'

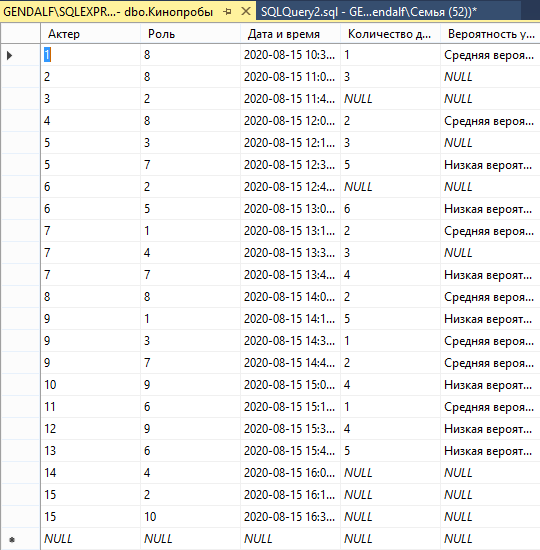
WHERE [Количество дублей]<3

GO

UPDATE Кинопробы

SET [Вероятность утверждения на роль] = 'Низкая вероятность'

WHERE [Количество дублей]>3



*Рисунок 62 – Результат запроса 55.*

**56.** Добавить в таблицу Утверждены дополнительный столбец «Гонорар в миллионах». Разработать запрос, который позволяет занести в столбец значение гонорара в миллионах.

ALTER TABLE Утверждены

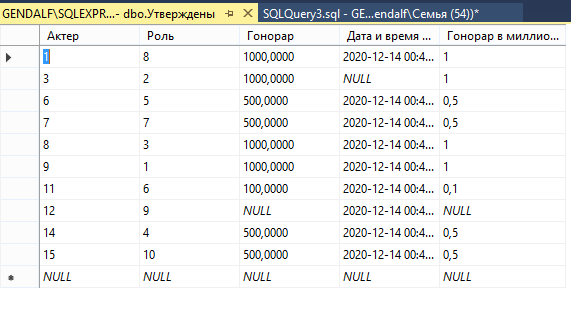
ADD [Гонорар в миллионах] FLOAT

GO

UPDATE Утверждены

SET [Гонорар в миллионах] = [Гонорар]/1000

FROM Утверждены



*Рисунок 63 – Результат запроса 56.*

**57.** Создать таблицу «Все о женихе», поместив в нее информацию о роли жениха.

CREATE TABLE [Все о женихе]

(Номер int,

Название\_роли varchar(20) not null,

Вид varchar(30) null,

Описание varchar(30) null)

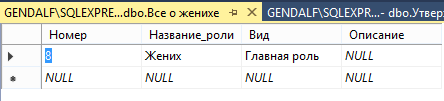
INSERT INTO [Все о женихе]

SELECT \*

FROM Роль

WHERE [Название роли] = ANY (SELECT [Название роли] FROM Роль

WHERE [Название роли] = 'Жених')



*Рисунок 64 – Результат запроса 57.*

**58.** Вывести информацию об актерах , ФИО которых начинаются с буквы К, Р или У.

SELECT \*

FROM Актеры

WHERE ФИО LIKE 'К%'

UNION

SELECT \*

FROM Актеры

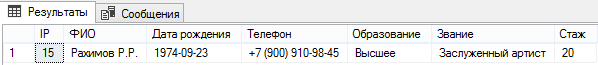
WHERE ФИО LIKE 'Р%'

UNION

SELECT \*

FROM Актеры

WHERE ФИО LIKE 'У%'



*Рисунок 65 – Результат запроса 58.*

**59.** Вывести информацию об утверждении актеров, изменив гонорар актеров в соответствии со следующей таблицей:

|  |  |  |
| --- | --- | --- |
|  | **IP актера** | **Уменьшение гонорара** |
| 1 | <4 | 30% |
| 2 | 4...9 | 10% |
| 3 | >9 | 0% |

SELECT [Актер],[Гонорар] AS [Старый гонорар], '30%' AS Уменьшение ,

[Гонорар]\*0.7 AS [Новый гонорар]

FROM Утверждены

WHERE [Актер]<4

UNION

SELECT [Актер],[Гонорар] AS [Старый гонорар],'10%' AS Уменьшение ,

[Гонорар]\*0.9 AS [Новый гонорар]

FROM Утверждены

WHERE [Актер] BETWEEN 4 AND 9

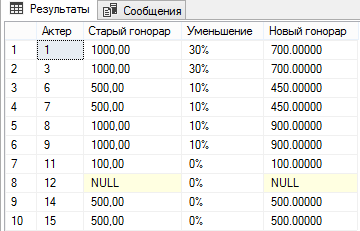
UNION

SELECT [Актер],[Гонорар] AS [Старый гонорар], '0%' AS Уменьшение ,

[Гонорар]\*1 AS [Новый гонорар]

FROM Утверждены

WHERE [Актер]>9;



*Рисунок 66 – Результат запроса 59.*

**60.** Вывести номера ролей, на которые пробовались и актер с IP= 1, и актер с IP= 4.

SELECT Кинопробы.Роль

FROM Кинопробы

WHERE Кинопробы.Актер='1'

INTERSECT

SELECT Кинопробы.Роль

FROM Кинопробы

WHERE Кинопробы.Актер='4'



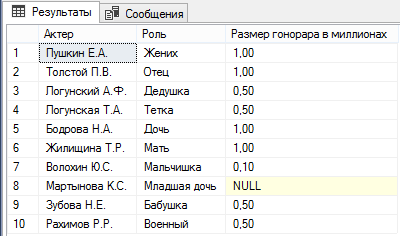
*Рисунок 67 – Результат запроса 60.*

**61.** Вывести информацию об утвержденных, с указанием ФИО, название роли, и гонораре в миллионе.

SELECT Актеры.ФИО AS [Актер], Роль.[Название роли] AS [Роль],ROUND(Утверждены.Гонорар/1000, 2) AS [Размер гонорара в миллионах]

FROM ([Актеры] INNER JOIN Утверждены ON Актеры.IP= Утверждены.Актер) INNER JOIN

Роль ON Роль.Номер = Утверждены.Роль



*Рисунок 68 – Результат запроса 61.*

**62.** Вывести список актеров, которые пробовались ТОЛЬКО на роль «Жених»

SELECT DISTINCT Кинопробы.Актер

FROM Актеры INNER JOIN Кинопробы ON Актеры.IP=Кинопробы.Актер

GROUP BY Кинопробы.Актер

EXCEPT

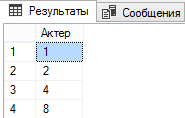
SELECT Кинопробы.Актер

FROM Кинопробы INNER JOIN Роль ON Кинопробы.Роль=Роль.Номер

WHERE Роль.Номер NOT IN (SELECT Роль.Номер

FROM Роль

WHERE [Название роли] = 'Жених')



*Рисунок 69 – Результат запроса 62.*

# ЗАКЛЮЧЕНИЕ

В результате данной курсовой работы была создана база данных Кинопроба.

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. Использование диалекта Transact-SQL позволило наиболее полно и качественно создать базу данных.

# СПИСОК ЛИТЕРАТУРЫ

1. М. Грабер. Введение в SQL. Изд. Лори, 2004. – 382с.
2. Ицик Бен-Ган Microsoft SQL Server 2008 Основы T-SQL:Пер. с англ. – СПб:БВХ-Петербург, 2009. – 432с.:ил.
3. Библиотека MSDN Microsoft SQL Server

<<http://msdn.microsoft.com/ru-ru/library/bb418431(SQL.10).aspx> >

1. Раздаточный материал Гринченко Н.Н.