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:

«Справочная система театра»

Выполнил студент группы 135

Шандала С.Д.

дата сдачи на проверку, подпись

Руководитель работы

ассистент каф. ЭВМ

Баранова С. Н.

оценка дата защиты, подпись

Содержание

[Введение 5](#_Toc122468005)

[1 Семантическое описание предметной области 6](#_Toc122468006)

[2 Разработка базы данных 7](#_Toc122468007)

[2.1 Разработка логической модели данных 7](#_Toc122468008)

[2.2 Определение типов хранимых данных 13](#_Toc122468009)

[2.3 Задание ограничений целостности данных 14](#_Toc122468010)

[2.4 Построение схемы БД 17](#_Toc122468011)

[2.5 Разработка сценария для создания БД и основных объектов структуры БД 18](#_Toc122468012)

[3 Разработка основных операций реляционной алгебры 21](#_Toc122468013)

[4 Разработка запросов к БД для реализации информационных потребностей пользователя 25](#_Toc122468014)

[Заключение 49](#_Toc122468015)

[Библиографический список 50](#_Toc122468016)

**Вариант задания.**

База данных «Справочная система театра» должна хранить информацию о расписании спектаклей и участвующих в них актерах. Требуется хранить данные:

1. Об актерах, а именно:

ФИО, дата рождения, адрес, стаж, звание, оклад.

1. О спектаклях, а именно:

название спектакля, жанр, возрастная категория, автор пьесы, режиссер, дата премьеры.

1. Занятость в спектакле, а именно:

актер, спектакль, роль.

1. Расписание спектаклей с указанием даты и времени начала и стоимости билета.

# **Введение**

В современном мире базы данных применяются практически везде. Язык SQL широко применяется в работе с базами данных. База данных – это упорядоченный набор структурированной информации или данных, которые обычно хранятся в электронном виде в компьютерной системе. База данных обычно управляется системой управления базами данных (СУБД).

Значительная сфера IT-сферы так или иначе связана с ним. Несмотря на свою кажущуюся простоту, язык SQL предоставляет огромные возможности для работы с базами данных и применяется во многих информационных системах. Информационная система (ИС) — [система](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0), предназначенная для хранения, поиска и обработки [информации](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D1%8F), и соответствующие организационные ресурсы (человеческие, технические, финансовые и т. д.), которые обеспечивают и распространяют информацию. ИС предназначена для своевременного обеспечения надлежащих людей надлежащей информацией, то есть для удовлетворения конкретных информационных потребностей в рамках определённой [предметной области](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D0%B4%D0%BC%D0%B5%D1%82%D0%BD%D0%B0%D1%8F_%D0%BE%D0%B1%D0%BB%D0%B0%D1%81%D1%82%D1%8C), при этом результатом функционирования информационных систем является информационная продукция.

В данной курсовой работе с помощью языка SQL была создана база данных для предметной области «Справочная система театра». Было разработано описание области, создание БД: создание логической модели данных, определение их типов, задание ограничений, построение схемы БД, а также создание БД с помощью языка SQL; разработка запросов с помощью реляционной алгебры, а также создание запросов к БД на языке SQL.

# **1 Семантическое описание предметной области**

Справочной системой театра называется система, которая предоставляет посетителям различную информацию о театре. Кроме зрителей справочная система также будет удобна работникам театра, чтобы быстро и точно узнавать расписание своей работы.

Каждый актер задействован в некотором количестве спектаклей. У каждого актера есть имя и дата рождения. Актерские звания ограничены некоторым набором. Если у актера не указано значение оклада, то оно приводится к минимальной заработной плате. Об актерах требуется хранить следующую информацию: ФИО, дата рождения, адрес, стаж, звание, оклад. Могут быть записи об актерах которые не задействованы ни в одном спектакле.

В одном спектакле может быть задействовано множество актеров. Возрастные категории спектаклей представлены следующим множеством чисел с какого возраста можно смотреть данный спектакль: {0, 3, 6, 10, 12, 14, 16,18}. Спектакли имеют ограниченный набор жанров, которые позволяют довольно точно охарактеризовать его направленность и целевую аудиторию. Не у каждой пьесы по которой ставятся спектакли есть четко известный автор. О каждом спектакле требуется хранить следующую информацию: название спектакля, жанр, возрастная категория, автор пьесы, режиссер, дата премьеры. Один актер может играть роль в одном спектакле в разные даты. Могут быть спектакли, которые сейчас не ставят.

О занятости в спектакле требуется хранить следующую информацию: актер, спектакль, роль. Один актер может быть задействован в одном спектакле в нескольких ролях. Может быть спектакль в котором задействован только один актер.

У спектаклей должно быть четкое расписание. Для каждого спектакля нужно хранить его дату и время начала и цену билета на него. Двух спектаклей в одно время не может идти. Но может быть несколько спектаклей в один день. Если цена не указана, то указывается цена в тысячу рублей.

# **2 Разработка базы данных**

# **2.1 Разработка логической модели данных**

***Описание основных сущностей ПО.***

На основании проведенного анализа предметной области «Справочная система театра» можно выделить следующие сущности: Актер, Спектакль, Занятость в спектакле, Расписание спектаклей. Представим список сущностей предметной области (таблица 1).

Таблица 1 – Список сущностей

|  |  |  |
| --- | --- | --- |
| № | Название | Назначение |
| 1 | Актер | Перечень актеров работающих в театре с информацией о них |
| 2 | Спектакль | Список спектаклей, которые ставятся в этом театре, с информацией о них |
| 3 | Занятость в спектакле | Список актеров и их роли в спектаклях |
| 4 | Расписание спектаклей | Расписание спектаклей и цены на билеты на них |

Для каждой сущности необходимо привести 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 сущности актер нет атрибута, с помощью которого можно однозначно идентифицировать его. Поэтому требуется ввести дополнительный атрибут, например, Табельный номер, который и будет являться первичным ключом.

Таблица 2 – Список атрибутов таблицы «Актер»

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Название | Назначение |
| ПК (первичный ключ) | Табельный номер | Ключевое поле.  Представляет собой первичный ключ. Это уникальное значение, соответствующее каждому актеру. Значения автоматически генерируется СУБД при вставке новой записи в таблицу. |
|  | ФИО | Хранит в себе полные фамилию имя и отчество актера. |
|  | Дата рождения | Хранит в себе информацию о дате рождения актера. |
|  | Адрес | Хранит в себе информацию о месте работы актера. |
|  | Стаж | Хранит в себе информацию о стаже работы актера. |
|  | Звание | Хранит в себе информацию о том какое звание имеет актер. |
|  | Оклад | Хранит в себе оклад, который платится актеру. |

Для сущности спектакль также нет атрибута, с помощью которого можно однозначно определить его. Поэтому для него также требуется ввести дополнительный атрибут. Например, номер код спектакля, который будет являться первичным ключом.

Таблица 3 – Список атрибутов таблицы «Спектакль»

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Название | Назначение |
| ПК (первичный ключ) | Код спектакля | Ключевое поле.  Представляет собой первичный ключ. Это уникальное значение, соответствующее каждому спектаклю. Значения автоматически генерируется СУБД при вставке новой записи в таблицу. |
|  | Название спектакля | Хранит в себе название спектакля. |
|  | Жанр | Хранит в себе информацию о том к какому жанру принадлежит спектакль. |
|  | Возрастная категория | Хранит в себе информацию о том с какого возраста можно смотреть данный спектакль. |
|  | Автор пьесы | Хранит в себе информацию о том кто был автором исходного текста, по которому ставится спектакль. |
|  | Режиссер | Хранит информацию о том кто взял исходный текст, поработал с ним и поставил спектакль. |
|  | Дата премьеры | Хранит в себе дату, в которую спектакль впервые был представлен зрителям. |

В таблице Занятость в спектакле отсутствует уникальное поле, которое можно было бы использовать в качестве первичного ключа. Исходя из предметной области один актер может играть в одном спектакле несколько ролей, поэтому сочетание из актера, роли и спектакля дает составной первичный ключ. Так же поле актер будет внешним ключом.

Таблица 4 – Список атрибутов таблицы «Занятость в спектакле»

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Название | Назначение |
| ПК (первичный ключ)  ВК (внешний ключ) | Актер | Является частью составного первичного ключа.  С помощью данного ключа можно определить какой актер занят в спектакле. |
| ПК (первичный ключ)  ВК (внешний ключ) | Спектакль | Является частью составного первичного ключа.  С помощью данного ключа можно определить в каком спектакле занят актер |
| ПК (первичный ключ) | Роль | Является частью составного первичного ключа.  С помощью данного ключа можно определить какой актер занят в спектакле. |

В таблице расписание спектаклей первичный ключ также будет составным. Он будет состоять из спектакля и даты и времени его начала. Так как один спектакль может ставиться много раз в разное время. В качестве внешнего ключа можно взять спектакль.

Таблица 5 – Список атрибутов таблицы «Расписание спектаклей»

|  |  |  |
| --- | --- | --- |
| Ключевое поле | Название | Назначение |
| ПК (первичный ключ)  ВК (внешний ключ) | Спектакль | Является частью составного первичного ключа.  С его можно определит какой спектакль будет ставиться. |
| ПК (первичный ключ) | Дата и время начала | Является частью составного первичного ключа.  С его можно определит какой спектакль будет ставиться. |
|  | Стоимость билета | Хранит в себе информацию о цене за билет на конкретный сеанс спектакля. |

***Построение логической модели***

На основании семантического описания предметной области и списка атрибутов из таблиц 2-5 опишем классы объектов (сущностей) и их свойства, расставим существующие связи между ними и приведем обоснование типов связей (1:1, 1:М и т.д.).

Результат требуется представим в виде таблицы Список связей с описанием характеристик связей (таблица 6) и логической модели в графическом виде, где будут изображены все таблицы и связи между ними (рисунок 1).

Таблица 6 – Список связей

|  |  |  |  |
| --- | --- | --- | --- |
| № | Сущности, участвующие в связи | Тип связи | Обоснование |
| 1 | Актер – Занятость в спектакле | 1:N | Один актер может играть во множестве спектаклей разные роли, но в конкретном спектакле и конкретную роль может играть только он. |
| 2 | Спектакль – Занятость в спектакле | 1:N | Один спектакль может ставиться много раз, в нем могут играть разные актеры, и может быть разный состав и разное распределение по ролям |
| 3 | Спектакль – Расписание спектаклей | 1:N | Один спектакль может ставиться много раз в разное время, но в одно время может проходить только один спектакль. |

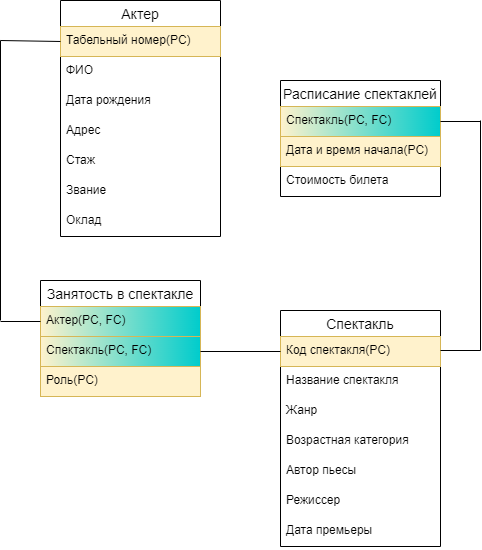


Рисунок 1 – Логическая модель предметной области «Справочная система театра»

# **2.2 Определение типов хранимых данных**

Для каждой таблицы приведем список всех атрибутов с указанием типа данных. При выборе типов данных следует учитывать, что далее для реализации будет использоваться СУБД Microsoft SQL Server. Результаты приведены в таблицах 7-10.

Таблица 7 – Список атрибутов таблицы «Актер»

|  |  |
| --- | --- |
| Название | Тип данных |
| Табельный номер | int,  нач. значение = 1, приращение = 1 |
| ФИО | varchar(40) |
| Дата рождения | date |
| Адрес | varchar(40) |
| Стаж | int |
| Звание | varchar(30) |
| Оклад | money |

Таблица 8 – Список атрибутов таблицы «Спектакль»

|  |  |
| --- | --- |
| Название | Тип данных |
| Код спектакля | int,  нач. значение = 1, приращение = 1 |
| Название спектакля | varchar(40) |
| Жанр | varchar(20) |
| Возрастная категория | varchar(5) |
| Автор пьесы | varchar(20) |
| Режиссер | varchar(20) |
| Дата премьеры | date |

Таблица 9 – Список атрибутов таблицы «Занятость в спектакле»

|  |  |
| --- | --- |
| Название | Тип данных |
| Актер | int |
| Спектакль | int |
| Роль | varchar(30) |

Таблица 10 – Список атрибутов таблицы «Расписание спектаклей»

|  |  |
| --- | --- |
| Название | Тип данных |
| Спектакль | int,  нач. значение = 1, приращение = 1 |
| Дата и время начала | datetime |
| Стоимость билета | money |

# **2.3 Задание ограничений целостности данных**

Для каждой таблицы необходимо выделить ограничения, накладываемые на значения атрибутов. Это могут быть ограничения на значения, значения по умолчанию, запрет/разрешение использования NULL-значений, поддержка уникальности значений и др. Результаты представлены в таблицах 11-14.

Таблица 11 – Ограничения таблицы «Актер». начало

|  |  |
| --- | --- |
| Название | Ограничение |
| Табельный номер | Первичный ключ. Это уникальное значение, не может принимать NULL-значения. Значения автоматически генерируются СУБД, начиная от 1 с шагом 1. |
| ФИО | Не может принимать NULL значения. |

Таблица 11 – Ограничения таблицы «Актер». завершение

|  |  |
| --- | --- |
| Дата рождения | ДД-ММ-ГГГГ. Не может принимать NULL значения. |
| Адрес | Единственное возможное значения – адрес театра для которого делается справочная система(г. Липецк, пл. Константиновой, д. 3) |
| Стаж | Не может принимать NULL значения. Ограничен от 0 до 100. |
| Звание | Ограничено театральными званиями(ведущий мастер сцены, артист первой категории, артист второй категории, артист высшей категории). |
| Оклад | В случае отсутствия информации приводится по умолчанию к значению минимальной заработной платы(14 000 рублей). Не может быть меньше 0. |

Таблица 12 – Ограничения таблицы «Спектакль». начало

|  |  |
| --- | --- |
| Название | Ограничение |
| Код спектакля | Первичный ключ. Это уникальное значение, не может принимать NULL-значения. Значения автоматически генерируются СУБД, начиная от 1 с шагом 1. |

Таблица 12 – Ограничения таблицы «Спектакль». завершение

|  |  |
| --- | --- |
| Название спектакля | Не может принимать NULL значения. Должно являться уникальным. |
| Жанр | Ограничено стандартным набором жанров(драма, комедия, трагедия, мюзикл, монодрама, трагикомедия). |
| Возрастная категория | {0, 3, 6, 10, 12, 14, 16,18}+. |
| Автор пьесы | В случае если неизвестен автор значение по умолчанию «Неизвестно». |
| Режиссер | Не может принимать NULL значения. |
| Дата премьеры | ДД-ММ-ГГГГ. Не может принимать NULL значения. |

Таблица 13 – Ограничения таблицы «Занятость в спектакле»

|  |  |
| --- | --- |
| Название | Ограничение |
| Актер | Атрибут, входящий в составной первичный ключ.  Атрибут, входящий во внешний ключ |
| Спектакль | Атрибут, входящий в составной первичный ключ.  Атрибут, входящий во внешний ключ |
| Роль | Атрибут, входящий в составной первичный ключ. Не может принимать NULL значения. |

Таблица 14 – Ограничения таблицы «Расписание спектаклей»

|  |  |
| --- | --- |
| Название | Ограничение |
| Спектакль | Атрибут, входящий в составной первичный ключ.  Атрибут, входящий во внешний ключ. |
| Дата и время начала | Атрибут, входящий в составной первичный ключ.  Спектакли могут начинаться с 8 до 20 часов.  ДД-ММ-ГГГГ ЧЧ:ММ:СС.000 |
| Стоимость билета | В случае отсутствия информации приводится по умолчанию к значению в 1000 рублей. Не может быть меньше 0 рублей. Для одного спектакля, проводящегося в разное время может различаться. |

# **2.4 Построение схемы БД**

Схема БД представляет собой набор заголовков таблиц. То есть на схеме БД указываются имена таблиц (сущностей), их атрибуты и связи между таблицами. На рисунке 2 представлена схема базы данных «Справочная система театра».

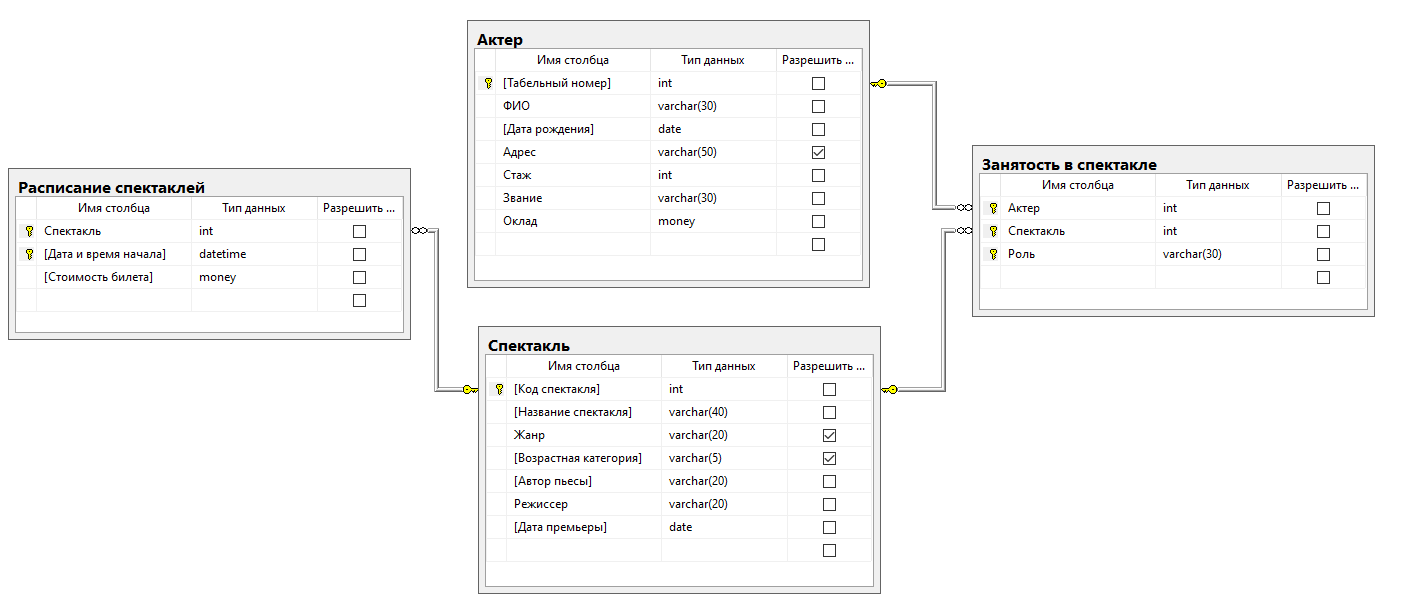


Рисунок 2 – Схема базы данных «Справочная система театра»

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

**--Создание файлов БД**

USE master

GO

CREATE DATABASE СправочнаяСистемаТеатра

ON

(NAME = 'СправочнаяСистемаТеатра',

FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL12.SQLEXPRESS\MSSQL\DATA\СправочнаяСистемаТеатра.mdf',

SIZE = 10,

MAXSIZE = 30,

FILEGROWTH = 1)

LOG ON

(NAME = 'СправочнаяСистемаТеатра\_log',

FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL12.SQLEXPRESS\MSSQL\DATA\СправочнаяСистемаТеатра\_log.ldf'

,

SIZE = 10,

MAXSIZE = 20,

FILEGROWTH = 1)

GO

**--Создание и заполнение таблиц**

USE СправочнаяСистемаТеатра

GO

CREATE TABLE Актер

([Табельный номер] int not null identity (1, 1),

ФИО varchar(40) not null,

[Дата рождения] date not null,

Адрес varchar(50) default 'г. Липецк, пл. Константиновой, д. 3',

Стаж int not null check (Стаж between 0 and 100),

Звание varchar(30)not null check (Звание in ('Ведущий мастер сцены','Артист первой категории', 'Артист второй категории', 'Артист высшей категории')),

Оклад money not null check (Оклад > 0) default 14000)

GO

ALTER TABLE Актер

ADD CONSTRAINT PK\_Актер primary key

([Табельный номер])

GO

CREATE TABLE Спектакль

([Код спектакля] int not null identity (1, 1),

[Название спектакля] varchar(40) not null unique,

Жанр varchar(20) check (Жанр in ('Драма','Комедия', 'Трагедия', 'Мюзикл','Монодрама','Трагикомедия')),

[Возрастная категория] varchar(5) check([Возрастная категория] in ('0+','3+', '6+', '10+','12+','14+','16+','18+')),

[Автор пьесы] varchar(20) not null default 'Неизвестно',

Режиссер varchar(20) not null,

[Дата премьеры] date not null,

)

GO

ALTER TABLE Спектакль

ADD CONSTRAINT PK\_Спектакль primary key

([Код спектакля])

GO

CREATE TABLE [Занятость в спектакле]

(Актер int not null,

Спектакль int not null,

Роль varchar(30) not null,)

GO

ALTER TABLE [Занятость в спектакле]

ADD CONSTRAINT PK\_Занятость\_в\_спектакле primary key (Актер, Спектакль, Роль)

GO

ALTER TABLE [Занятость в спектакле]

ADD CONSTRAINT FK\_Занятость\_в\_спектакле\_Актер foreign key

(Актер) references Актер([Табельный номер])on delete cascade

GO

ALTER TABLE [Занятость в спектакле]

ADD CONSTRAINT FK\_Занятость\_в\_спектакле\_Спектакль foreign key

(Спектакль) references Спектакль([Код спектакля])on delete cascade

GO

CREATE TABLE [Расписание спектаклей]

(Спектакль int not null,

[Дата и время начала] datetime not null check (datepart(hour, [Дата и время начала]) between 8 and 20),

[Стоимость билета] money not null check ([Стоимость билета] > 0) default 1000)

GO

ALTER TABLE [Расписание спектаклей]

ADD CONSTRAINT PK\_Расписание\_спектаклей primary key (Спектакль , [Дата и время начала])

GO

ALTER TABLE [Расписание спектаклей]

ADD CONSTRAINT FK\_Расписание\_спектаклей\_Спектакль foreign key

(Спектакль) references Спектакль([Код спектакля])on delete cascade

GO

DBCC CHECKIDENT('Актер', RESEED, 1)

GO

INSERT INTO Актер VALUES

('Мельгер Михаил Михайлович','17-02-1966', default,35 ,'Ведущий мастер сцены',110000),

('Ароньел Григорий Олегович','03-03-1993', default,3 ,'Артист первой категории',80000),

('Леверьярд Дмитрий Андреевич','26-11-1984',default, 10 ,'Артист высшей категории',90000),

('Крайт Роман Евгеньевич','20-07-1957',default,47 ,'Артист высшей категории',90000),

('Иловаев Игорь Константинович','01-12-1991', default,6 ,'Артист первой категории',80000),

('Иловаева Дарья Андреевна','02-12-1995', default,2 ,'Артист высшей категории',100000),

('Анкелонов Владислав Игоревич','16-10-1949',default,51 ,'Артист высшей категории',65000),

('Грэхович Михаил Евгеньевич','05-06-1975',default, 22 ,'Артист высшей категории',90000),

('Вельхуров Сергей Сергеевич','13-05-1997',default, 0 ,'Артист второй категории',60000),

('Вельхурова Наталья Михайловна','29-04-1997', default, 0 ,'Артист второй категории',60000)

GO

DBCC CHECKIDENT('Спектакль', RESEED, 1)

GO

INSERT INTO Спектакль VALUES

('Фауст','Трагедия', '16+','Иоганн Гете' ,'Николай Гриммов','12-05-2020'),

('Манфред','Трагедия', '14+','Джордж Байрон' ,'Вячеслав Виртаров','20-10-2022'),

('Рюи Блаз','Драма ','16+', 'Виктор Гюго' ,'Вячеслав Виртаров','26-03-2020'),

('Король Лир','Трагедия','16+','Уильям Шекспир' ,'Даниил Ксихров','22-08-2021'),

('Гамлет','Трагедия', '16+','Уильям Шекспир' ,'Даниил Ксихров','11-06-2021'),

('Троил и Крессида','Трагикомедия', '18+','Уильям Шекспир' ,'Елена Троянова','06-01-2022'),

('Венецианский купец','Комедия','16+','Уильям Шекспир' ,'Вячеслав Виртаров','06-06-2020'),

('Юлий Цезарь','Трагедия','14+', 'Уильям Шекспир' ,'Николай Гриммов','03-11-2022'),

('Макбет','Трагедия','16+', 'Уильям Шекспир' ,'Вячеслав Виртаров','17-02-2022'),

('Сердце кошмара','Драма', '18+', default,'Николай Гриммов','30-08-2022')

GO

INSERT INTO [Занятость в спектакле] VALUES

(9,4, 'Эдгар, сын Глостера'),

(6,4, 'Регана, дочь Лира'),

(4,4,'Король Лир'),

(9,5,'Гамлет, принц датский'),

(10,5, 'Офелия, дочь Полония'),

(2,5,'Лаэрт, сын Полония'),

(1,5, 'Клавдий, король датский'),

(6,5, 'Гертруда, королева датская'),

(3,8, 'Марк Брут, заговорщик'),

(5,8,'Марк Антоний, триумвир'),

(2,3,'Рюи Блаз'),

(1,10,'Мастер труппы'),

(1,7, 'Антонио, венецианский купец'),

(7,7, 'Шейлок, богатый еврей'),

(3,1, 'Фауст, талантливый ученый'),

(1,1,'Мефистофель, злой дух'),

(3,2,'Манфред, чернокнижник'),

(4,6, 'Уллис, греческий герой'),

(9,6,'Троил, сын царя Приама'),

(6,6,'Крессида, знатная троянка')

GO

INSERT INTO [Расписание спектаклей] VALUES

(10,'01.09.2023 10:00:00',2500),

(2,'14.03.2023 13:30:00',1700),

(2,'13.02.2023 19:30:00',2000),

(5,'13.06.2023 15:00:00',4000),

(5,'12.03.2023 17:30:00',3600),

(5,'06.10.2023 20:00:00',3400),

(7,'30.05.2023 12:00:00',default),

(7,'22.11.2023 13:00:00',default),

(4,'03.07.2023 18:00:00',2900),

(4,'19.05.2023 15:30:00',2300),

(4,'15.10.2023 15:00:00',2900),

(4,'29.06.2023 18:30:00',2900),

(1,'26.06.2023 17:30:00',2100),

(1,'03.05.2023 20:00:00',2100),

(1,'20.05.2023 11:00:00',2100),

(1,'10.07.2023 14:00:00',2300),

(1,'07.01.2023 13:30:00',2400),

(3,'31.01.2023 17:00:00',default),

(6,'24.06.2023 14:30:00',1400),

(6,'02.04.2023 12:00:00',1400)

GO

# **3 Разработка основных операций реляционной алгебры**

1. Вывести полную информацию о пьесах, автор которых Шекспир.

R = Спектакль WHERE Автор\_пьесы = 'Уильям Шекспир'

1. Вывести название спектакля, жанр и авторов пьес, режиссер которых Виртаров.

R = (Спектакль WHERE Режиссер = 'Вячеслав Виртаров') [Название\_спектакля, Жанр, Автор\_пьесы]

1. Вывести фамилии актеров с окладом больше 80 тысяч

R = (Актер WHERE Оклад > 80000) [ФИО]

1. Вывести полную информацию о занятости в концерте с указанием ФИО и звания актера и названием и жанром спектакля.

R1 = Актер RENAME Табельный\_номер AS Актер

R2 = Спектакль RENAME Код\_спектакля AS Спектакль

R3 = ((R1 JOIN Занятость\_в\_спектакле) JOIN R2) [Роль, ФИО, Звание, Название\_спектакля, Жанр]

1. Вывести полную информацию обо всех актерах. Если актер играет в каком-то спектакле, то указать его роль

R1 = Занятость\_в\_спектакле RENAME Актер AS Табельный\_номер

R2 = (Актер LEFT JOIN R1) [ФИО, Дата\_рождения, Адрес, Стаж, Звание, Оклад, Роль]

1. Вывести всю информацию о спектаклях с возрастной категорией 14+ и 16+.

R1 = Спектакль WHERE Возрастная\_ категория = '14+'

R2 = Спектакль WHERE Возрастная\_ категория = '16+'

R3 = R1 UNION R2

1. Вывести всю информацию о спектаклях с жанром трагедия, автор пьесы которых не Шекспир.

R1 = Спектакль WHERE Жанр = 'Трагедия'

R2 = Спектакль WHERE Автор\_пьесы != 'Уильям Шекспир'

R3 = R1 INTERSECT R2

1. Вывести ФИО и стаж актеров, которые не играют в спектаклях.

R1 = Актер [Табельный\_номер]

R2 = Занятость\_в\_спектакле [Актер] RENAME Актер AS Табельный\_номер

R3 = R1 EXCEPT R2

R4 = (R3 JOIN Актер) [ФИО, Стаж]

1. Вывести полную информацию об актерах, которые не играли ни в одной драме.

R1 = Актер [Табельный\_номер]

R2 = Спектакль RENAME Код\_спектакля AS Спектакль

R3 = R2 JOIN Занятость\_в\_спектакле

R4 = (R3 WHERE Жанр = 'Драма') [Актер]

R5 = R4 RENAME Актер AS Табельный\_номер

R6 = R1 EXCEPT R5

R7 = R6 JOIN Актер

1. Вывести название спектаклей в которых играют актеры высшей категории, но не играет ведущий мастер сцены.

R1 = Занятость\_в\_спектакле RENAME Актер AS Табельный\_номер

R2 = R1 JOIN Актер

R3 = R2 WHERE Звание = 'Актер высшей категории'

R4 = R2 WHERE Звание = 'Ведущий мастер сцены'

R5 = R3 EXCEPT R4

R6 = R5 RENAME Спектакль AS Код\_спектакля

R7 = (R6 JOIN Спектакль) [Название\_спектакля]

1. Вывести ФИО и дату рождения актеров которые не играли ни у Трояновой, ни у Гриммова.

R1 = Актер [Табельный\_номер]

R2 = Занятость\_в\_спектакле RENAME Спектакль AS Код\_спектакля

R3 = R2 JOIN Спектакль

R4 = (R3 WHERE Режиссер = 'Елена Троянова') [Актер]

R5 = (R3 WHERE Режиссер = 'Николай Гриммов') [Актер]

R6 = R4 RENAME Актер AS Табельный\_номер

R7 = R5 RENAME Актер AS Табельный\_номер

R8 = R1 EXCEPT R6

R9 = R8 EXCEPT R7

R10 = (R9 JOIN Актер) [ФИО, Дата\_рождения]

1. Вывести название спектакля в котором играют все актеры.

R1 = Актер [Табельный\_номер]

R2 = Занятость\_в\_спектакле RENAME Актер AS Табельный\_номер

R3 = R2 [Спектакль, Табельный\_номер]

R4 = R3 DIVIDEBY R1

R5 = R4 RENAME Спектакль AS Код\_спектакля

R6 = (R5 JOIN Спектакль) [Название\_спектакля]

1. Вывести ФИО актеров, которые играют только в комедиях.

R1 = Спектакль RENAME Код\_спектакля AS Спектакль

R2 = R1 JOIN Занятость\_в\_спектакле

R3 = (R2 WHERE Жанр = 'Комедия') [Актер]

R4 = (R2 WHERE Жанр != 'Комедия') [Актер]

R5 = R3 EXCEPT R4

R6 = R5 RENAME Актер AS Табельный\_номер

R7 = (R6 JOIN Актер) [ФИО]

1. Вывести информацию о спектаклях в которых играют актеры второй категории в виде: ФИО актера, название спектакля, автор исходной пьесы, режиссер, роль.

R1 = Занятость\_в\_спектакле RENAME Актер AS Табельный\_номер, Спектакль AS Код\_спектакля

R2 = (R1 JOIN Актер) RENAME ФИО AS ФИО\_актера

R3 = (R2 JOIN Спектакль) RENAME Автор\_пьесы AS Автор\_исходной\_пьесы

R4 = R3 WHERE Звание = 'Актер второй категории'

R5 = R5 [ФИО\_актера, Название\_спектакля, Автор\_исходной\_пьесы, Режиссер, Роль]

1. Вывести название и режиссера пьес категории 16+, в которых играют актеры с окладом меньше 90 тысяч и цена на билеты на который больше 2 тысяч.

R1 = Спектакль RENAME Код\_спектакля AS Спектакль

R2 = Актер RENAME Табельный\_номер AS Актер

R3 = (((R1 JOIN Расписание спектаклей) JOIN Занятость\_в\_спектакле) JOIN R2)

R4 = R3 WHERE Возрастная\_ категория = '16+'

R5 = R4 WHERE Оклад < 90000

R6 = R5 WHERE Стоимость\_билета > 2000

R7 = R6 [Название\_спектакля, Режиссер]

# **4 Разработка запросов к БД для реализации информационных потребностей пользователя**

Таблицы базы данных, к которым будут создаваться запросы представлены на рисунках 3-6.

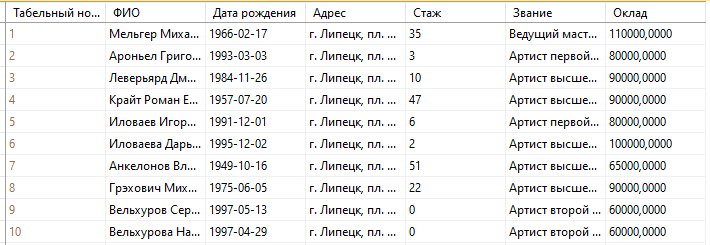


Рисунок 3 – Таблица "Актер"

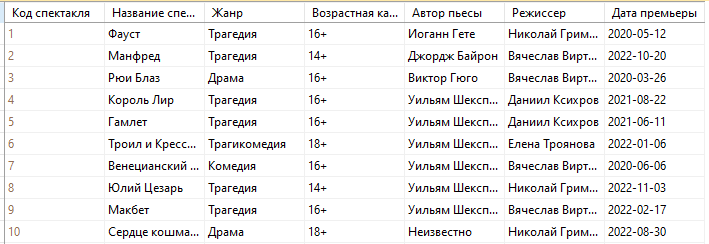


Рисунок 4 – Таблица "Спектакль"



Рисунок 5 – Таблица "Занятость в спектакле"

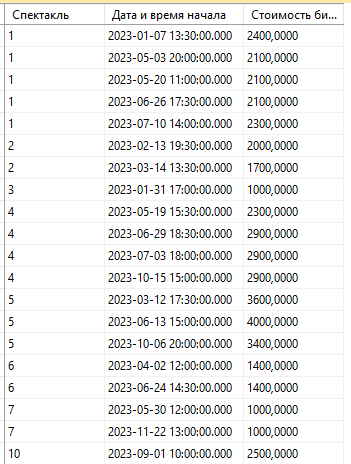


Рисунок 6 – Таблица "Расписание спектаклей"

1. Запросы с выбором строк по условию:

Запрос с использованием операторов сравнения – Вывести всю информацию о спектаклях, автор пьесы которых не Шекспир(рисунок 7).

SELECT \*

FROM Спектакль

WHERE [Автор пьесы] != 'Уильям Шекспир'

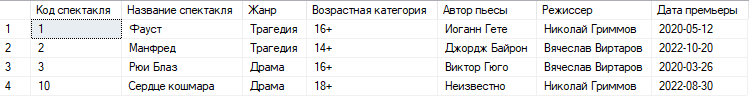


Рисунок 7 – Запрос с использованием операторов сравнения

Запрос с использованием логических операторов AND, OR и NOT – Вывести всех ведущих мастеров сцены или тех актеров чей стаж меньше 10 лет(рисунок 8).

SELECT \*

FROM Актер

WHERE Звание = 'Ведущий мастер сцены' Or Стаж < 10

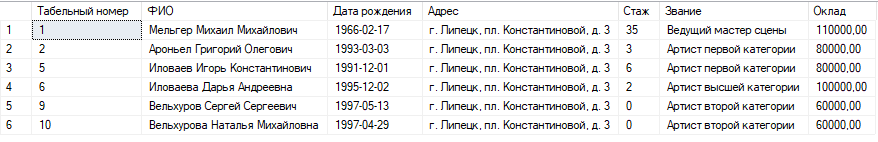


Рисунок 8 – Запрос с использованием логических операторов AND, OR и NOT

Запрос на использование комбинации логических операторов – Вывести все спектакли режиссер которых Виртаров и которые не относятся к трагедиям или драмам(рисунок 9).

SELECT \*

FROM Спектакль

WHERE Режиссер = 'Вячеслав Виртаров' AND NOT (Жанр = 'Трагедия' OR Жанр = 'Драма')



Рисунок 9 – Запрос на использование комбинации логических операторов

Запрос на использование выражений над столбцами – Вывести всю информацию об актерах с увеличением им оклада в 1.2 раза, стажа на 5 лет и добавлению перед званием слова профессиональный(рисунок 10).

SELECT [Табельный номер], ФИО, [Дата рождения], Адрес,

Стаж + 5 AS Стаж, 'Профессиональный ' + Звание AS Звание, Оклад \* 1.2 AS Оклад

FROM Актер

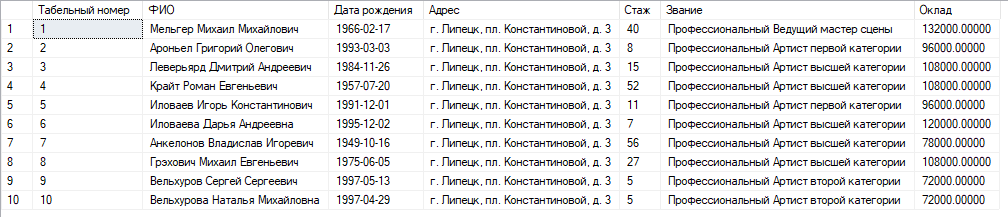


Рисунок 10 – Запрос на использование выражений над столбцами

Запрос с проверкой на принадлежность множеству – Вывести всю информацию о спектаклях, авторы пьес которых Гюго и Гете(рисунок 11).

SELECT \*

FROM Спектакль

WHERE [Автор пьесы] IN ('Иоганн Гете','Виктор Гюго')



Рисунок 11 – Запрос с проверкой на принадлежность множеству

Запрос с проверкой на принадлежность диапазону значений – Вывести всю информацию о расписании спектаклей которые стоят между 2 и 3 тысячами рублей(рисунок 12).

SELECT \*

FROM [Расписание спектаклей]

WHERE [Стоимость билета] BETWEEN 2000 AND 3000

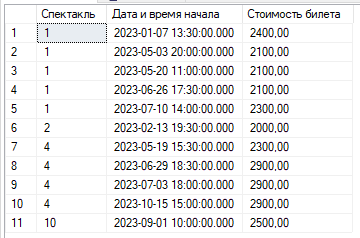


Рисунок 12 – Запрос с проверкой на принадлежность диапазону значений

Запрос с проверкой на соответствие шаблону – Вывести всю информацию о занятости в спектакле для ролей начинающихся на буквы М, Г и О( рисунок 13).

SELECT \*

FROM [Занятость в спектакле]

WHERE Роль LIKE'[МГО]%'

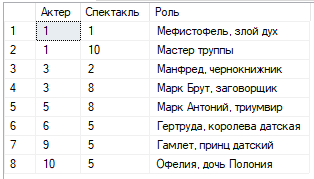


Рисунок 13 – Запрос с проверкой на соответствие шаблону

Запрос с проверкой на неопределенное значение – Вывести всю информацию о спектаклях жанр которых определен( рисунок 14).

SELECT \*

FROM Спектакль

WHERE Жанр IS NOT NULL



Рисунок 14 – Запрос с проверкой на неопределенное значение

2. Запросы на теоретико-множественные реляционные операции:

Запрос на объединение таблиц – Вывести всю информацию о спектаклях с возрастной категорией 14+ и 16+(рисунок 15).

SELECT \*

FROM Спектакль

WHERE [Возрастная категория] = '14+'

UNION

SELECT \*

FROM Спектакль

WHERE [Возрастная категория] = '16+'

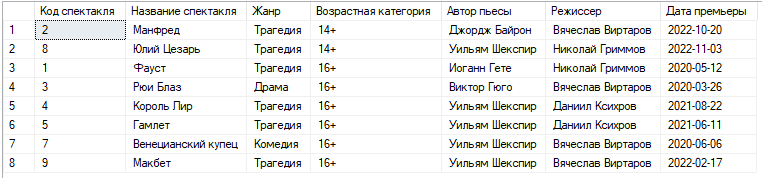


Рисунок 15 – Запрос на объединение таблиц

Запрос на пересечение таблиц – Вывести всю информацию о спектаклях с жанром трагедия, автор пьесы которых Шекспир(рисунок 16).

SELECT \*

FROM Спектакль

WHERE Жанр = 'Трагедия'

INTERSECT

SELECT \*

FROM Спектакль

WHERE [Автор пьесы] = 'Уильям Шекспир'

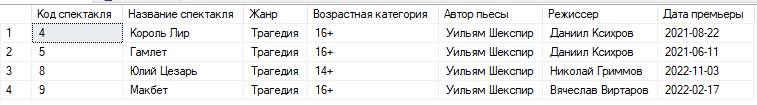


Рисунок 16 – Запрос на пересечение таблиц

Запрос на вычитание таблиц – Вывести всю информацию только об актерах с окладом меньше 90 тысяч, не входящих в артистов первой категории(рисунок 17).

SELECT \*

FROM Актер

WHERE Оклад < 90000

EXCEPT

SELECT \*

FROM Актер

WHERE Звание = 'Артист первой категории'

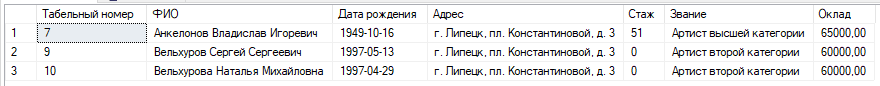


Рисунок 17 – Запрос на вычитание таблиц

Запрос с использованием декартового произведения таблиц – Для артистов Мельгера и Грэховича вывести список всех возможных ролей в спектаклях с номерами 3, 4 и 8(рисунок 18).

SELECT ФИО, Спектакль, Роль

FROM Актер,[Занятость в спектакле]

WHERE ФИО IN ('Мельгер Михаил Михайлович','Грэхович Михаил Евгеньевич') AND Спектакль IN (3,4,8)



Рисунок 18 – Запрос с использованием декартового произведения таблиц

3. Многотабличные запросы:

Запрос с использованием соединения двух таблиц по равенству и условием отбора – Вывести все роли для спектаклей по пьесам Шекспира(рисунок 19).

SELECT Роль

FROM [Занятость в спектакле] JOIN Спектакль ON [Занятость в спектакле].Спектакль = Спектакль.[Код спектакля]

WHERE [Автор пьесы] = 'Уильям Шекспир'

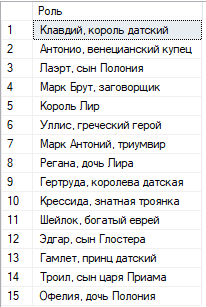


Рисунок 19 – Запрос с использованием соединения двух таблиц по равенству и условием отбора

Запрос с использованием соединения трех таблиц по равенству и условием отбора – Вывести ФИО актеров, которые играли в спектаклях режиссера Ксихрова(рисунок 20).

SELECT DISTINCT ФИО

FROM [Занятость в спектакле] JOIN Спектакль ON [Занятость в спектакле].Спектакль = Спектакль.[Код спектакля]

JOIN Актер ON [Занятость в спектакле].Актер = Актер.[Табельный номер]

WHERE Режиссер = 'Даниил Ксихров'

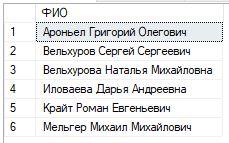


Рисунок 20 – Запрос с использованием соединения трех таблиц по равенству и условием отбора

Запрос с использованием соединения по четырем таблицам – Вывести ФИО актеров которые играли в спектаклях с ценой билета от 3 до 4 тысяч(рисунок 21).

SELECT DISTINCT ФИО

FROM [Занятость в спектакле] JOIN Спектакль ON [Занятость в спектакле].Спектакль = Спектакль.[Код спектакля]

JOIN Актер ON [Занятость в спектакле].Актер = Актер.[Табельный номер]

JOIN [Расписание спектаклей] ON [Расписание спектаклей].Спектакль = Спектакль.[Код спектакля]

WHERE [Стоимость билета] BETWEEN 3000 AND 4000

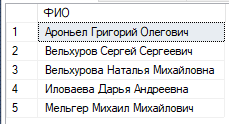


Рисунок 21 – Запрос с использованием соединения по четырем таблицам

Запрос с использованием левого внешнего соединения– вывести ФИО и стаж актера высшей категории, если задействован в спектакле указать роль(рисунок 22).

SELECT ФИО, Стаж, Роль

FROM Актер LEFT JOIN [Занятость в спектакле] ON [Занятость в спектакле].Актер = Актер.[Табельный номер]

WHERE Звание = 'Артист высшей категории'

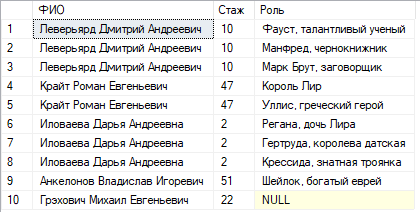


Рисунок 22 – Запрос с использованием левого внешнего соединения

Запрос на использование правого внешнего соединения – Вывести название и возрастную категорию спектаклей авторства Шекспира с указанием даты и времени начала(рисунок 23).

SELECT [Название спектакля], [Возрастная категория], [Дата и время начала]

FROM [Расписание спектаклей] RIGHT JOIN Спектакль ON [Расписание спектаклей].Спектакль = Спектакль.[Код спектакля]

WHERE [Автор пьесы] = 'Уильям Шекспир'

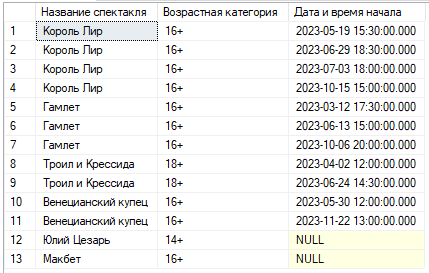


Рисунок 23 – Запрос на использование правого внешнего соединения

Запрос с использованием функции COUNT – Посчитать количество актеров, задействованных в спектаклях Гриммова(рисунок 24).

SELECT COUNT(DISTINCT ФИО) AS [Количество актеров]

FROM [Занятость в спектакле] JOIN Спектакль ON [Занятость в спектакле].Спектакль = Спектакль.[Код спектакля]

JOIN Актер ON [Занятость в спектакле].Актер = Актер.[Табельный номер]

WHERE Режиссер = 'Николай Гриммов'



Рисунок 24 – Запрос с использованием функции COUNT

Запрос с использованием функции SUM – Посчитать сумму стоимостей билетов на спектакли жанра драма или трагикомедия(рисунок 25).

SELECT SUM([Стоимость билета]) AS [Сумма]

FROM Спектакль JOIN [Расписание спектаклей] ON [Расписание спектаклей].Спектакль = Спектакль.[Код спектакля]

WHERE Жанр IN ('Драма ','Трагикомедия')



Рисунок 25 – Запрос с использованием функции SUM

3 запроса с использованием временных функций:

1. Получить названия спектаклей которые начинаются либо в 12 либо в 15 часов(рисунок 26).

SELECT DISTINCT [Название спектакля]

FROM Спектакль JOIN [Расписание спектаклей] ON [Расписание спектаклей].Спектакль = Спектакль.[Код спектакля]

WHERE DATEPART(HOUR, [Дата и время начала]) = 12 OR DATEPART(HOUR, [Дата и время начала]) = 15

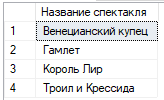


Рисунок 26 – Первый запрос с использованием временных функций

2. Получить ФИО актеров, которые играют в спектаклях, которые впервые были поставлены во второй половине месяца(рисунок 27).

SELECT DISTINCT ФИО

FROM [Занятость в спектакле] JOIN Спектакль ON [Занятость в спектакле].Спектакль = Спектакль.[Код спектакля]

JOIN Актер ON [Занятость в спектакле].Актер = Актер.[Табельный номер]

WHERE DATEPART(DAY, [Дата премьеры]) BETWEEN 16 AND 31

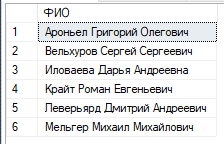


Рисунок 27 – Второй запрос с использованием временных функций

3. Вывести название спектаклей, которые будут ставиться весной и в которых будут играть актеры 1980-2000 года рождения(рисунок 28).

SELECT DISTINCT [Название спектакля]

FROM [Занятость в спектакле] JOIN Спектакль ON [Занятость в спектакле].Спектакль = Спектакль.[Код спектакля]

JOIN Актер ON [Занятость в спектакле].Актер = Актер.[Табельный номер]

JOIN [Расписание спектаклей] ON [Расписание спектаклей].Спектакль = Спектакль.[Код спектакля]

WHERE (DATEPART(MONTH, [Дата и время начала]) BETWEEN 3 AND 5) AND DATEPART(YEAR, [Дата рождения]) BETWEEN 1980 AND 2000

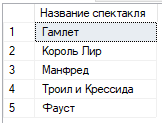


Рисунок 28 – Третий запрос с использованием временных функций

Запрос с использованием группировки по одному столбцу – Посчитать количество ролей в спектаклях каждого режиссера(Рисунок 29).

SELECT Режиссер, COUNT(Роль) AS [Количество ролей]

FROM [Занятость в спектакле] JOIN Спектакль ON [Занятость в спектакле].Спектакль = Спектакль.[Код спектакля]

GROUP BY Режиссер

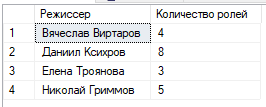


Рисунок 29 – Запрос с использованием группировки по одному столбцу

Запрос на использование группировки по нескольким столбцам – Посчитать количество ролей для каждой категории в каждом жанре(рисунок 30).

SELECT [Возрастная категория],Жанр, COUNT(Роль) AS [Количество ролей]

FROM [Занятость в спектакле] JOIN Спектакль ON [Занятость в спектакле].Спектакль = Спектакль.[Код спектакля]

GROUP BY [Возрастная категория], Жанр

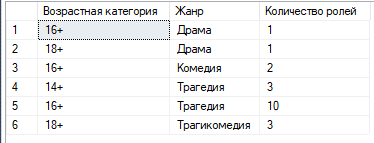


Рисунок 30 – Запрос на использование группировки по нескольким столбцам

Запрос с использованием условия отбора групп HAVING – Вывести звания, актеры которых сыграли в сумме 4 роли(рисунок 31).

SELECT Звание

FROM [Занятость в спектакле] JOIN Актер ON [Занятость в спектакле].Актер = Актер.[Табельный номер]

GROUP BY Звание

HAVING COUNT(Роль) = 4



Рисунок 31 – Запрос с использованием условия отбора групп HAVING

Запрос с использованием сортировки по столбцу – К каждому артисту вывести количество ролей, отсортировав их в алфавитном порядке(рисунок 32).

SELECT ФИО, COUNT(Роль) AS [Количество ролей]

FROM [Занятость в спектакле] RIGHT JOIN Актер ON [Занятость в спектакле].Актер = Актер.[Табельный номер]

GROUP BY ФИО

ORDER BY ФИО

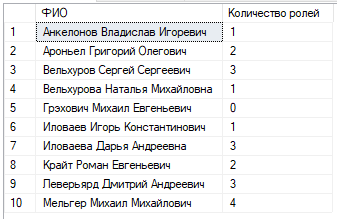


Рисунок 32 – Запрос с использованием сортировки по столбцу

4. Запросы на модификацию таблиц:

Запрос на добавление новых данных в таблицу – Добавить информацию о новом спектакле(рисунок 33).

INSERT INTO Спектакль

VALUES ('Царь Эдип','Трагедия','18+','Софокл','Даниил Ксихров','2021-09-12');

GO

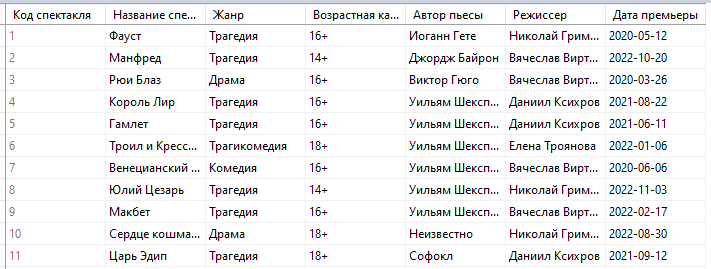


Рисунок 33 – Запрос на добавление новых данных в таблицу

Запрос на добавление новых данных по результатам выполненного запроса – Добавить в таблицу всех актеров первой категории(рисунок 34).

INSERT INTO Актер

SELECT ФИО, [Дата рождения], Адрес, Стаж, Звание, Оклад

FROM Актер

WHERE Звание = 'Артист первой категории'

GO

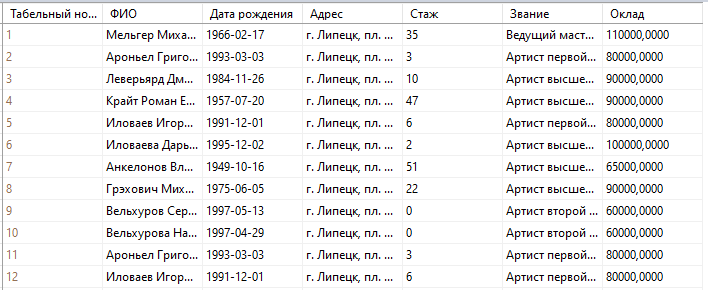


Рисунок 34 – Запрос на добавление новых данных по результатам выполненного запроса

Запрос на обновление существующих данных в таблице – Увеличить актерам стаж на 10 лет(рисунок 35).

UPDATE Актер

SET Стаж = Стаж + 10

GO

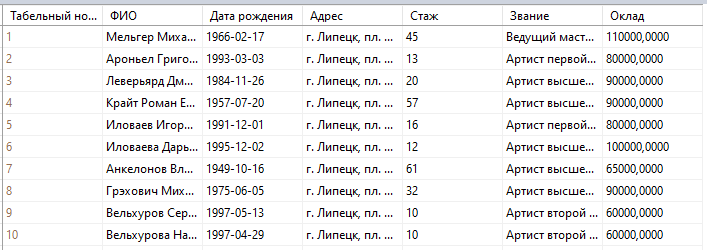


Рисунок 35 – Запрос на обновление существующих данных в таблице

Запрос на обновление существующих данных по результатам подзапроса во фразе WHERE – Увеличить на 20 тысяч оклад всем артистам, сыгравшим в спектаклях с номерами 1 и 5 рисунок 36).

UPDATE Актер

SET Оклад = Оклад + 20000

WHERE [Табельный номер] IN

(SELECT Актер

FROM [Занятость в спектакле]

WHERE Спектакль = 1 OR Спектакль = 5)

GO

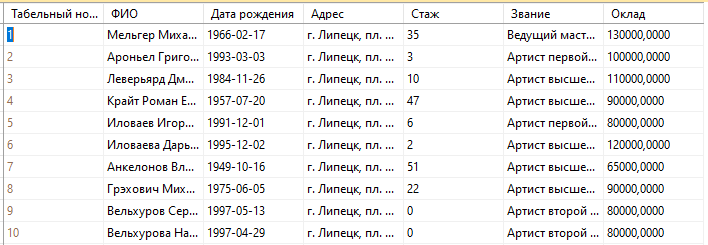


Рисунок 36 – Запрос на обновление существующих данных по результатам подзапроса во фразе WHERE

Запрос на удаление существующих данных – Удалить все данные из таблицы занятость в спектакле(рисунок 37).

DELETE FROM [Занятость в спектакле]

GO



Рисунок 37 – Запрос на удаление существующих данных

Запрос на удаление существующих данных из одной таблицы на основе связанных с ней таблиц – Удалить всю информацию об спектаклях в которых играют актеры с фамилиями начинающихся на буквы М, Г и И(рисунок 38).

DELETE FROM Спектакль

FROM Спектакль JOIN [Занятость в спектакле] ON Спектакль.[Код спектакля] = [Занятость в спектакле].Спектакль

JOIN Актер ON Актер.[Табельный номер] = [Занятость в спектакле].Актер

WHERE ФИО LIKE '[МГИ]%'

GO

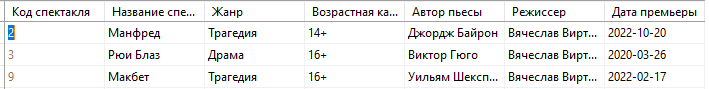


Рисунок 38 – Запрос на удаление существующих данных из одной таблицы на основе связанных с ней таблиц

Запрос на удаление существующих данных по результатам подзапроса во фразе WHERE – Удалить всю занятость в спектакле актеров высшей категории (рисунок 39).

DELETE FROM [Занятость в спектакле]

WHERE Актер IN

(SELECT [Табельный номер]

FROM Актер

WHERE Звание = 'Артист высшей категории')

GO

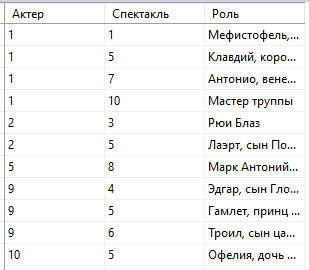


Рисунок 39 – Запрос на удаление существующих данных по результатам подзапроса во фразе WHERE

Запрос на удаление существующих данных из одной таблицы на основе связанных с ней таблиц – Удалить артистов которые играли в спектаклях жанра трагедия(рисунок 40).

DELETE FROM Актер

WHERE [Табельный номер] IN

(SELECT Актер

FROM [Занятость в спектакле]

WHERE Спектакль IN

(SELECT [Код спектакля]

FROM Спектакль

WHERE Жанр = 'Трагедия'))

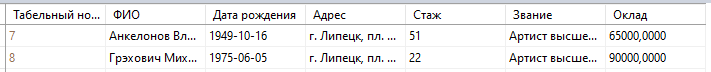


Рисунок 40 – Запрос на удаление существующих данных из одной таблицы на основе связанных с ней таблиц

5. Многотабличные запросы с использованием подзапросов:

3 запроса с использованием операций сравнения:

1. Вывести ФИО и стаж актеров с окладом больше чем средний оклад(рисунок 41).

SELECT ФИО, Стаж

FROM Актер

WHERE Оклад >

(SELECT AVG(Оклад) AS [Средний оклад]

FROM Актер)

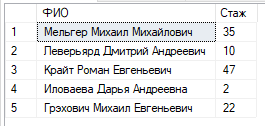


Рисунок 41 – Первый запрос с использованием операций сравнения

2. Вывести название спектакля, в котором сыграли все актеры(рисунок 42).

SELECT [Название спектакля]

FROM Спектакль

WHERE [Код спектакля] IN

(SELECT Спектакль

FROM [Занятость в спектакле]

GROUP BY Спектакль

HAVING COUNT(DISTINCT Актер) =

(SELECT COUNT([Табельный номер]) AS COUNT

FROM Актер))



Рисунок 42 – Второй запрос с использованием операций сравнения

3. Вывести всю информацию обо всех актерах, которые родились позже, чем Грэхович(рисунок 43).

SELECT \*

FROM Актер

WHERE [Дата рождения] >

(SELECT [Дата рождения]

FROM Актер

WHERE ФИО = 'Грэхович Михаил Евгеньевич')

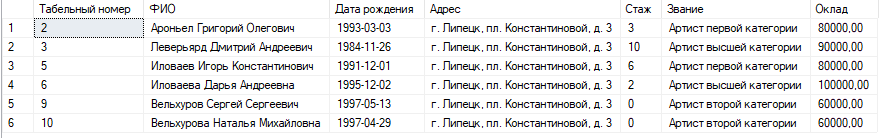


Рисунок 43 – Третий запрос с использованием операций сравнения

Запрос с использованием операции ANY – Вывести всю информацию о спектаклях, премьера которых была раньше хотя бы одного другого спектакля(рисунок 44).

SELECT \*

FROM Спектакль

WHERE [Дата премьеры] <ANY

(SELECT [Дата премьеры]

FROM Спектакль)

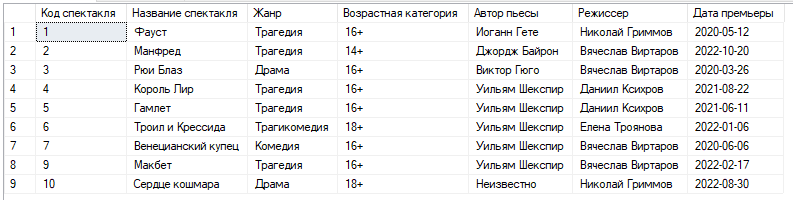


Рисунок 44 – Запрос с использованием операции ANY

Запрос с использованием операции ALL – Вывести фамилию актера который получает оклад меньше чем все остальные(рисунок 45).

SELECT ФИО

FROM Актер

WHERE Оклад <=ALL

(SELECT Оклад

FROM Актер)

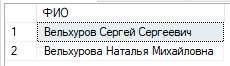


Рисунок 45 – Запрос с использованием операции ALL

Запрос с использованием операции IN – Вывести ФИО актеров задействованных в спектаклях меньше 3 раз(рисунок 46)

SELECT ФИО

FROM Актер

WHERE [Табельный номер] IN

(SELECT Актер

FROM (SELECT Актер, COUNT(Роль) AS [Количество ролей]

FROM [Занятость в спектакле]

GROUP BY Актер) AS [Количество ролей]

WHERE [Количество ролей] < 3)

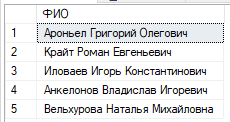


Рисунок 46 – Запрос с использованием операции IN

Запрос с использованием операции EXISTS – Вывести информацию о тех спектаклях, в которых никто не задействован(рисунок 47).

SELECT \*

FROM Спектакль

WHERE NOT EXISTS

(SELECT Спектакль

FROM [Занятость в спектакле]

WHERE [Занятость в спектакле].Спектакль = Спектакль.[Код спектакля])



Рисунок 47 – Запрос с использованием операции EXISTS

Запрос с использованием двух вложенных друг в друга подзапросов – Вывести ФИО актеров, игравших только в спектаклях у Ксихрова(рисунок 48).

SELECT ФИО

FROM Актер

WHERE [Табельный номер] IN

(SELECT Актер

FROM [Занятость в спектакле]

WHERE Спектакль IN

(SELECT [Код спектакля]

FROM Спектакль

WHERE Режиссер = 'Даниил Ксихров')

AND

[Табельный номер] NOT IN

(SELECT Актер

FROM [Занятость в спектакле]

WHERE Спектакль IN

(SELECT [Код спектакля]

FROM Спектакль

WHERE Режиссер != 'Даниил Ксихров')))



Рисунок 48 – Запрос с использованием двух вложенных друг в друга подзапросов

Запрос с использованием трех вложенных друг в друга подзапросов – Вывести количество жанров спектаклей, которых проходят наиболее рано или поздно по времени(рисунок 49).

SELECT COUNT(DISTINCT Жанр) AS Количество

FROM Спектакль

WHERE [Код спектакля] IN

(SELECT Спектакль

FROM(SELECT \*, DATEPART(HOUR, [Дата и время начала]) AS [Время начала]

FROM [Расписание спектаклей]) AS [Расписание спектаклей]

WHERE [Время начала] IN

(SELECT MAX([Время начала]) AS Максимум

FROM(SELECT DATEPART(HOUR, [Дата и время начала]) AS [Время начала]

FROM [Расписание спектаклей]) AS [Расписание спектаклей])

OR [Время начала] IN

(SELECT MIN([Время начала]) AS Минимум

FROM(SELECT DATEPART(HOUR, [Дата и время начала]) AS [Время начала]

FROM [Расписание спектаклей]) AS [Расписание спектаклей]))



Рисунок 49 – Запрос с использованием трех вложенных друг в друга подзапросов

6. Запросы на изменение структуры существующей таблицы:

Запрос на добавление нового столбца к таблице – Добавить артистам столбец номер телефона(рисунок 50).

ALTER TABLE Актер ADD

[Номер телефона] VARCHAR(20)

GO

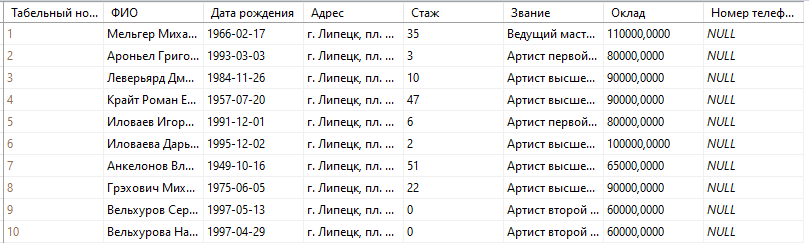


Рисунок 50 – Запрос на добавление нового столбца к таблице

Запрос на добавление нового ограничения к таблице – Добавить проверку на год рождения, он должен быть меньше чем 2000.

ALTER TABLE Актер

ADD CONSTRAINT Проверка\_на\_возраст CHECK (DATEPART(YEAR,[Дата рождения]) < 2000)

GO

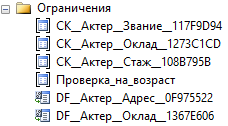


Рисунок 51 – Запрос на добавление нового ограничения к таблице

Запрос на удаление нового столбца к таблице – Удалить добавленный столбец с номером телефона.

ALTER TABLE Актер

DROP COLUMN [Номер телефона]

GO

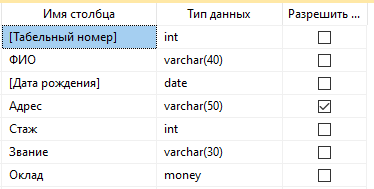


Рисунок 52 – Запрос на удаление нового столбца к таблице

Запрос на удаление нового ограничения к таблице – Удалить добавленное ограничение с проверкой на возраст.

ALTER TABLE Актер

DROP CONSTRAINT Проверка\_на\_возраст

GO

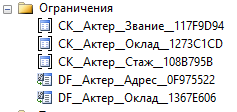


Рисунок 53 – Запрос на удаление нового ограничения к таблице

Запрос на изменение типа данных в таблице – Изменить тип данных даты рождения актера с date на varchar(рисунок 52).

ALTER TABLE Актер

ALTER COLUMN [Дата рождения] VARCHAR(15)

GO

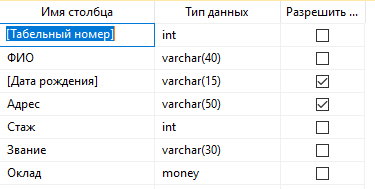


Рисунок 54 – Запрос на изменение типа данных в таблице

Запрос на добавление нового первичного ключа к таблице с сохранением целостности таблицы – В спектакле сделать первичным ключом столбец название спектакля.

ALTER TABLE [Расписание спектаклей] ADD

[Название спектакля] VARCHAR(40)

GO

UPDATE [Расписание спектаклей]

SET [Расписание спектаклей].[Название спектакля] = Спектакль.[Название спектакля]

FROM [Расписание спектаклей] JOIN Спектакль on Спектакль.[Код спектакля] = [Расписание спектаклей].Спектакль

GO

ALTER TABLE [Занятость в спектакле] ADD

[Название спектакля] VARCHAR(40)

GO

UPDATE [Занятость в спектакле]

SET [Занятость в спектакле].[Название спектакля] = Спектакль.[Название спектакля]

FROM [Занятость в спектакле] JOIN Спектакль on Спектакль.[Код спектакля] = [Занятость в спектакле].Спектакль

GO

ALTER TABLE [Расписание спектаклей]

DROP CONSTRAINT FK\_Расписание\_спектаклей\_Спектакль

ALTER TABLE [Занятость в спектакле]

DROP CONSTRAINT FK\_Занятость\_в\_спектакле\_Спектакль

ALTER TABLE Спектакль

DROP CONSTRAINT PK\_Спектакль

ALTER TABLE Спектакль

ADD CONSTRAINT PK\_Спектакль primary key

([Название спектакля])

GO

ALTER TABLE [Расписание спектаклей]

ADD CONSTRAINT FK\_Расписание\_спектаклей\_Спектакль foreign key

([Название спектакля]) references Спектакль([Название спектакля])on delete cascade

GO

ALTER TABLE [Занятость в спектакле]

ADD CONSTRAINT FK\_Занятость\_в\_спектакле\_Спектакль foreign key

([Название спектакля]) references Спектакль([Название спектакля])on delete cascade

GO

ALTER TABLE Спектакль

ALTER COLUMN [Код спектакля] int not null

GO

ALTER TABLE Спектакль

ADD CONSTRAINT unique\_Спектакль unique([Код спектакля])

GO

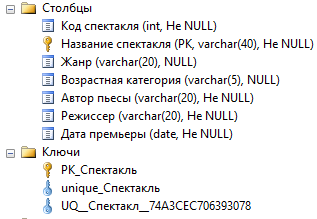


Рисунок 55 – Запрос на добавление нового первичного ключа к таблице с сохранением целостности таблицы

# **Заключение**

Была разработана и создана база данных, представляющая из себя базу данных для справочной системы театра. К данной базе были разработаны различные запросы, которые моделируют ту или иную ситуацию, когда реальный человек может обратиться к этой справочной системе по тому или иному вопросу.

Данная база данных работает исправно и выполняет поставленные задачи, но нынешняя версия может быть улучшена доработками в различных областях ее функционала. Например в уже существующие таблицы можно добавить новые поля, которые позволят более точно отразить различные ситуации и давать более точные сведенья. Или же можно создать новые таблицы, с новой информацией о той или иной области сферы постановки спектаклей, для создания большего, более полного и качественного объема информации.

В процессе работы над данной базой данных были получены и усовершенствованы навыки: анализ и построение логической модели предметной области, создание на основе анализа базы данных, создание на основе анализа ограничений для таблиц данной базы данных, заполнение базы данных, разработка и написание для базы данных запросов реляционной алгебры, разработка и написание запросов для одной таблицы, входящую в базу данных, разработка и написание запросов для нескольких таблиц, входящих в базу данных, реализация работы с несколькими таблицами методом их соединения или с помощью подзапросов, разработка и написание запросов на модификацию данных в таблицах, разработка и написание запросов на изменение структуры таблицы.

# **Библиографический список**

1. Базы данных: методические указания к курсовой работе/ Рязан. гос. радиотехн. ун-т; сост.: Н.Н. Гринченко, Н.И. Хизриева, С.Н. Баранова. – Рязань, 2020. – 23 с.
2. Волкова Т.В. Разработка систем распределенной обработки данных [Электронный ресурс]: учебно-методическое пособие/ Волкова Т.В., Насейкина Л.Ф.- Электрон. текстовые данные. - Оренбург: Оренбургский государственный университет, ЭБС АСВ, 2012. - 330 c.
3. Громов А.Ю. Современные технологии разработки интегрированных информационных систем: учеб. пособие / А.Ю. Громов, Н.Н. Гринченко, Н.В. Шемонаев; Рязан. гос. радиотехн. ун-т. - Рязань, 2015. - 48 с.
4. Введение в программные системы и их разработку [Электронный ресурс]/ С.В. Назаров [и др.]. - Электрон. текстовые данные.- М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. - 649 c
5. Гринченко Н.Н. Инструментальные средства поддержки баз данных: учеб. пособие / Н.Н. Гринченко, А.Ю. Громов; Рязан. гос. радиотехн. ун-т. - Рязань, 2015. - 48 с.