**ОТЧЁТ ПО ПРАКТИКЕ.**

**Содержание**

[**Введение**](#ВВЕДЕНИЕ)**(7-8)**

[**1.Предметная область**](#Предметнаяобласть)**(9-10)**

[**1.1 Описание предметной область**](#Описание_предметной_области)**(9)**

[**1.2 Анализ предметной области**](#Анализ_предметной_области)**(9-10)**

[**2.Модель 2НФ**](#_2.Модель_2_НФ)**(11)**

[**3.Модель 3НФ**](#Модель_3НФ)**(12-13)**

[**4.MVS, GitHub**](#MVS_Git_Github)**(14-16)**

[**5.Запросы SQL**](#_5.SQL)**(17-21)**

[**6.Экранные формы**](#ЭКРАННЫЕ_ФОРМЫ)**(21-40)**

[**7.Вывод(41)**](#вывод)

[**8.Список литературы(42)**](#СПИСОК_ЛИТЕРАТУРЫ)

[**9.Приложения(43)**](#вывод)

**ВВЕДЕНИЕ**

Персональный компьютер, как известно, является универсальным устройством для обработки информации. Персональные компьютеры могут выполнять любые действия по обработке информации. Для этого необходимо составить для компьютера на понятном ему языке точную и подробную последовательность инструкций – программу, как надо обрабатывать информацию. Соответственно, требуется все более разнообразное программное обеспечение для решения задач в новых областях применения ПК. Непрерывное повышение мощности персональных компьютеров, периферийных устройств, а также развитие средств связи дает разработчикам программного обеспечения все больше возможностей для максимально полного удовлетворения запросов конечных потребителей. Это и ставший стандартом графический интерфейс для любого ПО, и внедренные возможности для отправки документов и данных с помощью интернет, непосредственно из прикладной программы, и возможность использования компьютера, как хранилища информации благодаря появлению новых видов накопителей большой емкости и малым временем доступа к данным, а также многие другие возможности и сервисные функции. Количество программ, установленных на современном компьютере, исчисляется сотнями и даже тысячами. Именно они обеспечивают комфортную работу пользователя. Вся совокупность программ входит в так называемое программное обеспечение компьютера. Программой для ЭВМ является представленная в объективной форме совокупность данных и команд, предназначенных для функционирования ЭВМ и других компьютерных устройств в целях получения определённого результата, включая подготовительные материалы, полученные в ходе разработки программы для ЭВМ, и порождаемые ею аудиовизуальные отображения. Поэтому, во время прохождения практики была поставлена цель.

Целью прохождения учебной практики является:

1. Конструирование базы данных в MS SQL по заданной предметной области.
2. Создание программы, связанной с базой данных.
3. Освоение запросов с целью вывода, удаления, редактирования и добавления данных в базу данных.

Для достижения данной цели потребовалось решить следующие задачи:

* Провести анализ заданной предметной области;
* Провести анализ создания программы преподавателем;
* На основе анализа создать собственную программу;
* Ознакомиться с различной дополнительной литературой;

МИНИМАЛЬНЫЕ СИСТЕМНЫЕ ТРЕБОВАНИЯ:

Операционная система: Windows 7 64 Bit, Windows 8 64 Bit, Windows 7 64 Bit Service Pack 1, Windows Vista 64 Bit Service Pack Видеокарта: NVIDIA 9800 GT 1GB / AMD HD 4870 1GB (DX 10, 10.1, 11)

Процессор: Intel Core 2 Quad CPU Q6600 @ 2.40GHz (4 CPUs) / AMD Phenom 9850 Quad-Core Processor (4 CPUs) @ 2.5GHz Оперативная память

(ОЗУ): 4GB

Привод: DVD

РЕКОМЕНДОВАННЫЕ СИСТЕМНЫЕ ТРЕБОВАНИЯ:

Операционная система: Windows 8.1 64 Bit, Windows 8 64 Bit, Windows 7 64 Bit Service Pack 1

Видеокарта: NVIDIA GTX 660 2GB / AMD HD7870 2GB

Процессор: Intel Core i5 3470 @ 3.2GHZ (4 CPUs) / AMD X8 FX-8350 @ 4GHZ (8 CPUs) Оперативная память (ОЗУ): 8GB

Привод: DVD

**1.Предметная область.**

**1.1 Описание предметной области:**

Вы являетесь коммерческим директором театра и в ваши обязанности входит вся организационно-финансовая работа, связанная с привлечением актёров и заключением контрактов. Вы поставили дело следующим образом: каждый год театр осуществляет постановку различных спектаклей. Каждый спектакль имеет определённый бюджет. Для участия в конкретных постановках в определённых ролях вы привлекаете актёров. С каждым из актёров вы заключаете контракт на определённую сумму. Каждый из актёров имеет некоторый стаж работы и определённое звание.

**1.2 Анализ предметной области.**

**Театр-** (греч. Theatron) –(место для зрелищ) – род искусства, специфическим средством выражения которого является сценическое действие, возникающее в процессе игры актера перед публикой. Возник в Древней Греции. Источники театра – в древних охотничьих и сельскохозяйственных игрищах, массовых народных обрядах. **1)** род искусства, особенностью которого является художественное отражение явлений жизни посредством драматического действия, возникающего в процессе игры актеров перед зрителями; в ходе исторического развития определились три основные вида театра, отличающиеся специфическими признаками и средствами художественной выразительности – драматический, оперный и балетный; **2)** здание, где происходят театральные представления; **3)** представление, спектакль.

**Актёр** — исполнитель ролей в спектаклях и кино.

**Спектакль** — произведение сценического искусства.

**Жанр** — общее понятие, отражающее наиболее существенные свойства и связи явлений мира искусства, совокупность формальных и содержательных особенностей произведения.

**Стаж** — срок работы , продолжительность деятельности в какой-либо сфере, на какой-либо должности.

Выделим следующие базовые сущности нашей области:

1. Актёры -

(Со следующими атрибутами: [id актёра], [ФИО], [Звание], [id\_стажа ]);

1. Жанр-

(Со следующими атрибутами: [id жанра], [Название]);

1. Занятость актёров в спектакле-

(Со следующими атрибутами: [id актёра], [id спектакля], [Роль], [Стоимость годового контракта]);

1. Расписание спектакле-

(Со следующими атрибутами: [id театра], [id спектакля], [Стоимость билета], [Стоимость билета на премьеру], [Дата начала спектакля], [Дата окончания спектакля]);

1. Спектакли-

(Со следующими атрибутами: [id спектакля], [Название спектакля], [Год постановки], [Бюджет], [id\_жанра]);

1. Стаж-

(Со следующими атрибутами: [id стажа], [Стаж], [Дополнительная плата за стаж]);

1. Театр-

(Со следующими атрибутами: [id театра], [Название театра], [Город], [Наценка на билет], [Репертуар], [Дата окончания спектакля]);

В соответствии с предметной областью система строится с учетом следующих особенностей:

1. Каждый актёр имеет свой уникальный номер;
2. Актёры могут играть в разных спектаклях, в разных ролях, в разных театрах;
3. Каждый спектакль имеет свой уникальный номер;
4. Спектакль могут играть в разных театрах;
5. Каждое актерское звание имеет уникальный номер и свое название;
6. Каждый спектакль имеет жанр, а каждый жанр в свою очередь имеет уникальный номер и название;
7. За определённый стаж актёр получает вознаграждение;
8. У каждого театра есть определённая наценка на билет.

# **2.Модель 2 НФ (вторая нормальная форма)**

2 НФ (Вторая нормальная форма) -(англ. Second normal form; сокращённо 2NF)-одна из возможных нормальных форм отношения в реляционной базе данных. Переменная отношения находится во второй нормальной форме тогда и только тогда, когда она находится в первой нормальной форме и каждый не ключевой атрибут неприводимо зависит от (каждого) её потенциального ключа [2].

*Неприводимость* означает, что в составе потенциального ключа отсутствует меньшее подмножество атрибутов, от которого можно также вывести данную функциональную зависимость. Для неприводимой функциональной зависимости часто используется эквивалентное понятие «полная функциональная зависимость» [2].

Если потенциальный ключ является простым, то есть состоит из единственного атрибута, то любая функциональная зависимость от него является неприводимой (полной). Если потенциальный ключ является составным, то, согласно определению второй нормальной формы, в отношении не должно быть неключевых атрибутов, зависящих от *части* составного потенциального ключа [2].

Вторая нормальная форма по определению запрещает наличие неключевых атрибутов, которые вообще не зависят от потенциального ключа. Таким образом, вторая нормальная форма в том числе запрещает создавать отношения как несвязанные (хаотические, случайные) наборы атрибутов [2].

**3.Модель 3 НФ (третья нормальная форма)**

3 НФ (Третья нормальная форма**)** (англ. *Third normal form*; сокращённо 3NF) — одна из возможных нормальных форм отношения в реляционной базе данных.

Переменная отношения *R* находится в 3NF тогда и только тогда, когда выполняются следующие условия:

* *R* находится во второй нормальной форме.
* ни один не ключевой атрибут *R* не находится в транзитивной функциональной зависимости от потенциального ключа *R*.

Пояснения к определению:

*Не ключевой атрибут* отношения *R* — это атрибут, который не принадлежит ни одному из потенциальных ключей *R*.

*Функциональная зависимость* множества атрибутов **Z** от множества атрибутов **X** (записывается **X** → **Z**, произносится «икс определяет зет») является *транзитивной*, если существует такое множество атрибутов **Y**, что **X** → **Y** и **Y** → **Z**. При этом ни одно из множеств **X**, **Y** и **Z** не является подмножеством другого, то есть функциональные зависимости **X** → **Z**, **X** → **Y** и **Y** → **Z** не являются *тривиальными*, а также отсутствует функциональная зависимость **Y** → **X**.

Определение 3NF, эквивалентное определению Кодда, но по-другому сформулированное, дал Карло Заниоло в 1982 году. Согласно ему, переменная отношения находится в 3NF тогда и только тогда, когда для каждой из её функциональных зависимостей **X** → **A** выполняется хотя бы одно из следующих условий:

* **Х** содержит **А** (то есть **X** → **A** — тривиальная функциональная зависимость)
* **Х** — суперключ
* **А** — ключевой атрибут (то есть **А** входит в состав потенциального ключа).

Определение Заниоло чётко определяет разницу между 3NF и более строгой нормальной формой Бойса-Кодда (НФБК): НФБК исключает третье условие («**А** — ключевой атрибут»).

Запоминающееся и, по традиции, наглядное резюме определения 3NF Кодда было дано Биллом Кентом: каждый не ключевой атрибут «должен предоставлять информацию о ключе, полном ключе и ни о чём, кроме ключа»[3].

Условие зависимости от «полного ключа» неключевых атрибутов обеспечивает то, что отношение находится во второй нормальной форме; а условие зависимости их от «ничего, кроме ключа» — то, что они находятся в третьей нормальной форме. [3]

Крис Дейт говорит о резюме Кента как об «интуитивно привлекательной характеристике» 3NF, и замечает, что с небольшим изменением она может служить и как определение более строгой нормальной формы Бойса-Кодда: «каждый атрибут должен предоставлять информацию о ключе, полном ключе и ни о чём, кроме ключа». Вариант определения 3NF Кента является менее строгим, чем вариант нормальной формы Бойса-Кодда в формулировке Дейта, поскольку первая утверждает только то, что неключевые атрибуты зависят от ключей. Первичные атрибуты (которые являются ключами или их частями) вовсе не должны быть функционально зависимыми; каждый из них предоставляет информацию о ключе предоставлением самого ключа или его части. Здесь следует отметить, что это правило справедливо только для неключевых атрибутов, так как применение его ко всем атрибутам будет полностью запрещать все сложные альтернативные ключи, поскольку каждый элемент такого ключа будет нарушать условие «полного ключа». [3]

# **4.MVS, Git, Github.**

*Microsoft Visual Studio* - линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework и Silverlight.

Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных. Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода (как, например, Subversion и Visual SourceSafe), добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода на предметно-ориентированных языках программирования) или инструментов для прочих аспектов процесса разработки программного обеспечения (например, клиент Team Explorer для работы с Team Foundation Server).

**Git**  -(произносится «гит»)-система управления версиями. Проект был создан Линусом Торвальдсом для управления разработкой ядра Linux, первая версия выпущена 7 апреля 2005 года. На сегодняшний день его поддерживает Джунио Хамано.

Среди проектов, использующих Git — ядро Linux, Swift, Android, Drupal, Cairo, GNU Core Utilities, Mesa, Wine, Chromium, Compiz Fusion, FlightGear, jQuery, PHP, NASM, MediaWiki, DokuWiki, [Qt](https://ru.wikipedia.org/wiki/Qt), ряд дистрибутивов Linux.

Программа является свободной и выпущена под лицензией GNU GPL версии 2. По умолчанию используется TCP порт 9418.

Система спроектирована как набор программ, специально разработанных с учётом их использования в сценариях. Это позволяет удобно создавать специализированные системы контроля версий на базе Git или пользовательские интерфейсы. Например, Cogito является именно таким примером оболочки к репозиториям Git, а StGit использует Git для управления коллекцией исправлений (патчей).

Git поддерживает быстрое разделение и слияние версий, включает инструменты для визуализации и навигации по нелинейной истории разработки. Как и Darcs, BitKeeper, Mercurial, Bazaar и Monotone[en], Git предоставляет каждому разработчику локальную копию всей истории разработки, изменения копируются из одного репозитория в другой.

Удалённый доступ к репозиториям Git обеспечивается git-[демоном](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D0%BC%D0%BE%D0%BD_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0)), [SSH](https://ru.wikipedia.org/wiki/SSH)- или [HTTP](https://ru.wikipedia.org/wiki/HTTP)-сервером. TCP-сервис git-daemon входит в дистрибутив Git и является наряду с SSH наиболее распространённым и надёжным методом доступа. Метод доступа по HTTP, несмотря на ряд ограничений, очень популярен в контролируемых сетях, потому что позволяет использовать существующие конфигурации сетевых фильтров.

**Github** - Веб-сервис основан на системе контроля версий Git и разработан на Ruby on Rails и Erlang компанией GitHub, Inc (ранее Logical Awesome). Сервис бесплатен для проектов с открытым исходным кодом и (с 2019 года) небольших частных проектов, предоставляя им все возможности (включая SSL), а для крупных корпоративных проектов предлагаются различные платные тарифные планы.Слоган сервиса — «Social Coding» — на русский можно перевести как «Пишем код вместе». На футболках же печатают совсем другую фразу: «Fork you!» («Ответвись!»). С одной стороны, она созвучна с англоязычным ругательством и намекает на неформальную атмосферу. С другой, эти слова напоминают, что создавать новые форки с Git можно легко и безболезненно — традиционно, к созданию веток разработчики проектов с открытым исходным кодом относятся негативно — а также созвучна названию одной из возможностей GitHub — очереди форков. Создатели сайта называют GitHub «социальной сетью для разработчиков».Кроме размещения кода, участники могут общаться, комментировать правки друг друга, а также следить за новостями знакомых.С помощью широких возможностей Git программисты могут объединять свои репозитории — GitHub предлагает удобный интерфейс для этого и может отображать вклад каждого участника в виде дерева.

Для проектов есть личные страницы, небольшие [Вики](https://ru.wikipedia.org/wiki/%D0%92%D0%B8%D0%BA%D0%B8) и система отслеживания ошибок.

Прямо на сайте можно просмотреть файлы проектов с подсветкой синтаксиса для большинства языков программирования.

Ссылка на работу <https://github.com/Nikitabeepbeep>

# **5.SQL**

# 5.1 SQL Запросы

Запрос 1. Найти пары театров репертуар которых совпадает.

SELECT[id театра],[Название театра],[Репертуар]

FROM[dbo].[Театр]

WHERE [Репертуар] = 'Драма';

SELECT[id театра],[Название театра],[Репертуар]

FROM[dbo].[Театр]

WHERE [Репертуар] = 'Комедия';

SELECT[id театра],[Название театра],[Репертуар]

FROM[dbo].[Театр]

WHERE [Репертуар] = 'Опера';

SELECT[id театра],[Название театра],[Репертуар]

FROM[dbo].[Театр]

WHERE [Репертуар] = 'Фэнтези';

Данный запрос выводит пары театров у которых поле «Репертуар» из таблицы «Театр» имеет значение «Драма», «Комедия», «Опера», «Фэнтези».

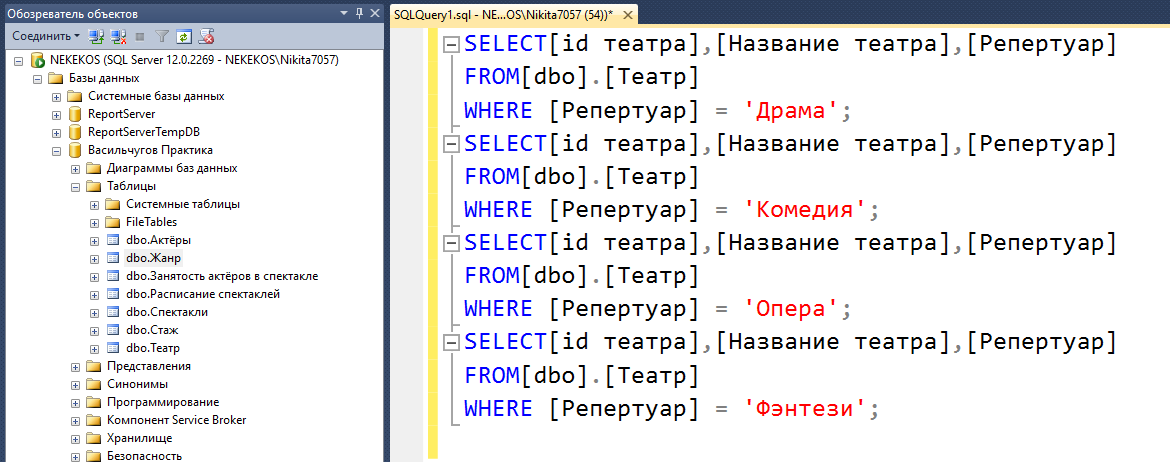


Рисунок 1. Так выглядит запрос в программе MSQL SMS.

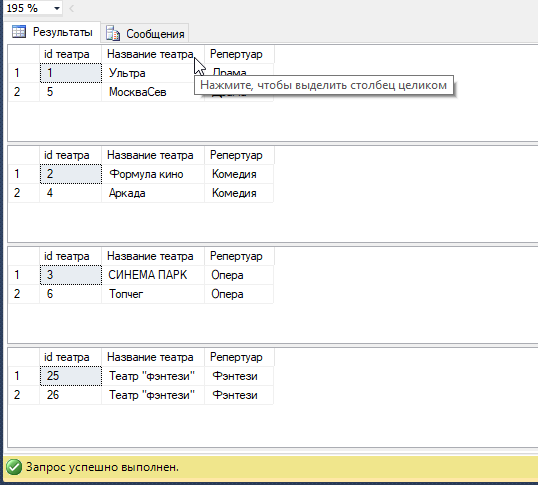


Рисунок 2. Так выглядит результат запроса.

Запрос 2. Найти всех актёров, которые исполнили заданную роль в разных спектаклях.

SELECT [Роль],[id актёра],[id спектакля]

FROM[dbo].[Занятость актёров в спектакле]

WHERE [Роль] = 'Клоун'

Данный запрос выводит всех актёров из таблицы «Занятость актёров в спектакле», которые исполнили определённую роль в разных театрах. Например, «Клоун».



Рисунок 3. Так выглядит сам запрос в программе MSQL SMS.

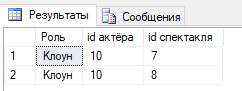


Рисунок 4. Так выглядит результат запроса.

Запрос 3. Для каждого театра посчитать среднюю стоимость билетов на премьеры спектаклей.

SELECT [Расписание спектаклей].[id театра],AVG ([Стоимость билета на премьеру]) AS [Стоиомсть]

FROM [Васильчугов Практика].[dbo].[Расписание спектаклей]

GROUP BY [Расписание спектаклей].[id театра]

Данный запрос вычисляет среднюю стоимость билетов на премьеры спектаклей из таблицы «Расписание спектаклей», в одном театре может играться не один спектакль, поэтому с помощью «GROUP BY» записи группируются по «id театра». Затем запрос выводит среднее значение для каждого театра (внутри данной группы).

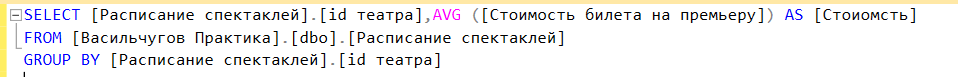


Рисунок 5. Так выглядит запрос в программе MSQL SMS.

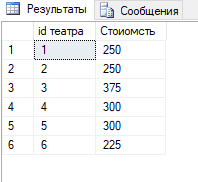


Рисунок 6. Так выглядит результат запроса.

Зарос 4. Какие театры не дают спектаклей на следующей неделе.

SELECT \*

FROM [Расписание спектаклей]

WHERE [Дата начала спектакля] < '2019-12-02'

AND [Дата окончания спектакля]< '2019-12-08'

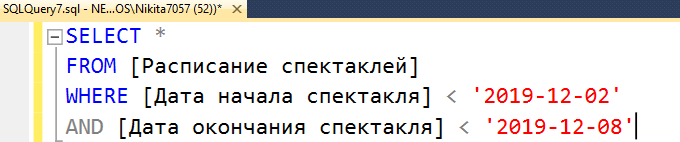
Данный запрос выводит все театры, которые не дают спектаклей на следующей неделе.

Рисунок 7. Так выглядит сам запрос.

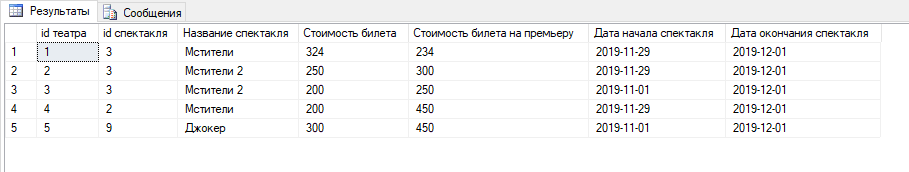


Рисунок 9. Так выглядит результат запроса.

Запрос 5. Какие актёры задействованы во всех спектаклях своих театров.

SELECT [id спектакля],[id актёра],[Роль],[Стоимость годового контракта]

FROM [Занятость актёров в спектакле]

WHERE [id актёра] = '1';

Данный запрос показывает актёра, который исполняют разные роли в разных спектаклях.

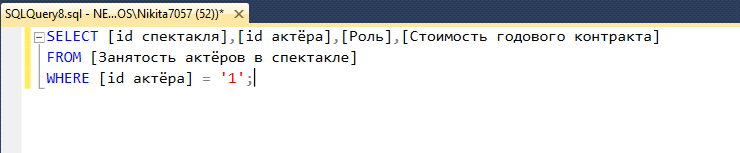


Рисунок 10. Так выглядит запрос в SQL.

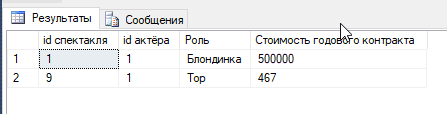


Рисунок 11. Так выглядит результат запроса.

Запрос 6. Найти театры, которые дают спектакли на следующей неделе.

SELECT [Дата начала спектакля],[Дата окончания спектакля],[id театра] FROM [dbo].[Расписание спектаклей]

WHERE [Дата начала спектакля] >= '2019-11-29'

AND [Дата окончания спектакля] <= '2019-12-31'

Данный запрос отображает все театры, которые дают спектакли на следующей неделе.

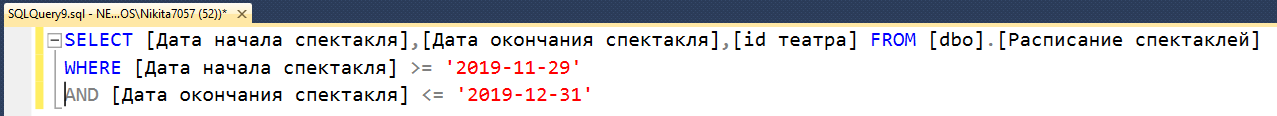


Рисунок 12. Так выглядит запрос в SQL.

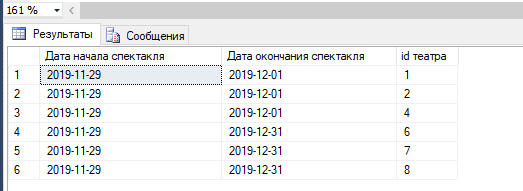


Рисунок 13. Так выглядит результат запроса.

**6.Экранные формы.**

Экранные формы программы по теме: «Занятость актёров театра».

Включив программу пользователь видит меню.

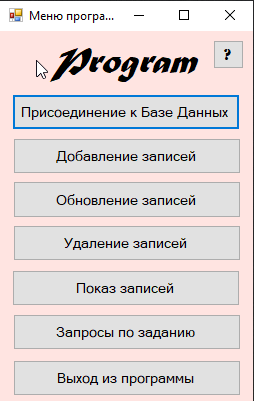


Рисунок 1. Меню программы.

Меню – это главное окно программы. С помощью меню можно осуществлять все представленные действия с базой данных.

В данном меню пользователь может нажимать на кнопки: «Подключение к Базе Данных», «Добавление записей», «Обновление записей», Удаление записей», «Показ записей», «Запросы по заданию», «Выход из программы», «?».

С помощью этих кнопок пользователь может подключиться к базе данных, добавлять новые записи в таблицы, обновлять записи в таблицах, удалять записи из таблиц, просматривать таблицы, выполнять некоторые запросы к БД, выходить из программы.

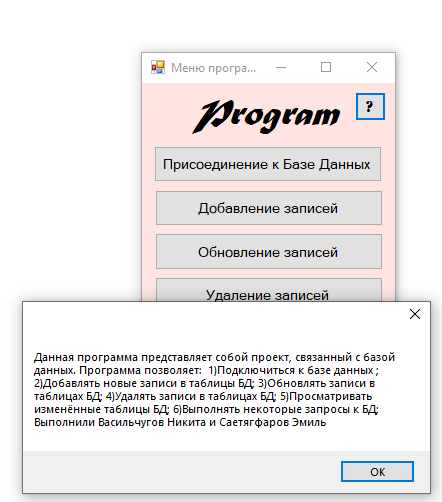
Нажав на кнопку «?» пользователь увидит данное сообщение:

Рисунок 2. «Что выводит программа, если нажать кнопку «?»»

Нажав на кнопку «Присоединение к Базе Данных» откроется окно, в котором пользователь должен заполнить данные для входа в базу данных. Чтобы подключиться к базе данных пользователь должен знать: Имя сервера (которое нужно вводить в поле для заполнения «Сервер»), Название базы данных (которое нужно вводить в поле для заполнения «База данных», Имя пользователя (это то имя, которое используется для входа в базу данных, вводить его нужно в поле для заполнения «Имя пользователя»), Пароль для входа в базу данных (вводится пароль в поле для заполнения «Пароль». Если пользователь не введёт ничего в строки для заполнения или где-нибудь забудет ввести данные, то он увидит следующие сообщение:(смотреть Рисунок 1. ,Рисунок 2.)

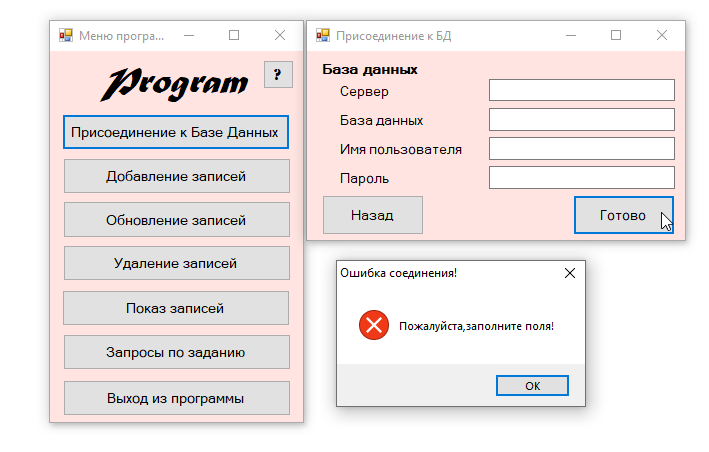


Рисунок 3. Ошибка подключения к БД.

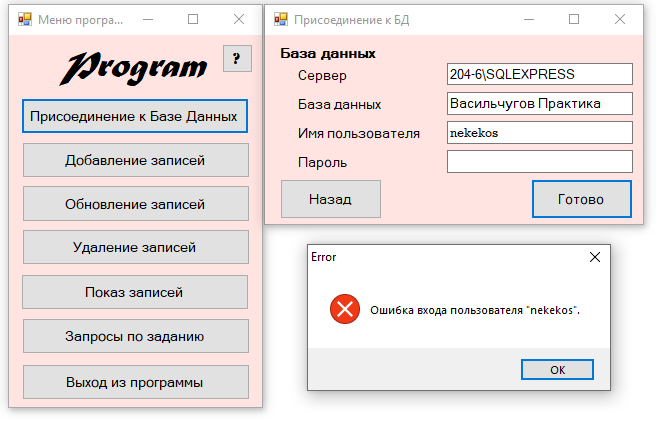


Рисунок 4. Ошибка входа пользователя (пользователь ввёл неверный пароль или забыл заполнить поле «Пароль»)

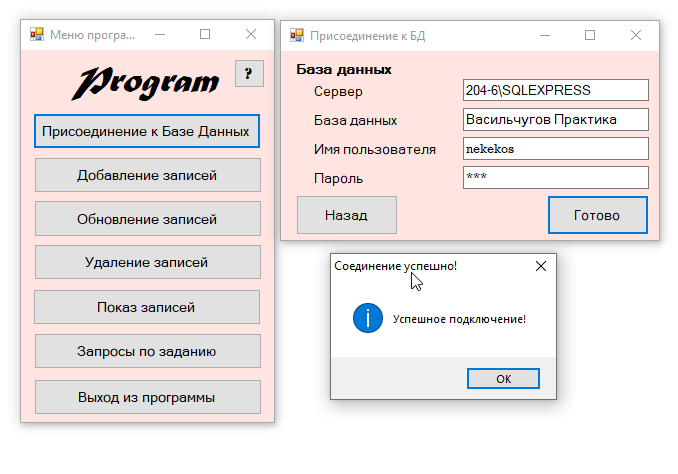


Рисунок 5. Успешное подключение к базе данных.

После успешного подключения пользователь возвращается в меню программы. Теперь пользователь может работать с базой данных.

Допустим, пользователь решил добавить записи в какую-нибудь таблицу, для этого ему следует нажать «Добавление записей».

Нажав на эту кнопку пользователь увидит следующее меню:

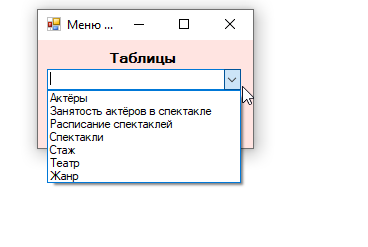


Рисунок 6. Меню добавления записей в таблицы.

С помощью данного меню пользователь сможет добавить новые строки в таблицы: «Актёры», «Занятость актёров в спектакле», «Расписание спектаклей», «Спектакли», «Стаж», «Театр», «Жанр».

Допустим, пользователь захотел добавить запись в таблицу «Актёры».

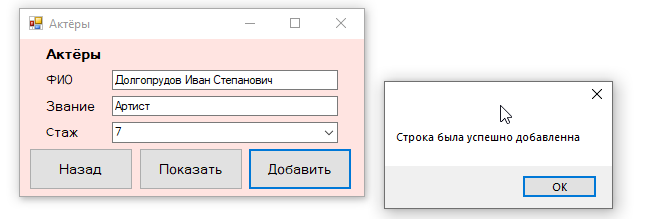


Рисунок 7. Добавление записей в таблицу «Актёры».

В данной таблице пользователь может «создать» нового актёра в базе данных, указать его ФИО, указать его звание, его стаж (стаж задаётся в таблице «Стаж», в таблице «Актёры» можно выбрать id заданного стажа).

Просмотр добавленных данных рассмотрим чуть позже.

Аналогично записи добавляются и для других таблиц.

Создадим новые записи в таблице «Жанр».

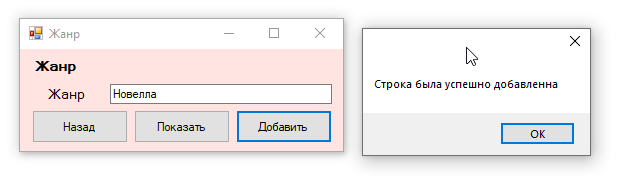


Рисунок 8. Добавление записей в таблицу «Жанр».

В данную таблицу пользователь может вписать название жанра.

Далее откроем таблицу «Спектакли». В ней пользователь может добавить год постановки спектакля, бюджет спектакля, название спектакля, и выбрать жанр из предложенных, а также свой ранее написанный, если, конечно, он добавил новый жанр раньше, чем зашёл в добавление записей в таблицу «Спектакли».

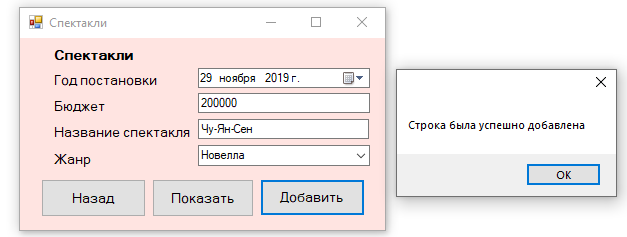


Рисунок 9. Добавление записей в таблицу «Спектакли».

В каждой форме добавления есть кнопка «Назад» и «Показать», которые соответственно могут вернуть пользователя к меню добавления или же сразу к форме «Показа записей».

Далее откроем форму добавления в таблицу «Театр».

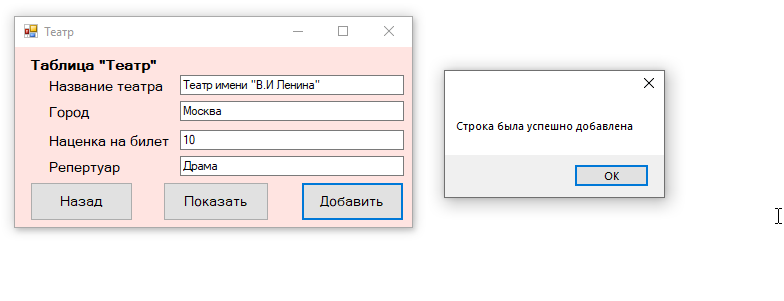


Рисунок 10. Добавление записей в таблицу «Театр».

В данной таблице пользователь может добавить название театра, город, наценку на билет, репертуар данного театра.

Далее откроем форму добавления в таблицу «Расписание спектаклей».

В ней пользователь может выбрать название театра, в котором будет происходить спектакль, выбрать название спектакля, задать стоимость билета, стоимость билета на премьеру, выбрать дату начала и дату окончания спектакля.

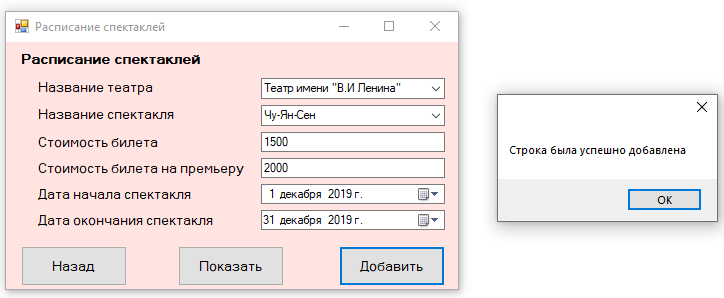


Рисунок 11. Добавление записей в таблицу «Расписание спектаклей».

Откроем добавление в таблицу «Занятость актёров в спектакле». В данной таблице можно выбрать название спектакля, ФИО актёра, которого нужно задействовать в спектакль, его роль и стоимость годового контракта.

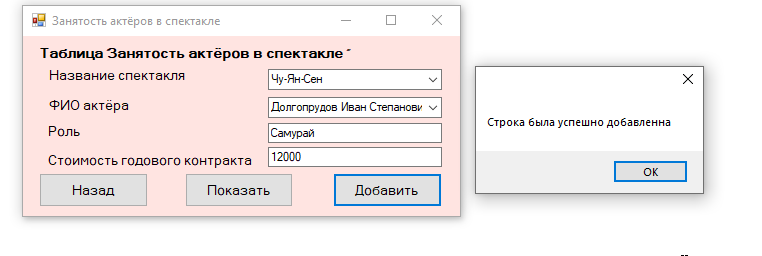


Рисунок 12. Добавление записей в таблицу «Занятость актёров в спектакле».

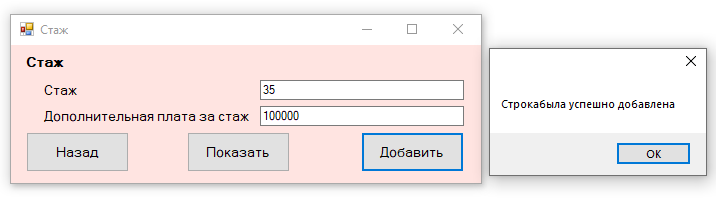
Ну и добавим новую запись в последнюю таблицу «Стаж». В данной таблице можно добавить стаж (сколько лет актёр работал) и дополнительную плату за этот стаж.

Рисунок 13. Добавление записей в таблицу «Стаж».

Теперь выберем кнопку «Показ записей» и посмотрим добавились ли наши новые данные.

Для просмотра каждой таблицы сделаны кнопки при нажатии на которые выведется нужная таблица.



Рисунок 14. Меню просмотра таблиц.

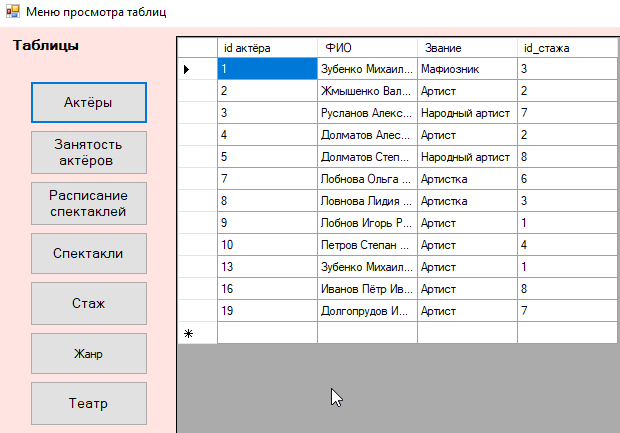
С помощью кнопки «Очистить», поле для отображения таблиц очищается, с помощью кнопки «Назад» возвращает пользователя в «Меню программы».

Рисунок 15. Просмотр таблицы «Актёры».

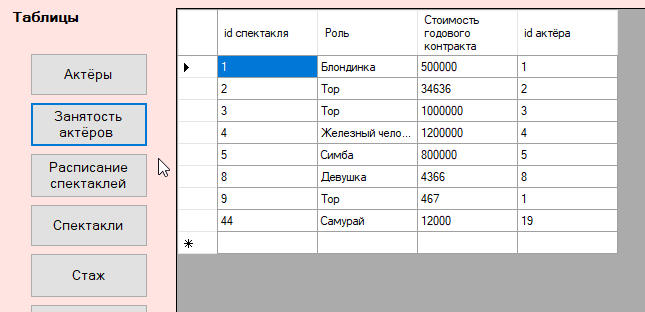


Рисунок 16. Просмотр таблицы «Занятость актёров театра».



Рисунок 17. Просмотр таблицы «Расписание спектаклей».

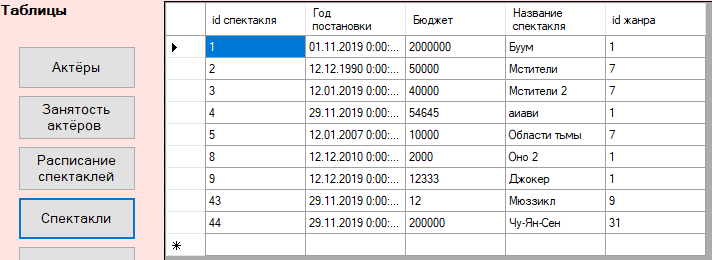


Рисунок 18. Просмотр таблицы «Спектакли».

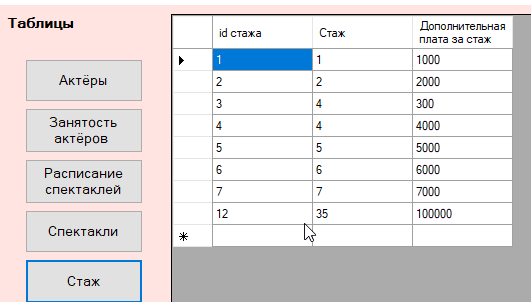


Рисунок 19. Просмотр таблицы «Стаж».

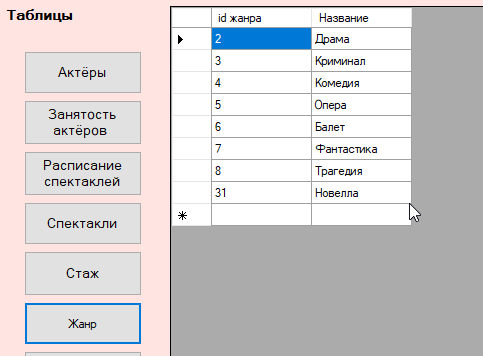


Рисунок 20. Просмотр таблицы «Жанр».

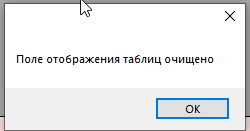


Рисунок 21. Нажатие на кнопку «Очистить таблицу».

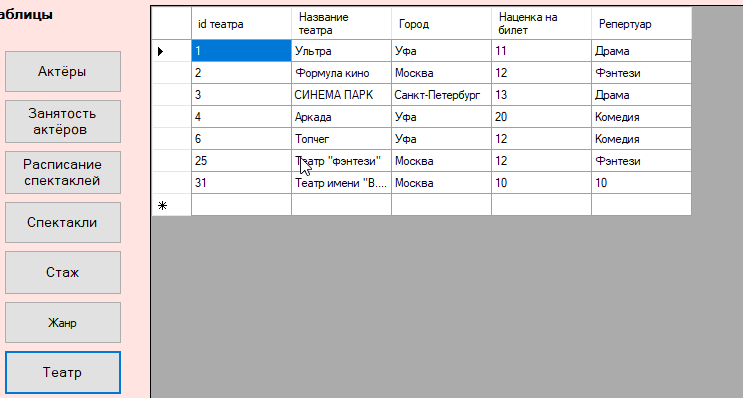


Рисунок 22. Просмотр таблицы «Театр».

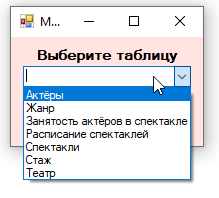


Рисунок 23. Меню обновления записей в таблицах.

С помощью этого меню пользователь может изменить записи в таблицах. Изменим все ранее добавленные записи, а потом выведем то, что получилось.

В формах обновления или изменения записей можно изменить все поля после выбора интересующего нас объекта.

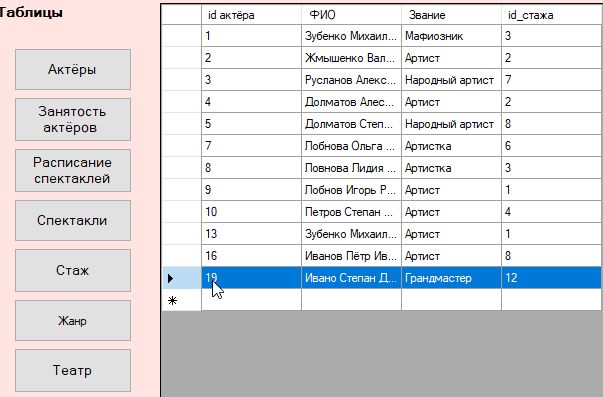


Рисунок24.Изменение записей в таблице «Актёры».

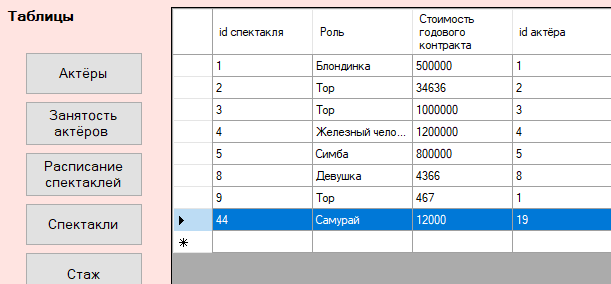


Рисунок 25. Изменение записей в таблице «Занятость актёров в спектакле».



Рисунок 26. Изменение записей в таблице «Расписание спектаклей».

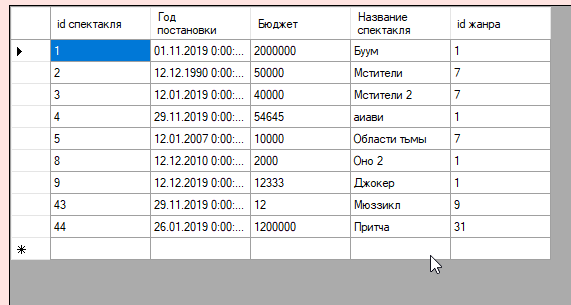


Рисунок 27. Изменение записей в таблице «Спектакли».



Рисунок 28. Изменение записей в таблице «Стаж».

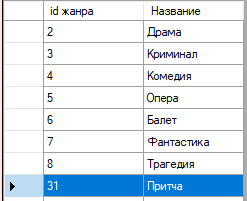


Рисунок 29. Изминение записей в таблице «Жанр».

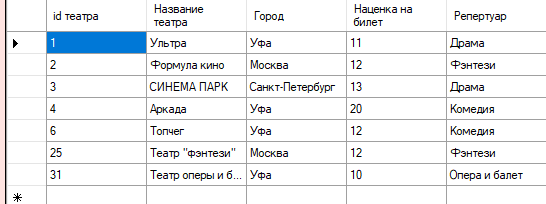


Рисунок 30. Изменение записей в таблице «Театр».

Возвращаемся с помощью кнопки «Назад» в «Меню программы», и с помощью кнопки «Удаление записей» откроем новую форму для удаления записей.

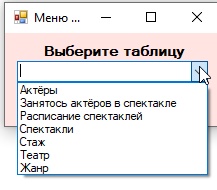


Рисунок 31. «Меню удаления записей».

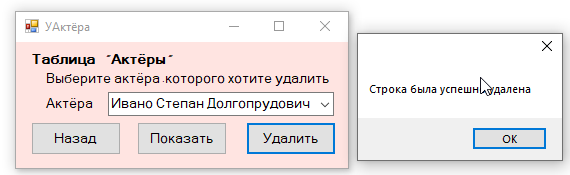


Рисунок 32. Удаление записей из таблицы «Актёры».

Открывая формы, пользователь может выбрать актёра для удаления из перечня возможных актёров для удаления, и, выбрав нужного актёра, может его удалить. В других формах аналогично выбирается: «Занятость актёра в спектакле»-«Актёр», «Расписание спектаклей»-«Название спектакля», «Спектакли»-«Название спектакля», «Стаж»-«Стаж», «Театр»-«Название театра», «Жанр»-«Название».

Возвращаемся в меню программы и с помощью показа записей убеждаемся, что все новые внесённые пользователем данные удалены. Далее снова возвращаемся в меню. Нажав на кнопку «Запросы по заданию» пользователь может увидеть список возможных запросов к БД, а при нажатии кнопки «ответ», вывести результат запроса, с помощью кнопки «Очистить» пользователь может очистить поле просмотра.

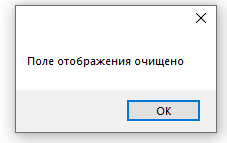


Рисунок 33. Кнопка «Очистить» в «Меню запросов».

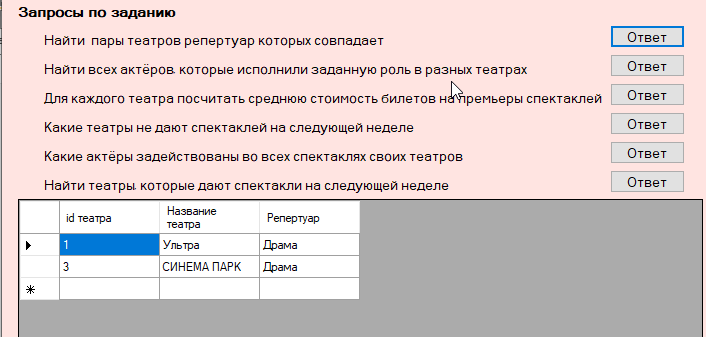


Рисунок 34. Результат запроса «Найти пары театров репертуар которых совпадает».

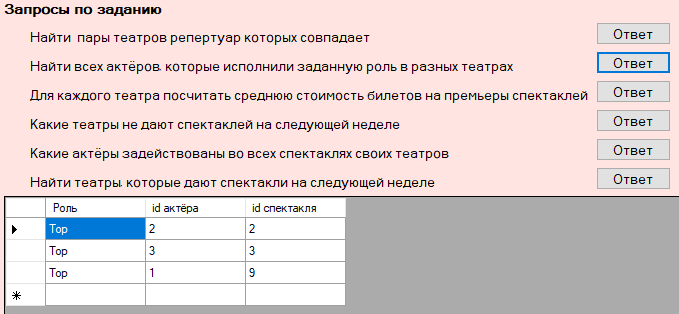


Рисунок 35. Результат запроса «Найти всех актёров, которые исполнили заданную роль в разных театрах».

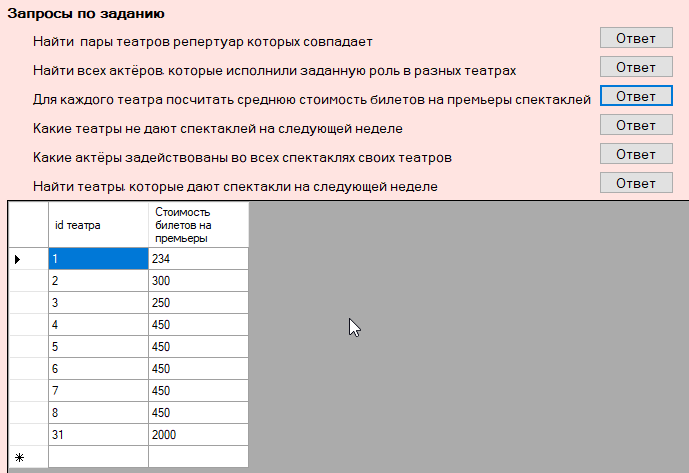


Рисунок 36. Результат запроса «Для каждого театра посчитать среднюю стоимость билетов на премьеры спектаклей».

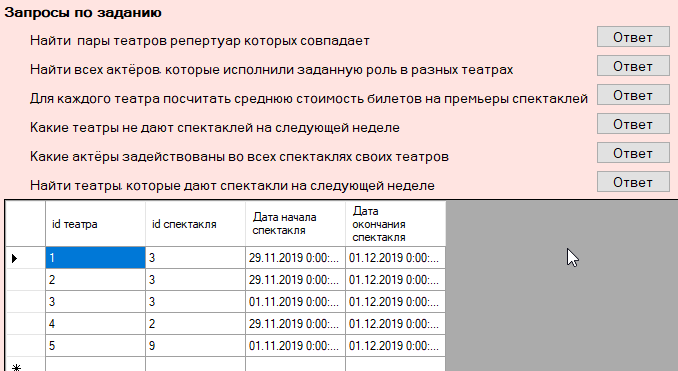


Рисунок 37. Результат запроса «Какие театры не дают спектаклей на следующей неделе».

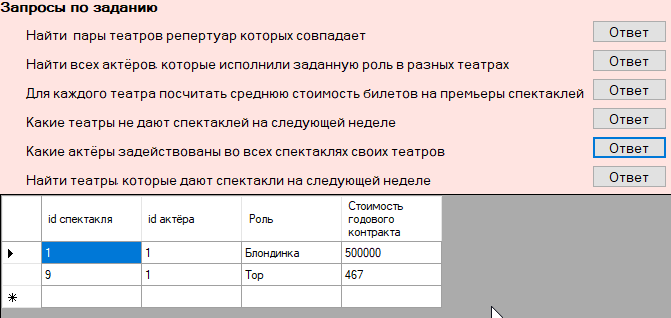


Рисунок 38. Результат запроса «Какие актёры задействованы во всех спектаклях своих театров».

Более подробно все запросы описаны ранее в разделе [5.1 SQL Запросы](#_5.1_SQL_Запросы).

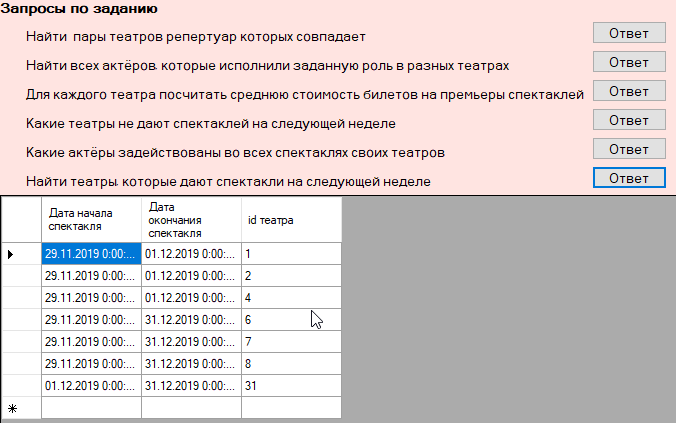
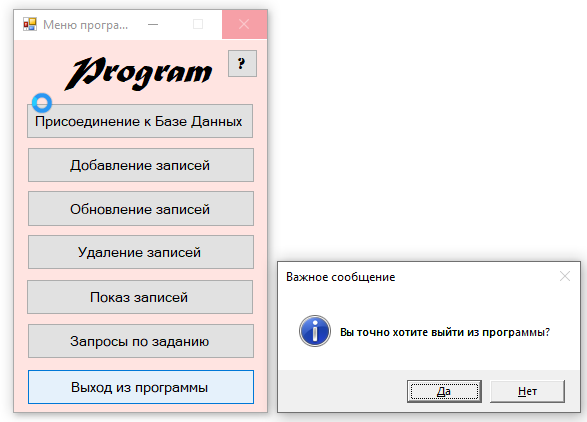


Рисунок 39. Результат запроса «Найти театры, которые дают спектакли на следующей неделе».

Далее вернёмся в «Меню программы» и нажмём на кнопку «Выход из программы».



Если нажать на «Да», то программа закроется, если нажать на «Нет», то пользователя перенесёт обратно в «Меню программы».

**7.Вывод:**   
В результате прохождения учебной практики значительно расширены профессиональные навыки, а именно изучено конструирование базы данных в MS SQL по заданной предметной области, изучено создание программы, связанной с базой данных, освоены запросы к базе данных с целью вывода, освоено удаление, редактирование, добавление записей в базу данных. Знания и навыки, полученные за период практики являются отличным стимулом и дальше изучать профессию. В период учебной практики удалось получить первый профессиональный опыт работы и сформировать общее представление о роде деятельности.

**8.Список литературы:**

Интернет-ресурсы:

<https://ru.wikipedia.org/wiki/%D0%92%D1%82%D0%BE%D1%80%D0%B0%D1%8F_%D0%BD%D0%BE%D1%80%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%84%D0%BE%D1%80%D0%BC%D0%B0>

<https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%B5%D1%82%D1%8C%D1%8F_%D0%BD%D0%BE%D1%80%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%84%D0%BE%D1%80%D0%BC%D0%B0>

<https://ru.wikipedia.org/wiki/Microsoft_Visual_Studio>

<https://ru.wikipedia.org/wiki/Git>

Книги:

Большой толковый словарь по культурологии, Кононенко Б.И 2003

«ГК РФ» Статья 1261. Программы для ЭВМ, глава 70 «Авторское право».