автономная некоммерческая образовательная организация высшего образования Центросоюза Российской Федерации

СИБИРСКИЙ УНИВЕРСИТЕТ ПОТРЕБИТЕЛЬСКОЙ КООПЕРАЦИИ

Кафедра информатики

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К курсовой работе по дисциплине: «Базы данных»

На тему: «Аренда Кинолент»

Студента очной формы обучения факультета экономики и управления 2 курса специальности 09.03.03 Прикладное программирование

ФИО

Группа Пиб- 11,

шифр шифр

Проверил:

Астапчук В.А. доцент, канд. техн. наук

Оценка после защиты:

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

Дата защиты:

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

Новосибирск 2022

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc137036600)

[1 Анализ предметной области 5](#_Toc137036601)

[2 Обоснование модели данных 7](#_Toc137036602)

[3 Обоснование выбора средств разработки 8](#_Toc137036603)

[4 Проектирование базы данных 13](#_Toc137036604)

[5 Описание функций управления данными 17](#_Toc137036605)

[6 Разработка приложения по работе с базой данных 20](#_Toc137036606)

[ЗАКЛЮЧЕНИЕ 21](#_Toc137036607)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 22](#_Toc137036608)

[ПРИЛОЖЕНИЕ А Руководство пользователя 23](#_Toc137036609)

[ПРИЛОЖЕНИЕ Б Код программы 32](#_Toc137036610)

# ВВЕДЕНИЕ

Объект исследования представленной работы – предприятие, которое занимается демонстрацией кинофильмов в кинотеатрах города.

Предмет исследования – учет сведений по аренде фильмов кинотеатрам города.

Цель представленной работы заключается в разработке информационной системы, которая будет реализовывать функции учета сведений по аренде фильмов кинотеатрам города.

В данный момент в предприятии отсутствует автоматизированная информационная система. В отсутствии автоматизированных средств данные о кинотеатрах обрабатываются вручную. Ручной учет неизбежно приводит к ошибкам и неточности в обработки информации. Кроме того, ручная обработка данных занимает много времени и требует большого количества ресурсов от сотрудников справочной службы. Поэтому тема работы является актуальной.

Таким образом, к автоматизации деятельности справочной службы кинотеатров города предрасполагает:

* необходимость хранения данных;
* необходимость управлять данными: поиск, добавление, изменение, удаление данных;
* необходимость быстрого получения отчетов, справок по запросу пользователя.

В ходе выполнения работы необходимо решить следующие задачи:

1. Характеристика объекта, процесса, предметной области как объекта информатизации.
2. Анализ информационного обеспечения объекта информатизации.
3. Постановка задач при разработке программного продукта.
4. Обоснование выбора программных средств решения задачи.
5. Разработка модели информационных потоков системы.
6. Проектирование схемы базы данных с помощью CASE-средства ERWin.
7. Описание функций групп пользователей.
8. Описание функций управления данными.
9. Организация защиты БД.
10. Построение схемы взаимосвязи программных модулей. Описание модулей программы.
11. Описание процесса отладки программы.
12. Разработка инструкции пользователя.
13. Разработка графика организации проектирования.

Методы исследования включают в себя основы проектирования информационных систем и баз данных.

# 1 Анализ предметной области

В качестве объекта автоматизации выбрано предприятие, которое занимается демонстрацией кинофильмов в кинотеатрах города.

По каждому кинотеатру в базе данных хранится следующая информация:

1. Название кинотеатра;
2. Категория;
3. Район, где расположен кинотеатр
4. Адрес и телефон кинотеатра;
5. Число посадочных мест;
6. Директор кинотеатра;
7. Владелец кинотеатра;
8. Банк кинотеатра и номер счета в банке;
9. ИНН кинотеатра.

После изучения ситуации на рынке кинофильмов и закупки кинолент в необходимо заносить следующие сведения:

1. Название кинофильма.
2. Категория фильма (боевик, триллер, комедия и др.).
3. Автор сценария.
4. Режиссер-постановщик.
5. Компания-производитель.
6. Год выхода на экран.
7. Поставщик киноленты.
8. Юридический адрес поставщика.
9. Банк поставщика и номер счета в банке.
10. ИНН поставщика; стоимость приобретения.

У одного поставщика может быть куплено несколько фильмов.

Закупленные фильмы сдаются в аренду кинотеатрам города.

Для контроля этой деятельности должна регистрироваться следующая информация:

1. Даты начала и окончания демонстрации фильма;
2. Сумма оплаты за аренду ленты;
3. Пени за несвоевременный возврат.

Необходимо учитывать, что один кинотеатр может одновременно арендовать несколько кинолент.

Таким образом можно выделить следующие функции приложения:

1. Авторизация пользователя в системе.
2. Работа со справочником категорий кинотеатров (категория, вместимость)
3. Работа со справочниками районов, производственных студий, режиссеров, сценаристов, жанров, банков.
4. Работа с данными о поставщиках.
5. Работа с данными о кинотеатрах. Фильтр и экспорт данных в Excel.
6. Работа с данными о фильмах. Фильтр и экспорт данных в Excel.
7. Работа с арендами фильмов. Фильтр и экспорт данных в Excel.

# 2 Обоснование модели данных

В качестве модели данных была выбрана реляционная модель базы данных. Перечислим преимущества реляционной модели базы данных:

1. Гибкость: реляционная модель позволяет создавать таблицы с различными связями между данными, что позволяет гибко управлять информацией.

2. Простота: реляционная модель проста в использовании и понимании, что делает ее доступной для любого пользователя.

3. Эффективность: реляционная модель обеспечивает быстрый доступ к данным, что позволяет повысить производительность базы данных.

4. Надежность: реляционная модель обеспечивает целостность данных, что позволяет избежать ошибок и повреждения информации.

5. Безопасность: реляционная модель обеспечивает управление доступом к данным, что позволяет защитить информацию от несанкционированного доступа.

6. Поддержка: реляционная модель имеет широкую поддержку со стороны различных программных продуктов, что облегчает ее использование и разработку.

# 3 Обоснование выбора средств разработки

В приведённых ниже таблицах 1 – 4 отмечено наличие или отсутствие тех или иных возможностей в некоторых популярных сегодня языках программирования.

Таблица 1– Условные обозначения

|  |  |
| --- | --- |
| Условные обозначения | |
| + | Указанная возможность присутствует |
| - | Указанная возможность отсутствует |
| +/- | Возможность поддерживается не полностью |
| -/+ | Возможность поддерживается очень ограниченно |
| ? | Нет данных |
| N/A | Постановка вопроса не применима к языку |

Таблица 2 – Типы и структуры данных

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Возможность | Язык | | | | | | | |
| [C](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)) | [C++](https://ru.wikipedia.org/wiki/C%2B%2B) | [C#](https://ru.wikipedia.org/wiki/CSharp) | [JavaScript](https://ru.wikipedia.org/wiki/JavaScript) | [Perl](https://ru.wikipedia.org/wiki/Perl) | [PHP](https://ru.wikipedia.org/wiki/PHP) | [VB.NET](https://ru.wikipedia.org/wiki/VB.NET) | [Delphi](https://ru.wikipedia.org/wiki/Delphi_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)) | |
| [Кортежи](https://ru.wikipedia.org/wiki/%D0%A1%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5_%D1%8F%D0%B7%D1%8B%D0%BA%D0%BE%D0%B2_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F#.D0.9A.D0.BE.D1.80.D1.82.D0.B5.D0.B6.D0.B8) | - | +/- | + | - | + | +/- | +/- | - | |
| [Алгебраические типы данных](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%B5%D0%B1%D1%80%D0%B0%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D1%82%D0%B8%D0%BF_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) | - | - | - | N/A | N/A | N/A | - | -/+ | |
| [Многомерные массивы](https://ru.wikipedia.org/wiki/%D0%A1%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5_%D1%8F%D0%B7%D1%8B%D0%BA%D0%BE%D0%B2_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F#.D0.9C.D0.BD.D0.BE.D0.B3.D0.BE.D0.BC.D0.B5.D1.80.D0.BD.D1.8B.D0.B5_.D0.BC.D0.B0.D1.81.D1.81.D0.B8.D0.B2.D1.8B) | + | + | + | +/- | +/- | +/- | + | + | |
| [Динамические массивы](https://ru.wikipedia.org/wiki/%D0%A1%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5_%D1%8F%D0%B7%D1%8B%D0%BA%D0%BE%D0%B2_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F#.D0.94.D0.B8.D0.BD.D0.B0.D0.BC.D0.B8.D1.87.D0.B5.D1.81.D0.BA.D0.B8.D0.B5_.D0.BC.D0.B0.D1.81.D1.81.D0.B8.D0.B2.D1.8B) | - | + | +/- | +/- | +/- | +/- | + | + | |
| [Ассоциативные массивы](https://ru.wikipedia.org/wiki/%D0%A1%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5_%D1%8F%D0%B7%D1%8B%D0%BA%D0%BE%D0%B2_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F#.D0.90.D1.81.D1.81.D0.BE.D1.86.D0.B8.D0.B0.D1.82.D0.B8.D0.B2.D0.BD.D1.8B.D0.B5_.D0.BC.D0.B0.D1.81.D1.81.D0.B8.D0.B2.D1.8B) | - | + | + | + | + | + | + | +/- | |
| [Контроль границ массивов](https://ru.wikipedia.org/wiki/%D0%A1%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5_%D1%8F%D0%B7%D1%8B%D0%BA%D0%BE%D0%B2_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F#.D0.9A.D0.BE.D0.BD.D1.82.D1.80.D0.BE.D0.BB.D1.8C_.D0.B3.D1.80.D0.B0.D0.BD.D0.B8.D1.86_.D0.BC.D0.B0.D1.81.D1.81.D0.B8.D0.B2.D0.BE.D0.B2) | - | +/- | + | N/A | N/A | +/- | + | + | |
| [Цикл foreach](https://ru.wikipedia.org/wiki/%D0%A1%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5_%D1%8F%D0%B7%D1%8B%D0%BA%D0%BE%D0%B2_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F#.D0.A6.D0.B8.D0.BA.D0.BB_foreach) | - | + | + | + | + | + | + | + | |
| [Списковые включения](https://ru.wikipedia.org/wiki/%D0%A1%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5_%D1%8F%D0%B7%D1%8B%D0%BA%D0%BE%D0%B2_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F#List_comprehensions) | - | - | -/+ | - | ? | - | + | - | |

Таблица 3 – Компилятор/интерпретатор

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Возможность | Язык | | | | | | | |
| [C](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)) | [C++](https://ru.wikipedia.org/wiki/C%2B%2B) | [C#](https://ru.wikipedia.org/wiki/CSharp) | [JavaScript](https://ru.wikipedia.org/wiki/JavaScript) | [Perl](https://ru.wikipedia.org/wiki/Perl) | [PHP](https://ru.wikipedia.org/wiki/PHP) | [VB.NET](https://ru.wikipedia.org/wiki/VB.NET) | [Delphi](https://ru.wikipedia.org/wiki/Delphi_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)) |
| [Open-source компилятор (интерпретатор)](https://ru.wikipedia.org/wiki/%D0%A1%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5_%D1%8F%D0%B7%D1%8B%D0%BA%D0%BE%D0%B2_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F#Open-source_.D0.BA.D0.BE.D0.BC.D0.BF.D0.B8.D0.BB.D1.8F.D1.82.D0.BE.D1.80_.28.D0.B8.D0.BD.D1.82.D0.B5.D1.80.D0.BF.D1.80.D0.B5.D1.82.D0.B0.D1.82.D0.BE.D1.80.29) | + | + | + | + | + | + | + | + |
| [Возможность компиляции](https://ru.wikipedia.org/wiki/%D0%A1%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5_%D1%8F%D0%B7%D1%8B%D0%BA%D0%BE%D0%B2_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F#.D0.92.D0.BE.D0.B7.D0.BC.D0.BE.D0.B6.D0.BD.D0.BE.D1.81.D1.82.D1.8C_.D0.BA.D0.BE.D0.BC.D0.BF.D0.B8.D0.BB.D1.8F.D1.86.D0.B8.D0.B8) | + | + | + | + | + | + | + | + |
| [Bootstrapping](https://ru.wikipedia.org/wiki/%D0%A1%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5_%D1%8F%D0%B7%D1%8B%D0%BA%D0%BE%D0%B2_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F#Bootstrapping) | + | + | + | + | ? | N/A | ? | + |
| [Многопоточная компиляция](https://ru.wikipedia.org/wiki/%D0%A1%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5_%D1%8F%D0%B7%D1%8B%D0%BA%D0%BE%D0%B2_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F#.D0.9C.D0.BD.D0.BE.D0.B3.D0.BE.D0.BF.D0.BE.D1.82.D0.BE.D1.87.D0.BD.D0.B0.D1.8F_.D0.BA.D0.BE.D0.BC.D0.BF.D0.B8.D0.BB.D1.8F.D1.86.D0.B8.D1.8F) | + | + | - | ? | ? | ? | + | ? |
| [Интерпретатор командной строки](https://ru.wikipedia.org/wiki/%D0%A1%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5_%D1%8F%D0%B7%D1%8B%D0%BA%D0%BE%D0%B2_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F#.D0.98.D0.BD.D1.82.D0.B5.D1.80.D0.BF.D1.80.D0.B5.D1.82.D0.B0.D1.82.D0.BE.D1.80_.D0.BA.D0.BE.D0.BC.D0.B0.D0.BD.D0.B4.D0.BD.D0.BE.D0.B9_.D1.81.D1.82.D1.80.D0.BE.D0.BA.D0.B8) | -/+ | +/- | + | + | + | + | + | +/- |
| [Условная компиляция](https://ru.wikipedia.org/wiki/%D0%A1%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5_%D1%8F%D0%B7%D1%8B%D0%BA%D0%BE%D0%B2_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F#.D0.A3.D1.81.D0.BB.D0.BE.D0.B2.D0.BD.D0.B0.D1.8F_.D0.BA.D0.BE.D0.BC.D0.BF.D0.B8.D0.BB.D1.8F.D1.86.D0.B8.D1.8F) | + | + | + | -/+ | + | + | + | + |

Таблица 4 – Функциональные возможности

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Возможность | Язык | | | | | | | |
| [C](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)) | [C++](https://ru.wikipedia.org/wiki/C%2B%2B) | [C#](https://ru.wikipedia.org/wiki/CSharp) | [JavaScript](https://ru.wikipedia.org/wiki/JavaScript) | [Perl](https://ru.wikipedia.org/wiki/Perl) | [PHP](https://ru.wikipedia.org/wiki/PHP) | [VB.NET](https://ru.wikipedia.org/wiki/VB.NET) | [Delphi](https://ru.wikipedia.org/wiki/Delphi_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)) |
| [Декларации чистоты функций](https://ru.wikipedia.org/wiki/%D0%A7%D0%B8%D1%81%D1%82%D0%BE%D1%82%D0%B0_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%B8) | - | - | - | - | - | - | - | - |
| [First class functions](https://ru.wikipedia.org/wiki/%D0%A1%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5_%D1%8F%D0%B7%D1%8B%D0%BA%D0%BE%D0%B2_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F#First_class_functions) | -/+ | + | + | + | + | - | ? | +/- |
| [Анонимные функции](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%BE%D0%BD%D0%B8%D0%BC%D0%BD%D1%8B%D0%B5_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%B8) | - | + | + | + | + | + | + | +/- |
| [Лексические замыкания](https://ru.wikipedia.org/wiki/%D0%A1%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5_%D1%8F%D0%B7%D1%8B%D0%BA%D0%BE%D0%B2_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F#.D0.9B.D0.B5.D0.BA.D1.81.D0.B8.D1.87.D0.B5.D1.81.D0.BA.D0.B8.D0.B5_.D0.B7.D0.B0.D0.BC.D1.8B.D0.BA.D0.B0.D0.BD.D0.B8.D1.8F) | - | + | + | + | + | + | + | +/- |
| [Частичное применение](https://ru.wikipedia.org/wiki/%D0%A1%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5_%D1%8F%D0%B7%D1%8B%D0%BA%D0%BE%D0%B2_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F#.D0.A7.D0.B0.D1.81.D1.82.D0.B8.D1.87.D0.BD.D0.BE.D0.B5_.D0.BF.D1.80.D0.B8.D0.BC.D0.B5.D0.BD.D0.B5.D0.BD.D0.B8.D0.B5) | - | +/- | ? | + | - | - | ? | ? |
| [Каррирование](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D1%80%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) | - | +/- | + | + | + | + | - | +/- |

Для выполнения данного проекта был выбран C#. В качестве интерактивной среды разработки использовалась Visual Studio, которая имеет весь необходимый функционал для создания графических интерфейсов пользователя.

Перечислим преимущества среды разработки Visual Studio:

1. Обширный набор инструментов - Visual Studio предоставляет разработчикам широкий спектр инструментов для создания, отладки и тестирования приложений, включая поддержку различных языков программирования и платформ.
2. Широкий спектр поддерживаемых языков программирования – Visual Studio поддерживает большое количество языков программирования, включая C++, C#, Visual Basic, F#, Python и многие другие.
3. Интеграция с другими инструментами - Visual Studio интегрируется с другими инструментами для разработки, такими как Git, Azure DevOps, и другими, что позволяет упростить и автоматизировать процесс разработки.
4. Поддержка многопоточности – Visual Studio предоставляет инструменты для разработки многопоточных приложений, что позволяет улучшить производительность приложений.
5. Расширяемость среды разработки - среда разработки Visual Studio может быть расширена с помощью плагинов и дополнений, которые добавляют новые функции и возможности.
6. Удобная отладка приложений- Visual Studio предлагает широкий спектр инструментов для отладки, включая возможность пошагового выполнения кода, просмотра значения переменных и многое другое.
7. Поддержка облачных сервисов - Visual Studio интегрируется с облачными сервисами, такими как Microsoft Azure, что позволяет разработчикам создавать, развертывать и управлять приложениями в облаке.
8. Безопасность разработки- Visual Studio обеспечивает высокий уровень безопасности и защиты данных при разработке приложений.

Сравнительный анализ систем управления базами данных представлен ниже в таблице 5.

Таблица 5 - Сравнительный анализ систем управления базами данных

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Критерий | MySql | Microsoft Access | interBase SQL Server | Microsoft SQL Server | Oracle Database |
| Производитель | Oracle | Microsoft | Embarcadero | Microsoft | Oracle |
| Платформа | Windows, Lunux | Windows | Windows, Lunux | Windows, Lunux | Windows, Lunux |
| Стоимость лицензии | Средняя | Средняя | Высокая | Высокая | Высокая |
| Клиент - серверная архитектура | Да | Нет | Да | Да | Да |
| Совместимость с другими системами | Нет | Да | Нет | Да | Нет |
| Сложность администрирования | Средняя | Низкая | Высокая | Средняя | Высокая |

Все указанные системы подходят для создания базы данных разрабатываемой автоматизированной информационной системы, однако наиболее целесообразно с точки зрения простоты администрирования, финансовой составляющей, и с учетом имеющейся материально-технической базы организации использовать систему управления базой данных Microsoft SQL Server.

Перечислим преимущества СУБД SQL Server:

1. Высокая производительность: SQL Server обеспечивает быстрый доступ к данным и обработку запросов.

2. Надежность и безопасность: SQL Server предоставляет многоуровневую защиту данных и легко обеспечивает резервное копирование и восстановление данных.

3. Широкий набор функциональных возможностей: SQL Server поддерживает аналитические функции, интеграцию с другими приложениями, а также поддержку различных типов данных.

4. Простота использования: SQL Server имеет удобный интерфейс управления и возможности автоматизации задач.

5. Масштабируемость: SQL Server легко масштабируется и поддерживает работу с большими объемами данных.

6. Поддержка современных технологий: SQL Server поддерживает работу с облачными технологиями, а также интеграцию с различными сервисами и платформами.

7. Обширное сообщество: SQL Server имеет большое сообщество пользователей и разработчиков, что обеспечивает широкую поддержку и доступность ресурсов для разработки и поддержки проектов.

# 4 Проектирование базы данных

Концептуальная модель базы данных представлена на рисунке 1.

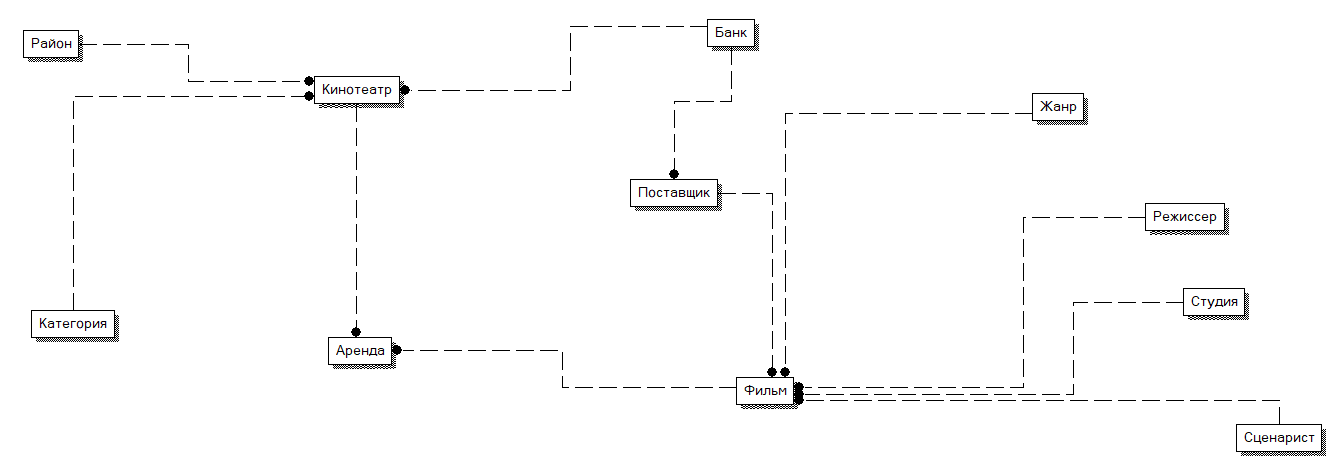


Рисунок 1 - Концептуальная модель базы данных

Логическая модель базы данных представлена на рисунке 2.

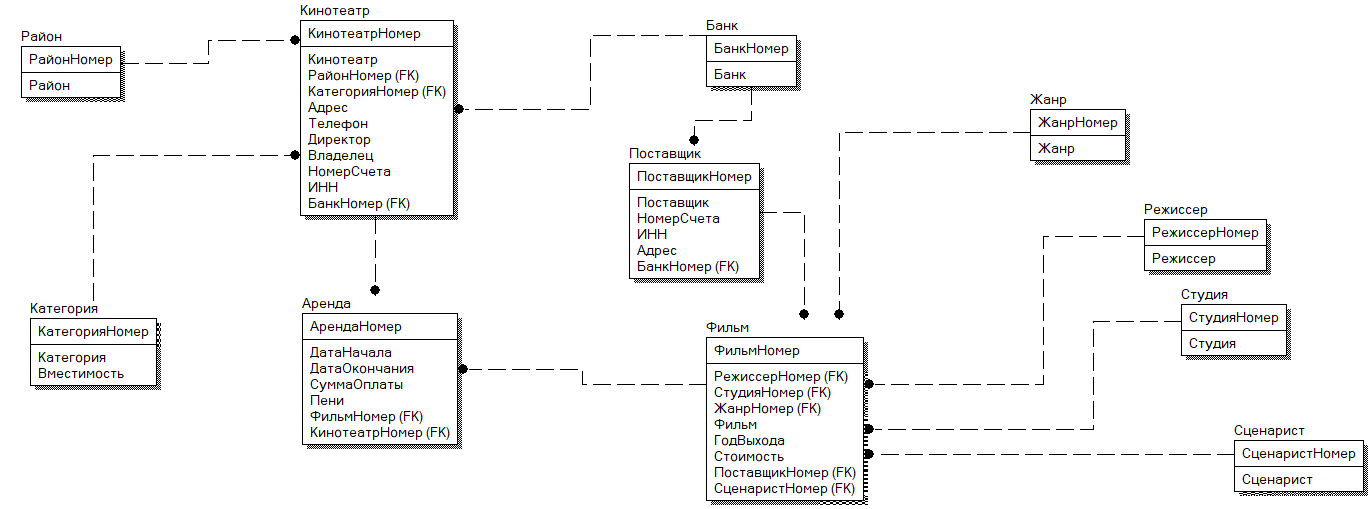


Рисунок 2 – Логическая модель

Физическая модель базы данных представлена на рисунке 3.

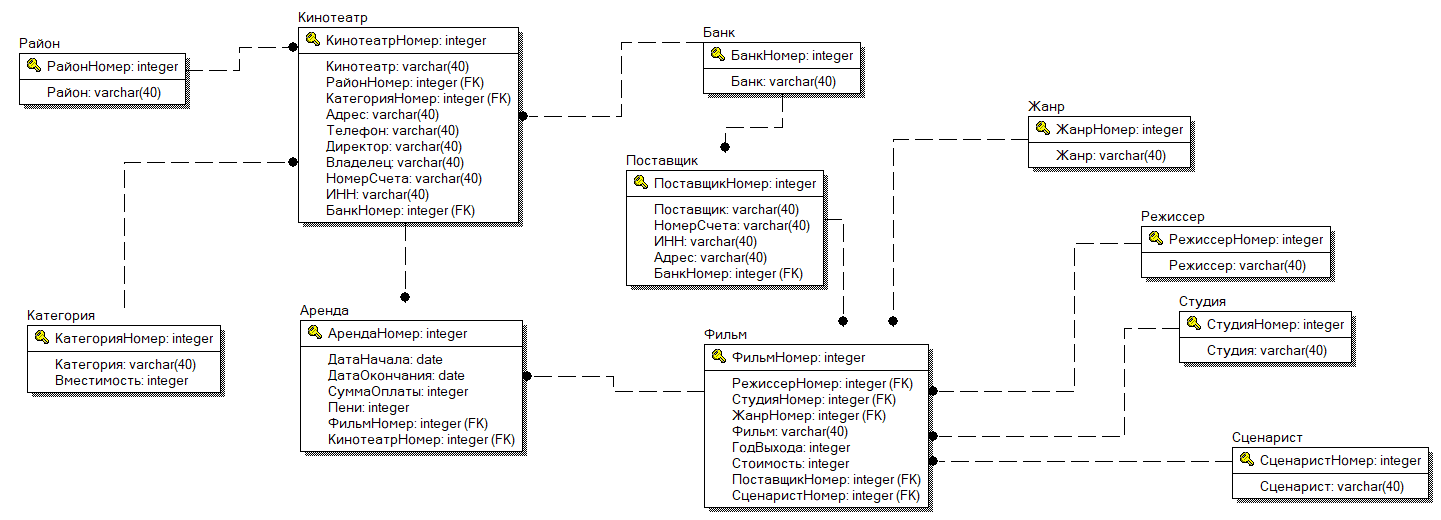


Рисунок 3 – Физическая модель

Описание модели приведено в таблицах 6-16.

Таблица 6 – Структура таблицы «Категория»

|  |  |  |
| --- | --- | --- |
| Наименование | Тип данных | Описание |
| КатегорияНомер | Автоинкремент | Первичный ключ |
| Категория | varchar (40) |  |
| Вместимость | integer |  |

Таблица 7 – Структура таблицы «Район»

|  |  |  |
| --- | --- | --- |
| Наименование | Тип данных | Описание |
| РайонНомер | Автоинкремент | Первичный ключ |
| Район | varchar(40) |  |

Таблица 8 – Структура таблицы «Студия»

|  |  |  |
| --- | --- | --- |
| Наименование | Тип данных | Описание |
| СтудияНомер | Автоинкремент | Первичный ключ |
| Студия | varchar(40) |  |

Таблица 9 – Структура таблицы «Режиссер»

|  |  |  |
| --- | --- | --- |
| Наименование | Тип данных | Описание |
| РежиссерНомер | Автоинкремент | Первичный ключ |
| Режиссер | varchar(40) |  |

Таблица 10 – Структура таблицы «Жанр»

|  |  |  |
| --- | --- | --- |
| Наименование | Тип данных | Описание |
| ЖанрНомер | Автоинкремент | Первичный ключ |
| Жанр | varchar(40) |  |

Таблица 11 – Структура таблицы «Сценарист»

|  |  |  |
| --- | --- | --- |
| Наименование | Тип данных | Описание |
| СценаристНомер | Автоинкремент | Первичный ключ |
| Сценарист | varchar(40) |  |

Таблица 12 – Структура таблицы «Банк»

|  |  |  |
| --- | --- | --- |
| Наименование | Тип данных | Описание |
| БанкНомер | Автоинкремент | Первичный ключ |
| Банк | varchar(40) |  |

Таблица 13 – Структура таблицы «Кинотеатр»

|  |  |  |
| --- | --- | --- |
| Наименование | Тип данных | Описание |
| КинотеатрНомер | Автоинкремент | Первичный ключ |
| РайонНомер | int | Внешний ключ (таблица «Район») |
| КатегорияНомер | int | Внешний ключ (таблица «Категория») |
| БанкНомер | int | Внешний ключ (таблица «Банк») |
| Адрес | varchar(40) |  |
| Телефон | varchar(40) |  |
| Директор | varchar(40) |  |
| Владелец | varchar(40) |  |
| НомерСчета | varchar(40) |  |
| ИНН | varchar(40) |  |
| БанкНомер | varchar(40) |  |

Таблица 14 – Структура таблицы «Фильм»

|  |  |  |
| --- | --- | --- |
| Наименование | Тип данных | Описание |
| ФильмНомер | Автоинкремент | Первичный ключ |
| РежиссерНомер | int | Внешний ключ (таблица «Режиссер») |
| СтудияНомер | int | Внешний ключ (таблица «Студия») |
| ЖанрНомер | int | Внешний ключ (таблица «Жанр») |
| ПоставщикНомер | int | Внешний ключ (таблица «Поставщик») |
| СценаристНомер | int | Внешний ключ (таблица «Сценарист») |
| Фильм | varchar(40) |  |
| ГодВыпуска | int |  |
| Стоимость | int |  |

Таблица 15 – Структура таблицы «Аренда»

|  |  |  |
| --- | --- | --- |
| Наименование | Тип данных | Описание |
| СеансНомер | Автоинкремент | Первичный ключ |
| ДатаНачала | date |  |
| ВремяОкончания | date |  |
| СуммаОплаты | int |  |
| Пени | int |  |
| КинотеатрНомер | int | Внешний ключ (таблица «Кинотеатр») |
| ФильмНомер | date | Внешний ключ (таблица «Фильм») |

Таблица 16 – Структура таблицы «Поставщик»

|  |  |  |
| --- | --- | --- |
| Наименование | Тип данных | Описание |
| ПоставщикНомер | Автоинкремент | Первичный ключ |
| БанкНомер | int | Внешний ключ (таблица «Банк») |
| Поставщик | varchar(40) |  |
| НомерСчета | varchar(40) |  |
| ИНН | varchar(40) |  |
| Адрес | varchar(40) |  |

# 5 Описание функций управления данными

Доступ к базе данных осуществляется с помощью ADO.NET. ADO.NET — это набор технологий в Visual Studio для доступа к базам данных. Ниже перечислены основные компоненты ADO.NET:

1. Connection (Подключение) — это компонент, который используется для установления соединения с базой данных.
2. Command (Команда) - используется для выполнения команд SQL на базе данных. Компоненты выполняет такие команды, как SELECT, INSERT, UPDATE и DELETE.
3. DataReader (Читатель данных) - обеспечивает доступ к результатам выполнения запросов на чтение (SELECT) по одной записи за раз.
4. DataAdapter (Адаптер данных) - служит промежуточным звеном между базой данных и DataSet. Используется для заполнения DataSet из базы данных и обновления данных в базе данных на основе изменений, внесенных в DataSet.
5. DataSet (Набор данных) - представляет данные из базы данных, полученные с помощью DataAdapter. DataSet содержит таблицы, которые представляют результаты SQL-запроса.
6. DataTable (Таблица) - представляет таблицу с данными из DataSet. Он может содержать столбцы и строки.
7. DataView (Представление данных) — это инструмент отображения данных из DataTable в виде таблицы. DataView может быть отфильтрован, сортирован или иначе изменен для представления данных определенным образом.
8. SqlParameter (Параметр запроса) - используется для передачи параметров SQL-запроса.

Для вывода данных на экран использовался компонент dataGridView.

Для управления таблицей использовался компонент BindingNavigator.

Схема взаимодействия доступа к базе данных представлена на рисунке 4.

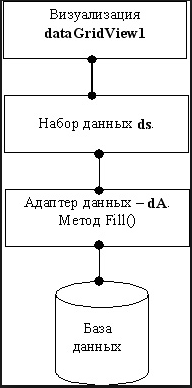


Рисунок 4 - Схема взаимодействия доступа к базе данных

При настройке подключения к базе данных в проекте, выбран источник данных, рисунок 5.

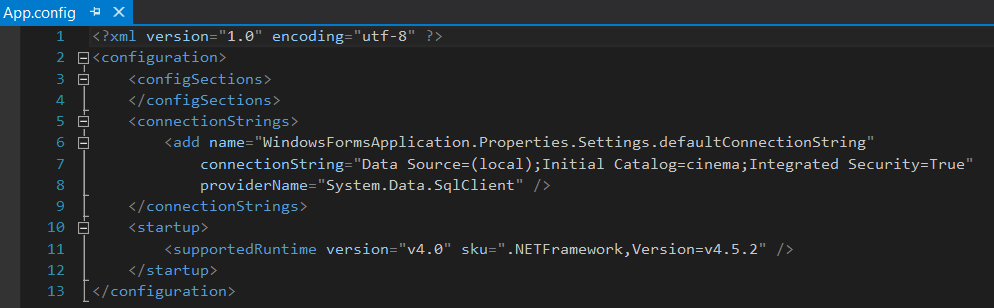


Рисунок 5 – Выбор источника данных

Настроенный источник данных представлен на рисунке 6

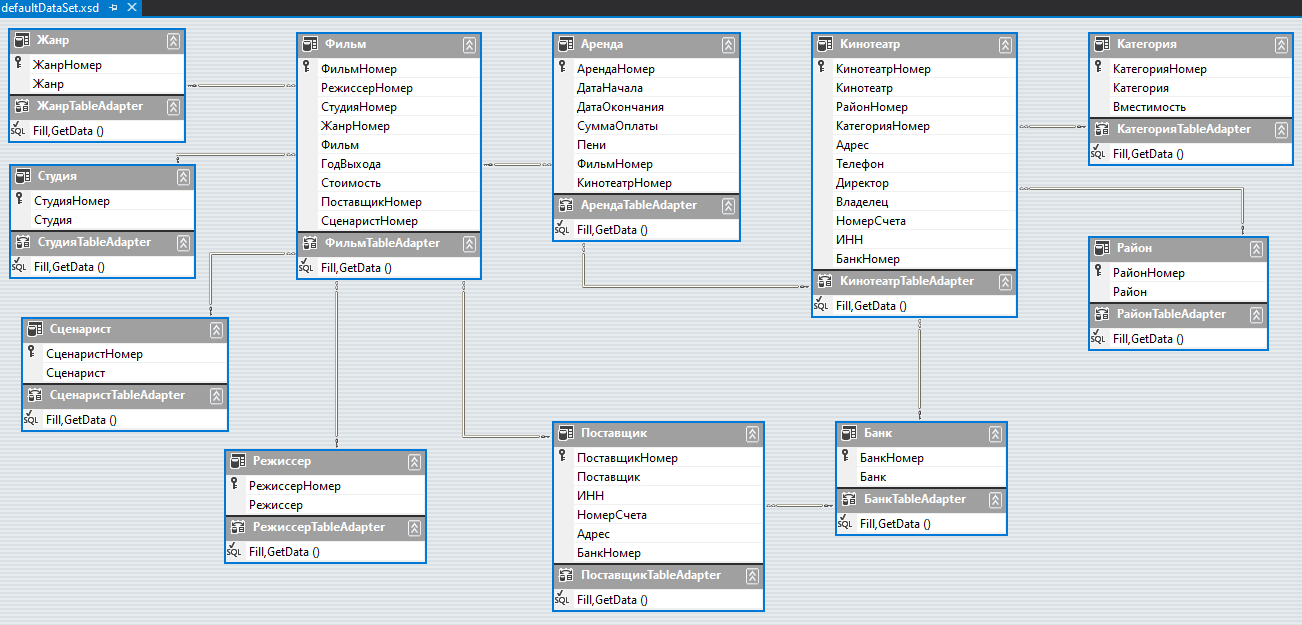


Рисунок 6 – Настроенный источник данных

# 6 Разработка приложения по работе с базой данных

Приложение реализовано в среде разработки Microsoft Visual Studio на языке C#. Графический интерфейс пользователя реализован с использованием интерфейса программирования Windows Forms.

Схема взаимосвязи программных модулей представлена на рисунке 7.



Рисунок 7 – Схема модулей системы

За работу главной формы приложения отвечает модуль frmMain.cs.

Модуль frmCategory.cs отвечает за работу со категориями кинотеатров.

Модуль frmCinema.cs отвечает за работу с кинотеатрами.

Модуль frmDistrict.cs отвечает за работу с районами города.

Модуль frmFilm.cs отвечает за работу с фильмами.

Модуль frmGenre.cs отвечает за работу с жанрами.

Модуль frmRent.cs отвечает за работу с арендами.

Модуль frmStudio.cs отвечает за работу со производственными студиями.

Модуль frmScriptwriter.cs отвечает за работу со сценаристами.

Модуль frmBank.cs отвечает за работу с банками.

Модуль frmDirector.cs отвечает за работу с режиссерами.

Модуль frmSupplier.cs отвечает за работу с поставщиками.

# ЗАКЛЮЧЕНИЕ

Спроектированная и разработанная информационная система «Кинотеатр» позволит эффективно осуществлять и учитывать сведения по аренде фильмов кинотеатрам города. Система спроектирована с использованием современных методик автоматизации бизнес-процессов.

Использование данной системы позволяет сократить временной цикл операции обработки сведений справочной службы за счёт перевода их в электронную форму и обеспечить достоверность информации на выходе.

В ходе выполнения работы были выполнены следующие задачи:

1. Описана предметную область и функции решаемых задач.
2. Описана входная и выходная информация.
3. Сформулирована постановка задачи.
4. Проведен анализ средств разработки информационных систем, в ходе анализа в качестве средств разработки выбрана среда разработки Visual Studio и СУБД MS SQL Server.
5. Выполнено проектирование базы данных. С помощью CASE-средства ERWin построена концептуальная, логическая и физическая модель БД.
6. Разработана база данных в СУБД MS SQL Server.
7. Разработана клиентская часть информационной системы.
8. Выработаны инструкции по эксплуатации информационной системы.

Все задачи работы выполнены. Целесообразность разработки данной информационной системы обуславливается широким спектром решаемой ею задач и возможностью получить существенную выгоду от применения данной системы.

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Астелс, Д. Практическое руководство по программированию / Д. Астелс, Г. – М.: Издательский дом «Вильямс», 2018. – 320 с.
2. Баженова, И.Ю. Основы проектирования приложений баз данных / И.Ю. Баженова. – М.: Бином. Лаборатория знаний, Интернет-университет информационных технологий, 2018. – 325 с.
3. Бекаревич, Ю.Б. Самоучитель Access 2010 / Ю.Б. Бекаревич, Н.В. Пушкина. – СПб.: БХВ-Петербург, 2019. – 432 с.
4. Вендров, А.М. CASE-технологии. Современные средства проектирования информационных систем / А.М. Вендров. – М.: Финансы и статистика, 2019. – 456 с.
5. Вигерсон, К. Разработка требований к программному обеспечению / К. Вигерс, Д. Битти. – М.: Издательство «Русская редакция», 2019. – 736 с.
6. Гайдамакин, Н.А. Автоматизированные системы и банки данных. Вводный курс: Учебное пособие / Н.А. Гайдамакин. – М.: Гелиос АРВ, 2018. – 368 с.
7. Гамма, Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования / Э. Гамма, Р. Хелм, Р. Джонсон, Д. Влиссидес. – СПб.: Питер, 2019. – 368 с.
8. Гущин А. Н. Базы данных: учебник. - М.: Директ-Медиа, 2019. - 266 с.
9. Карпова Т.С. Базы данных: модели, разработка, реализация. - СПб: Питер, 2018. – 142с.
10. Карпова, И.П. Базы данных: Учебное пособие / И.П. Карпова. - СПб: Питер, 2019. - 240 c.

# ПРИЛОЖЕНИЕ А Руководство пользователя

Перед запуском программы необходимо создать на сервере базы данных cinemarent, выполнить в неё скрипт для создания таблиц и заполнения их данными.

В SQL Server необходимо создать новую базу данных cinemarent, рисунок 8.

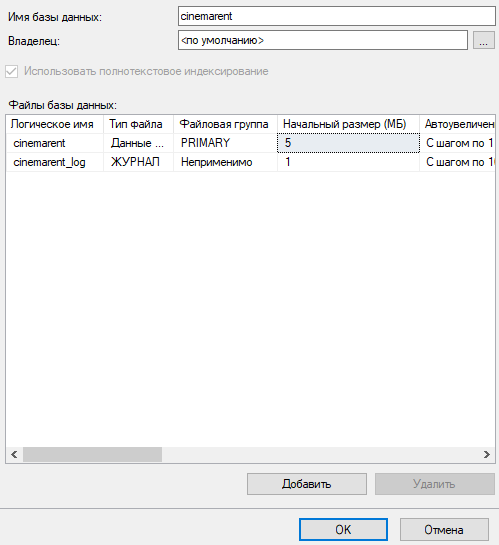


Рисунок 8 - Создание базы данных cinemarent

Выполнили в SQL Server Management Studio ddl-скрипт, рисунок 9.

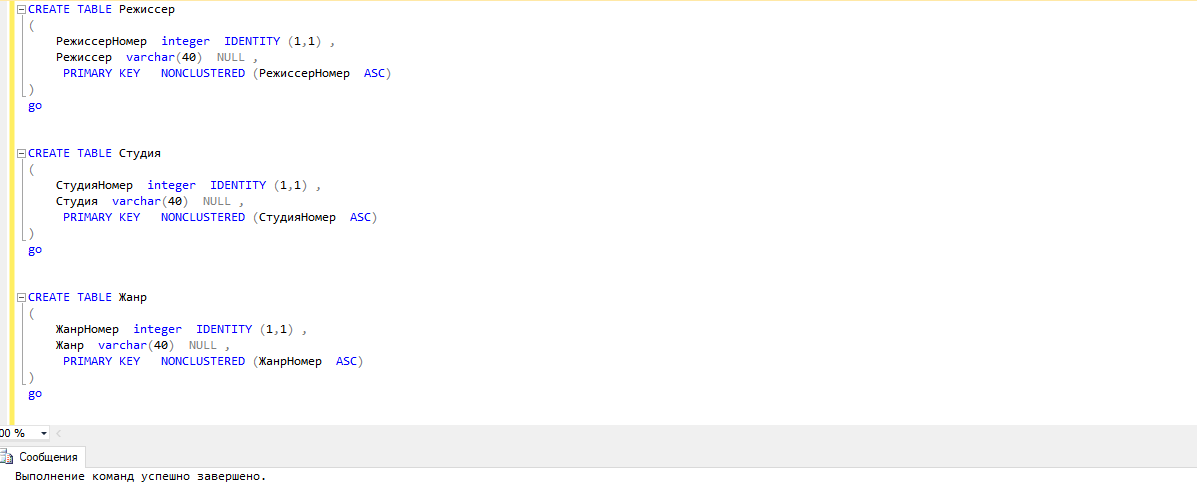


Рисунок 9 - Выполнение DDL-скрипта

В результате выполнения ddl-скрипта получили таблицы базы данных, которые по своей структуре соответствуют физической модели данных. Схема данных представлена на рисунке 10.



Рисунок 10 - Схема данных в SQL Server

После этого необходимо настроить config-файл подключения, задав имя сервера, рисунок 11.

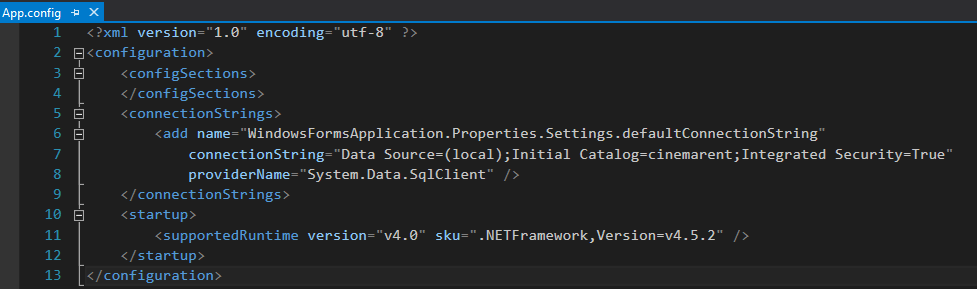


Рисунок 11 – Выбор источника данных

Запуск программы осуществляется посредством открытия exe-файла.

Главная форма программы представлена на рисунке 12.

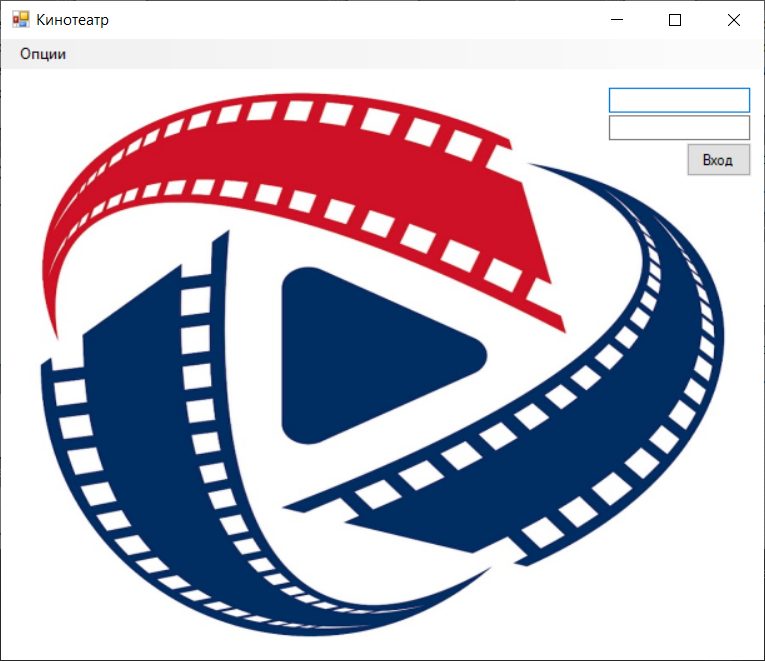


Рисунок 12 – Главная форма приложения

Рассмотрим полный доступ к БД в режиме администратора. После успешной авторизации в распоряжении администратора оказывается главное меню программы, рисунок 13.

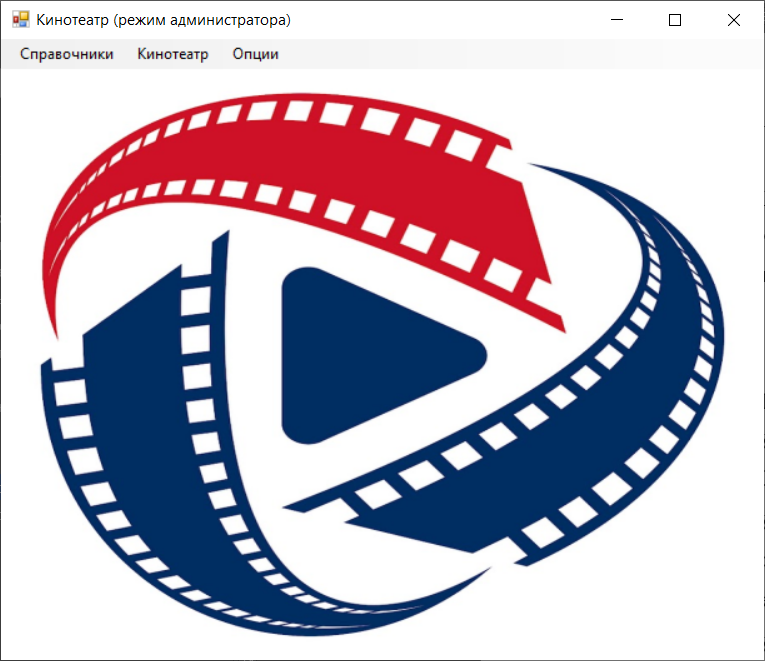


Рисунок 13 – Главное меню программы

Меню программы включает в себя следующие пункты:

1. Пункт меню «Справочники» включает в себя следующие подпункты: «Режиссеры», «Сценаристы», «Студии», «Жанры», «Районы» и «Банки».
2. Пункт меню «Кинотеатр» включает в себя следующие подпункты: «Поставщики», «Категории», «Фильмы», «Кинотеатры» и «Аренды».
3. Пункт меню «Опции» включает в себя подпункт «Выход».

Рассмотрим работу со справочниками.

Форма «Режиссеры» отвечает за отображение, редактирование, обновление и удаление данных таблицы «Режиссеры», рисунок 14.

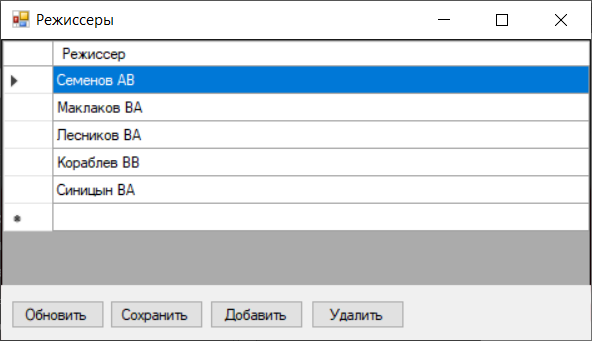


Рисунок 14 – Форма «Режиссеры»

Форма «Студии» отвечает за отображение, редактирование, обновление и удаление данных таблицы «Студии», рисунок 15.

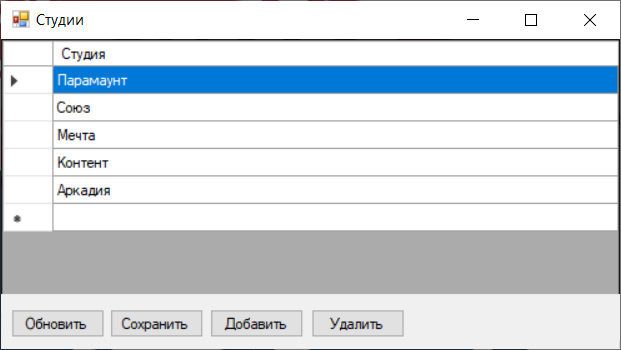


Рисунок 15 – Форма «Студии»

Форма «Жанры» отвечает за отображение, редактирование, обновление и удаление данных таблицы «Жанры», рисунок 16.

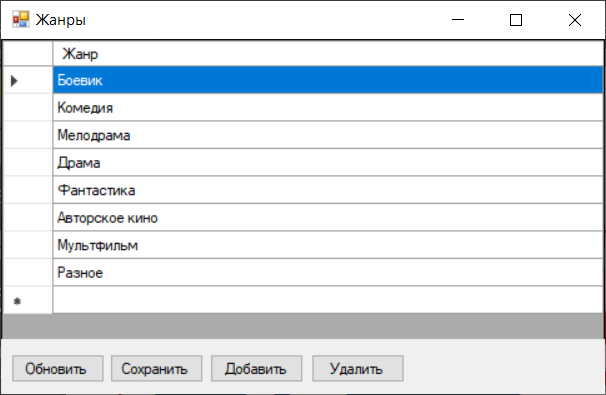


Рисунок 16 – Форма «Жанры»

Форма «Районы» отвечает за отображение, редактирование, обновление и удаление данных таблицы «Районы», рисунок 17.

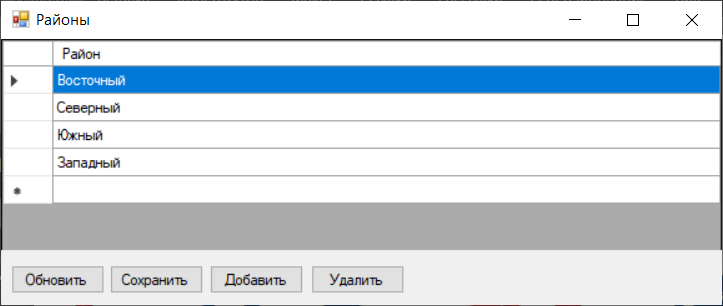


Рисунок 17 – Форма «Районы»

Форма «Категории» отвечает за отображение, редактирование, обновление и удаление данных таблицы «Категории», рисунок 18.

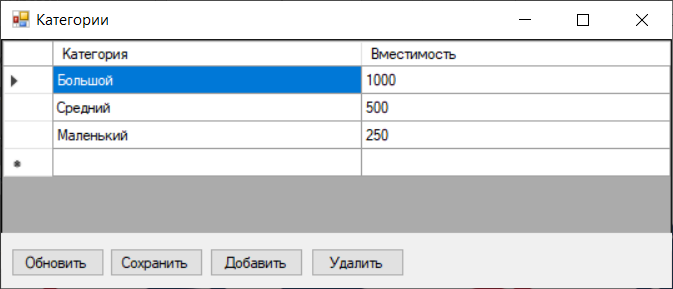


Рисунок 18 – Форма «Категории»

Форма «Поставщики» отвечает за отображение, редактирование, обновление и удаление данных таблицы «Поставщики», рисунок 19.

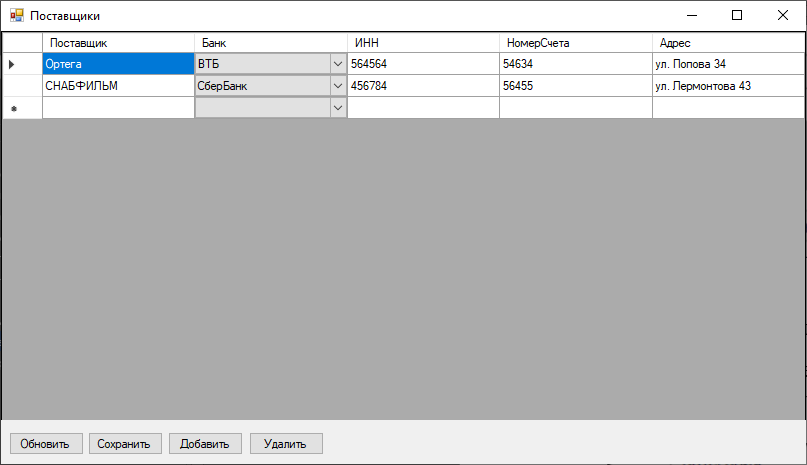


Рисунок 19 – Форма «Поставщики»

Форма «Фильмы» отвечает за отображение, редактирование, обновление и удаление данных таблицы «Фильмы», рисунок 20.

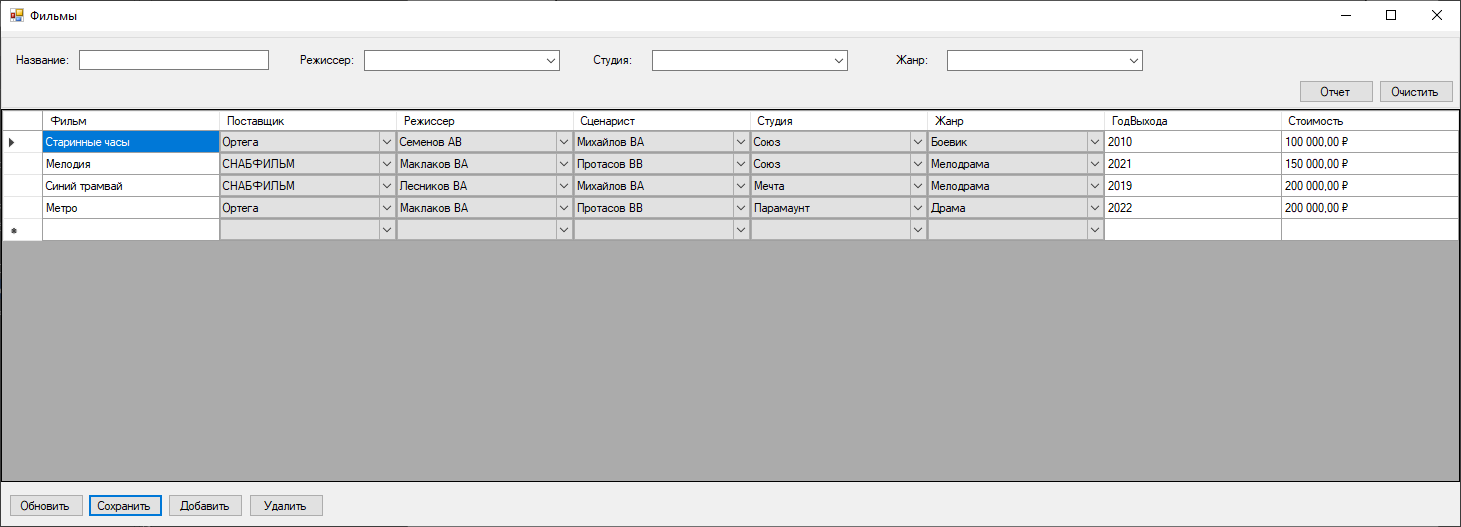


Рисунок 20 – Форма «Фильмы»

На форме доступен фильтр и экспорт данных в Excel.

Фильтр доступен по столбцам таблицы:

1. Название (перечень фильтруется после каждого введенного символа).
2. Режиссер (выбор из списка).
3. Студия (выбор из списка).
4. Жанр (выбор из списка).

Фильтр фильмов по режиссеру представлен на рисунке 21.

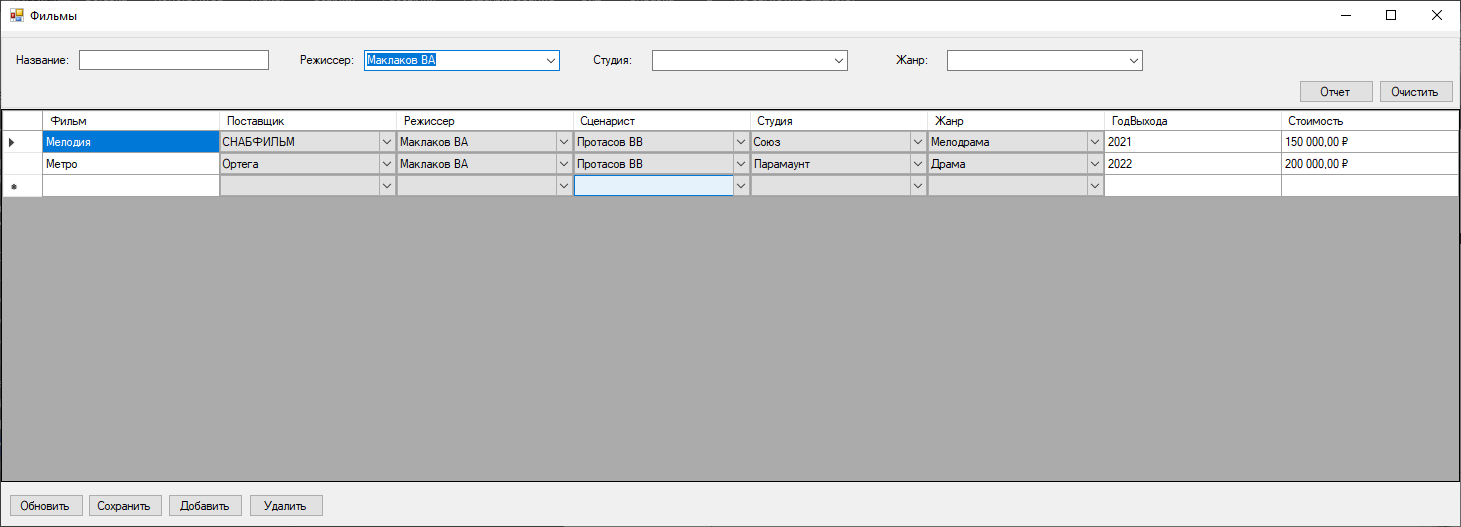


Рисунок 21 - Фильтр фильмов по режиссеру

Форма «Кинотеатры» отвечает за отображение, редактирование, обновление и удаление данных таблиц «Кинотеатры», рисунок 22.

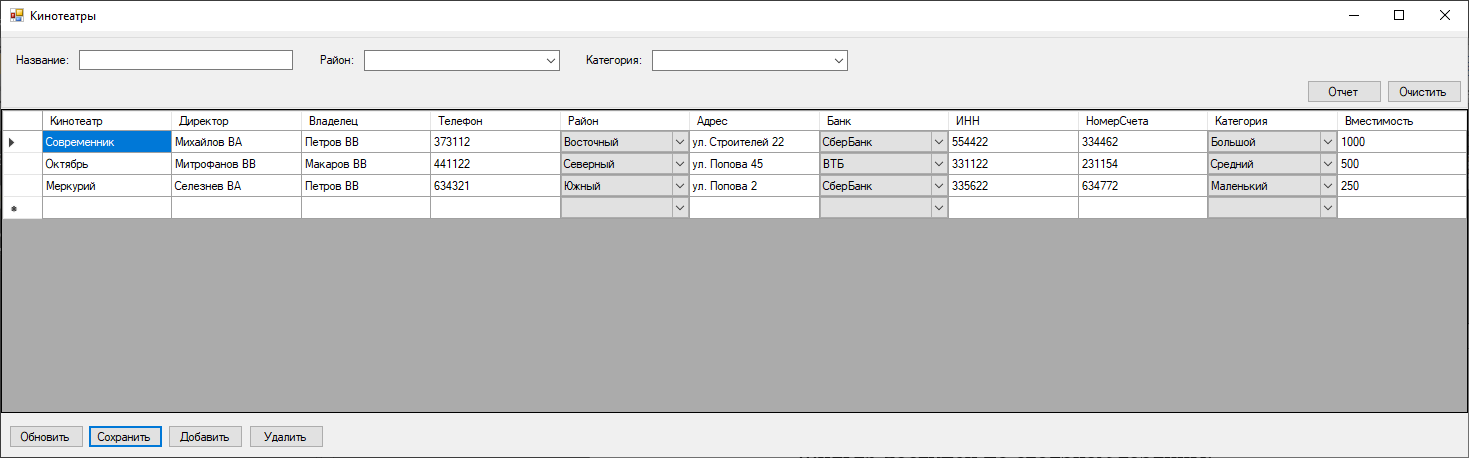


Рисунок 22 – Форма «Кинотеатры»

На форме доступен фильтр и экспорт данных в Excel.

Фильтр доступен по столбцам таблицы:

1. Название (перечень фильтруется после каждого введенного символа).
2. Район (выбор из списка).
3. Категория (выбор из списка).

Работа с арендами представлена на рисунке 23.

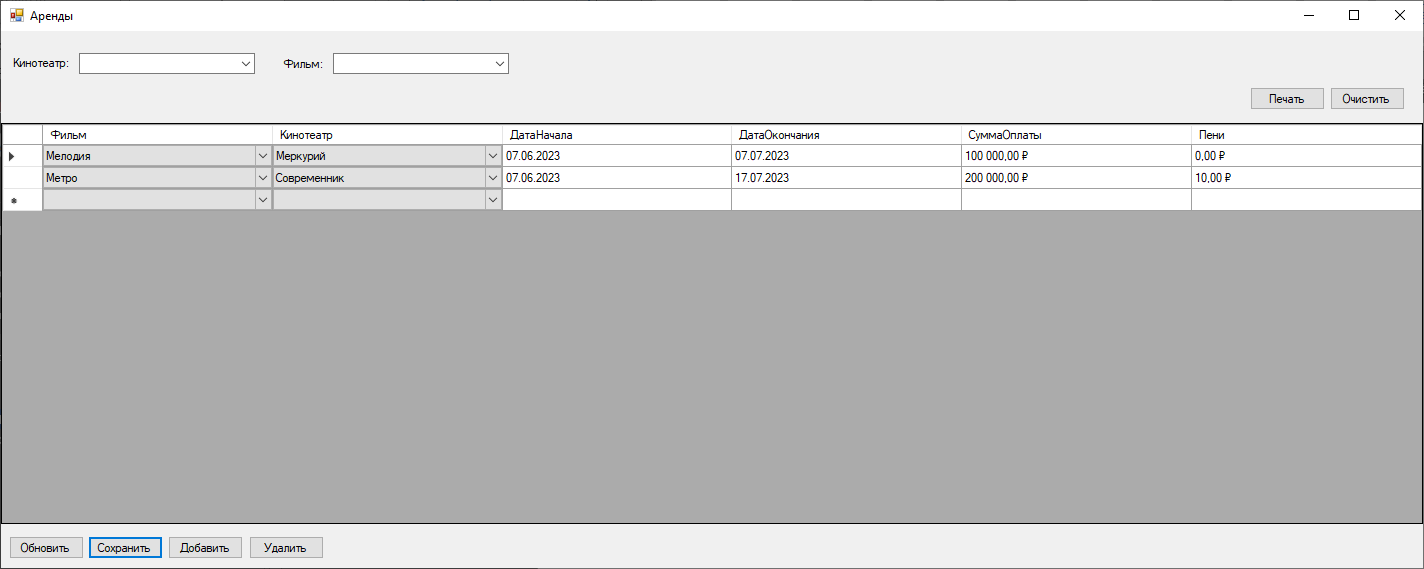


Рисунок 23 – Работа с арендами

Фильтр доступен по столбцам таблицы:

1. Кинотеатр (выбор из списка).
2. Фильм (выбор из списка).

Экспорт сеансов в Excel представлен на рисунке 24.

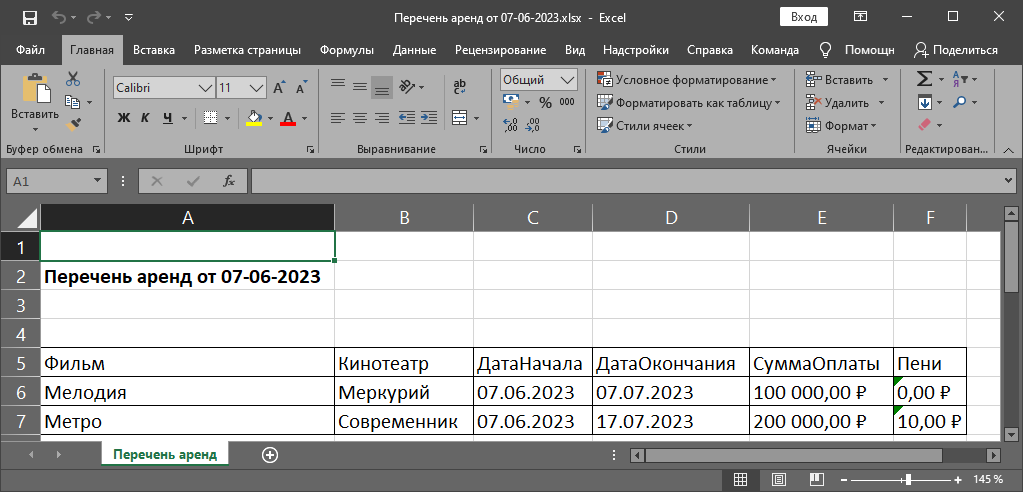


Рисунок 24 - Экспорт аренд в Excel

Для выхода из программы необходимо выбрать пункт меню «Опции» - «Выход».

# ПРИЛОЖЕНИЕ Б Код программы

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApplication

{

public partial class frmRent : Form

{

public frmRent()

{

InitializeComponent();

}

private void frmRent\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "defaultDataSet.Кинотеатр". При необходимости она может быть перемещена или удалена.

this.кинотеатрTableAdapter.Fill(this.defaultDataSet.Кинотеатр);

// TODO: данная строка кода позволяет загрузить данные в таблицу "defaultDataSet.Фильм". При необходимости она может быть перемещена или удалена.

this.фильмTableAdapter.Fill(this.defaultDataSet.Фильм);

// TODO: данная строка кода позволяет загрузить данные в таблицу "defaultDataSet.Аренда". При необходимости она может быть перемещена или удалена.

this.арендаTableAdapter.Fill(this.defaultDataSet.Аренда);

button6\_Click(sender, e);

}

private void button4\_Click(object sender, EventArgs e)

{

frmRent\_Load(sender, e);

}

private void button1\_Click(object sender, EventArgs e)

{

this.арендаTableAdapter.Update(this.defaultDataSet.Аренда);

}

private void button2\_Click(object sender, EventArgs e)

{

арендаBindingSource.AddNew();

DataRowView drw = арендаBindingSource.Current as DataRowView;

drw["ДатаНачала"] = DateTime.Today.ToString();

drw["ДатаОкончания"] = DateTime.Today.ToString();

}

private void button3\_Click(object sender, EventArgs e)

{

DialogResult result = MessageBox.Show("Удалить запись?", "Диалог удаления записи", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (result == DialogResult.Yes) //Если нажал Да

{

арендаBindingSource.RemoveCurrent();

}

}

private void button7\_Click(object sender, EventArgs e)

{

{// создаем приложение Excel

Microsoft.Office.Interop.Excel.\_Application app = new Microsoft.Office.Interop.Excel.Application();

// создаем новую книгу

Microsoft.Office.Interop.Excel.\_Workbook workbook = app.Workbooks.Add(Type.Missing);

// создаем новый лист

Microsoft.Office.Interop.Excel.\_Worksheet worksheet = null;

// приложение видимо

app.Visible = true;

// связь с листом

worksheet = (Microsoft.Office.Interop.Excel.\_Worksheet)workbook.ActiveSheet;

// переименование листа

worksheet.Name = "Перечень аренд";

worksheet.Cells[2, 1] = "Перечень аренд от " + DateTime.Today.ToString("dd-MM-yyyy");

int offset = 4;

int i = 0, j = 0;

// выводим заголовок

for (i = 1; i < dataGridView1.Columns.Count + 1; i++)

{ worksheet.Cells[1 + offset, i] = dataGridView1.Columns[i - 1].HeaderText; }

// экспорт данных

for (i = 0; i < dataGridView1.Rows.Count - 1; i++)

{

for (j = 0; j < dataGridView1.Columns.Count; j++)

{ worksheet.Cells[i + 2 + offset, j + 1] = dataGridView1.Rows[i].Cells[j].FormattedValue.ToString(); }

}

worksheet.Range["A" + Convert.ToString(1 + offset) + ":F" + Convert.ToString(dataGridView1.Rows.Count + offset)].Borders.LineStyle = 1;

// устанавливаем жирный шрифт для заголовков

Microsoft.Office.Interop.Excel.Range rng = (Microsoft.Office.Interop.Excel.Range)worksheet.Rows[2];

rng.Font.Bold = true;

// авто-размер столбцов

worksheet.Columns.AutoFit();

// сохранение файла

workbook.SaveAs("Перечень аренд от " + DateTime.Today.ToString("dd-MM-yyyy") + ".xlsx", Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing, Microsoft.Office.Interop.Excel.XlSaveAsAccessMode.xlExclusive, Type.Missing, Type.Missing, Type.Missing, Type.Missing);

}

}

private void button6\_Click(object sender, EventArgs e)

{

comboBox1.Text = "";

comboBox5.Text = "";

арендаBindingSource.Filter = "АрендаНомер=АрендаНомер";

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

//фильтр данных в зависимости от введенных параметров

арендаBindingSource.Filter = "АрендаНомер=АрендаНомер";

if (comboBox1.Text.Length != 0)

арендаBindingSource.Filter = арендаBindingSource.Filter + " and КинотеатрНомер=" + Convert.ToString(comboBox1.SelectedValue);

if (comboBox5.Text.Length != 0)

арендаBindingSource.Filter = арендаBindingSource.Filter + " and ФильмНомер=" + Convert.ToString(comboBox5.SelectedValue);

}

}

}