|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  | МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ | | | |
|  |  |  |  | федеральное государственное бюджетное образовательное учреждение | | |
|  |  |  |  |  | высшего профессионального образования | |
|  |  |  | **«Балтийский государственный технический университет «ВОЕНМЕХ» им. Д.Ф. Устинова»** | | | |
|  |  |  |  |  | **(БГТУ «ВОЕНМЕХ» им. Д.Ф. Устинова»)** | |
|  |  |  |  |  | БГТУ.СМК-П-4.2-12 | |
| Факультет |  |  | И |  | Информационные и управляющие системы |  |
|  |  |  | шифр |  | наименование | |
| Кафедра |  |  | И5 |  | Информационные системы и программная инженерия |  |
|  |  |  | шифр |  | наименование | |
| Дисциплина |  |  | Базы данных | | |  |



КУРСОВОЙ ПРОЕКТ

на тему

|  |
| --- |
| «Разработка модуля Планы НИР |
| для ЭИОС магистратуры» |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Выполнил студент группы | | | |  |  | И967 |
|  | Васильев Н.А. | | |  |  |  |
|  | Фамилия И.О. | | |  |  |  |
|  |  | **РУКОВОДИТЕЛЬ** | | | | |
| Вальштейн К.В. | | | |  |  |  |
| Фамилия И.О. | |  |  | Подпись | | |
| Оценка | | | |  |  |  |
| «\_\_\_\_\_» |  |  |  |  |  | 2019г. |

САНКТ-ПЕТЕРБУРГ

2019 г.

РЕФЕРАТ

Пояснительная записка к курсовому проекту содержит 32 страниц, 23 иллюстрацию, 3 таблицы, 1 приложение.

Объектом исследования является ИПК для директора кинотеатра.

Цель работы – разработка ИПК для директора кинотеатра.

* ходе работы над курсовым проектом требовалось разработать информационно-программный комплекс для директора кинотеатра. Была описана предметная область, исходные данные и были составлены соответствующие бизнес-правила.

Далее было произведено инфологическое моделирование с описанием спецификации сущностей. Также было произведено даталогическое моделирование, где были описаны спецификации отношений, полей, связей, ключей. Также был разработан пользовательский интерфейс с использованием СУБД Microsoft Access 2003. Также в конце работы были приведены примеры отчетов, полученных при работе с ИПК.

В результате исследования был создан требуемый информационно-программный комплекс.

Основные конструктивные показатели: высокая скорость работы, простота в использовании, удобная система общения с пользователем.

СОДЕРЖАНИЕ

[РЕФЕРАТ 2](#_Toc10545506)

[ВВЕДЕНИЕ 4](#_Toc10545507)

[1 Постановка задачи 7](#_Toc10545508)

[2 Инфологическая и даталогическая модель 8](#_Toc10545509)

[2.1 Инфологическая модель 8](#_Toc10545510)

[2.2 Даталогическая модель 12](#_Toc10545511)

[3 Структуры таблиц 13](#_Toc10545512)

[4 Формы и их работа 15](#_Toc10545513)

[4.1 Главная форма готовой базы данных «Кинотеатр» 15](#_Toc10545514)

[4.2 Форма «Залы» с подчинёнными формами «Секторы зала» и «Места» 16](#_Toc10545515)

[4.3 Форма «Фильмы» 16](#_Toc10545516)

[4.4 Форма «Сеансы» с подчинёнными формами «Билеты» и «Цены на билеты» 17](#_Toc10545517)

[4.5 Форма «Продажа бронирование» 17](#_Toc10545518)

[4.6 Форма «Движение билетов»: 21](#_Toc10545519)

[4.7 Форма «Должность» с подчинённой формой «Сотрудники» 22](#_Toc10545520)

[5 Запросы (режим SQL) 23](#_Toc10545521)

[5.1 Запрос «Создание билетов» 23](#_Toc10545522)

[5.2 Запрос «Итоги за год» 23](#_Toc10545523)

[5.3 Запрос «Движение билетов по дате» 24](#_Toc10545524)

[5.4 Запрос «Добавление брони» 24](#_Toc10545525)

[5.5 Запрос «Добавление возврата» 25](#_Toc10545526)

[5.6 Запрос «Добавление продажи» 25](#_Toc10545527)

[5.7 Запрос «Обновление брони» 25](#_Toc10545528)

[5.8 Запрос «Обновление оплачено» 25](#_Toc10545529)

[5.9 Запрос «Обновление не оплачено» 26](#_Toc10545530)

[6 Отчёты 27](#_Toc10545531)

[6.1 Отчёт «Движение билетов по дате» 27](#_Toc10545532)

[6.2 Отчёт «Итоги за месяц» 27](#_Toc10545533)

[6.3 Отчёт «Итоги за год» 28](#_Toc10545534)

[6.4 Отчёт «Итоги по сотрудникам за месяц» 28](#_Toc10545535)

[6.5 Отчёт «Итоги проката» 29](#_Toc10545536)

[ЗАКЛЮЧЕНИЕ 30](#_Toc10545537)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 31](#_Toc10545538)

[ПРИЛОЖЕНИЕ 32](#_Toc10545539)

ВВЕДЕНИЕ

Задача курсового проекта заключается в создании базы данных (информационной системы к ней), которая должна демонстрировать возможности, принципы, основы, требовательность баз данных в одной из сред (FoxPro, ACCESS, MySQL) для их создания. В данном проекте будет использована среда MS ACCESS 2003.

Элементы базы данных:

1. Таблицы. В базе данных информация хранится в виде двумерных таблиц. Можно так же импортировать и связывать таблицы из других СУБД или систем управления электронными таблицами. Одновременно могут быть открыты 1024 таблицы [1].
2. Запросы. При помощи запросов можно произвести выборку данных по какому-нибудь критерию из разных таблиц. В запрос можно включать до 255 полей [2].
3. Формы. Формы позволяют отображать данные из таблиц и запросов в более удобном для восприятия виде. С помощью форм можно добавлять и изменять данные, содержащиеся в таблицах. В формы позволяют включать модули [3].
4. Отчёты. Отчёты предназначены для печати данных, содержащихся в таблицах и запросах, в красиво оформленном виде. Отчёты так же позволяют включать модули [4].

Поле - это минимальный элемент базы данных, содержащий один неделимый квант информации, например, поле "Номер" при создании нового документа. Каждое поле характеризуется именем и типом хранящихся в нем данных [5].

Существует четыре вида связей (отношений между таблицами):

- один к одному,

- один ко многим,

- многие к одному,

- многие ко многим.

Связь один к одному означает, что каждой записи первой таблицы соответствует только одна, связанная с ней запись второй таблицы и наоборот. Такой тип отношений используется крайне редко, так как фактически все данные могут быть помещены в одной таблице [6].

Связь один ко многим характерна тем, что запись одной таблицы связана с несколькими записями другой таблицы. В то же время запись второй таблицы не может быть связана более чем с одной записью первой таблицы [7].

Связь многие ко многим или непрямая табличная связь определяет связь одной записи первой таблицы с несколькими записями второй таблицы, в то же время как одна запись второй таблицы может быть связана с несколькими записями первой таблицы [8].

Главная кнопочная форма создается с целью навигации по базе данных. Эта форма может использоваться в качестве главного меню базы данных. Элементами главной кнопочной формы являются объекты форм и отчетов (запросы и таблицы не являются элементами главной кнопочной формы) [9].

Небольшой экскурс по самой среде:

Microsoft Office Access или просто Microsoft Access — реляционная система управления базами данных (СУБД) корпорации Microsoft. Входит в состав пакета Microsoft Office. Имеет широкий спектр функций, включая связанные запросы, связь с внешними таблицами и базами данных. Благодаря встроенному языку VBA, в самом Access можно писать приложения, работающие с базами данных.

Основные компоненты MS Access:

* построитель таблиц;
* построитель экранных форм;
* построитель SQL-запросов (язык SQL в MS Access не соответствует стандарту ANSI);
* построитель отчётов, выводимых на печать.

Они могут вызывать скрипты на языке VBA, поэтому MS Access позволяет разрабатывать приложения и БД практически «с нуля» или написать оболочку для внешней БД.

Microsoft Jet Database Engine, которая используется в качестве движка базы данных MS Access, является файл-серверной СУБД и потому применима лишь к приложениям, работающим с небольшими объёмами данных и при небольшом числе пользователей, одновременно работающих с этими данными. Непосредственно в Access отсутствует ряд механизмов, необходимых в многопользовательских базах данных, таких, например, как триггеры.

**1 Постановка задачи**

Требуется реализовать ИПК для работы директора кинотеатра, который должен быть ориентирован на пользователя-непрограммиста, имеющего навык работы в среде операционной системы Windows.

ИПК должен обеспечивать работу с данными о фильмах, о билетах, о сеансах, находящихся в прокате, об отчётах, о продаже билетов, о бронирование билетов, ведение отчётов. Диалог с пользователем, работа с базами данных комплекса (ввод, корректировка, удаление и просмотр записей), выполнение ряда стандартных запросов к базам данных, выпуск стандартных отчетов должен осуществляться с помощью разработанного интерфейса.

**2 Инфологическая и даталогическая модель**

2.1 Инфологическая модель

В результате инфологического проектирования БД должна быть создана инфологическая модель. Эта модель строится путём анализа и определения объектов и связей между ними в предметной области.

В инфологическом проектировании существуют функциональный и объектный подход. Функциональный подход реализует принцип “от задач”. Объектный подход не фиксирует количество решаемых задач, а включает в инфологическую модель только объекты и связи между ними. Смешанный подход объединяет предметный и функциональный.

Рассмотрим инфологическое проектирование для объектного подхода. Для описания инфологической модели используются диаграммы “объекты - связи” или, по-другому, ER - диаграммы.

Описание предметной области предполагает рассмотрение ее в качестве совокупности связанных реальных сущностей, которые имеют набор свойств или атрибутов, характеризующих только эту сущность. Группы всех подобных сущностей образуют объект, в котором фиксируются только значимые для данной ИС свойства. Связь представляет собой отношение между двумя объектами.

Объект должен иметь один или несколько ключевых атрибутов, то есть таких атрибутов, которые однозначно или уникально определяют экземпляр объекта среди всех других экземпляров объектов указанного типа.

Совокупность объектов и связей между ними определяет структуру предметной области, на основе которой строится инфологическая модель.

Каждому объекту можно сопоставить экземпляр объекта, а тот в свою очередь образуется совокупностью конкретных значений атрибутов, и должны однозначно определяться с помощью ключа.

Ключ - один или несколько атрибутов, которые однозначно определяют экземпляр объекта.

Cпецификация сущностей представлена в таблице 1.

Таблица 1 - Спецификация сущностей

|  |  |  |
| --- | --- | --- |
| Сущность | Идентификатор | Описательные атрибуты |
| Директор | Код директора | Название, Адрес, Телефон, ИНН |
| Кинотеатр | Код кинотеатра | Код директора, Название, Адрес, Телефон, Номер счета, ИНН |
| Фильм | Код фильма | Название, Длительность, Дата начала проката, Компания-прокатчик |
| Сеанс | Код сеанса | Зал, Дата, Время, Фильм |
| Зал | Код зала | Название, Количество мест, Описание, Число рядов, Число мест |
| Секторы зала | Код сектора | Зал, Название, Описание, Нач. ряд, Кон. ряд. |
| Места | Код места | Зал, № ряда, № места. |
| Билеты | Код билета | Дата выпуска, Сеансы, Место, Оплачено, Бронь, Уничтожен. |
| Движение билетов | Код движения | № билета, Дата, Операция, Сотрудник. |
| Сотрудники | Код сотрудника | ФИО, Должность, Пароль. |
| Должность | Код должности | Название должности. |
| Цены на билеты | Сеанс, Сектор | Цена. |

Cпецификация связей представлена в таблице 2.

Таблица 2 - Спецификация связей

|  |  |  |  |
| --- | --- | --- | --- |
| Тип связи | От сущности | К сущности | Имя связи |
| 1:M | Директор | Кинотеатр | Управляет |
| 1:M | Кинотеатр | Фильм | Показывает |
| 1:M | Фильм | Сеанс | Содержит |
| 1:M | Сеанс | Цены на билеты | Оплачивается |
| 1:M | Сеанс | Билеты | Продаётся по |
| 1:M | Зал | Сеанс | Включает в себя |
| 1:M | Зал | Места | Имеет |
| 1:M | Зал | Секторы зала | Имеет |
| 1:M | Секторы зала | Цены на билеты | Продаются исходя из |
| 1:M | Места | Билеты | Указаны |
| 1:M | Билеты | Движение билетов | Отслеживаются |
| 1:M | Сотрудники | Движение билетов | Оперирует |
| 1:M | Должность | Сотрудники | Имеет |

Инфологическая модель для данного ИПК представлена на рисунке 1.

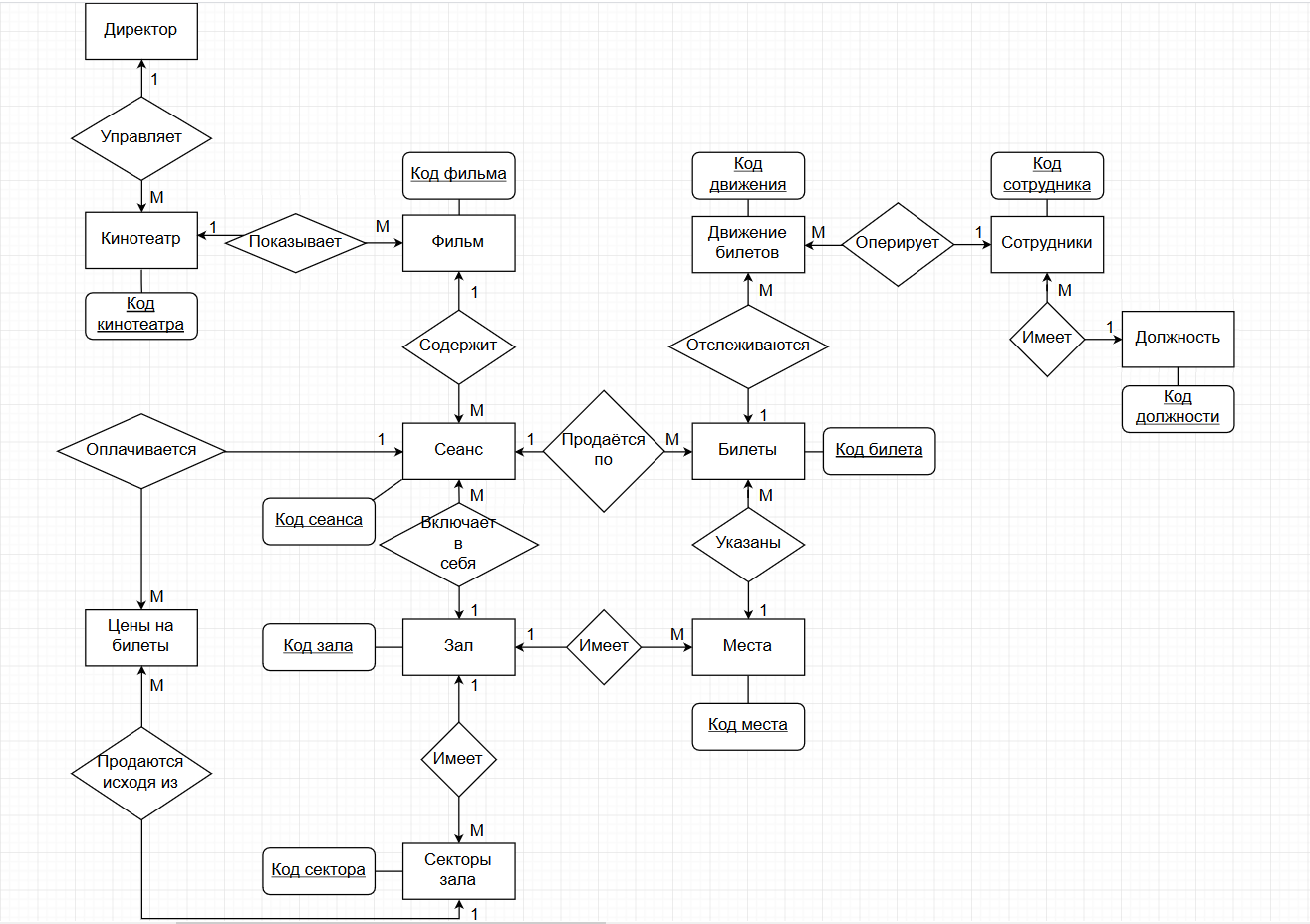


Рисунок 1 – Инфологическая схема

2.2 Даталогическая модель

Даталогическая модель готовой базы данных «Кинотеатр» отображает связи таблиц: Фильмы, Залы, Секторы зала, Места, Сеансы, Билеты, Цены на билеты, Движение билетов, Сотрудники, Должность. Даталогическая модель, разработанная в среде Access 2003, представлена на рисунке 2.

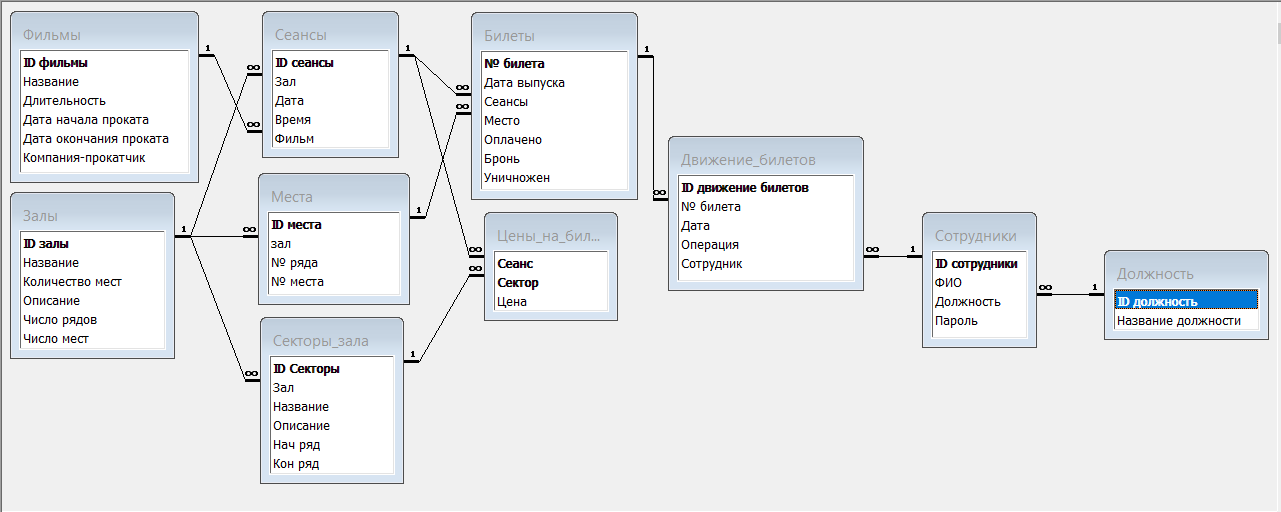


Рисунок 2 – Даталогическая схема

**3 Структуры таблиц**

Структуры таблиц представлены в таблице 3. В первом столбце таблицы указано название таблице. Во втором – наименование атрибута. В третьем столбце указывается тип атрибута. В четвертом – размер, в пятом – ограничения.

Таблица 3 – Структуры таблиц

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Таблица | Наименование | Тип | Размер | Ограничения |
| Билеты | 🔑 № билета | Счетчик | 8 bytes |  |
| Дата выпуска | Дата/время | 8 bytes |  |
| Сеансы | Числовой | 8 bytes |  |
| Место | Числовой | 8 bytes |  |
| Оплачено | Логический | 1 bit |  |
| Бронь | Логический | 1 bit |  |
| Уничтожен | Логический | 1 bit |  |
| Движение билетов | 🔑 ID движение билетов | Счетчик | 8 bytes |  |
| № билета | Числовой | 8 bytes |  |
| Дата | Дата/время | 8 bytes |  |
| Операция | Текстовый | 255 characters | <=15 |
| Сотрудник | Числовой | 8 bytes |  |
| Должность | 🔑 ID должность | Числовой | 8 bytes |  |
| Название должности | Текстовый | 255 characters | <=25 |
| Залы | 🔑 ID залы | Числовой | 8 bytes |  |
| Название | Текстовый | 255 characters | <=25 |
| Количество мест | Числовой |  |  |
| Описание | Текстовый | 255 characters | <=50 |
| Число рядов | Числовой | 4 bytes |  |
| Число мест | Числовой | 4 bytes |  |
| Сеансы | 🔑 ID сеансы | Счетчик | 8 bytes |  |
| Зал | Числовой | 8 bytes |  |
| Дата | Дата/время | 8 bytes |  |
| Время | Дата/время | 8 bytes |  |
| Фильм | Числовой | 8 bytes |  |
| Секторы зала | 🔑 ID Секторы | Числовой | 8 bytes |  |
| Зал | Числовой | 8 bytes |  |
| Название | Текстовый | 255 characters | <=20 |
| Описание | Текстовый | 255 characters | <=45 |
| Нач. ряд | Числовой | 4 bytes |  |
| Кон. ряд | Числовой | 4 bytes |  |
| Сотрудники | 🔑 ID сотрудники | Числовой | 8 bytes |  |
| ФИО | Текстовый | 255 characters | <=30 |
| Должность | Числовой | 8 bytes |  |
| Пароль | Текстовый | 255 characters | <=15 |
| Фильмы | 🔑 ID фильмы | Счетчик | 8 bytes |  |
| Название | Текстовый | 255 characters | <=50 |
| Длительность | Дата/время | 8 bytes |  |
| Дата начала проката | Дата/время | 8 bytes |  |
| Дата окончания проката | Дата/время | 8 bytes |  |
| Компания-прокатчик | Текстовый | 255 characters | <=30 |
| Места | 🔑 ID места | Числовой | 8 bytes |  |
| Зал | Числовой | 8 bytes |  |
| № ряда | Числовой | 4 bytes |  |
| № места | Числовой | 4 bytes |  |
| Цены на билеты | 🔑 Сеанс | Числовой | 8 bytes |  |
| 🔑 Сектор | Числовой | 8 bytes |  |
| Цена | Денежный | 8 bytes |  |

**4 Формы и их работа**

4.1 Главная форма готовой базы данных «Кинотеатр»

Данная форма представлена на рисунке 3.

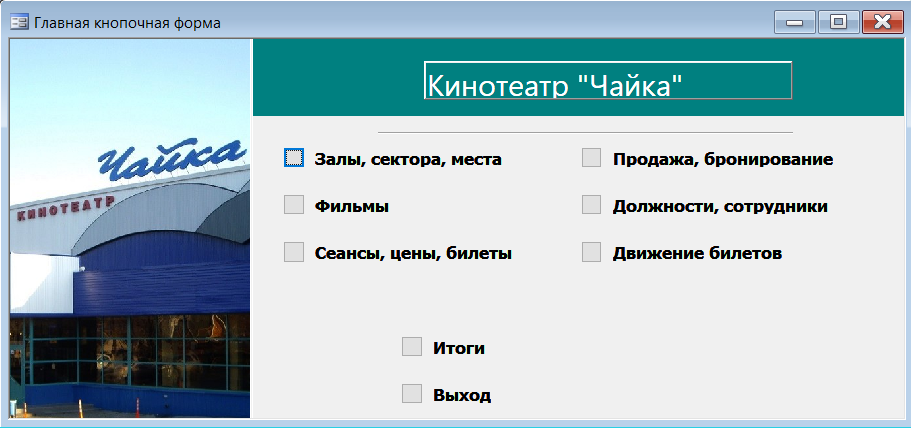


Рисунок 3 – Главная форма «Кинотеатр»

Страница «Итоги» главной кнопочной формы базы данных «Кинотеатр», представлена на рисунке 4.

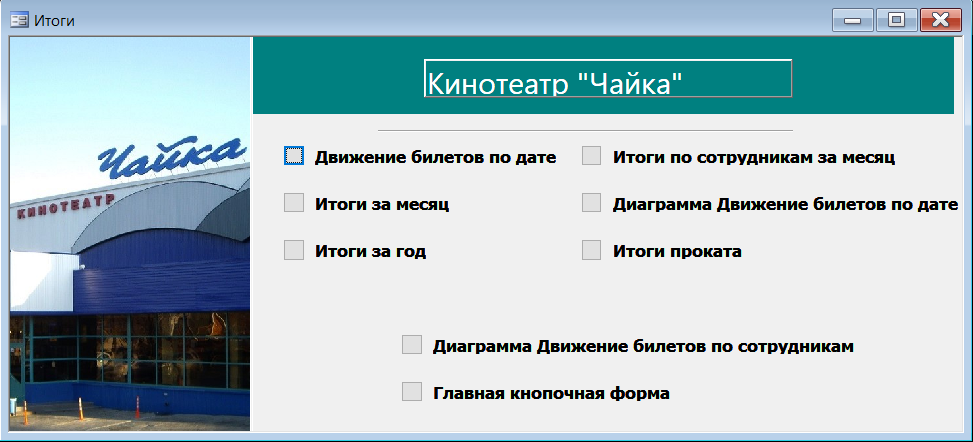


Рисунок 4 – Страница «Итоги»

4.2 Форма «Залы» с подчинёнными формами «Секторы зала» и «Места»

Данная форма представлена на рисунке 5.

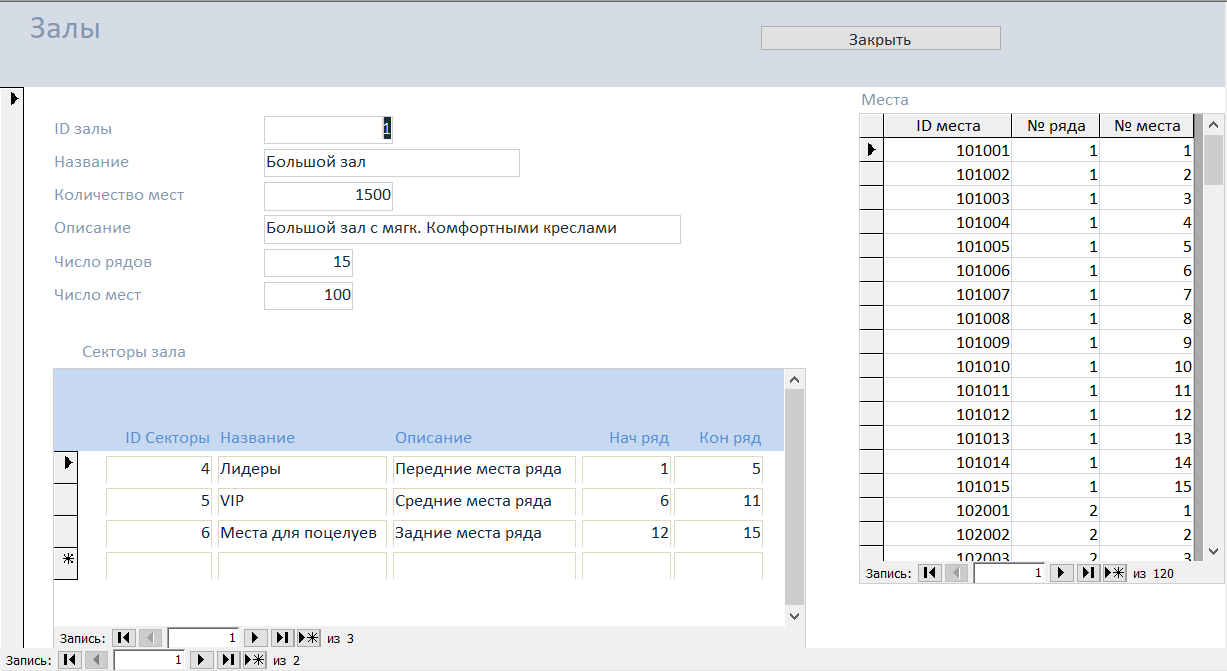


Рисунок 5 – Форма «Залы»

4.3 Форма «Фильмы»

Данная форма представлена на рисунке 6.

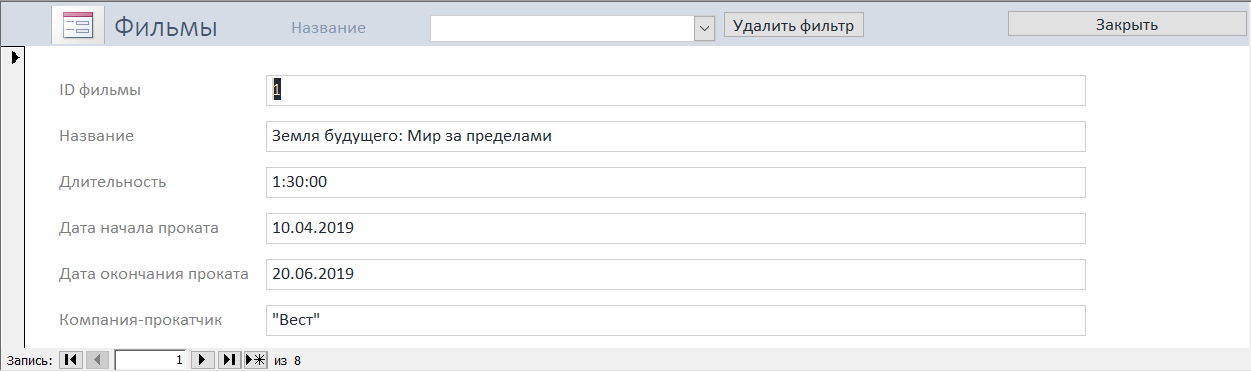


Рисунок 6 – Форма «Фильмы»

4.4 Форма «Сеансы» с подчинёнными формами «Билеты» и «Цены на билеты»

Данная форма представлена на рисунке 7.

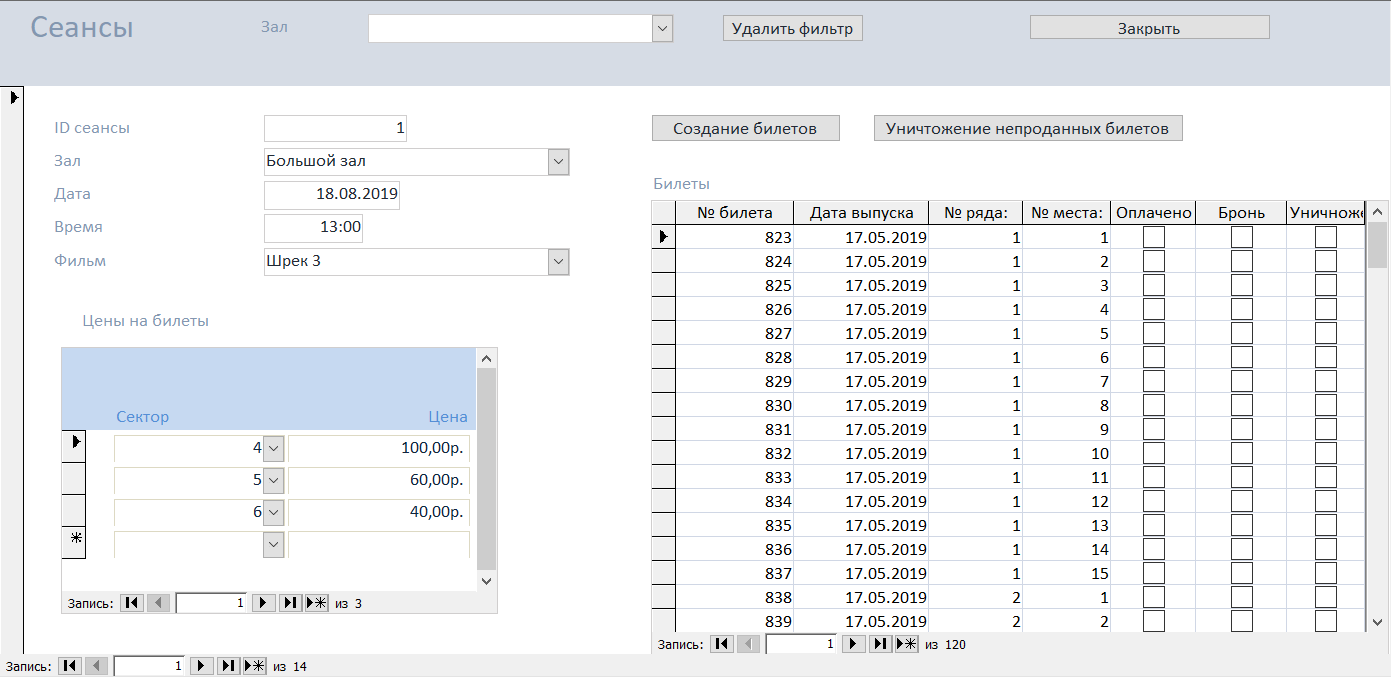


Рисунок 7 – Форма «Сеансы»

На данный сеанс можно автоматически создать билеты и уничтожить непроданные билеты.

4.5 Форма «Продажа бронирование»

Следующая форма представлена на рисунке 8 для всех операций (кроме уничтожения) с билетами:

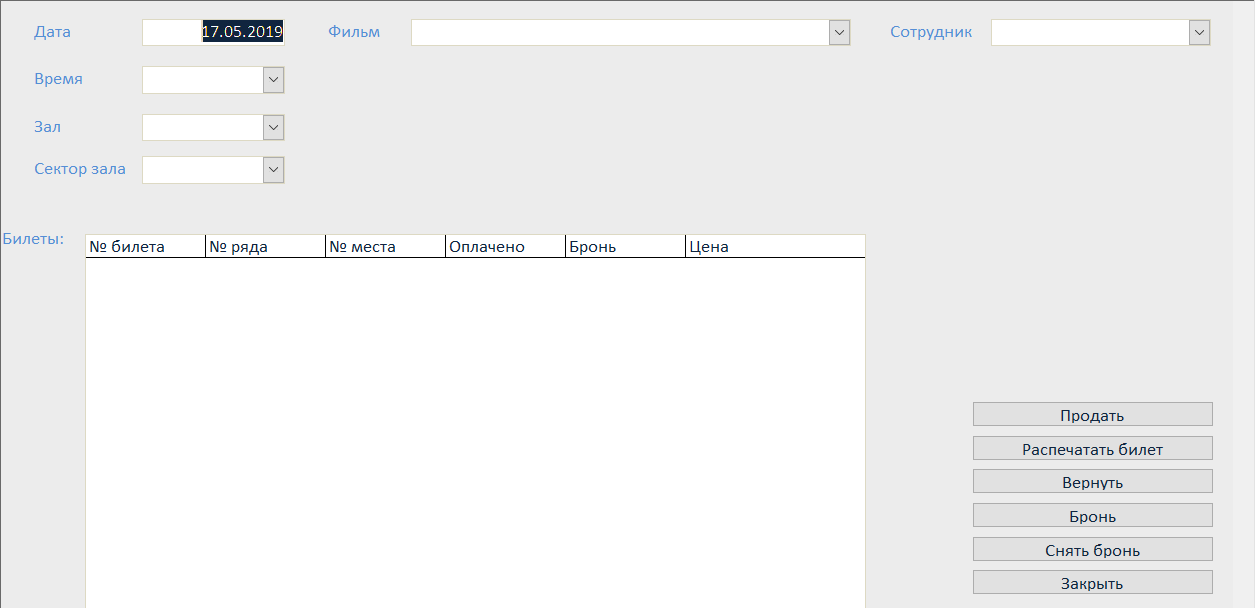


Рисунок 8 – Форма «Продажа бронирование»

После выбора даты можно выбрать фильм, прокат которого соответствует дате. После выбора фильма можно выбрать время, когда есть сеансы, потом зал и сектор зала. Появятся билеты, которые уже созданы, но не уничтожены, представлено на рисунке 9:

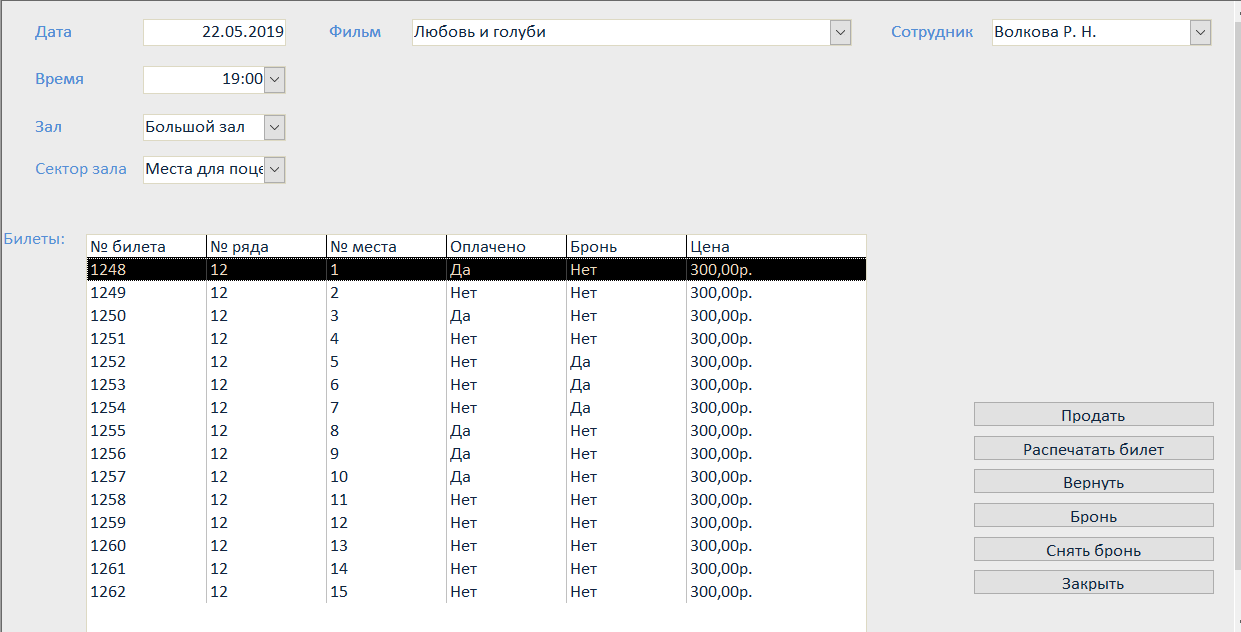


Рисунок 9 – Форма «Продажа бронирование»

Если билет не оплачен, его можно продать (запрос на добавление в «Движение билетов», запрос на обновление в «Билеты»), иначе сообщение, представлено на рисунке 10:

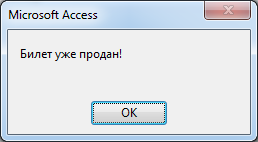


Рисунок 10 – Сообщение «Билет уже продан!»

Если билет не оплачен и нет брони, его можно забронировать (запрос на добавление в «Движение билетов», запрос на обновление в «Билеты»), иначе сообщение, представлено на рисунке 11:

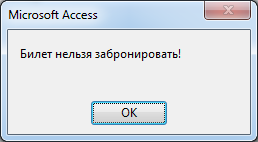


Рисунок 11 – Сообщение «Билет нельзя забронировать!»

Если билет оплачен, его можно вернуть (запрос на добавление в «Движение билетов», запрос на обновление в «Билеты»), иначе сообщение, представлено на рисунке 12:

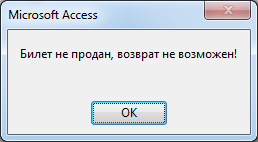


Рисунок 12 – Сообщение «Билет не продан, возврат не возможен!»

Если билет не оплачен и забронирован, можно снять бронь (запрос на добавление в «Движение билетов», запрос на обновление в «Билеты»), иначе сообщение, представлено на рисунке 13:

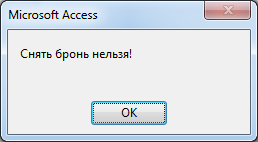
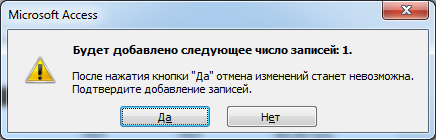


Рисунок 13 – Сообщение «Снять бронь нельзя!»

Если проверка прошла успешно, выполняются 2 запроса (нажимаем «да»), представлено на рисунке 14:



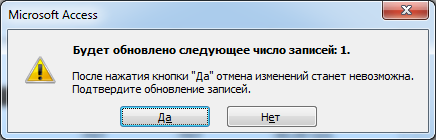


Рисунок 14 – Сообщение на запросы

Если билет не продан, распечатать его не получится, представлено на рисунке 15:

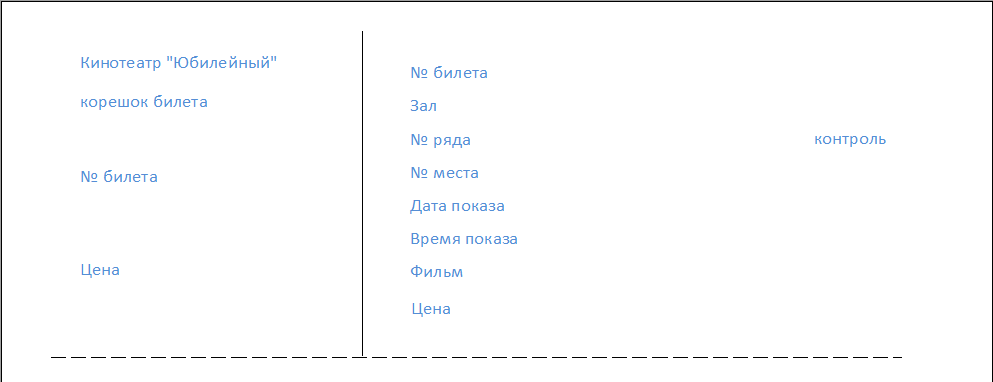


Рисунок 15 – Билет не продан

Если продан, представлено на рисунке 16:

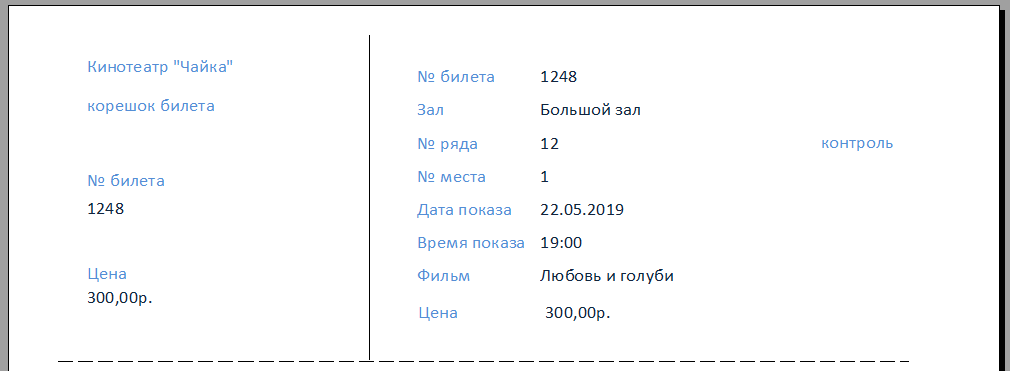


Рисунок 16 – Билет продан

4.6 Форма «Движение билетов»:

Данная форма представлена на рисунке 17.

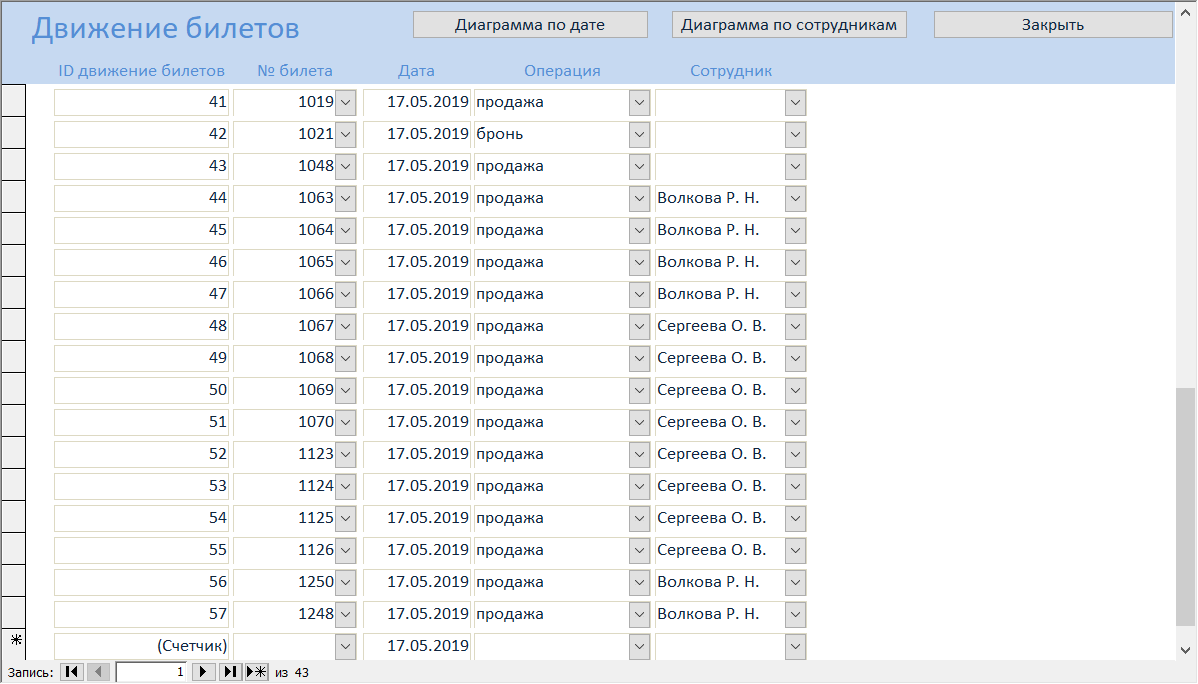


Рисунок 17 – Форма «Движение билетов»

4.7 Форма «Должность» с подчинённой формой «Сотрудники»

Данная форма представлена на рисунке 18.

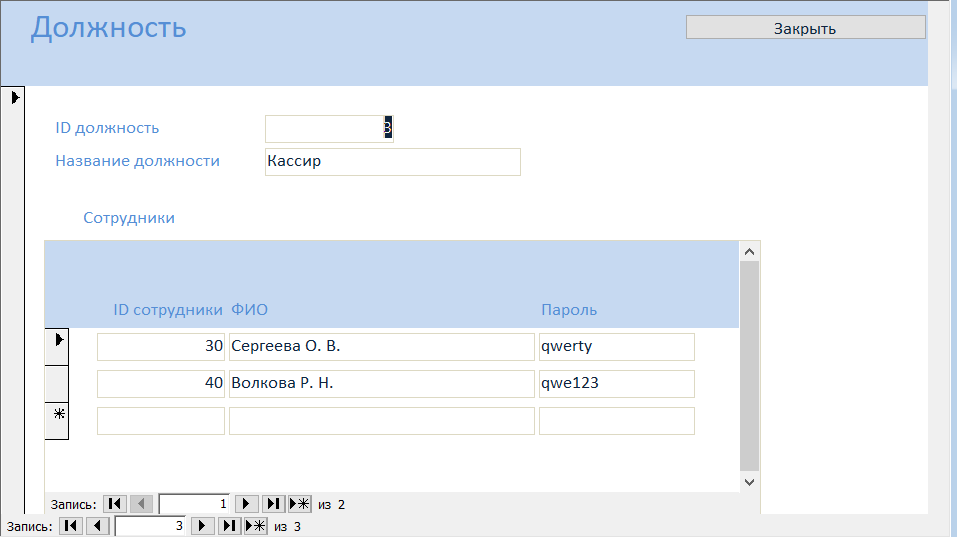


Рисунок 18 – Форма «Должность»

**5 Запросы (режим SQL)**

5.1 Запрос «Создание билетов»

INSERT INTO Билеты ( [Дата выпуска], Сеансы, Место )

SELECT Date() AS Выражение1, Сеансы.[ID сеансы], Места.[ID места]

FROM (Места LEFT JOIN Сеансы ON Места.зал = Сеансы.Зал) LEFT JOIN [Кол билетов на сеанс] ON Сеансы.[ID сеансы] = [Кол билетов на сеанс].Сеансы

WHERE (((Сеансы.[ID сеансы])=[Формы]![Сеансы]![ID сеансы]) AND (NZ([Кол билетов на сеанс].[Count-№ билета])=0));

5.2 Запрос «Итоги за год»

SELECT Залы.Название AS Зал, Фильмы.Название AS Фильм, Month([Сеансы.Дата]) AS Месяц, Year([Сеансы.Дата]) AS Год, Движение\_билетов.Операция, Count(Движение\_билетов.[№ билета]) AS [Кол билетов], Sum(IIf([Движение\_билетов].[Операция]="продажа",[Цены\_на\_билеты].[Цена],IIf([Движение\_билетов].[Операция]="возврат",-[Цены\_на\_билеты].[Цена],0))) AS Стоимость, MonthName(Month([Сеансы.Дата])) AS [Название месяца]

FROM Фильмы INNER JOIN ((Залы INNER JOIN Сеансы ON Залы.[ID залы] = Сеансы.Зал) INNER JOIN ((([Сектор билета] INNER JOIN Билеты ON [Сектор билета].[№ билета] = Билеты.[№ билета]) INNER JOIN Цены\_на\_билеты ON [Сектор билета].[ID Секторы] = Цены\_на\_билеты.Сектор) INNER JOIN Движение\_билетов ON Билеты.[№ билета] = Движение\_билетов.[№ билета]) ON (Сеансы.[ID сеансы] = Цены\_на\_билеты.Сеанс) AND (Сеансы.[ID сеансы] = Билеты.Сеансы)) ON Фильмы.[ID фильмы] = Сеансы.Фильм

GROUP BY Залы.Название, Фильмы.Название, Month([Сеансы.Дата]), Year([Сеансы.Дата]), Движение\_билетов.Операция, MonthName(Month([Сеансы.Дата]))

HAVING (((Year([Сеансы.Дата]))=[Введите год]) AND ((Движение\_билетов.Операция)="продажа" Or (Движение\_билетов.Операция)="возврат"));

5.3 Запрос «Движение билетов по дате»

SELECT Движение\_билетов.Дата, Движение\_билетов.Операция, Движение\_билетов.[№ билета], IIf([Движение\_билетов].[Операция]="продажа",[Цены\_на\_билеты].[Цена],IIf([Движение\_билетов].[Операция]="возврат",-[Цены\_на\_билеты].[Цена],0)) AS Выражение1

FROM Сеансы INNER JOIN ((([Сектор билета] INNER JOIN Цены\_на\_билеты ON [Сектор билета].[ID Секторы] = Цены\_на\_билеты.Сектор) INNER JOIN Билеты ON [Сектор билета].[№ билета] = Билеты.[№ билета]) INNER JOIN Движение\_билетов ON Билеты.[№ билета] = Движение\_билетов.[№ билета]) ON (Сеансы.[ID сеансы] = Цены\_на\_билеты.Сеанс) AND (Сеансы.[ID сеансы] = Билеты.Сеансы)

WHERE (((Движение\_билетов.Дата)=[Введите дату]));

5.4 Запрос «Добавление брони»

INSERT INTO Движение\_билетов ( [№ билета], Дата, Операция, Сотрудник )

SELECT Билеты.[№ билета], Date() AS Выражение2, "бронь" AS Выражение3, [Формы]![Продажа бронирование]![ПолеСоСписком24] AS Выражение4

FROM Билеты

WHERE (((Билеты.[№ билета])=[Формы]![Продажа бронирование]![Список10]) AND ((Билеты.Бронь)=No) AND ((Билеты.Оплачено)=No));

5.5 Запрос «Добавление возврата»

INSERT INTO Движение\_билетов ( [№ билета], Дата, Операция, Сотрудник )

SELECT Билеты.[№ билета], Date() AS Выражение2, "возврат" AS Выражение3, [Формы]![Продажа бронирование]![ПолеСоСписком24] AS Выражение4

FROM Билеты

WHERE (((Билеты.[№ билета])=[Формы]![Продажа бронирование]![Список10]) AND ((Билеты.Оплачено)=Yes));

5.6 Запрос «Добавление продажи»

INSERT INTO Движение\_билетов ( [№ билета], Дата, Операция, Сотрудник )

SELECT Билеты.[№ билета], Date() AS Выражение2, "продажа" AS Выражение3, [Формы]![Продажа бронирование]![ПолеСоСписком24] AS Выражение4

FROM Билеты

WHERE (((Билеты.[№ билета])=[Формы]![Продажа бронирование]![Список10]) AND ((Билеты.Оплачено)=No));

5.7 Запрос «Обновление брони»

UPDATE Билеты SET Билеты.Бронь = Yes

WHERE (((Билеты.Бронь)=No) AND ((Билеты.Оплачено)=No) AND ((Билеты.[№ билета])=[Формы]![Продажа бронирование]![Список10]));

5.8 Запрос «Обновление оплачено»

UPDATE Билеты SET Билеты.Оплачено = Yes

WHERE (((Билеты.Оплачено)=No) AND ((Билеты.[№ билета])=[Формы]![Продажа бронирование]![Список10]));

5.9 Запрос «Обновление не оплачено»

UPDATE Билеты SET Билеты.Оплачено = No

WHERE (((Билеты.Оплачено)=Yes) AND ((Билеты.[№ билета])=[Формы]![Продажа бронирование]![Список10]));

**6 Отчёты**

6.1 Отчёт «Движение билетов по дате»

Данный отчёт представлен на рисунке 19.

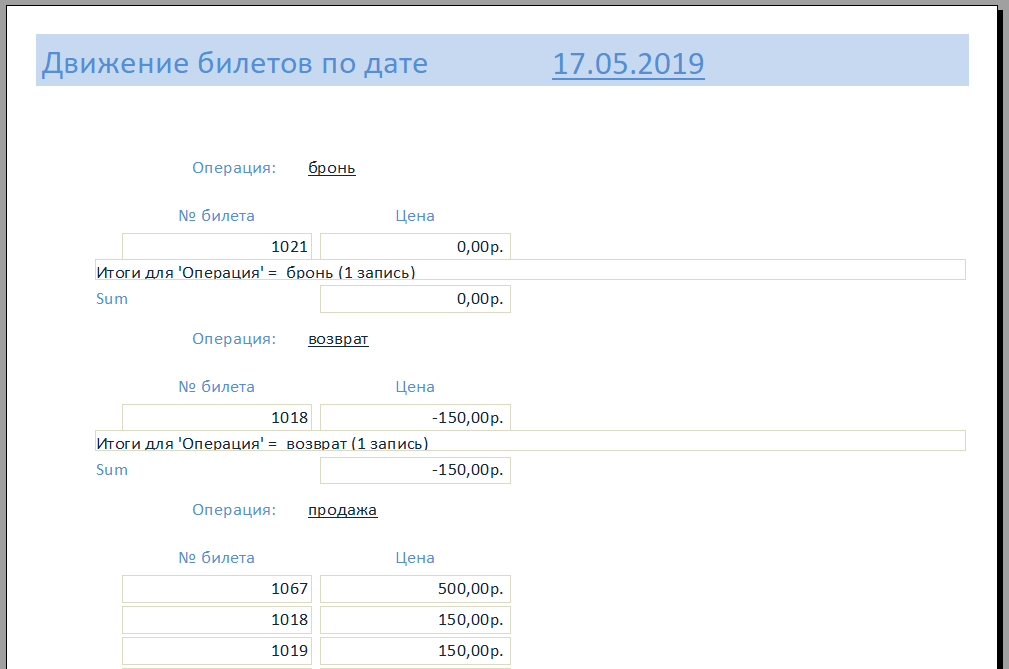


Рисунок 19 – Отчёт «Движение билетов по дате»

6.2 Отчёт «Итоги за месяц»

Данный отчёт представлен на рисунке 20.

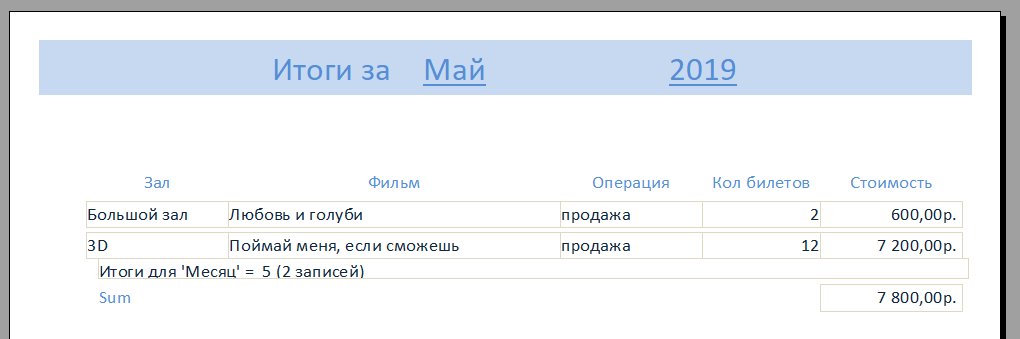


Рисунок 20 – Отчёт «Итоги за месяц»

6.3 Отчёт «Итоги за год»

Данный отчёт представлен на рисунке 21.



Рисунок 21 – Отчёт «Итоги за год»

6.4 Отчёт «Итоги по сотрудникам за месяц»

Данный отчёт представлен на рисунке 22.



Рисунок 22 – Отчёт «Итоги по сотрудникам за месяц»

6.5 Отчёт «Итоги проката»

Данный отчёт представлен на рисунке 23.

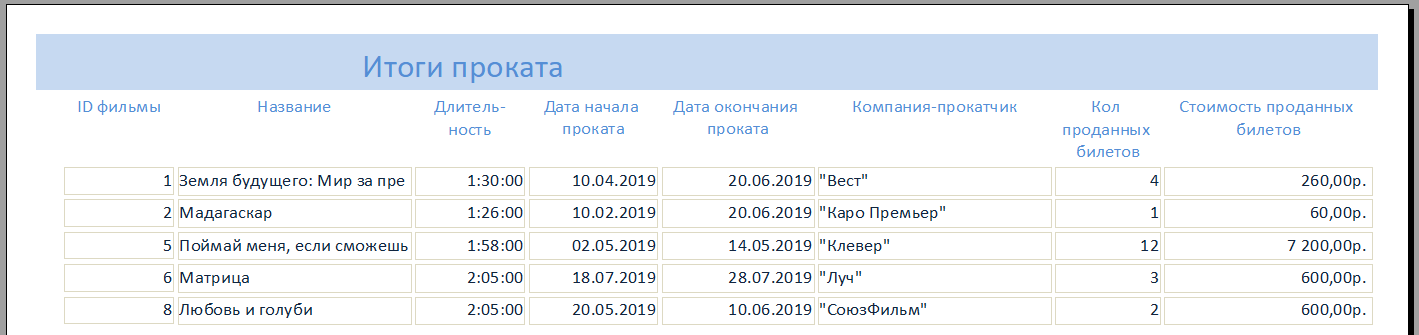


Рисунок 23 – Отчёт «Итоги проката»

ЗАКЛЮЧЕНИЕ

В результате курсового проекта достигнуты следующе цели:

1. Был реализован ИПК, который ориентирован на пользователя-непрограммиста, имеющего навык работы в среде операционной системы Windows. ИПК был спроектирован с использованием инфологического и даталогического моделирования. В ходе инфологического моделирования в инфологическую модель были включены все необходимые типы сущностей, которые нужны для функционирования кинотеатра. В ходе даталогического моделирования был разработан ИПК работы директора кинотеатра, который автоматизирует и облегчит работу директора.
2. База данных была сделана в среде MS ACCESS 2003, которая обратно совместима с новыми версиями MS ACCESS (2007-2016(диаграммы работать не будут)).
3. В базе данных присутствуют формы, отчёты, макросы, таблицы, запросы и диаграммы.
4. Сделана база данных, которая демонстрирует возможности среды MS ACESS, а также основные принципы построения баз данных в целом.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Голицына, О. Л. Базы данных / О.Л. Голицына, Н.В. Максимов, И.И. Попов. - М.: Форум, 2015. - 400 c.

2. Илюшечкин, В. М. Основы использования и проектирования баз данных / В.М. Илюшечкин. - М.: Юрайт, Юрайт, 2013. - 224 c.

3. Карпова, И. П. Базы данных / И.П. Карпова. - М.: Питер, 2013. - 240 c.

4. Кириллов, В.В. Введение в реляционные базы данных (+ CD-ROM) / В.В. Кириллов. - М.: БХВ-Петербург, 2016. - 318 c.

5. Кузнецов, С. Д. Базы данных. Модели и языки / С.Д. Кузнецов. - М.: Бином-Пресс, 2013. - 720 c.

6. Кузнецов, С. Д. Основы баз данных / С.Д. Кузнецов. - М.: Бином. Лаборатория знаний, Интернет-университет информационных технологий, 2017. - 488 c.

7. Латыпова, Р. Р. Базы данных. Курс лекций / Р.Р. Латыпова. - Москва: Высшая школа, 2016. - 177 c.

8. Мартишин, С. А. Базы данных. Практическое примечание СУБД SQL и NoSOL. Учебное пособие / С.А. Мартишин, В.Л. Симонов, М.В. Храпченко. - М.: Форум, Инфра-М, 2016. - 368 c.

9. Миркин, Б. Г. Введение в анализ данных. Учебник и практикум / Б.Г. Миркин. - М.: Юрайт, 2015. - 176 c.

ПРИЛОЖЕНИЕ

Диск CD-RW (700MB) с отчётом и с самой базой данных. Исходные тексты макросов и обработчики кнопок на формах можно посмотреть во встроенном редакторе Visual Basic.