|  |
| --- |
| **МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ** |
| **федеральное государственное бюджетное образовательное учреждение высшего образования «Российский экономический университет имени Г.В. Плеханова»**  **Московский приборостроительный техникум** |

**ОТЧЕТ**

по учебной практике

УП.06.01 «Учебная практика»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Профессионального модуля ПМ.06 «Сопровождение информационных систем»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Специальность 09.02.07 «Информационные системы и программирование»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Квалификация «Специалист по информационным системам»\_\_\_\_\_\_\_\_\_

|  |  |  |  |
| --- | --- | --- | --- |
| Студент |  |  | Шатохин Роман Евгеньевич |
|  | *подпись* |  | *фамилия, имя, отчество* |
| Группа | ИС50-1-20 |  |  |

Руководитель по практической подготовке от техникума

|  |  |  |
| --- | --- | --- |
|  |  | Павлова Анастасия Витальевна |
| *подпись* |  | *фамилия, имя, отчество* |
|  |  | Жданцев Кирилл Николаевич |
| *подпись* |  | *фамилия, имя, отчество* |

«\_\_\_\_» \_\_\_\_\_\_\_\_\_ 202\_ года

Оглавление

[Практическая №1 «Анализ предметной области» 3](#_Toc102512581)

[Практическая работа №2 «Создание бизнес-процессов IDEF0» 9](#_Toc102512582)

[Практическая работа №3 «Создание бизнес-процессов IDEF3» 15](#_Toc102512583)

[Практическая работа №4 «Создание бизнес-процессов DFD» 21](#_Toc102512584)

[Практическая работа №5 «Нормализация БД» 27](#_Toc102512585)

[Практическая №6 «ER-модель» 31](#_Toc102512586)

[Практическая №7 «Создание базы данных в MS Access» 42](#_Toc102512587)

[Практическая №8 «Запросы и отчеты в Access» 49](#_Toc102512588)

[Практическая работа №9 «Генерация БД. Создание резервной копии» 57](#_Toc102512589)

[Практическая работа №10 «Создание информационной системы. Устранение ошибок в программном коде» 63](#_Toc102512590)

# Практическая №1 «Анализ предметной области»

Цель работы: произвести анализ предметной области на тему «Кинотеатр».

**Теоретические сведения к выполнению практической работы:**

*Предметная область* – часть реального мира, которая подлежит изучению и дальнейшем интерпретированию в базе данных.

Под предметной областью в частном случае понимают следующее: некоторое предприятие, выполняющее ту или иную деятельность.

Примеры предметных областей:

• Учебное заведение;

• Автомобильный салон;

• Торговая точка;

• Банк и т.д.

**Целью** этапа анализа является преобразование общих, расплывчатых знаний об исходной предметной области (требований заказчика) в точные определения и спецификации для разработчиков, а также генерация функционального описания системы.

*Подсистема* – часть предметной области, направленная на выполнение более конкретного круга задач.

Пример подсистем:

• Отдел кадров;

• Отдел бухгалтерии;

• Продажа товара;

• Учёт товара на складах;

• Обслуживание клиентов и т.д.

*Информационный объект* – часть подсистемы предметной области, который хранит и передаёт информацию о том или ином объекте, событии, действии предметной области.

Примеры информационного объекта:

• Персонал;

• Должность;

• Оплата;

• Заказ и т.д.

**Этапы выполнения:**

1. **Описание предметной области**

1.1. Наименование организации – кинотеатр «КАРО», видом предприятия является отдых и развлечение.

1.2. Род деятельности организации – предоставление услуг кинопоказа, предоставление услуг досуга и развлечения.

1.3. Круг потребителей – кругом потребителей считаются люди любого возраста и любого пола, но чаще всего посещают круг лиц от 16 до 30 лет.

1.4. Выпуск продукции – готовой продукцией будет являться показ кинофильма.

1.5. Нормативные документы:

* Внутренние документы – правила внутреннего трудового распорядка, должностные инструкции, устав предприятия, штатное расписание;
* Внешние документы – Законом Российской Федерации "О защите прав потребителей", Федеральным законом "О защите детей от информации, причиняющей вред их здоровью и развитию".

1.6. Организационная структура предприятия - состав, взаимосвязь подразделений и отдельных должностей, управляющих предприятием. (создать с помощью smartart)

Рисунок 1 - Диаграмма структуры предприятия

2. **Описание подсистем**

2.1. Наименование подсистем – для данной предметной области используется 3 подсистемы:

* Составление расписание;
* Покупка билета;
* Работа бара.

2.2. Описать роль подсистемы в предметной области и её функциональные задачи:

Составление расписания – Администратор составляет расписание на определённый день работы кинотеатра, где будет указываться в каком зале будет идти какой фильм и во сколько начала сеанса.

Покупка билета – клиент приходит в кинотеатр или через сторонний сервис, выбирает место и оформляет покупку.

Работа бара – клиент перед походом на сеанс кинофильма может зайти в кино бар и выбрать себе снеки и напитки для более приятного просмотра фильма.

2.3. Определить информационные объекты необходимые для работы подсистемы:

1. **Составление расписания;**
   1. Залы;
   2. Фильмы;
   3. Расписание;
   4. Администратор;
2. **Процесс покупки билета;**
   1. Кассир;
   2. Список свободных мест;
   3. Скидочная карта (если имеется);
3. **Процесс покупки кино бара;**
   1. Продавец;
   2. Ассортимент товара;
   3. Скидочная карта (если имеется).

**3. Описание информационных объектов**

3.1. Описать назначение объекта в рамках подсистемы –

* Залы – Номер зала, Количество мест в зале;
* Фильм – Название, Длительность;
* Расписание – Длительность с учётом рекламы;
* Администратор – ФИО, дата рождения, Должность;
* Кассир – ФИО, дата рождения, должность;
* Список свободных мест – Номер зала, номер места, ряд места;
* Скидочная карта – Срок действия, номер карты, процент скидки, сумма для улучшения процента скидки;
* Продавец - ФИО, дата рождения, должность;
* Ассортимент – Наименования, размеры, цена;
* Скидочная карта – Срок действия, номер карты, процент скидки, сумма для улучшения процента скидки.

3.2. Описать взаимосвязь информационных объектов с подсистемами– в табличном виде расположить информационные объекты с подсистемами.

Таблица 1 - Взаимосвязи информационных объектов

|  |  |  |
| --- | --- | --- |
| Составление расписание | Покупка билета | Покупка бара |
| Залы | Скидочная карта | |
| Фильмы | Список свободных мест | Ассортимент |
| Сотрудник | | |
| Расписание | - | - |

**Вывод**

В ходе работы был произведён анализ предметной области на тему «Кинотеатр»

# Практическая работа №2 «Создание бизнес-процессов IDEF0»

Цель работы: создать бизнес - процессы с помощью методологии IDEF0, произвести декомпозицию до 3 уровня.

**Теоретические сведения к выполнению практической работы:**

Этапы жизненного цикла:

Анализ – определение потребностей клиента и специфики предметной области;

Проектирование – логическая разработка спецификаций программного продукта;

Реализация – физическая реализация логических спецификаций;

Тестирование – выявление не корректной работы программного обеспечения;

Внедрение – передача программного продукта заказчику и введение в эксплуатацию.

SADT (Structured Analysis and Design Technique) – метод структурного анализа и проектирования. Данная методология направлена на определение и выявление:

* Основных функций проектируемой программной системы;
* Составные части программной системы;
* Позволяет в виде функциональной диаграммы отображать взаимосвязанные функции.

На основе методологии SADT была разработана методология:

IDEF0 — методология функционального моделирования (англ. function modeling) и графическая нотация, предназначенная для формализации и описания бизнес-процессов.

Направлена на выполнение основных задач:

* Пояснение к функциям предметной области;
* Определение объектов предметной области;
* Определение механизмов выполнения той или иной функции.

В IDEF0 выделяют главные составляющие:

* Блок имени функций – определяет процесс, действие, функцию, которая выполняется в предметной области, подсистеме или в процессе;
* Исходные данные (Входные данные) – данные об объектах, людях, событиях, которые подлежат обработке процессом и дальнейшем преобразованием в результат;
* Управление – объекты стандартов, правил, нормативных документов предметной области, обеспечивающие выполнение процессов/функций;
* Механизм – объекты за счёт чего выполняются процессы, штатные или структурные единицы, предприятия, некое оборудование или физические механизмы;
* Результат (Выходные данные) – объект, который является продуктом исходных данных, за счёт выполнения функции некими механизмами на основе объектов управления.

В анализе предметной области декомпозиция понимается следующим образом: чем больше уровней присутствует в бизнес – моделировании по методологии SADT стандарта IDEF 0, тем проще будет проходить протекание проектирования и реализация программного продукта.

Процесс «углубления» в каждый родительский процесс именуется – декомпозицией.

Уровень декомпозиции – детализирование/разбиение одного процесса на более детальные подпроцессы.

Этапы выполнения:

1. Определение протекающих бизнес-процессов в предметной области.

1.2. Используя инструментальное средство BP-Win составить схемы бизнес-процессов. Все бизнес-процессы должны быть взаимосвязаны между собой и вытекать один из другого.

1.3. Отобразить общую схему предметной области и описать все элементы

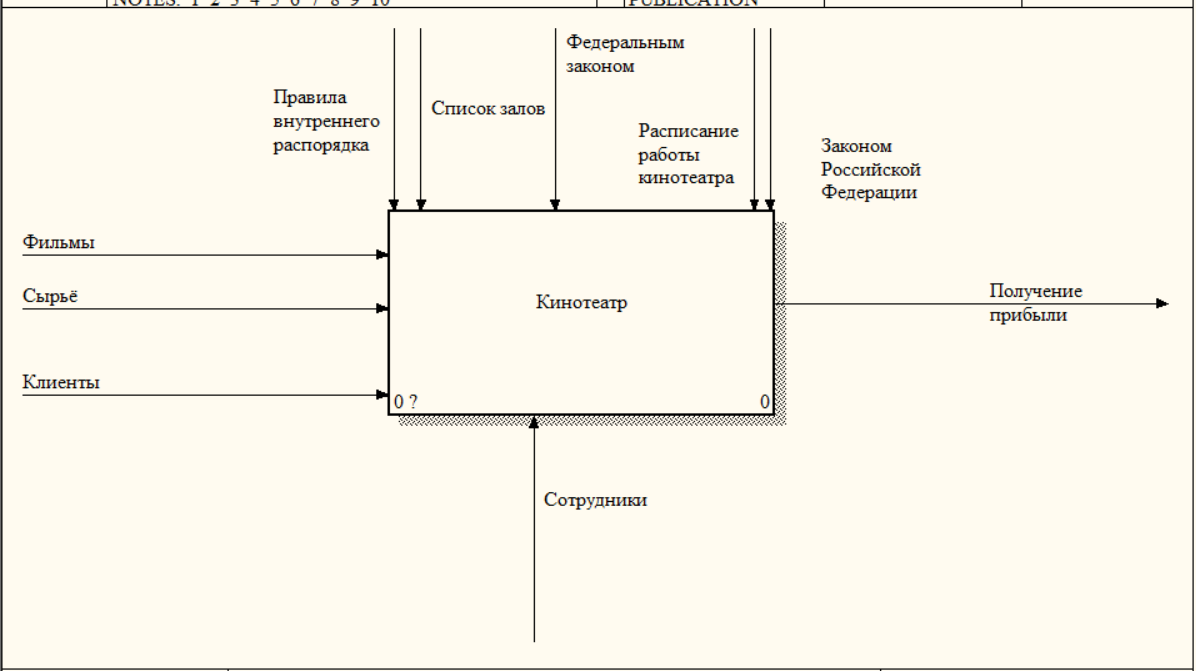


Рисунок 2 - Главный блок

Входными данные: Фильм, Сырьё. Клиенты;

Выходными данные: Получение прибыли;

Механизмы: Зал и Сотрудники;

Управление: Правила внутреннего распорядка, Список залов, Федеральный закон, Расписание работы кинотеатра, Закон Российской Федерации;

А главным блоком выступает: Кинотеатр.

1.4. Отобразить схему подсистем

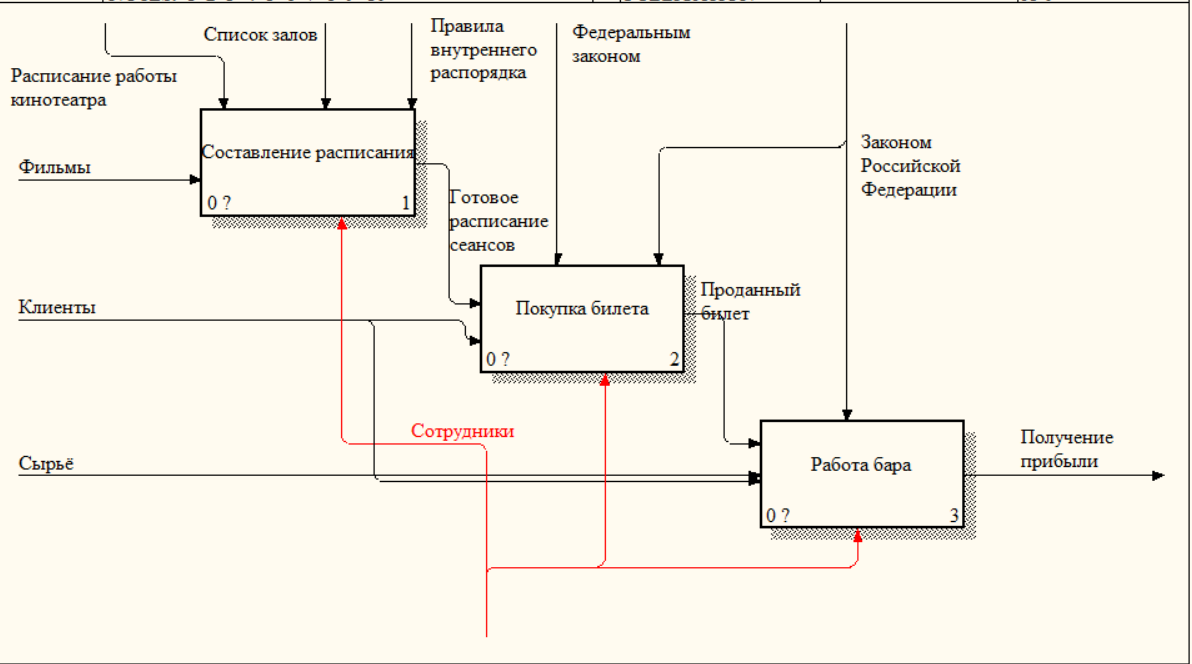


Рисунок 3 - Декомпозиция главного блока

1.5. Отобразить схему бизнес-процессов подсистем

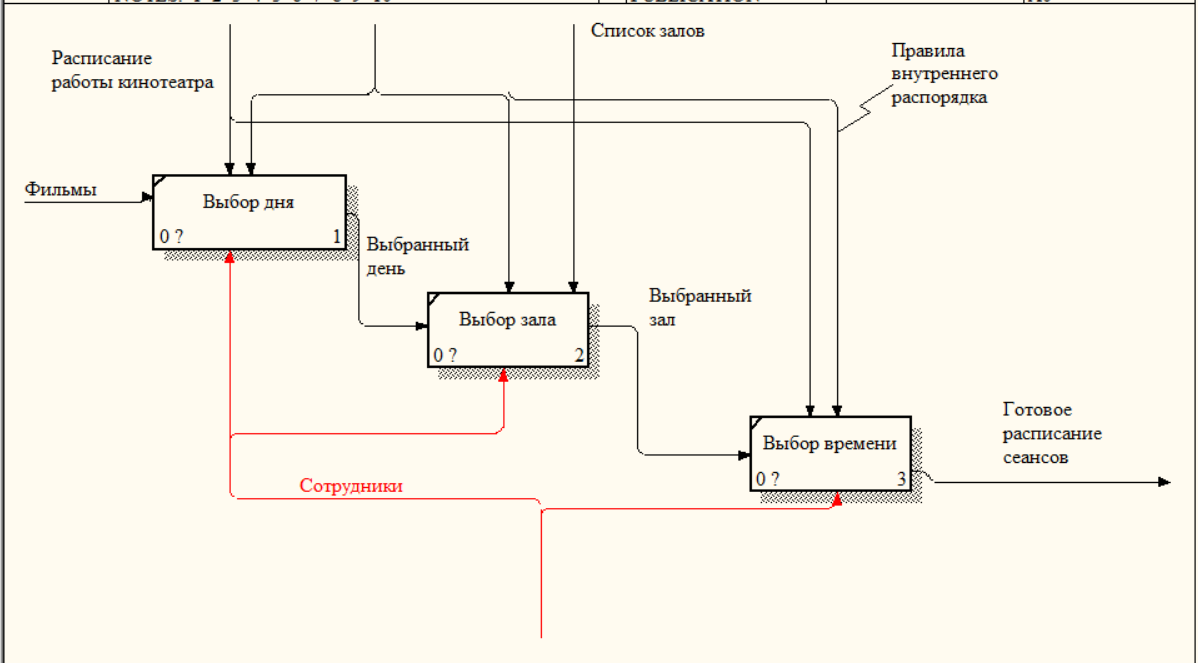


Рисунок 4 - Декомпозиция первого блока

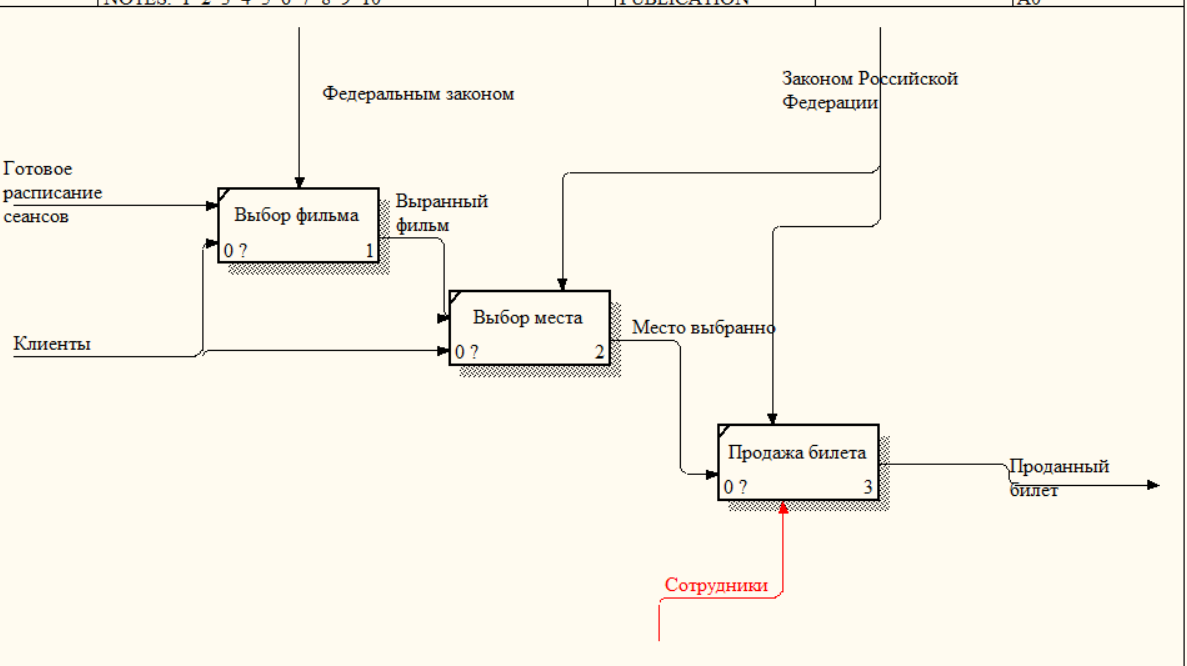


Рисунок 5 - Декомпозиция второго блока

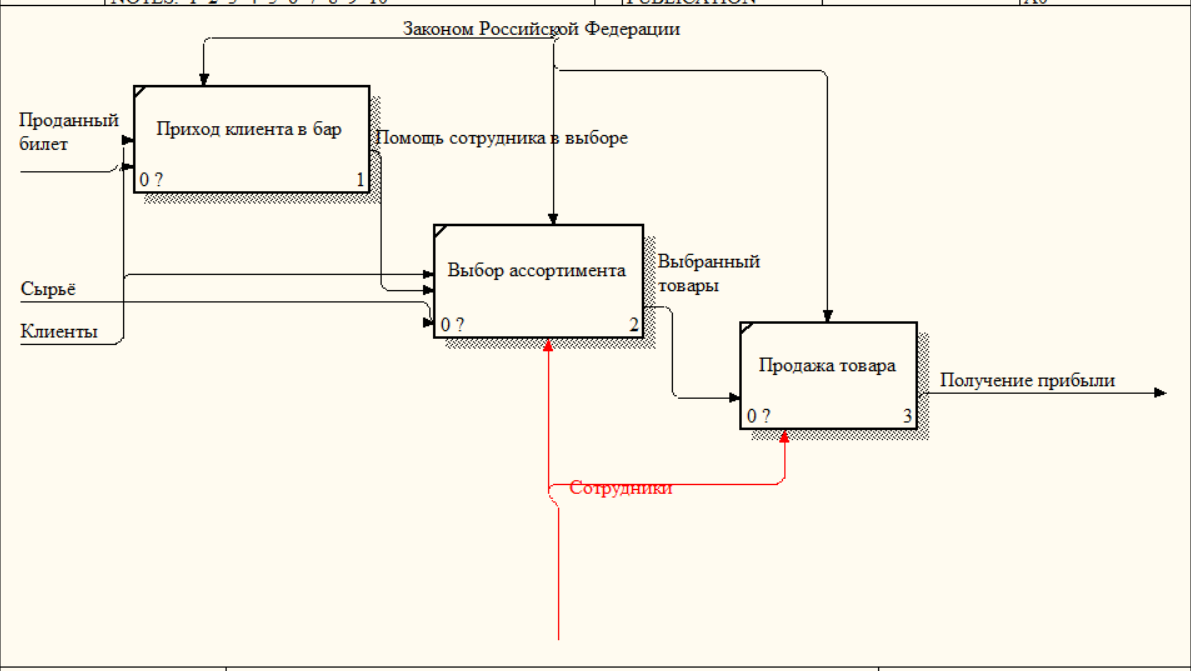


Рисунок 6 - Декомпозиция третьего блока

**Вывод**

В ходе работы был создан бизнес-процессы с помощью методологии IDEF0.

# Практическая работа №3 «Создание бизнес-процессов IDEF3»

Цель работы: создать бизнес-процессы с помощью методологии IDEF3.

**Теоретические сведения к выполнению практической работы:**

Этапы жизненного цикла:

Анализ – определение потребностей клиента и специфики предметной области;

Проектирование – логическая разработка спецификаций программного продукта;

Реализация – физическая реализация логических спецификаций;

Тестирование – выявление не корректной работы программного обеспечения;

Внедрение – передача программного продукта заказчику и введение в эксплуатацию.

SADT (Structured Analysis and Design Technique) – метод структурного анализа и проектирования. Данная методология направлена на определение и выявление:

* Основных функций проектируемой программной системы;
* Составные части программной системы;
* Позволяет в виде функциональной диаграммы отображать взаимосвязанные функции.

Процесс «углубления» в каждый родительский процесс именуется – декомпозицией.

Уровень декомпозиции – детализирование/разбиение одного процесса на более детальные подпроцессы.

Этапы выполнения:

1. Определение протекающих бизнес-процессов в предметной области.

* Составление расписания;
* Покупка билета;
* Работа бара.

1.2. Используя инструментальное средство BP-Win составить схемы бизнес-процессов. Все бизнес-процессы должны быть взаимосвязаны между собой и вытекать один из другого.

1.3. Отобразить общую схему предметной области и описать все элементы.

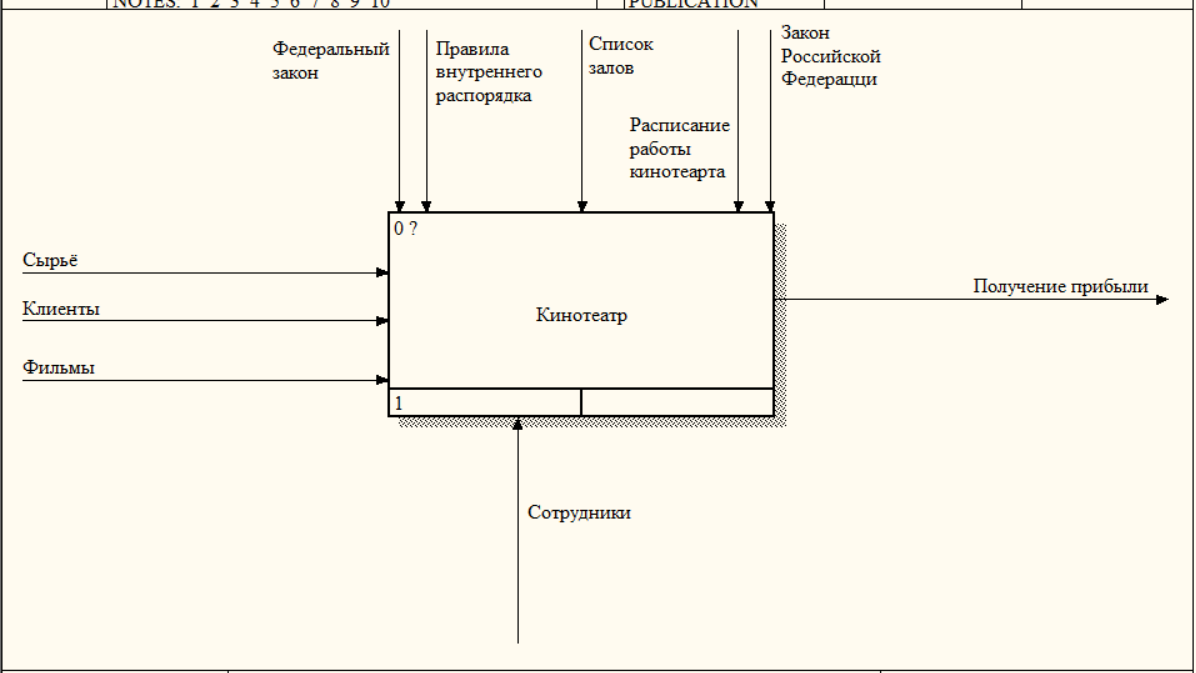


Рисунок 7 - Главный блок

Входными данные: Фильм, Сырьё. Клиенты;

Выходными данные: Получение прибыли;

Механизмы: Сотрудники;

Управление: Правила внутреннего распорядка, Список залов, Закон Российской Федерации "О защите прав потребителей", Федеральный закон "О защите детей от информации, причиняющей вред их здоровью и развитию", Расписание работы кинотеатра;

Главным блоком выступает: Кинотеатр.

1.4. Отобразить схему подсистем

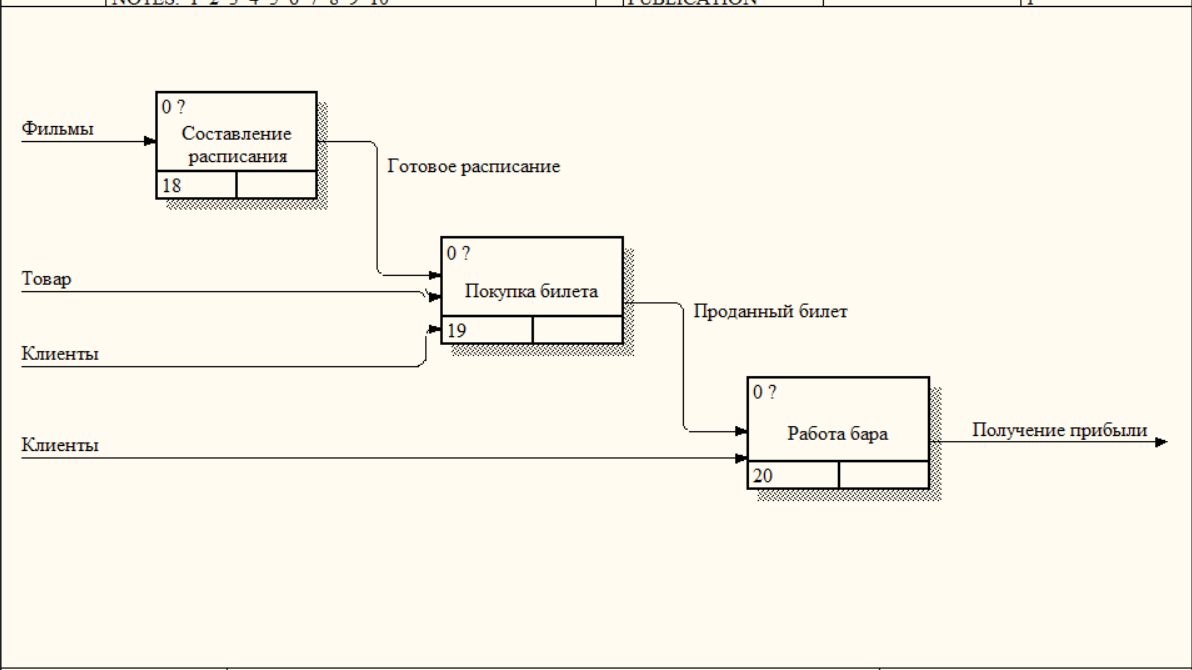


Рисунок 8 - Декомпозиция главного блока

На данной схеме описывается декомпозиция главного блок «Кинотеатр» на 3 подсистемы: Составление расписания, Покупка билета, Работа бара. К каждой из этих подсистем входят входные данные и из последней подсистемы выходит итог.

1.5. Отобразить схему бизнес-процессов подсистем

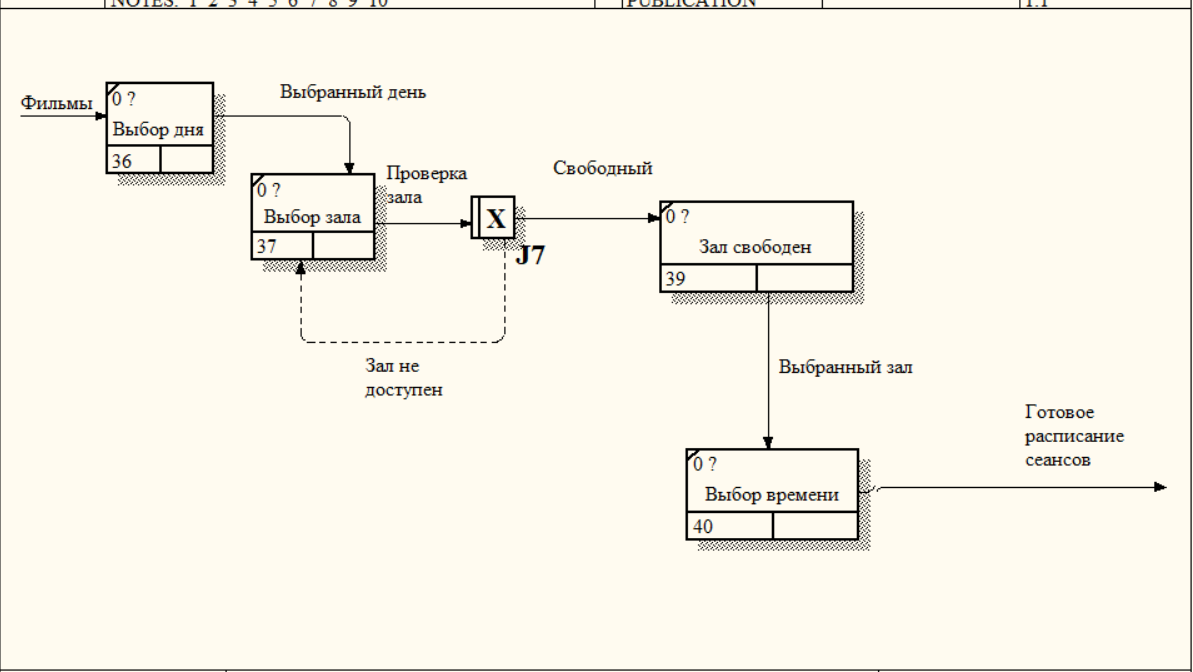


Рисунок 9 - Декомпозиция первого блока

На рис.9 описывается первая подсистема «Составление расписания». На ней сначала выбирают день, затем выбирают в котором будет показываться фильм. Если зал свободен, то дальше выбирают день, а если зал не доступен, то выбирают новый. После выбора времени в итоге получается готовое расписание сеансов.

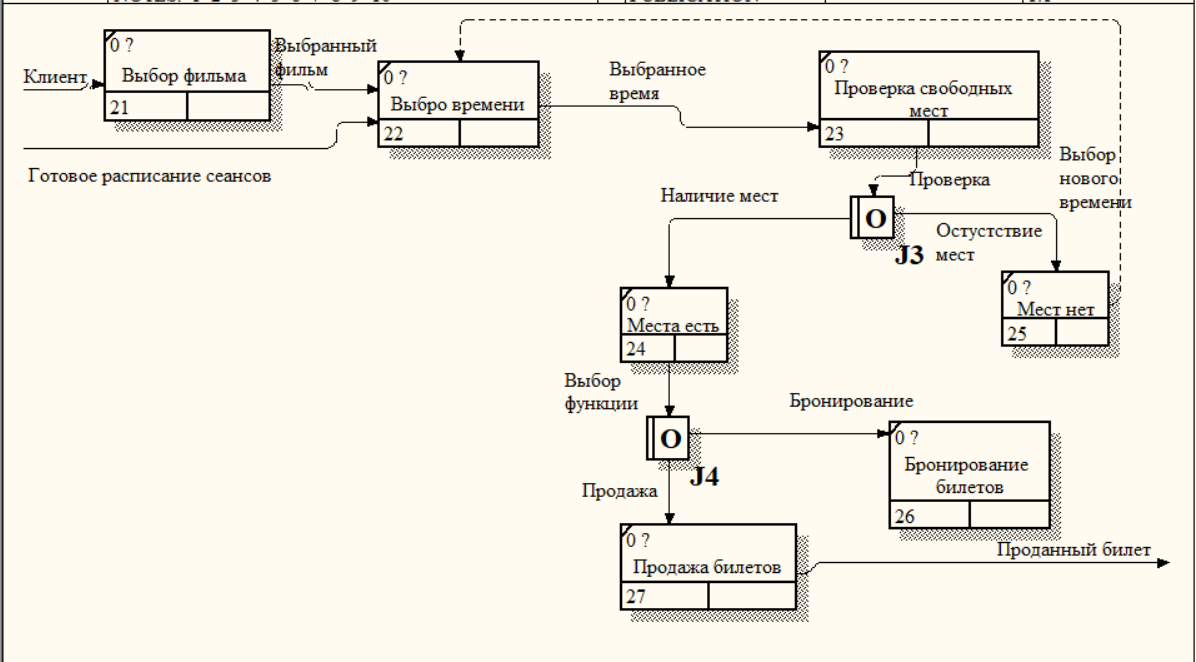


Рисунок 10 - Декомпозиция второго блока

На рис.10 происходит декомпозиция второго блока «Покупка билета». На ней клиент выбирает фильм и время. Затем проверяется есть ли свободные места на данное время. Если мест нет, то клиенту надо будет выбрать новое время на сеанс. А если места есть, то происходит выбор между бронированием билета или продажа билета.

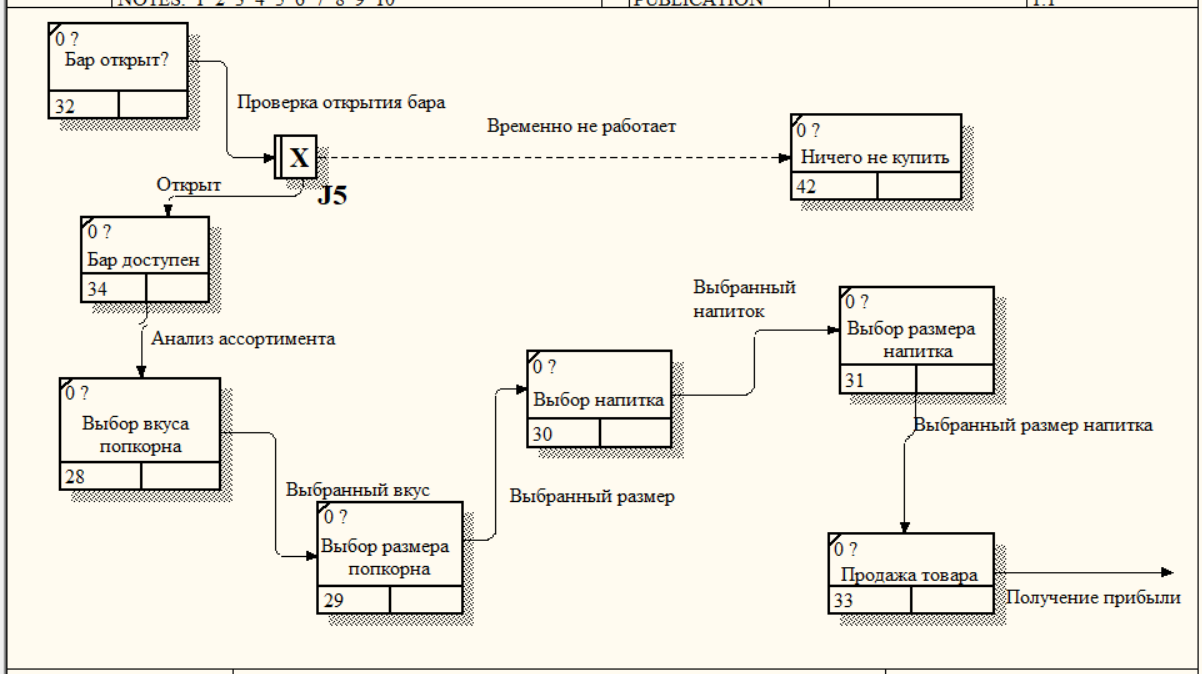


Рисунок 11 - Декомпозиция третьего блока

На рис.11 описывается третья подсистема «Работа бара». Для начала проверяется работает ли бар или нет. Если бар работает, то клиент выбирает вкус попкорна затем размер попкорна. Так же происходит и с напитком сначала какой напиток, а потом размер. И после этого происходит продажа товара и в итоге компания получает прибыль.

**Вывод**

В ходе работы был создан бизнес-процессы с помощью методологии IDEF3.

# Практическая работа №4 «Создание бизнес-процессов DFD»

Цель работы: создать бизнес-процессы с помощью методологии DFD.

**Теоретические сведения к выполнению практической работы:**

Этапы жизненного цикла:

Анализ – определение потребностей клиента и специфики предметной области;

Проектирование – логическая разработка спецификаций программного продукта;

Реализация – физическая реализация логических спецификаций;

Тестирование – выявление не корректной работы программного обеспечения;

Внедрение – передача программного продукта заказчику и введение в эксплуатацию.

SADT (Structured Analysis and Design Technique) – метод структурного анализа и проектирования. Данная методология направлена на определение и выявление:

* Основных функций проектируемой программной системы;
* Составные части программной системы;
* Позволяет в виде функциональной диаграммы отображать взаимосвязанные функции.

Процесс «углубления» в каждый родительский процесс именуется – декомпозицией.

Уровень декомпозиции – детализирование/разбиение одного процесса на более детальные подпроцессы.

Этапы выполнения:

1. Определение протекающих бизнес-процессов в предметной области.

* Составление расписания;
* Покупка билета;
* Работа бара.

1.2. Используя инструментальное средство BP-Win составить схемы бизнес-процессов. Все бизнес-процессы должны быть взаимосвязаны между собой и вытекать один из другого.

1.3. Отобразить общую схему предметной области и описать все элементы.

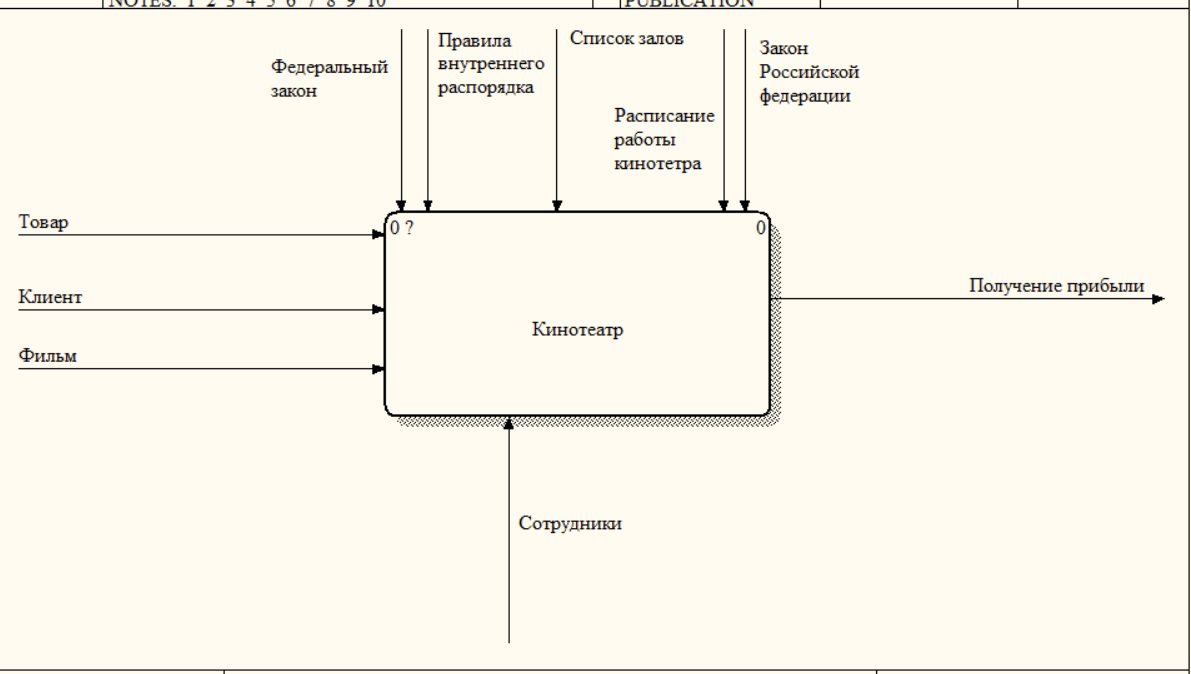


Рисунок 12 - Главный блок

Входными данные: Фильм, Сырьё. Клиенты;

Выходными данные: Получение прибыли;

Механизмы: Сотрудники;

Управление: Правила внутреннего распорядка, Список залов, Закон Российской Федерации "О защите прав потребителей", Федеральный закон "О защите детей от информации, причиняющей вред их здоровью и развитию", Расписание работы кинотеатра;

Главным блоком выступает: Кинотеатр.

1.4. Отобразить схему подсистем

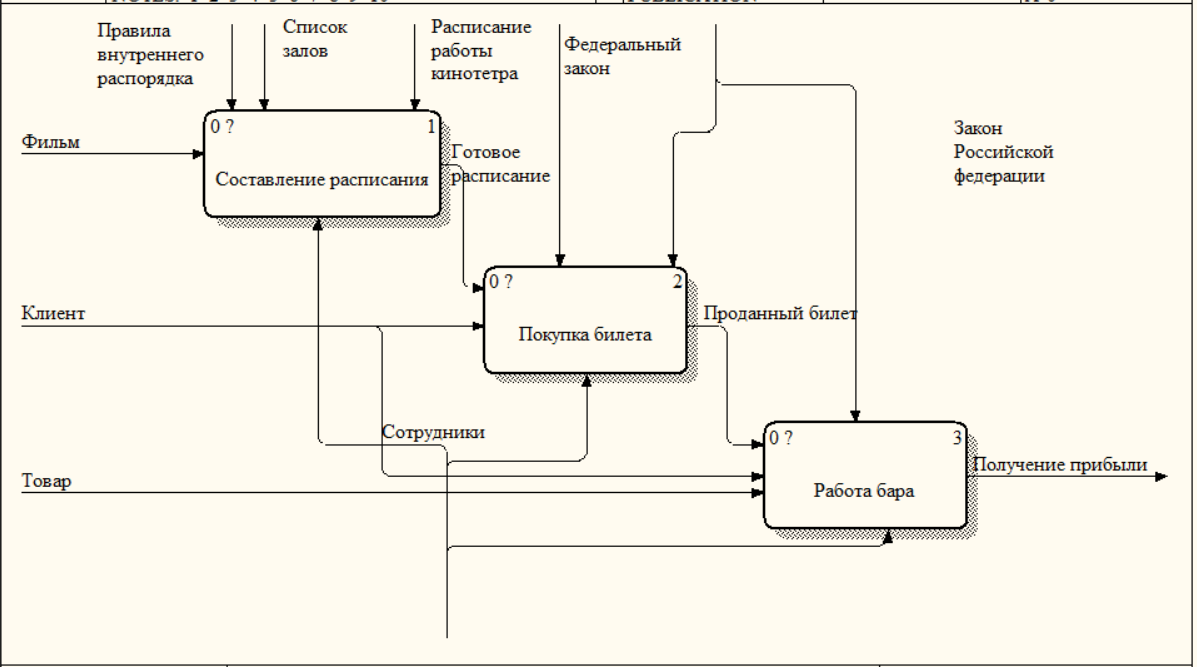


Рисунок 13 - Декомпозиция главного блока

На данной схеме описывается декомпозиция главного блок «Кинотеатр» на 3 подсистемы: Составление расписания, Покупка билета, Работа бара. К каждой из этих подсистем входят входные данные и из последней подсистемы выходит итог.

1.5. Отобразить схему бизнес-процессов подсистем

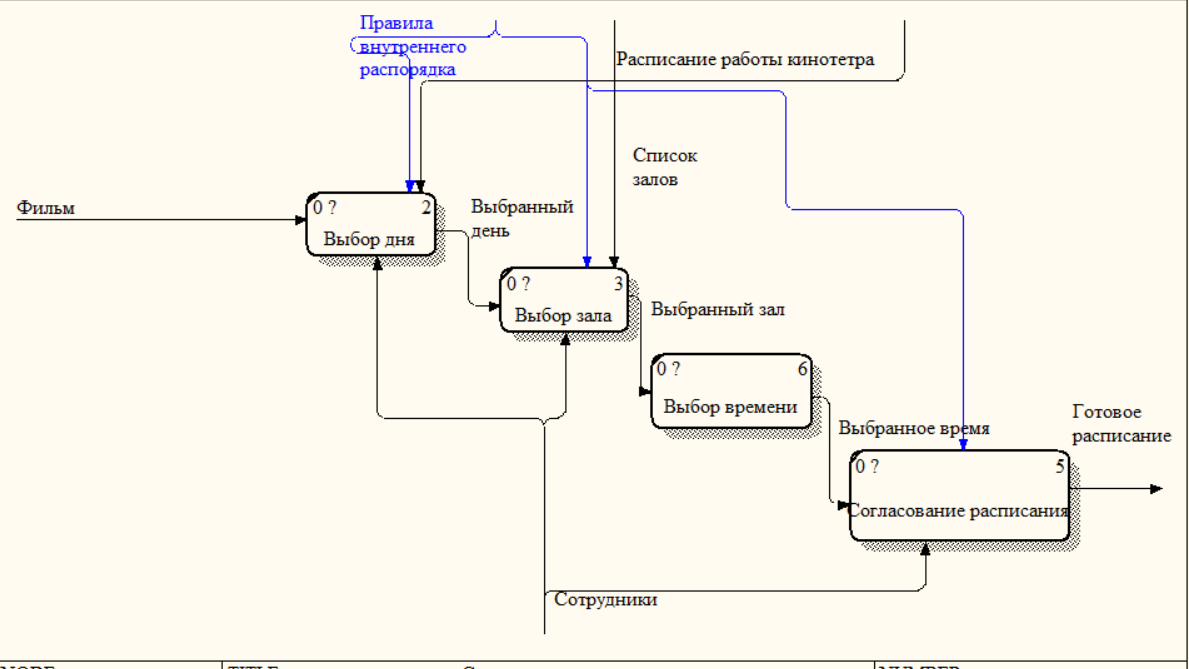


Рисунок 14 - Декомпозиция первого блока

На рис.14 описывается первая подсистема «Составление расписания». На ней сначала выбирают день, затем выбирают в котором будет показываться фильм. После выбирают зал, в котором будет показываться фильм. После выбора времени, согласовывают расписание. И в итоге получается готовое расписание сеансов.

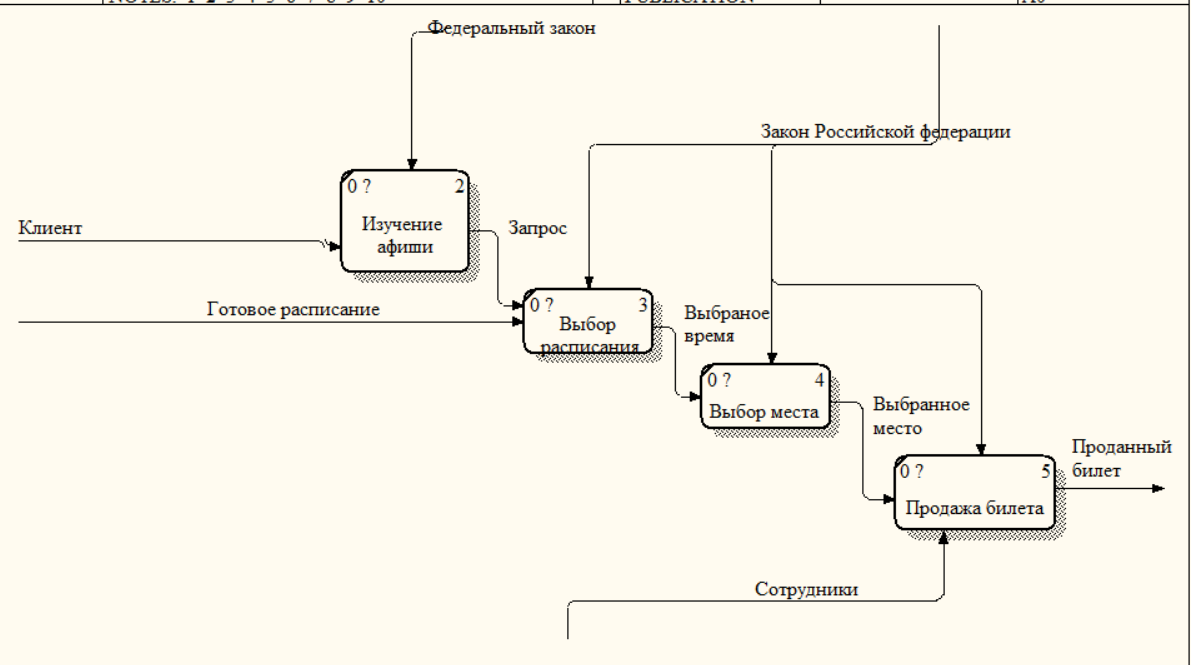


Рисунок 15 - Декомпозиция второго блока

На рис.15 происходит декомпозиция второго блока «Покупка билета». На ней клиент изучает афишу и выбирает время. Затем происходит выбор места на данное время. После идёт продажа билета.

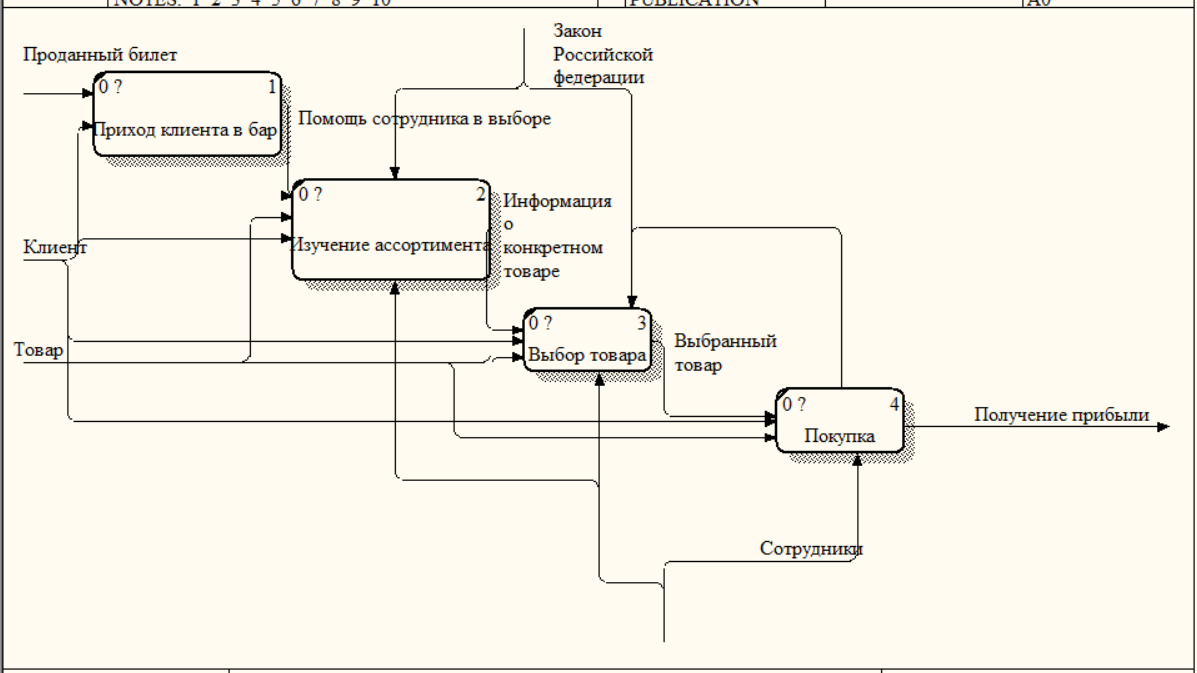


Рисунок 16 - Декомпозиция третьего блока

На рис.16 описывается третья подсистема «Работа бара». Для начала клиент приходит в бар и изучает ассортимент. Затем он выбирает товар, который хочет купить. И после этого происходит продажа товара и в итоге компания получает прибыль.

**Вывод**

В ходе работы был создан бизнес-процессы с помощью методологии DFD.

# Практическая работа №5 «Нормализация БД»

Цель работы: определить и описать объекты предметной области, провести нормализацию бд.

**Теоретическая часть:**

База данных — это структурированная хранилище данных которая присуща определённой предметной области.

Адекватность данных — это определенный уровень соответствия, создаваемого с помощью полученной информации образа реальному объекту, процессу, явлению и т.п.

Достоверность данных — это степень соответствия данных об объектах в БД реальным значениям свойств объектов в данный момент времени, определяющиеся изменениями самих объектов, некорректностями записей об их состоянии или некорректностями расчетов их характеристик.

Целостность данных — это поддержание и обеспечение точности и согласованности данных на протяжении всего их жизненного цикла. Это важнейший аспект разработки, внедрения и использования любой системы, которая хранит, обрабатывает или извлекает данные.

Первичный ключ — это поле или набор полей со значениями, которые являются уникальными для всей таблицы.

Внешний ключ (FK) — это столбец или сочетание столбцов, которое применяется для принудительного установления связи между данными в двух таблицах с целью контроля данных, которые могут храниться в таблице внешнего ключа.

1 к ∞ (один ко многим) – когда один элемент одной таблицы принадлежит множеству элементов другой таблице. (Чаще всего используется)

1 клиент в множество заказов

1 к 1 (один к одному) – когда один элементу одной таблицы принадлежит одному элементу другой таблицы. (Редко используется)

1Vim-номер к 1 авто

∞ к ∞ (множество ко множеству) - когда множество элементов одной таблицы принадлежит множеству элементов другой таблицы. (Физически не реализуемо). Чтобы реализовать эту таблицу физически необходимо добавить ещё одну таблицу что бы через неё были связаны таблицы

Множество голубей едят множество хлеба.

Этап выполнения:

1. Создать нормализацию 1НФ + заполнить таблицы по 4 записи

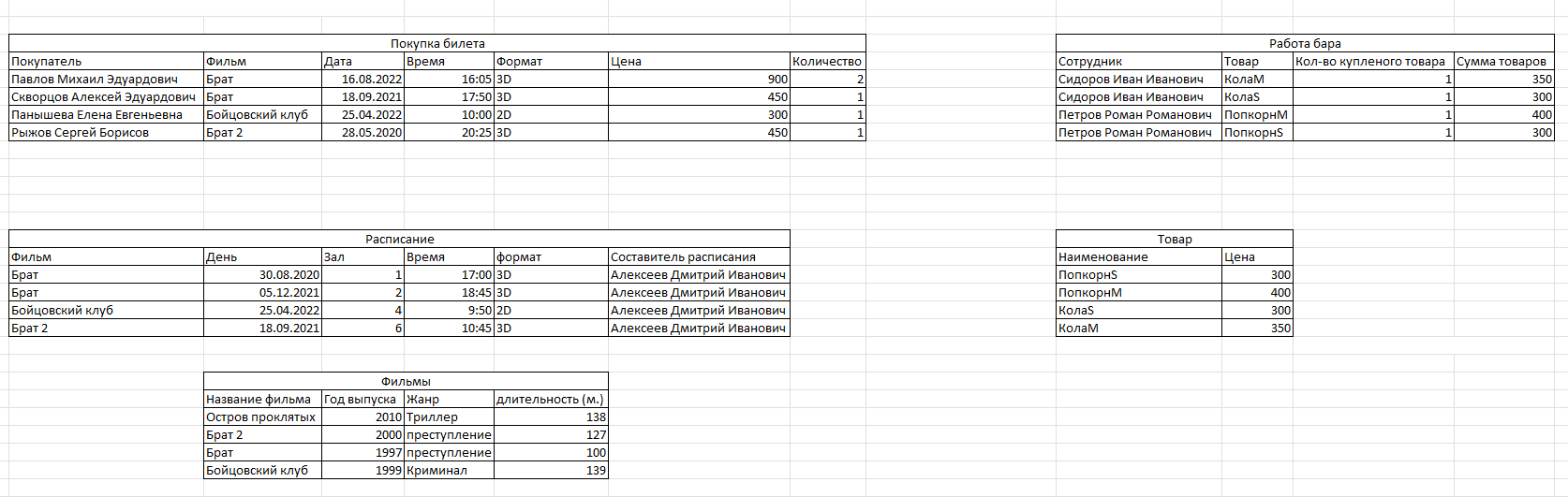


Рисунок 17 - 1 нормальная форма

1. Создать нормализацию 2НФ + заполнить таблицы по 4 записи

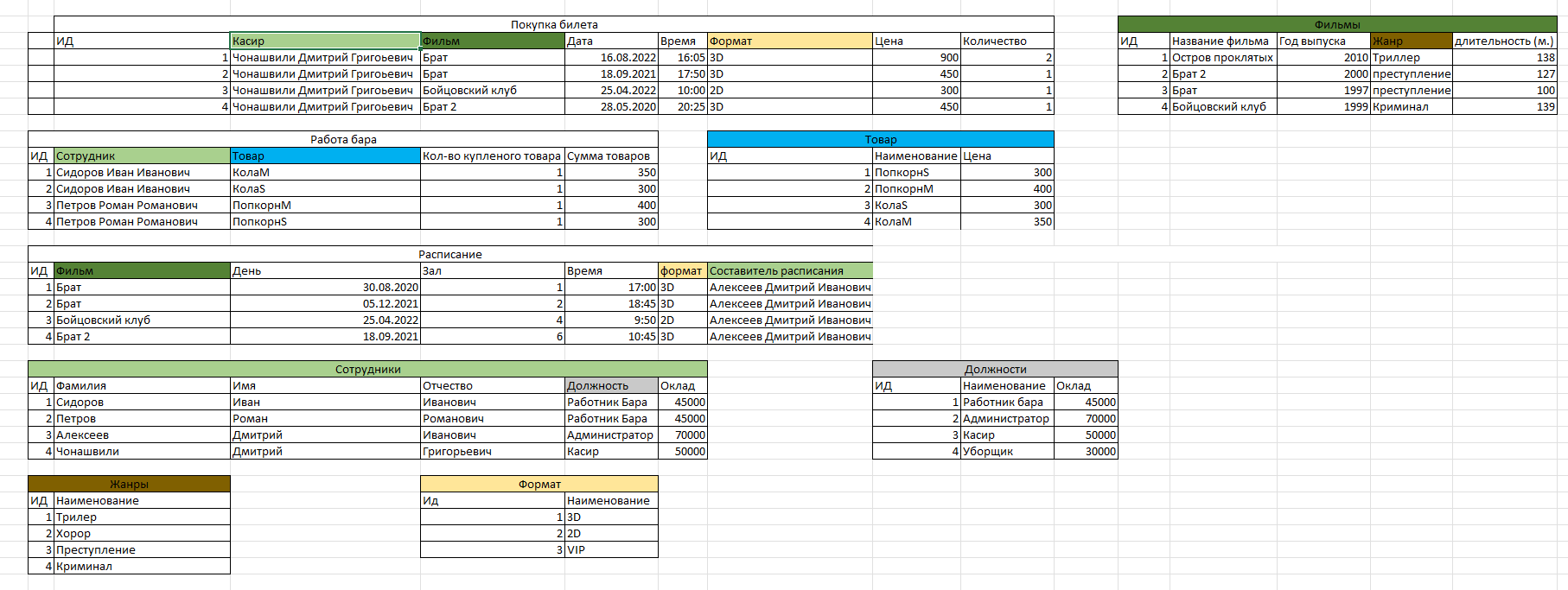


Рисунок 18 - 2 нормальная форма

1. Создать нормализацию 3НФ + заполнить таблицы по 4 записи

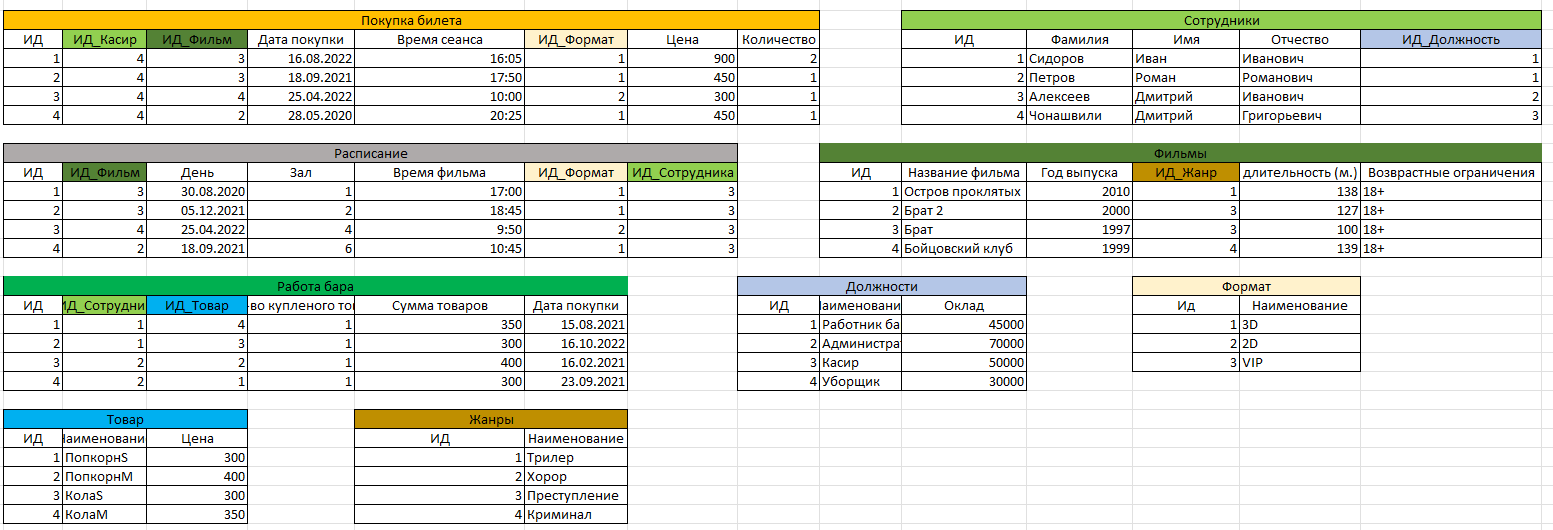


Рисунок 19 - 3 нормальная форма

1. Определить цвета таблиц и выделить атрибуты, относительно данных.

Главными таблицами моих подсистем является:

* Покупка билета;
* Расписание;
* Работа бара.

Связи:

1. **Один ко многим;**
   1. Таблица должность связанна с таблицей сотрудники;
   2. Таблица Формат связанна с таблицей расписание;
   3. Таблица Формат связанна с таблицей покупка билета;
   4. Таблица сотрудники связана с таблицей покупка билета;
   5. Таблица сотрудники связана с таблицей работа бара;
   6. Таблица сотрудники связана с таблицей расписание;
   7. Таблица Фильм связанна с таблицей покупка билета;
   8. Таблица Фильм связанна с таблицей расписание;
   9. Таблица Товар связанна с таблицей работа бара;
   10. Таблица жанр связанна с таблицей фильм;
2. **Один ко одному не используется;**
3. **Многие ко многим не используется.**

**Вывод**

В ходе работы определена и описаны объекты предметной области и проведена нормализация БД.

# Практическая №6 «ER-модель»

Цель работы: проектирование структуры базы данных, создание ER – моделей.

Теоретическая часть:

ER-модель – это представление базы данных в виде наглядных графических диаграмм. ER-модель визуализирует процесс, который определяет некоторую предметную область. Диаграмма «сущность»-«связь» – это диаграмма, которая представляет в графическом виде сущности, атрибуты и связи.

1. Логическая или Инфологическая модель данных – это отображение предметной области без привязки к аппаратной или программной части, на понятном пользователю языке, где связь строится от названия родительской сущности к внешнему атрибуту дочерней сущности.

2. Физическая или Даталогическая модель данных – это отображение

инфологической модели с привязкой к программной и аппаратной части, составляется на языке понятном СУБД, где связь идёт от первичного ключа главной таблицы к внешнему ключу подчинённой таблицы.

Особенности построения модели:

1. Все таблицы должны быть связаны

2. Связи не должны пересекаться между собой

3. Связь должна идти строго от PK к FK

4. Избегать замыкание сущностей с помощью распределения связей («по кольцу)»

Практическая часть:

1. На основании проделанной нормализации БД, составить:

1.1. Логическую модель данных

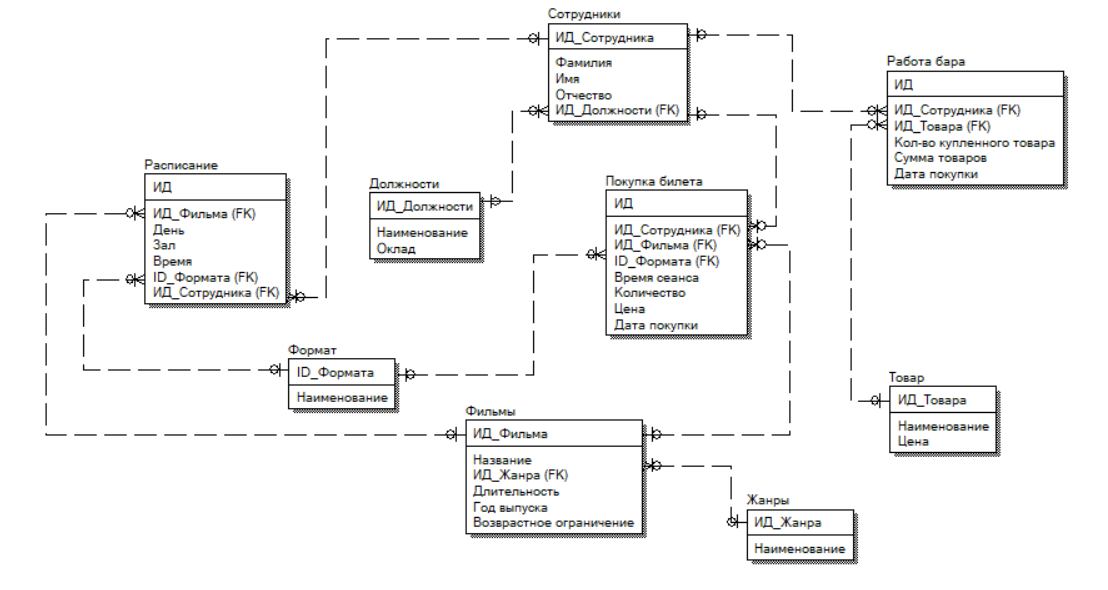


Рисунок 20 - Логическая модель

1.2. Физическую модель данных

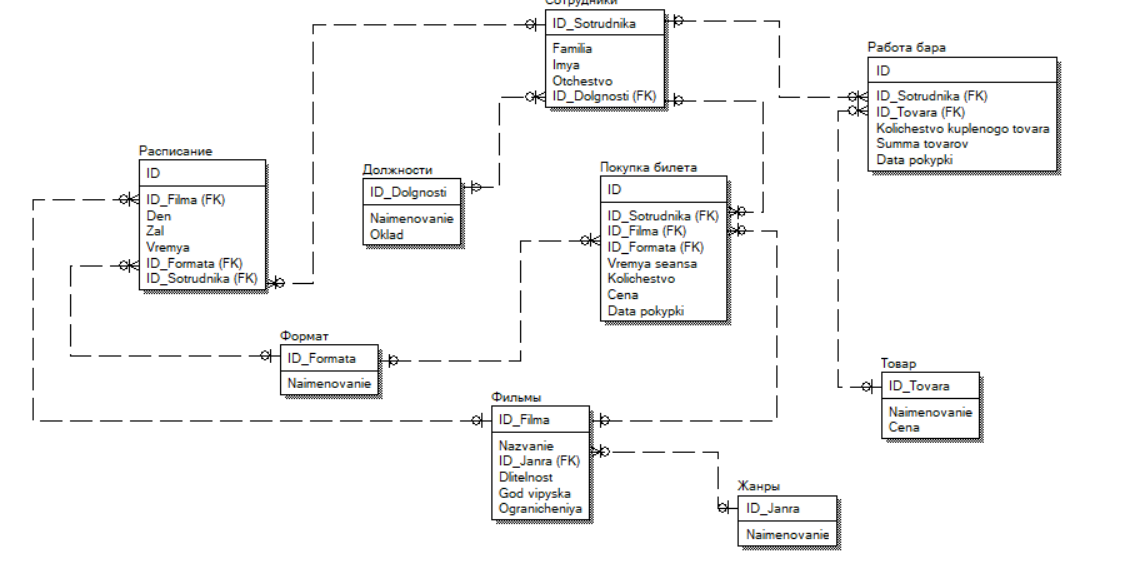


Рисунок 21 - Физическая модель

2. Описать все сущности с атрибутами, указать связи и к каким подсистемам относятся.

Пример:

* Сущность «Сотрудники» хранит в себе такие данные, как Фамилия сотрудника, Имя сотрудника, Отчество сотрудника. Данная сущность «Сотрудник» связана с сущностью «Должность». Сущность «Сотрудник» участвует в подсистеме «Продажа билетов», «Работа бара» и «Расписание» связью 1:М.
* Сущность «Должность» хранит в себе такие данные, как Название должности, Оклад. Данная сущность «Должность» связана с сущностью «Сотрудник» связью 1:М.
* Сущность «Работа бара» хранит в себе такие данные, как Количество купленного товара, Сумма товаров и дата покупки. Данная сущность «Работа бара» связана с сущностями «Сотрудники» и «Товар».
* Сущность «Расписание» хранит в себе такие данные, как День, Зал, Время. Данная сущность «Расписание» связана с сущностями «Формат», «Фильмы», «Сотрудники».
* Сущность «Покупка билета» хранит в себе такие данные, как Дата покупки, время сеанса, цена, количество. Данная сущность «Покупка билета» связана с сущностями «Формат», «Фильмы», «Сотрудники».
* Сущность «Формат» хранит в себе такие данные, как Название. Данная сущность «Формат» связанна с сущностями «Покупка билета», «Расписание» связью 1:М.
* Сущность «Фильмы» хранит в себе такие данные, как Название, Год выпуска, Длительность, Возвратное ограничение. Данная сущность «Фильмы» связанна с сущностями «Жанры». Сущность «Фильмы» участвует в подсистеме «Продажа билетов» и «Расписание связью» 1:М.
* Сущность «Жанры» хранит в себе такие данные, как Название. Данная сущность «Жанры» участвует, в сущности, «Фильмы» связью 1:М.
* Сущность «Товар» хранит в себе такие данные, как Название, Цена. Данная сущность «Товар» участвует в подсистеме «Работа бара» связью 1:М.

3. Указать тип данных для всех атрибутов сущностей

Таблица 2 - Тип данных «Сотрудники»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Логическое  название поля | Физическое  название поля | Тип  данных | Ограничения  по символам | Комментарий |
| ИД\_Сотрудника | ID\_sotrudnika | Number | - | Уникальный номер  (Идентификатор) записи  Сотрудников |
| Фамилия | Familia | String | (20) | Фамилия Сотрудника |
| Имя | Imya | String | (15) | Имя Сотрудника |
| Отчество | Otchestvo | String | (20) | Отчество сотрудника |
| ИД\_Должности | ID\_Dolgnosti | Number | - | Уникальный номер  (Идентификатор) записи  должностей |

Таблица 3 - Тип данных «Работа бара»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Логическое  название поля | Физическое  название поля | Тип  данных | Ограничения  по символам | Комментарий |
| ИД | ID | Number | - | Уникальный номер  (Идентификатор) записи  Работы бара |
| ИД\_Сотрудника | ID\_sotrudnika | Number | - | Уникальный номер  (Идентификатор) записи  Сотрудников |
| ИД\_Товара | ID\_Tovara | Number | - | Уникальный номер  (Идентификатор) записи  Товара |
| Дата покупки | Data pokupki | Datetime | (8) | Дата совершения покупки |
| Сумма товаров | Summa tovarov | Number | (10) | Общая сумма покупки |
| Кол-во купленного товара | Kolichestvo kuplenogog tovara | Number | (6) | Количество купленного товара |

Таблица 4 - Тип данных «Расписание»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Логическое  название поля | Физическое  название поля | Тип  данных | Ограничения  по символам | Комментарий |
| ИД | ID | Number | - | Уникальный номер  (Идентификатор) записи  Составленого расписания |
| День | Den | Datetime | (8) | Выбранный день, в который будут составлять расписание |
| Зал | Zal | Number | (2) | Выбранный зал на кинофильм |
| Время | Vremya | Datetime | (4) | Выбранное время, на которое будет показываться фильм |
| ИД\_Фильма | ID\_Filma | Number | - | Уникальный номер  (Идентификатор) записи  Фильмов |
| ИД\_Формата | ID\_Formata | Number | - | Уникальный номер  (Идентификатор) записи  Форматов |
| ИД\_Сотрудника | ID\_sotrudnika | Number | - | Уникальный номер  (Идентификатор) записи  Сотрудников |

Таблица 5 - Тип данных «Должности»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Логическое  название поля | Физическое  название поля | Тип  Данных | Ограничения  по символам | Комментарий |
| ИД\_Должности | ID\_Dolgnosti | Number | - | Уникальный номер  (Идентификатор) записи  Должностей |
| Наименование | Naimenovanie | String | (30) | Наименование должностей |
| Оклад | Oklad | Number | (8) | Оклад на конкретную должность |

Таблица 6 - Тип данных «Покупка билета»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Логическое  название поля | Физическое  название поля | Тип  Данных | Ограничения  по символам | Комментарий |
| ИД | ID | Number | - | Уникальный номер  (Идентификатор) записи  Покупки билета |
| ИД\_Сотрудника | ID\_sotrudnika | Number | - | Уникальный номер  (Идентификатор) записи  Сотрудников |
| ИД\_Фильма | ID\_Filma | Number | - | Уникальный номер  (Идентификатор) записи  Фильмов |
| ИД\_Формата | ID\_Formata | Number | - | Уникальный номер  (Идентификатор) записи  Формата |
| Дата покупки | Data pokupki | Datetime | (8) | Дата в которую была совершена покупка билета |
| Время сеанса | Vremya seansa | Datetime | (4) | Точное время начала сеанса |
| Цена | Cena | Number | (8) | Цена билета |
| Количество | Kolichestvo | Number | (3) | Количество купленных билетов |

Таблица 7 - Тип данных «Формат»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Логическое  название поля | Физическое  название поля | Тип  Данных | Ограничения  по символам | Комментарий |
| ИД\_Формата | ID\_Formata | Number | - | Уникальный номер  (Идентификатор) записи  Формата |
| Наименование | Naimenovanie | String | (10) | Наименование Форматов для фильма |

Таблица 8 - Тип данных «Фильмы»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Логическое  название поля | Физическое  название поля | Тип  Данных | Ограничения  по символам | Комментарий |
| ИД\_Фильма | ID\_Filma | Number | - | Уникальный номер  (Идентификатор) записи  Фильмов |
| Название | Nazvanie | String | (30) | Полное название Фильма |
| ИД\_Жанра | ID\_Janra | Number | - | Уникальный номер  (Идентификатор) записи  Жанра |
| Длительность | Dlitelnost | Number | (3) | Длительность фильма в минутах |
| Год выпуска | God vipyska | Datetime | (8) | Год выпуска фильма |
| Возвратное ограничение | Ogranicheniya | Number | (3) | Возврастное ограничение для фильма |

Таблица 9 - Тип данных «жанры»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Логическое  название поля | Физическое  название поля | Тип  данных | Ограничения  по символам | Комментарий |
| ИД\_Жанра | ID\_Janra | Number | - | Уникальный номер  (Идентификатор) записи  Жанра |
| Наименование | Naimenovanie | String | (20) | Наименование Жанра |

Таблица 10 - Тип данных «Товар»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Логическое  название поля | Физическое  название поля | Тип  данных | Ограничения  по символам | Комментарий |
| ИД\_Товара | ID\_Tovara | Number | - | Уникальный номер  (Идентификатор) записи  Товара |
| Наименование | Naimenovanie | String | (30) | Наименование Товара |
| Цена | Cena | Number | (6) | Цена товара |

Вывод: в ходе работы была спроектирована структура базы данных, создана ER – моделей.

# Практическая №7 «Создание базы данных в MS Access»

Цель работы: создать базу данных в MS Access и произвести работу с данными при помощи запросов.

Практическая часть:

1. Создать в MS Access базу данных по созданной ER-модели. (скрин 1 окна с созданием)

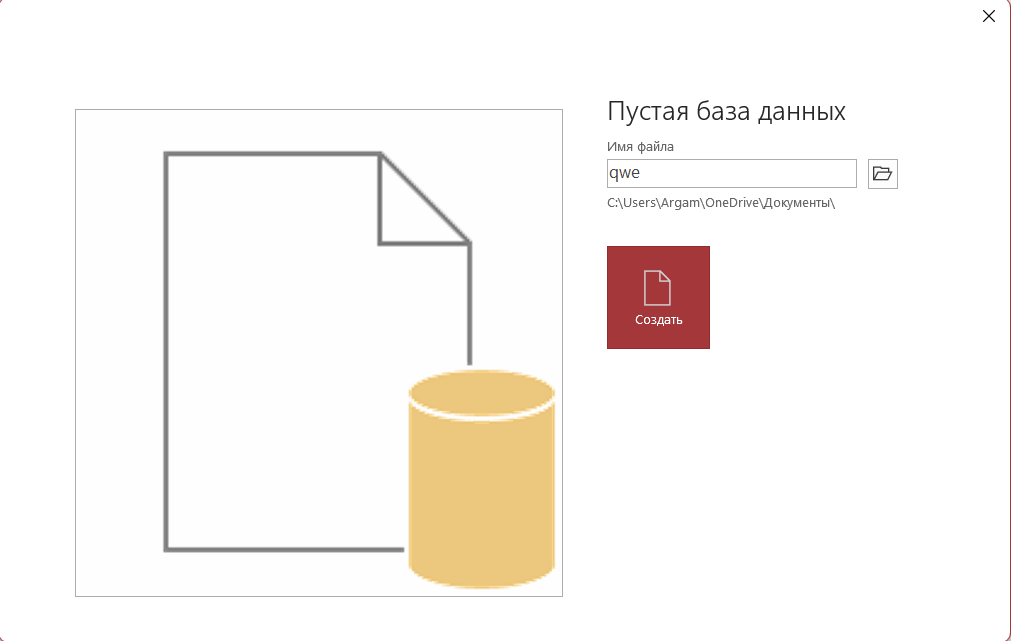


Рисунок 22 - Создание базы данных access

2. Во всех сущностях настроить типы данных атрибутов. (через конструктор таблиц) (скрины)

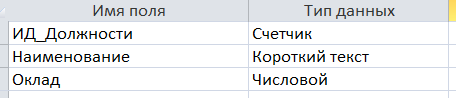


Рисунок 23 - Сущность Должности

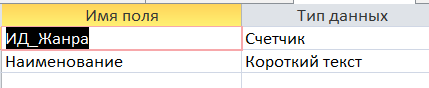


Рисунок 24 - Сущность Жанры

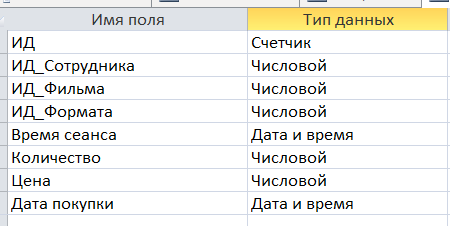


Рисунок 25 - Сущность Покупка билета

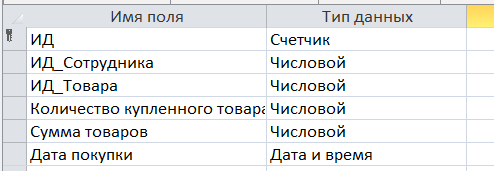


Рисунок 26 - Сущность Работа бара

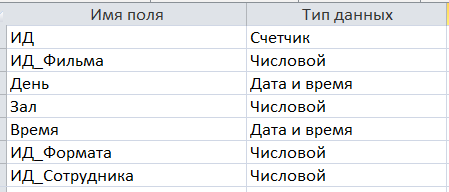


Рисунок 27 - Сущность Расписание

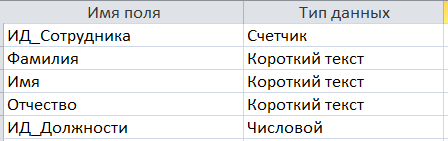


Рисунок 28 - Сущность Сотрудники

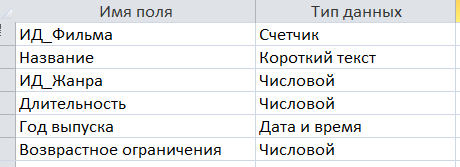


Рисунок 29 - Сущность Фильмы

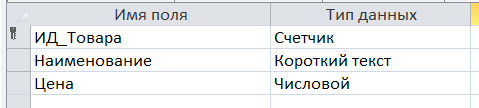


Рисунок 30 - Сущность товар

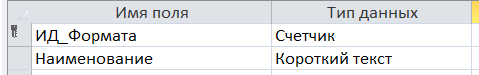
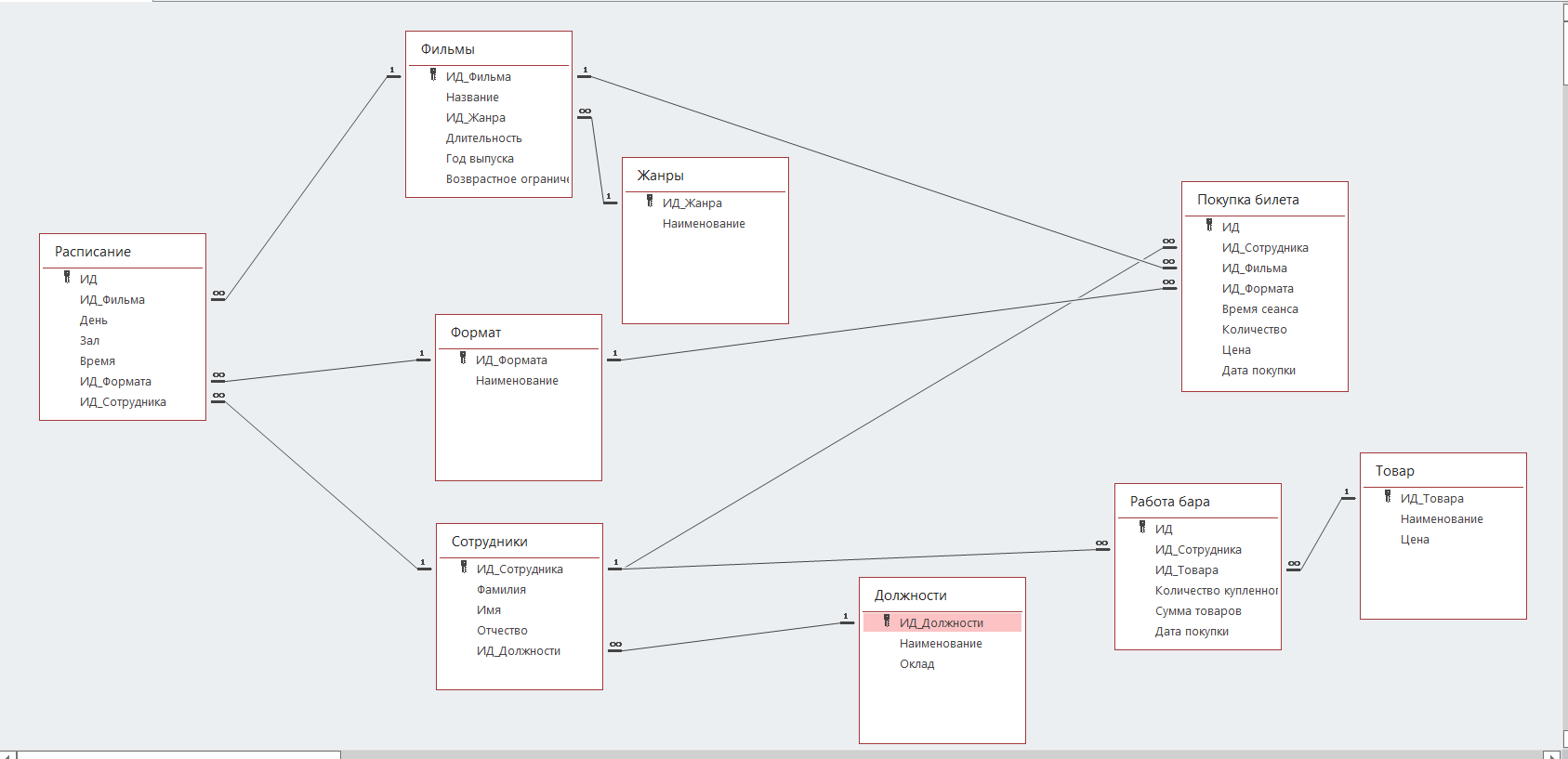


Рисунок 31 - Сущность Формат

3. Настроить взаимосвязи между таблицами (при настройке поставить галочки в разделе Обеспечение целостности данных: каскадное обновление и каскадное удаление связанных записей) (скрин модели с настроенными связями)



4. Заполнить все таблицы по 15 записей (скрины)



Рисунок 32 - Таблица Должности

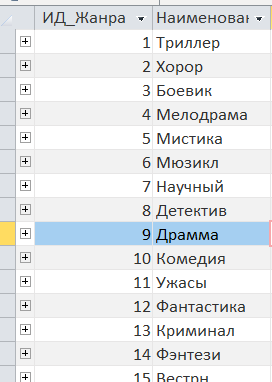


Рисунок 33 - Таблица Жанры

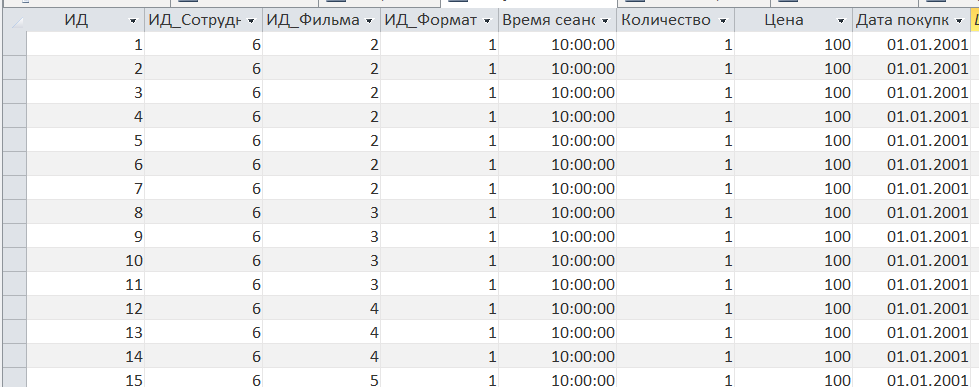


Рисунок 34 - Таблица Покупка билета

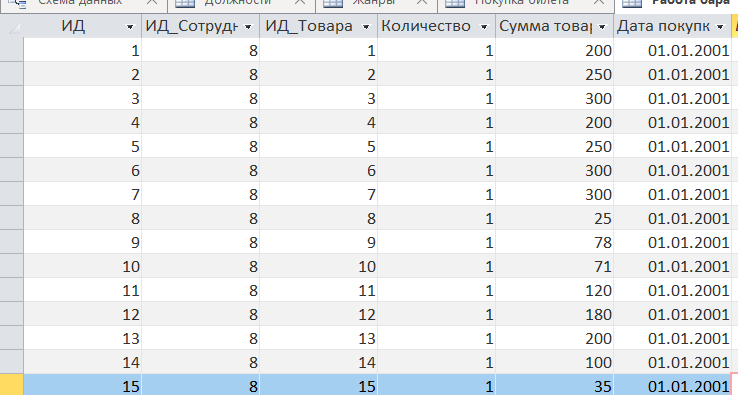


Рисунок 35 - Таблица Работа бара

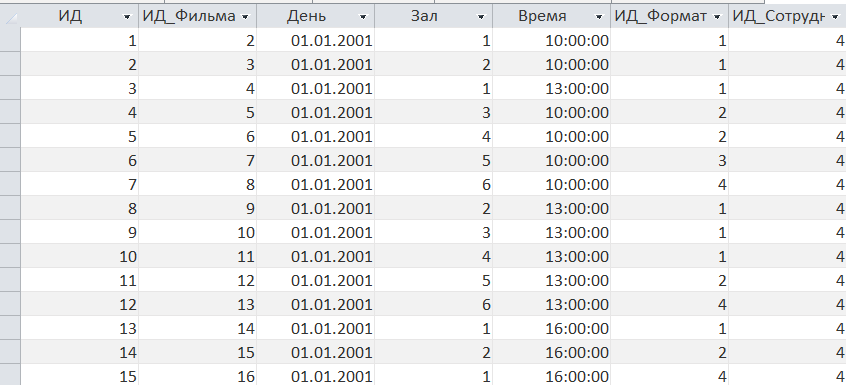


Рисунок 36 - Таблица Расписание

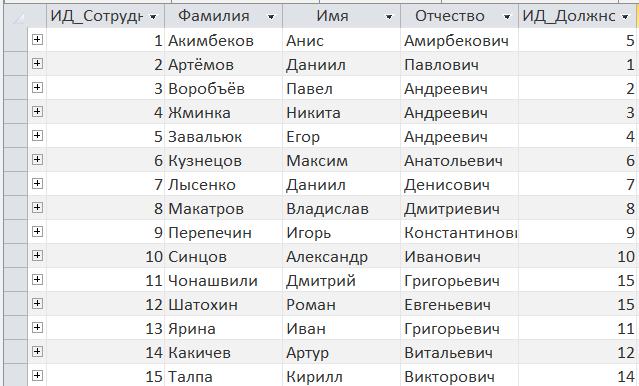


Рисунок 37 - Таблица Сотрудники

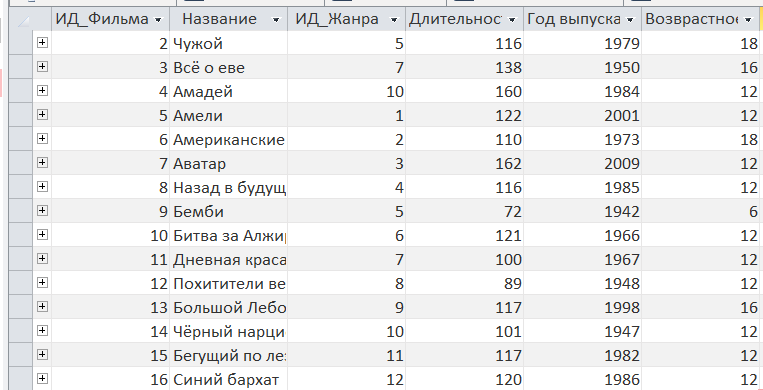


Рисунок 38 - Таблица Фильмы

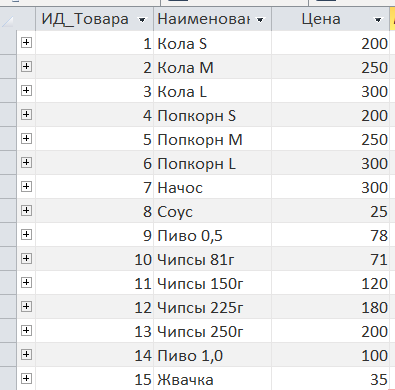


Рисунок 39 - Таблица Товар

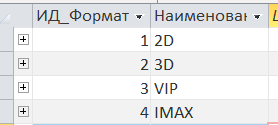


Рисунок 40 - Таблица Формат

Вывод: в ходе работы была создана база данных в MS Access.

# Практическая №8 «Запросы и отчеты в Access»

Цель работы: создать несколько запросов для вывода необходимой информации из таблиц, а также сформировать отчеты для вывода данных на печать.

Запросы:

1. Создать запрос на выборку (через конструктор запросов)

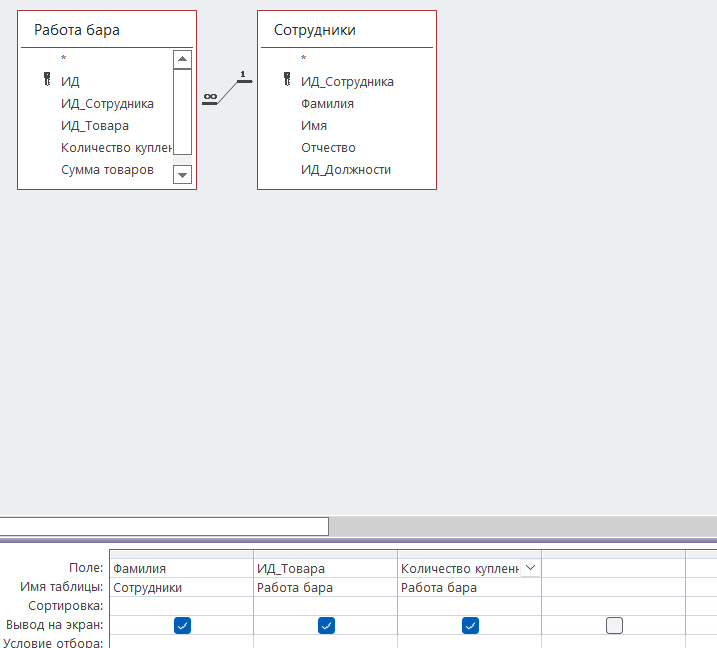


Рисунок 41 - Создание запроса на покупку бара

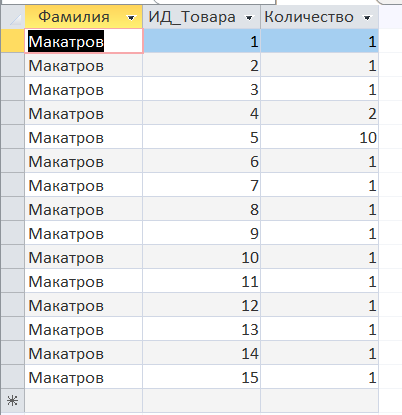


Рисунок 42 - Запрос на покупку бара

2. Создать запрос на выборку с условием

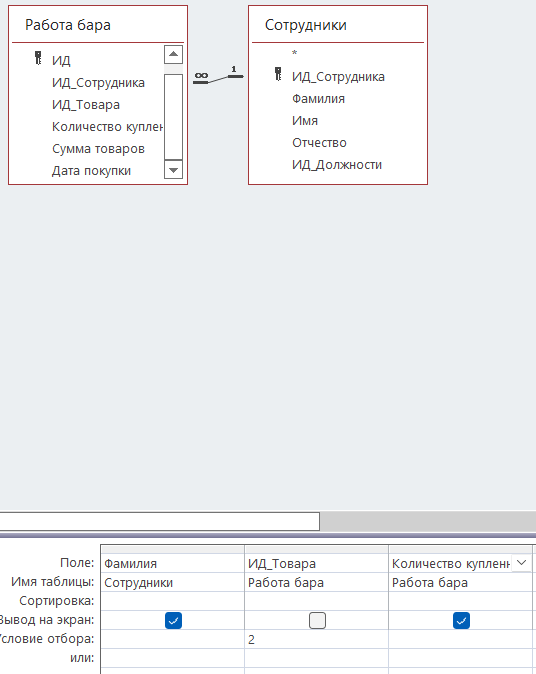


Рисунок 43 - Создание запроса на выбранную покупку товара

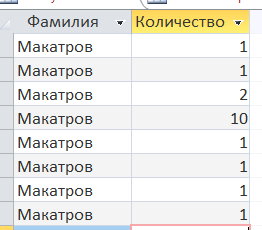


Рисунок 44 - Запрос на покупку выбранного товара

3. Создать запрос на групповые операции

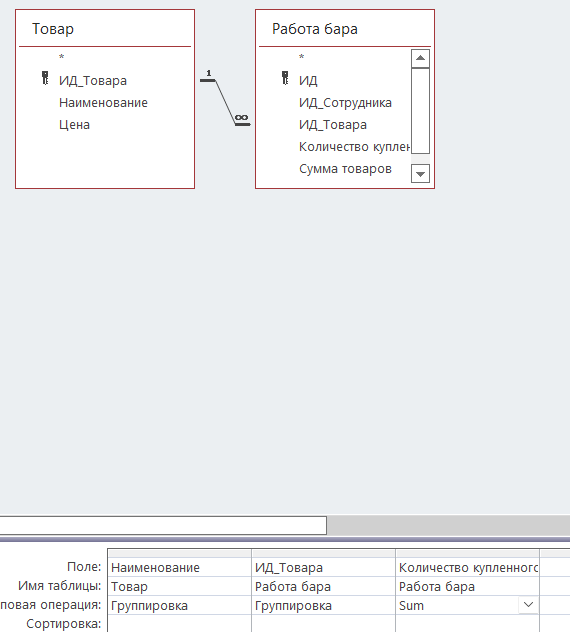


Рисунок 45 - Создание запроса на количество купленных товаров

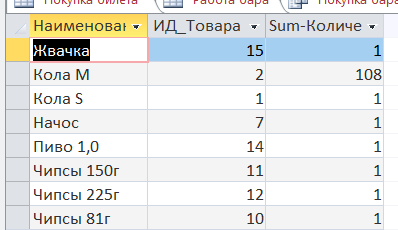


Рисунок 46 - Запрос на количество купленных товаров

Отчеты: (в своем файле с БД сохраняете все созданные отчеты)

1. Создать 3 отчета по таблицам (скрины отчетов)

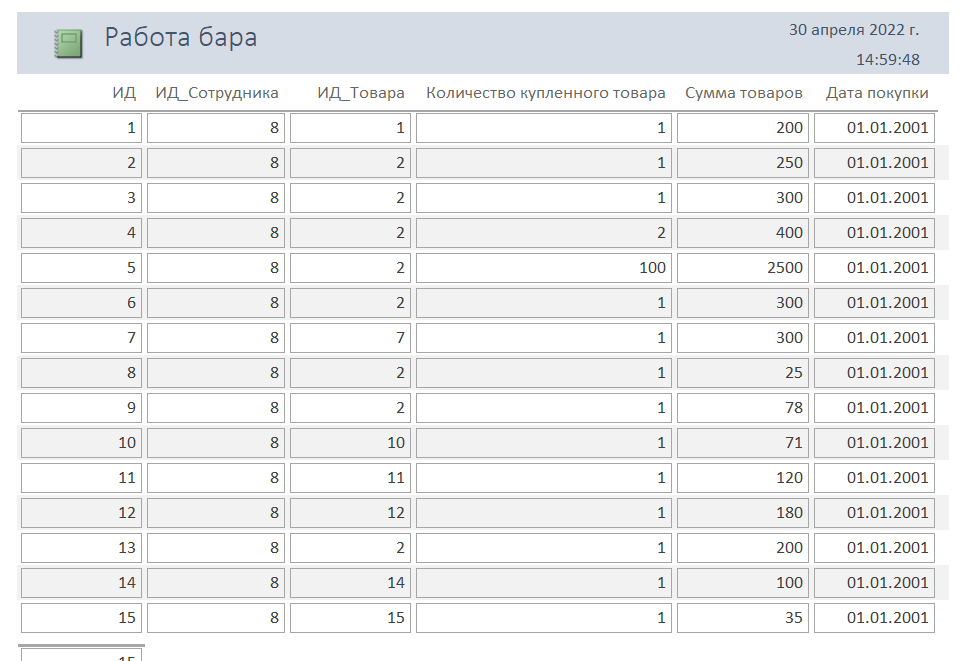


Рисунок 47 - Отчёт по работе бара

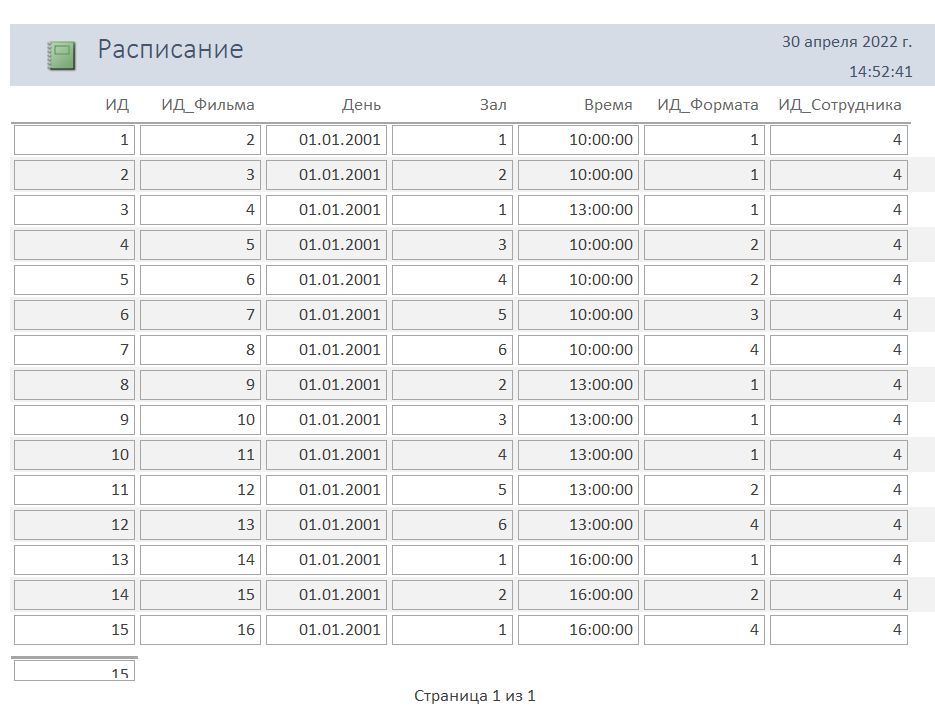


Рисунок 48 - Отчёт по расписанию

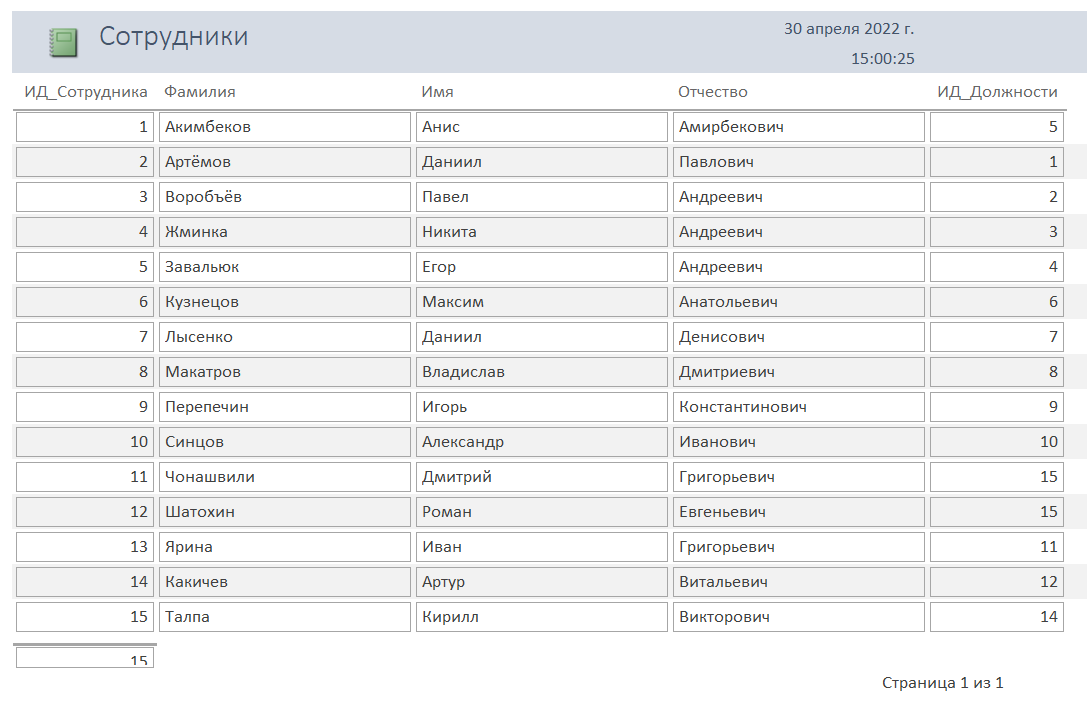


Рисунок 49 - Отчёт о сотрудниках

2. Создать 2 отчета с помощью мастера отчетов, объединив данные из разных таблиц

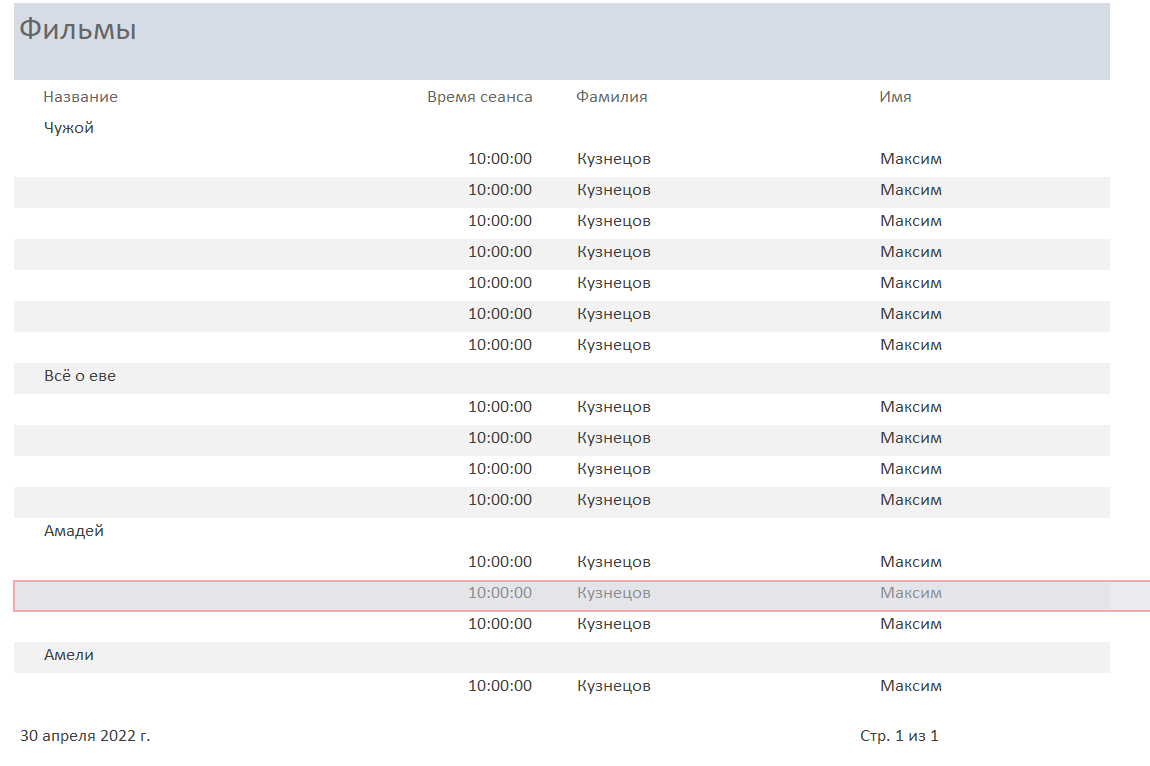


Рисунок 50 - Отчёт о покупки билетов

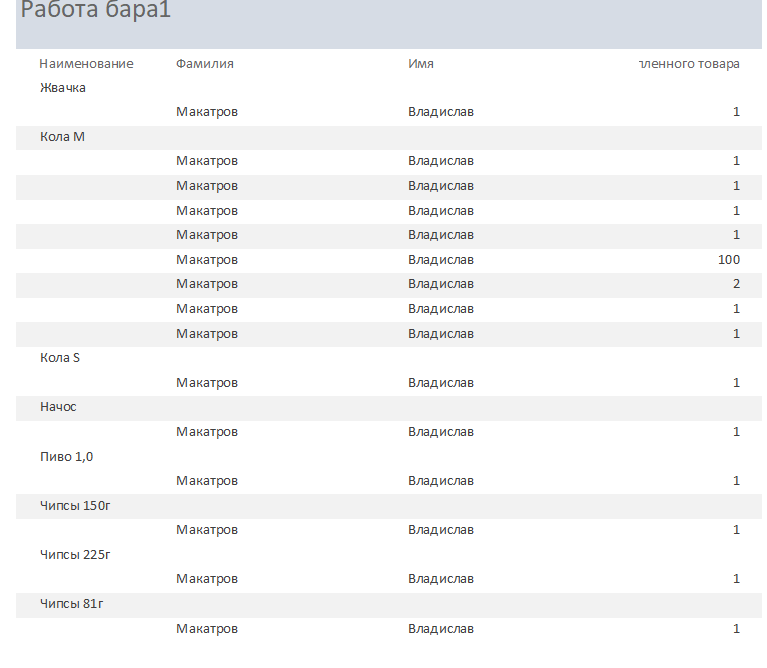


Рисунок 51 - Отчёт о покупке бара

3. Создать 1 отчет по созданному ранее запросу (скрин отчета)



Рисунок 52 - Отчёт по запросу

Вывод: в ходе работы была созданы несколько запросов для вывода необходимой информации из таблиц, а также сформированы отчеты для вывода данных на печать.

# Практическая работа №9 «Генерация БД. Создание резервной копии»

Цель работы: сгенерировать базу данных в Access с помощью созданной er-модели.

Выполнение:

1. Создать файл Access.

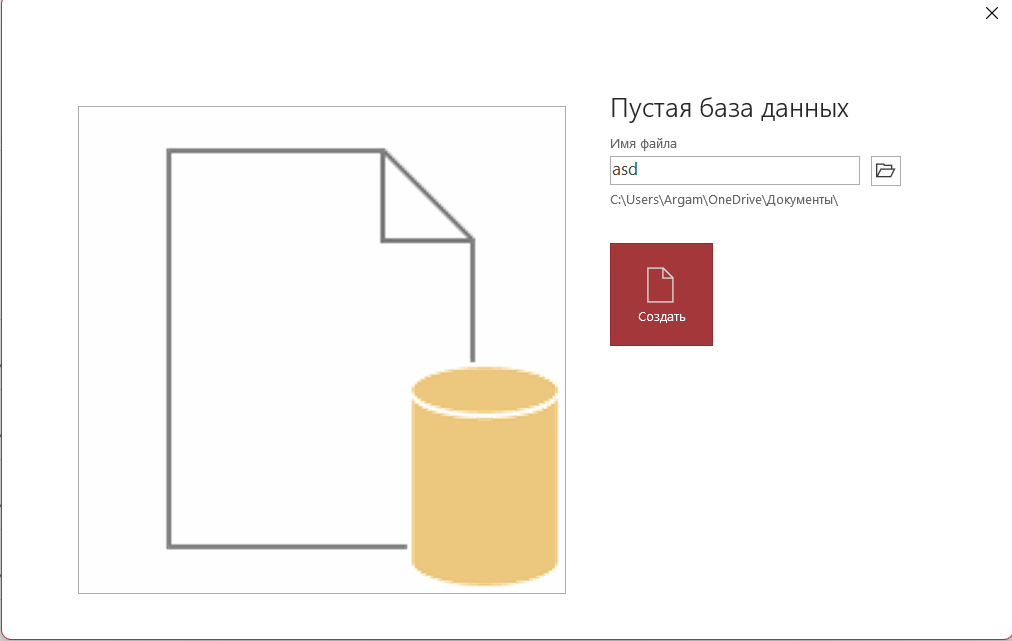


Рисунок 53 - Создание БД

2. Сохранить в формате 2002-2003(\*.mdb).

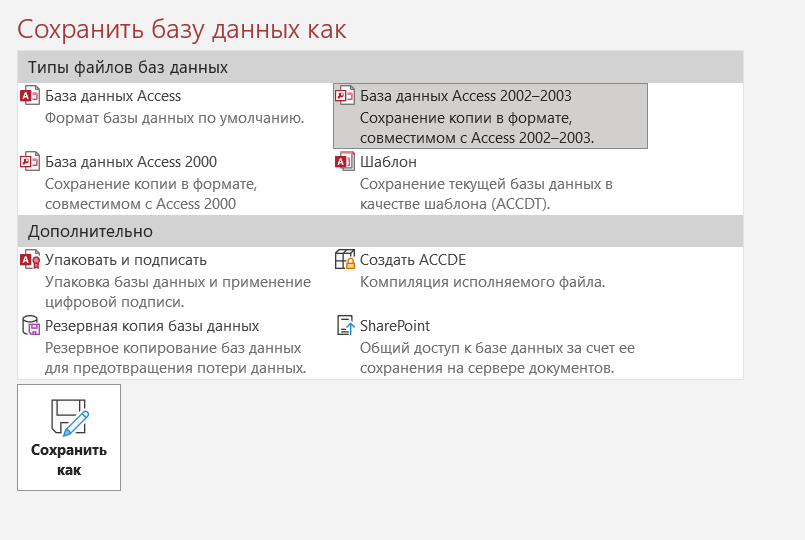


Рисунок 54 - Сохранение в формате mbd

3. Открыть ER-диаграмму и с помощью списка выбора инструментов произвести переключение с логической на физическую моделью.

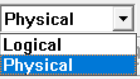


Рисунок 55 - Физическая модель

4. Выбрать СУБД, в которой производиться генерация БД физического уровня. Для этого выбираем команду Database-Choose Database. В раскрывшемся диалоговом окне выбрать СУБД Access 2003 и щелкнуть по кнопке «ОК».

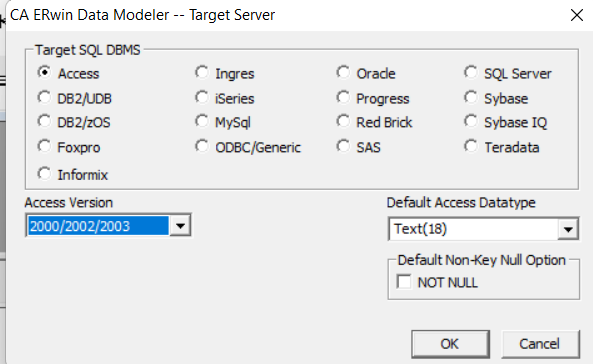


Рисунок 56 - Выбор СУБД

5. Далее осуществить подключение через Database Connection. Выставить user name: admin. В столбце «Value» напротив Database выбрать созданную ранее базу. После чего нажать на кнопку «Connect».

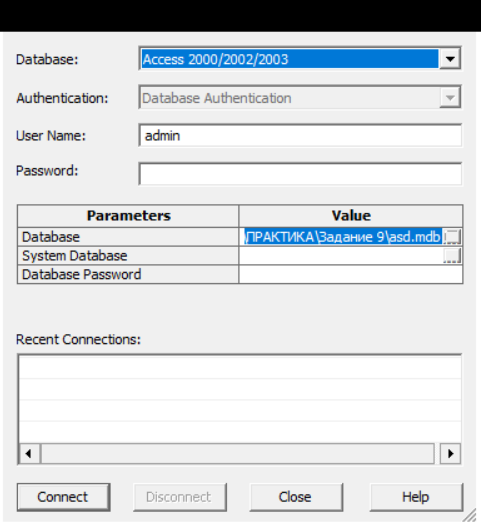


Рисунок 57 - Подключение

6. Настроить генерацию БД в Access. Forward Engineer-Schema Generation. Во вкладке Options переходим в Index и ставим галочку напротив ForeignKey.

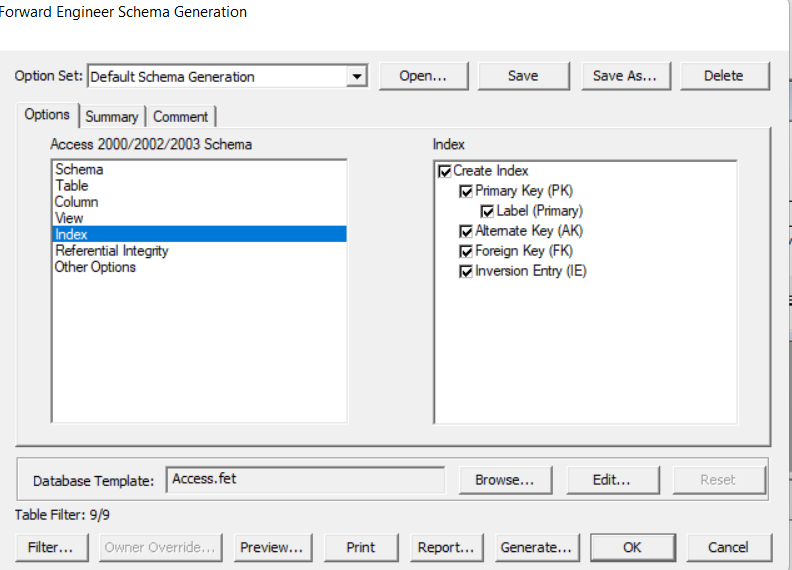


Рисунок 58 - Настройка генерации

7. Нажимаем на кнопку «Generate». После успешного подключения БД нажимаем «ОК».

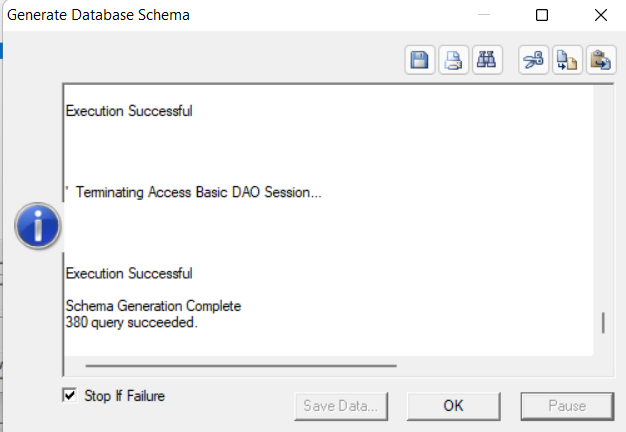


Рисунок 59 - Проверка подключения

8. Заходим в Access и проверяем созданные таблицы

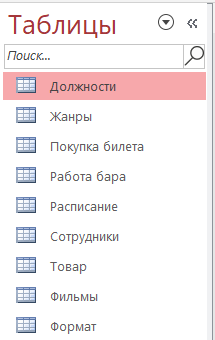


Рисунок 60 - Проверка таблиц

9. Далее просматриваем созданную схему данных, в ней автоматически проставятся связи.

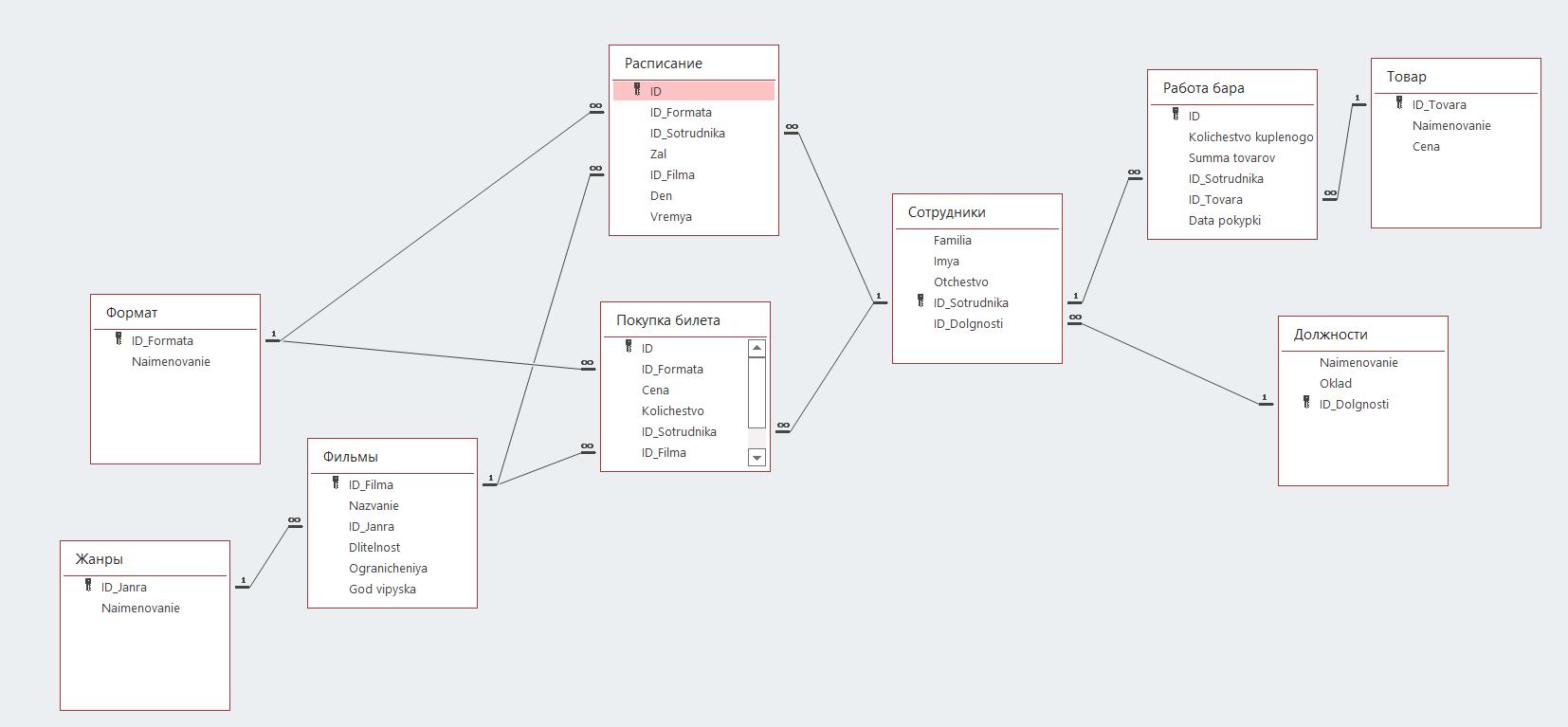


Рисунок 61 - Проверка связей

Резервное копирование БД

Теоретическая часть:

Вам потребуется резервная копия базы данных Рабочего стола Access, чтобы восстановить всю базу данных в случае сбоя системы или восстановить объект, если команды &quot;Отменить&quot; недостаточно для исправления ошибки.

Если вам кажется, что резервная копия базы данных не используется впустую, подумайте, на что можно сэкономить время, чтобы избежать потери данных и проектирования. Регулярное создание резервных копий особенно важно при обновлении базы данных несколькими пользователями. Без резервной копии невозможно восстановить поврежденные или отсутствующие объекты, а также изменения в дизайне

базы данных.

Выполнение:

1. Выберите Файл - Сохранить как. В разделе «Типы файлов» выберите «Сохранить базу данных как» (рис. 62).

2. В разделе «Дополнительно» выберите элемент «Резервная копия базы данных», а затем — Сохранить как. Имя файла по умолчанию содержит имя исходной базы данных и время резервного копирования (рис. 62).

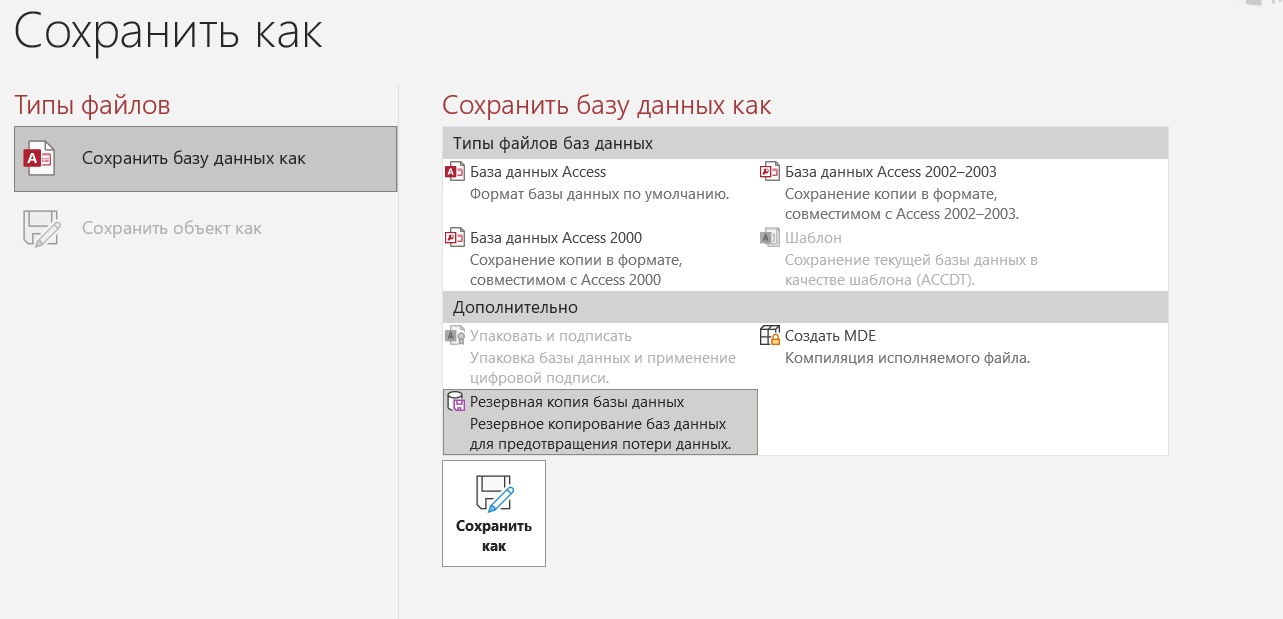


Рисунок 62 - Создание резервной копии

4. Нажмите кнопку «Сохранить».

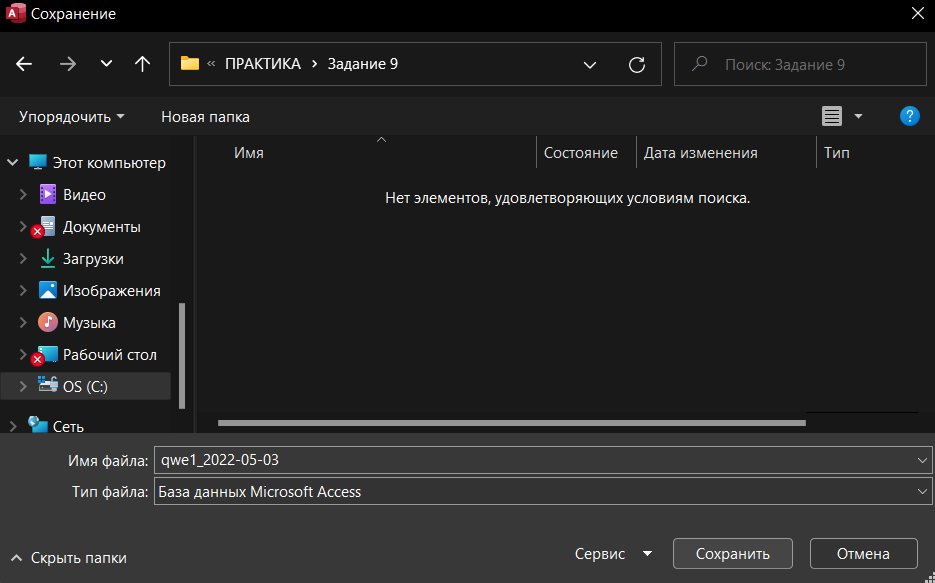


Рисунок 63 - Сохранение резервной копии

Вывод: в ходе работы была сгенерирована база данных в Access с помощью созданной er-модели.

# Практическая работа №10 «Создание информационной системы. Устранение ошибок в программном коде»

Цель работы: создать информационную систему, подключить базу данных, реализовать основные функции добавления, изменения, удаления данных, а также устранить ошибки в программном коде, возникающих в процессе создания ИС.

Выполнение:

1. Создать информационную систему (скрины справочников)

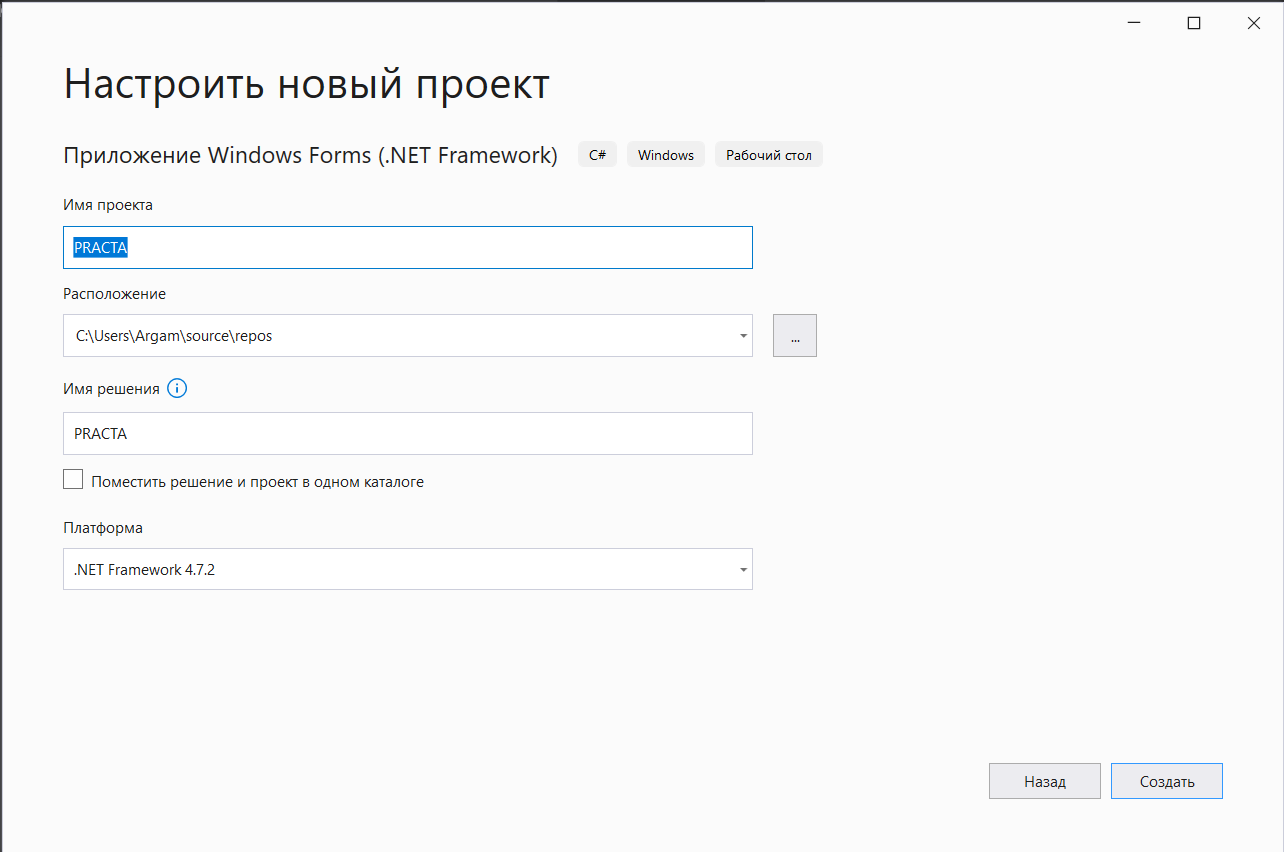


Рисунок 64 - Создание проекта

2. Подключить базу данных (поэтапные скриншоты подключения)

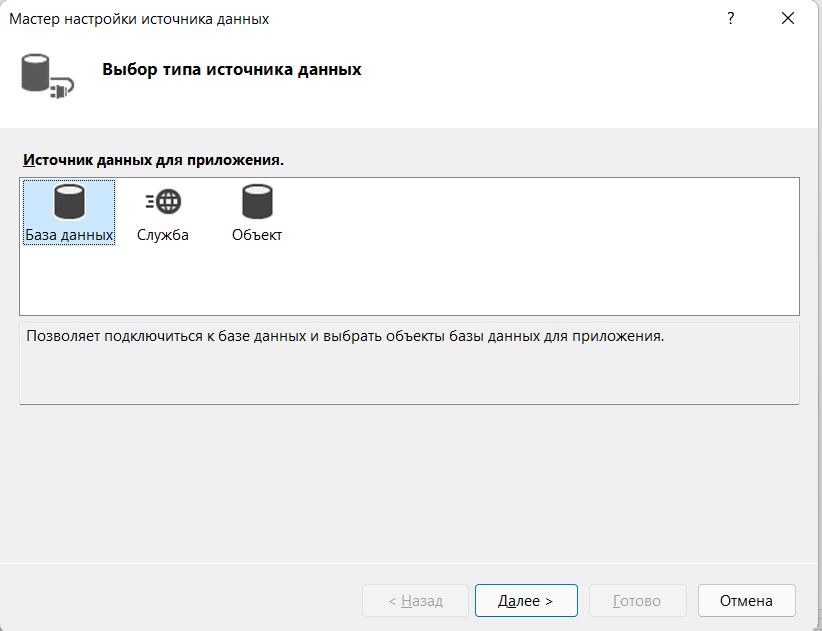


Рисунок 65 - Первый шаг подключения бд

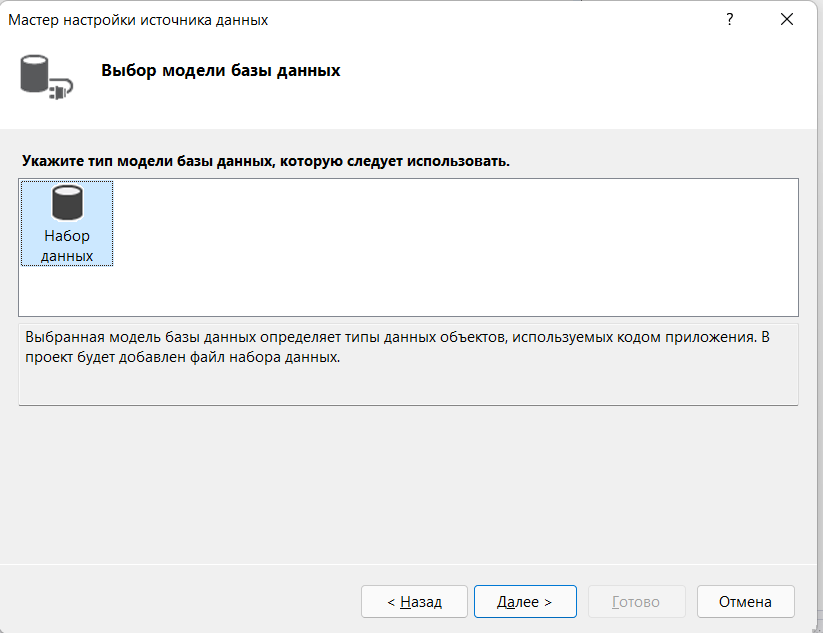


Рисунок 66 - Второй шаг подключения бд

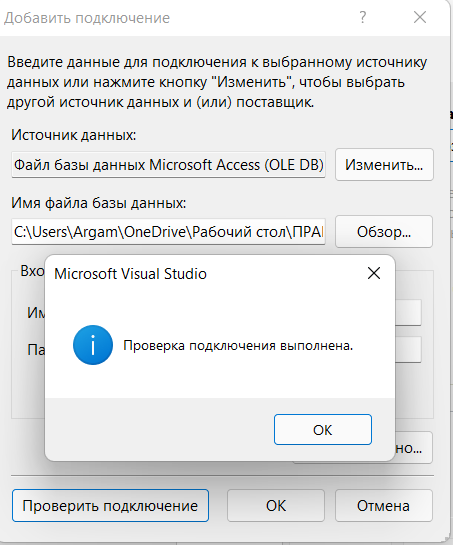


Рисунок 67 - Проверка на подключение бд

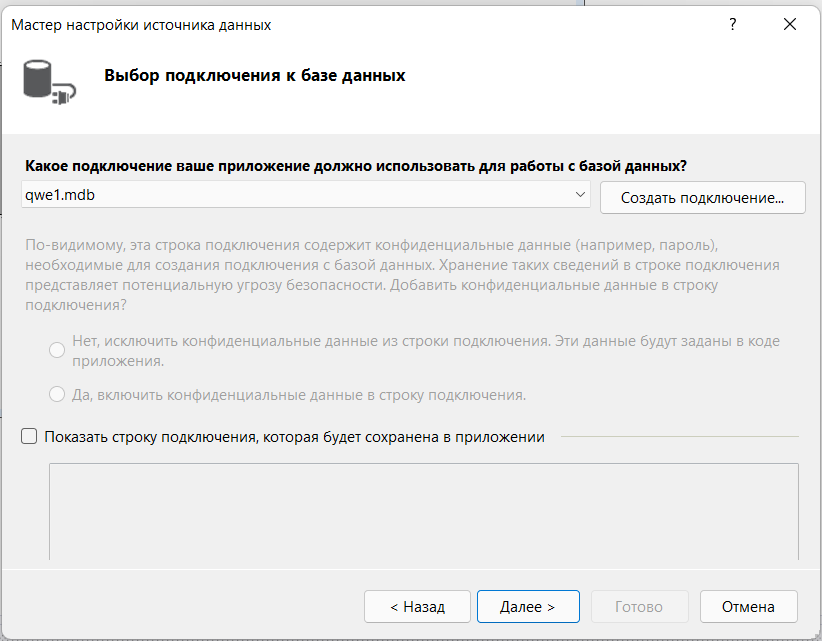


Рисунок 68 - Третий шаг на подключение бд

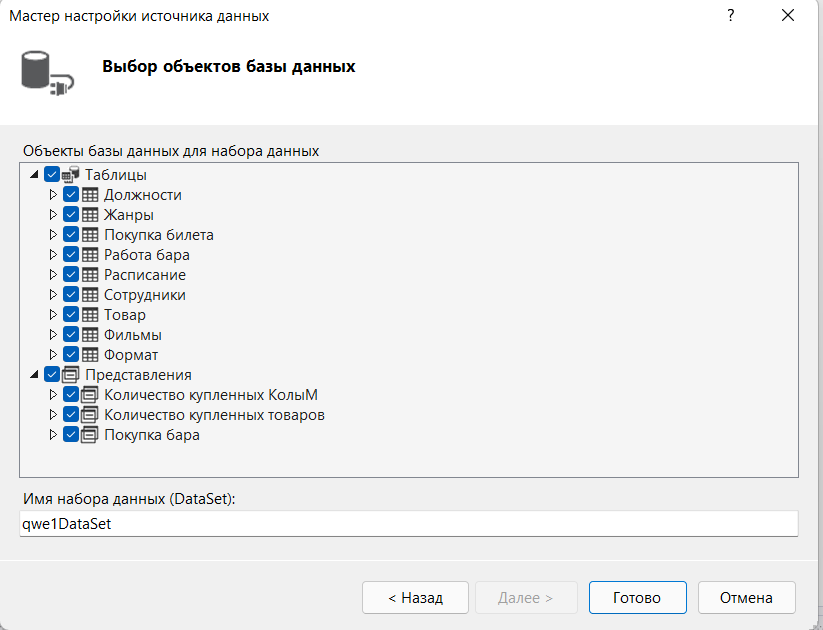


Рисунок 69 - Финальный шаг на подключение бд

3. Реализовать функцию добавления данных (описание процесса, код + скрин результата)

Ставим кнопку на экран и пишем обработчик, который привязывает её к новой форме, которую создаём.

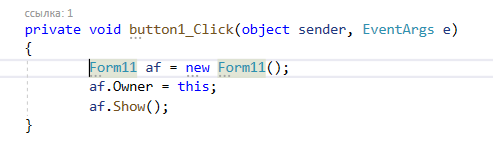


Рисунок 70 - Обработчик на открытие формы

Затем на форме расставляем элементы, которые понадобятся в таблице так же создаём две кнопки одну на выход вторую на добавление данных в таблицу.

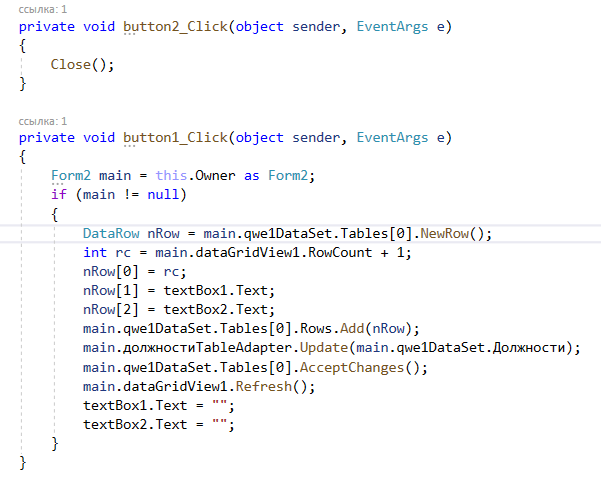


Рисунок 71 - Обработчик на закрытие формы и на добавление данных в таблицу

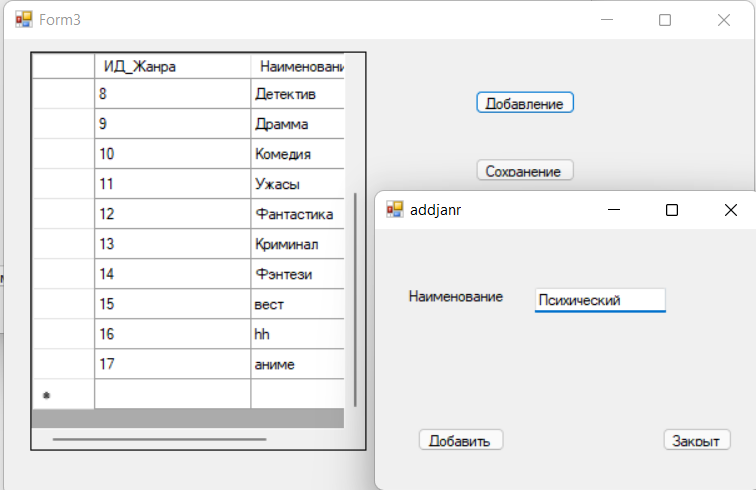


Рисунок 72 – До

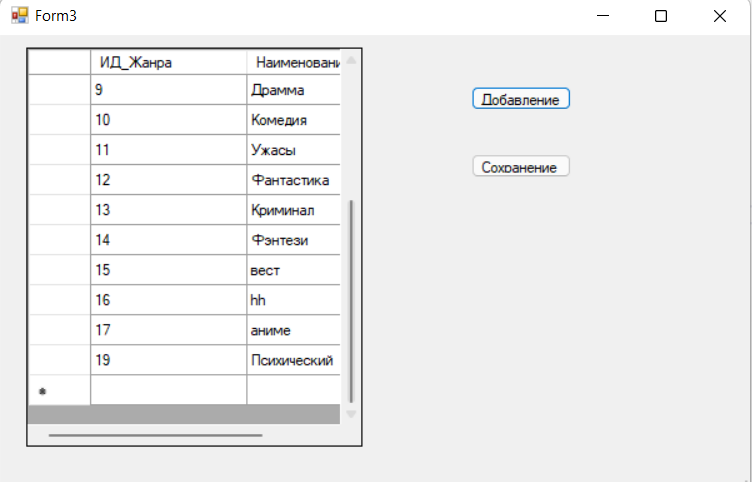


Рисунок 73 - После

4. Реализовать функцию изменения данных (описание процесса, код + скрин результата)

Изначально мы создаём на форме с таблицей кнопку и пишем на неё обработчик.

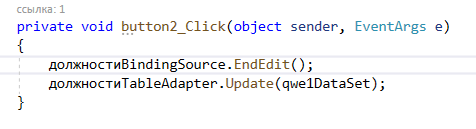


Рисунок 74 - Обработчик на сохранение данных

Ещё главное не забыть, что бы программа открывала нашу базу данных последней версией. Если этого не сделать, то не будут сохранятся наши изменённые данные.

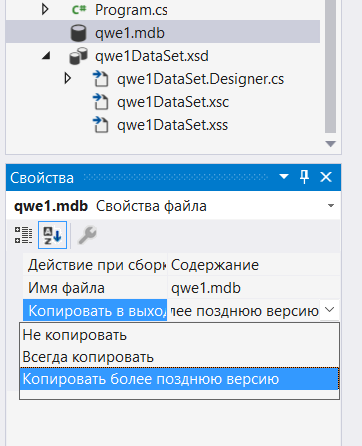


Рисунок 75 - Последняя версия БД

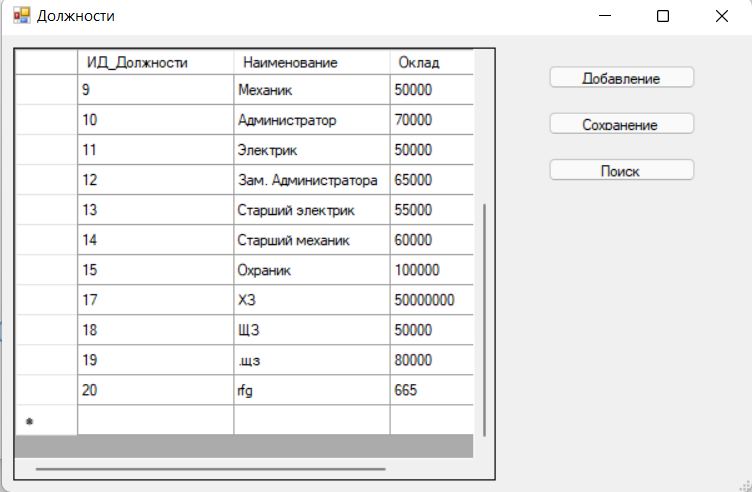


Рисунок 76 – До

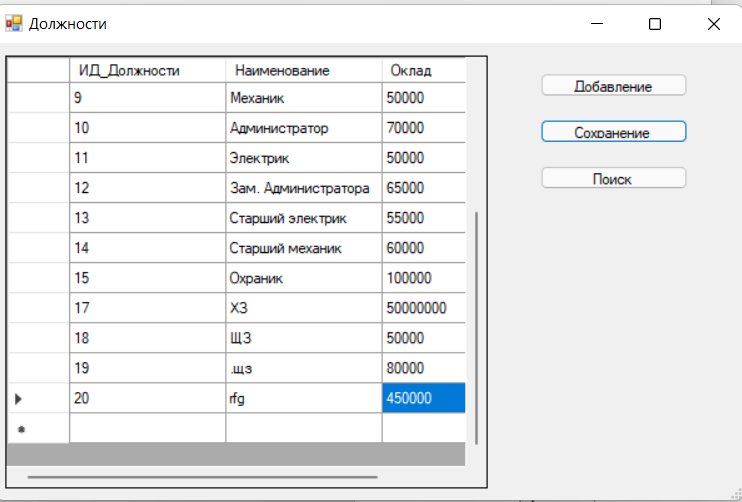


Рисунок 77 - После

5. Реализовать функцию удаления данных (описание процесса, код + скрин результата)

Функция удаления не реализована, и она в принципе не нужна ведь можно просто выбрать строку, которую надо удалить и нажать на клавиатуре кнопке «Delete»

6. Реализовать функцию поиска данных (описание процесса, код + скрин

результата)

Что бы реализовать функцию поиска данных надо на главной форме с таблицей создать кнопку и написать в ней обработчик, который откроет нам заранее созданную новою форму.

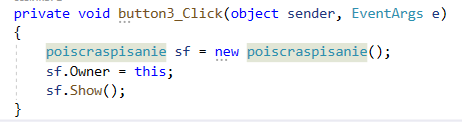


Рисунок 78 - Обработчик на открытие формы

Затем в ней добавляем textbox и 2 кнопки которые будут отвечать за выход и поиск. В textbox мы пишем то, что надо найти из таблицы и он показывает данные, которые сходятся с нашими.

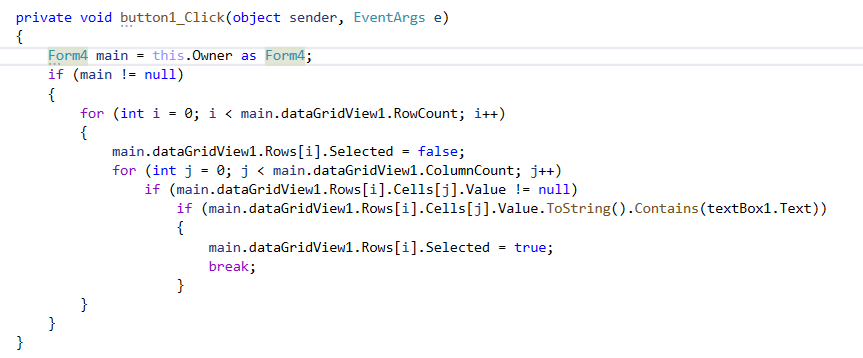


Рисунок 79 - Обработчик на поиск

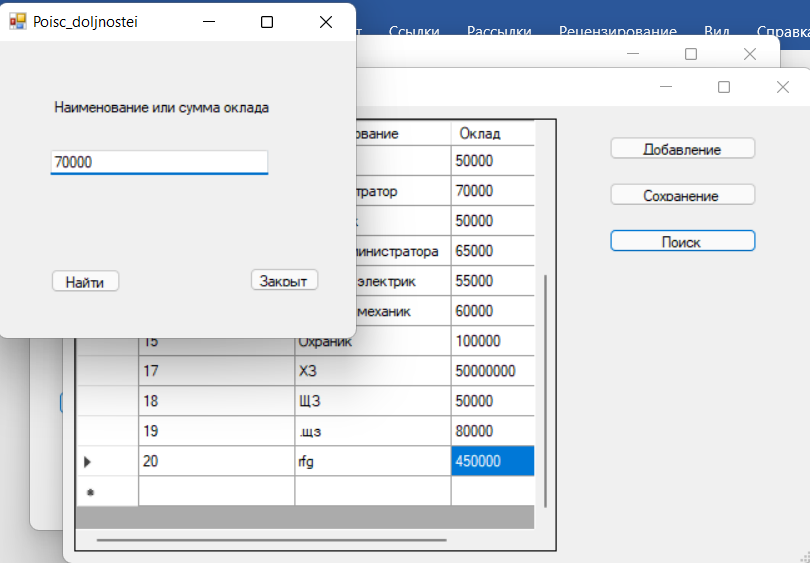


Рисунок 80 - Поиск должностей с окладом в 70000

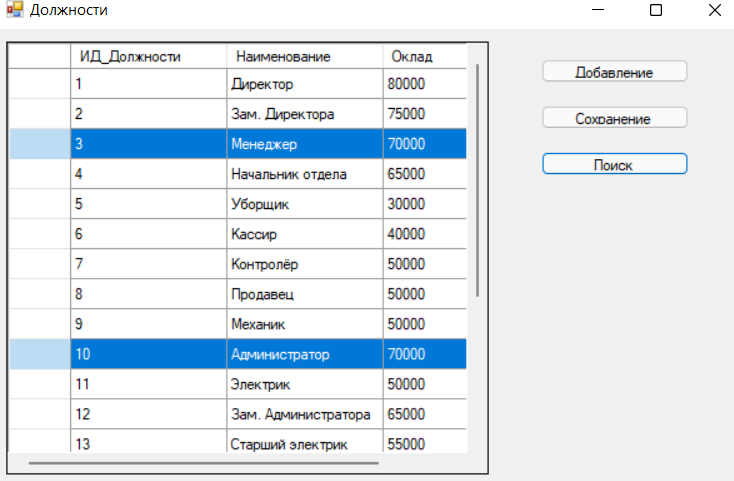


Рисунок 81 - Итог

7. Выявить минимум 5 ошибок, возникающих в написании программного кода (вставить скрин ошибки, описать ее, описать решение проблемы)

* При создании обработчика на изменение данных он делал свою работу, но при новом запуске программы он выводил старую информацию. Что бы решить эту проблему надо было что бы наша база данных всегда запускала более позднюю версию.

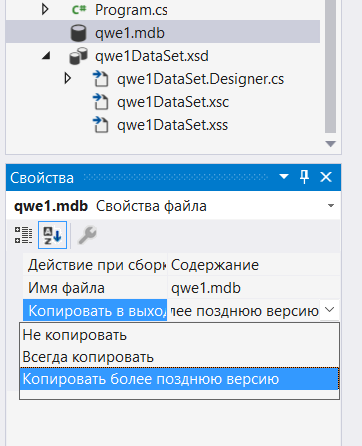


Рисунок 82 - Решение первой проблемы

* Второй ошибкой было то, что обработчик добавить не работал и было не понятно, из-за чего он не хотел функционировать. Решением проблемы стало то, что надо было на форме где таблица в form.designer.cs изменить 4 строки с private на public

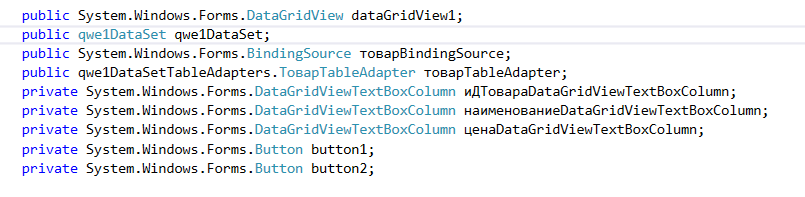


Рисунок 83 - Решение второй проблемы

* Третьей ошибкой снова стал обработчик добавить. Ошибкой служило то, что он добавлял данные не в ту таблицу, а иногда не хотел вообще работать так как не хватает столбцов или не правильный формат столбцов. Решением стало то, что надо изменять счёт обработчика на +1 с каждым новым обработчиком.

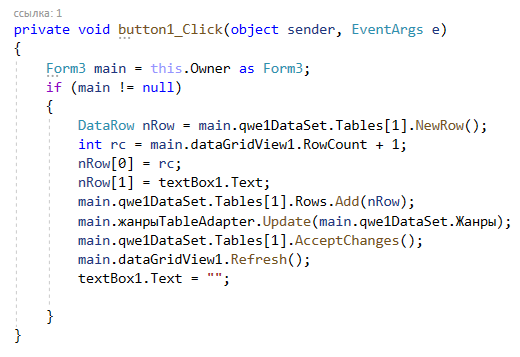


Рисунок 84 - Решение третьей проблемы часть 1

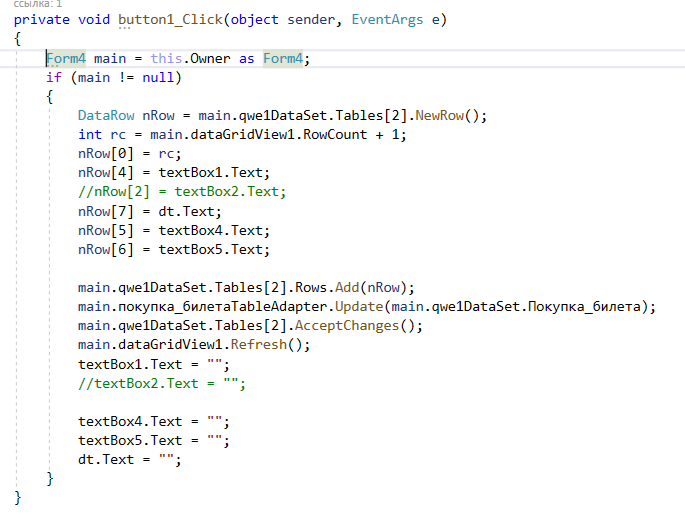


Рисунок 85 - Решение третьей проблемы часть 2

* Четвёртой проблемой стал снова обработчик добавить. Проблемой служило то, что не должны быть в таблице индикаторы вторичных ключей и поэтому надо было удалять и заново создавать атрибут с этими же данными, но другого формата. И поэтому он менял свою поочередность. Это и стало проблемой ведь таблица считает по атрибутам базы данных из access и поэтому в обработчике добавить выходили ошибки на счёт не того атрибута, а из-за это не могла водится данные из-за неправильного формата ячейки. Решением проблемы стало то, что надо было смотреть поочередность атрибутов а access и затем писать правильны обработчик добавить
* Пятой ошибкой стало то, когда создавал обработчик поиск не менял форму в которой находилась таблица и из-за этого он не хотел работать. Решением проблемы стало то, что надо было выбрать нужную форму, в которой он будет искать данные.

Вывод: в ходе работы была создана информационная система, подключена база данных, реализованы основные функции добавления, изменения, удаления данных, а также устранены ошибки в программном коде, возникающих в процессе создания ИС.