**СОДЕРЖАНИЕ**

[**ВВЕДЕНИЕ** 2](#_Toc135664772)

[**1 Постановка задачи и анализ предметной области** 3](#_Toc135664773)

[**2 Назначение проектируемой базы данных** 5](#_Toc135664774)

[**3 Проектирование инфологической модели данных** 6](#_Toc135664775)

[**3.1 Информационные потребности пользователя (анализ запросов)** 6](#_Toc135664776)

[**3.2 Определение сущностей и связей** 6](#_Toc135664777)

[**3.3 Определение сущности, атрибутов, ключей** 7](#_Toc135664778)

[**3.4 Выявление и описание ограничений целостности** 9](#_Toc135664779)

[**3.5 Разработка инфологической модели предметной области** 12](#_Toc135664780)

[**3.6 Доказательство того, что все отношения (таблицы) находятся в 1-ой, 2-ой, 3-ей нормальных формах** 15](#_Toc135664781)

[**4 Проектирование даталогической модели БД** 17](#_Toc135664782)

[**4.1 Выбор СУБД** 17](#_Toc135664783)

[**4.2 Отображение инфологической модели на даталогическую модель** 17](#_Toc135664784)

[**4.3 Схема даталогической модели базы данных (схема данных)** 24](#_Toc135664785)

[**5 Реализация БД** 25](#_Toc135664786)

[**5.1 Разработка средств реализации ограничений целостности** 25](#_Toc135664787)

[**5.2 Разработка процедур ведения БД** 26](#_Toc135664788)

[**5.2.1 Представления** 26](#_Toc135664789)

[**5.2.2 Функции** 29](#_Toc135664790)

[**5.2.3 Триггеры** 32](#_Toc135664791)

[**5.2.4 Процедуры** 34](#_Toc135664792)

[**ЗАКЛЮЧЕНИЕ** 37](#_Toc135664793)

[**СПИСОК ЛИТЕРАТУРЫ** 38](#_Toc135664794)

[**ПРИЛОЖЕНИЕ А** 39](#_Toc135664795)

[**ПРИЛОЖЕНИЕ Б** 43](#_Toc135664796)

# **ВВЕДЕНИЕ**

В настоящее время практически невозможно найти организацию или предприятие, которое не использовало бы в своей деятельности программное обеспечение. Поток информации достаточно велик, и для быстроты и легкости в работе с ней зачастую используют базы данных. Именно база данных позволяет хранить неограниченное количество информации, легко редактировать и постоянно дополнять ее. Основным объектом нашего рассмотрения является ремонт компьютерной техники. Большинство организаций имеют кучу данных по поставке и продаже товаров. Примером реальной реализации может служить любой сервис по ремонту компьютерной техники.

В процессе работы над проектом будут рассмотрены основные принципы проектирования баз данных, выбор подходящих инструментов для реализации проекта и создание функционала для работы с базой данных. Таким образом целью курсового проекта является проектирование базы данных предметной области «Ремонт компьютерной техники». В качестве средства разработки используется СУБД – SQL Server 2022.

# **1 Постановка задачи и анализ предметной области**

Целью курсового проекта является проектирование базы данных предметной области «Ремонт компьютерной техники». Подсистема «Обработка заказов на ремонт».

Ремонт компьютерной техники – это сервис по ремонту, где любой желающий, представив необходимую информацию о себе и о своих устройствах, может сдать технику на ремонт, также, при необходимости, заказать нужные для ремонта детали через сервис. Для реализации БД необходимо:

1. Оформление ремонта, в которое входит:

— Информация о устройстве,

— Информация о сотруднике, который производит ремонт,

— Информация о дате начала ремонта,

— Информация о дате окончания ремонта,

— Информация о стоимости ремонта,

— Информация о проблеме,

— Информация о проделанной работе.

Для реализации данных задач необходимо выполнить следующие действия:

* 1. Создать процедуру для заполнения информации о клиенте.
  2. Создать процедуру для заполнения информации о устройстве.
  3. Создать процедуру для заполнения информации о сотруднике.
  4. Создать процедуру для заполнения информации о комплектующем.
  5. Создать процедуру для заполнения информации о поставщике.

Также необходимо реализовать выдачу основной информации по ремонтам устройств при помощи представления.

1. Необходимо указать дату заказа

Для реализации данных задач необходимо выполнить следующие действия:

Можно использовать процедуру для заполнения всей информации по заказу ремонта.

Еще необходимо выдавать численную информацию по определенным запросам.

1. Нужно реализовать функцию для работы получения численных значений.

Для реализации данных задач необходимо выполнить следующие действия:

* 1. Создать функцию, которая будет считать суммарную стоимость заказов ремонтов каждого клиента.

Также создать и другие функции, различные запросы, процедуры и представления, которые могут пригодиться в работе с заказами.

# **2 Назначение проектируемой базы данных**

Спроектированная БД предназначена для автоматизированного учета работы «Обработки заказов на ремонт», которая содержит следующие данные:

— Информация о клиенте

— Информация о устройстве

— Информация о ремонте

— Информация о сотруднике

Данная разработка должна использоваться для ПО ОС Windows 10/11, СУБД SQL Server 2022. Для ПК не ниже Intel Core i3.

# **3 Проектирование инфологической модели данных**

# **3.1 Информационные потребности пользователя (анализ запросов)**

При разработке данного курсового проекта была выбрана предметная область «Ремонт компьютерной техники», подсистема «Обработка заказов на ремонт».

В ней необходимо отразить:

— Данные о клиенте (Имя, фамилия, отчество, адрес, номер телефона, электронная почта);

— Данные об устройстве (Модель, производитель, тип устройства, год выпуска, серийный номер, клиент);

— Данные о ремонтах (Устройство, сотрудник, дата приёма, дата окончания, стоимость ремонта, описание проблемы, описание проделанной работы);

— Данные о сотруднике (Имя, фамилия, отчество, стаж, должность, зарплата, дата приема на работу);

— Данные о заказе комплектующих (Клиент, комплектующее, количество, стоимость, дата заказа, статус заказа);

— Данные о комплектующих (запчастях) (Название, описание, производитель, стоимость, поставщик);

— Данные о поставщиках (Название компании, контактное лицо, номер телефона, адрес, электронная почта);

# **3.2 Определение сущностей и связей**

**Сущность** — любой различимый объект (который мы можем различить от другого), информация о котором должна быть представлена в БД.

*Пример:* Сотрудники

**Экземпляр сущности** — это информация о конкретном представителе объекта.

*Пример*: Сотрудник «Калинин Иван Сергеевич»

**Связь** — соединение между двумя и более сущностями.

**Экземпляр связи** — конкретная связь между конкретными представителями объектов.

Сущности, представленные в данном курсовом проекте:

— Клиенты (информация о клиентах);

— Устройства (информация об устройствах);

— Ремонты (информация о ремонтах);

— Сотрудники (информация о сотрудниках);

— Заказы комплектующих (информация о заказах комплектующих);

— Комплектующие/Запчасти (информация о комплектующих (запчастях));

— Поставщики (информация о поставщиках).

# **3.3 Определение сущности, атрибутов, ключей**

Каждая сущность обладает набором атрибутов.

**Атрибут** — отдельная характеристика сущности. Его наименование должно быть уникальным для конкретного типа сущности, но может быть одинаковым для различного типа сущностей. Атрибуты используются для определения того, какая информация должна быть собрана о сущности. *Пример*: Имя клиента, номер телефона клиента

Сущность состоит из экземпляров, каждый из которых должен отличаться от другого экземпляра.

**Ключ** — это минимальный набор атрибутов, по значениям которых можно однозначно найти требуемый экземпляр сущности. Минимальность означает что исключение из набора любого атрибута не позволяет идентифицировать сущность по оставшимся.

*Пример*: для сущности «Клиенты» ключом является атрибут «Код»

Каждая сущность обладает хотя бы одним возможным ключом. Один из них принимается за **первичный ключ**. Если все возможные ключи являются длинными и поэтому неудобными для идентификации экземпляров, в сущность вводят дополнительный атрибут с ролью первичного ключа.

Первичный ключ из одной сущности можно добавить в другую сущность для создания связи между ними. В другой таблице он называется **внешним ключом**. Первичный ключ сущности позволяет идентифицировать ее экземпляры, а внешний — экземпляры сущности, которая находится в связи с данной сущностью.

*Пример*: сущность «Клиенты» имеет первичный ключ «Код», данный ключ будет являться внешним для сущности «Устройства», которая имеет атрибут «Код клиента».

Ключи и атрибуты в данном курсовом проекте:

— Сущность **Клиенты** содержит атрибуты: Код, Имя, Фамилия, Отчество, Адрес, Номер телефона, Электронная почта. Код является ключом.

— Сущность **Устройства** содержит атрибуты: Код, Модель, Производитель, Тип устройства, Год выпуска, Серийный номер, Код клиента. Код является ключом.

— Сущность **Ремонты** содержит атрибуты: Код, Код устройства, Код сотрудника, Дата приёма, Дата окончания, Стоимость ремонта, Описание проблемы, Описание проделанной работы. Код является ключом.

— Сущность **Сотрудники** содержит атрибуты: Код, Имя, Фамилия, Отчество, Стаж, Должность, Зарплата, Дата приёма на работу. Код является ключом.

— Сущность **Заказы комплектующих** содержит атрибуты: Код, Код клиента, Код комплектующего, Количество, Стоимость, Дата заказа, Статус заказа. Код является ключом.

— Сущность **Комплектующие/Запчасти** содержит атрибуты: Код, Название, Описание, Производитель, Стоимость, Код поставщика. Код является ключом.

— Сущность **Поставщики** содержит атрибуты: Код, Название компании, Контактное лицо, Номер телефона, Адрес, Электронная почта. Код является ключом.

# **3.4 Выявление и описание ограничений целостности**

Под целостностью данных понимаются ссылочные ограничения, т.е. те ограничения, которые нужно соблюдать для сохранения целостности связи между таблицами, в случае если в них будут изменяться или удаляться записи.

Для обеспечения целостности данных в SQL Server есть 4 варианта:

1. Если не указано каскадное обновление связей, то предотвращается изменение значений первичного ключа в главной таблице, если существуют связанные записи в подчиненной таблице.

2. Если указано каскадное обновление связей, то при изменении значений первичного ключа будут изменяться соответствующие значения в связанной таблице.

3. Если не указано каскадное удаление связанных записей, то предотвращается удаление связанных записей из главной таблицы, если имеются связанные с ней записи в подчиненной.

4. Если указано каскадное удаление, то связанные записи подчиненной таблицы удаляются автоматически.

В данном курсовом проекте у каждой связи установлены такие параметры как каскадное обновление и каскадное удаление связей. Свойства этих параметров описаны выше.

Также к ограничениям целостности можно отнести ограничения на столбец и на таблицу, а точнее на значения данных в них. К таким ограничениям можно отнести следующие:

• Запрещение null значения: данные, заносимые в столбец или таблицу, не должны равняться нулю.

• Ограничения на допустимые значения полей: условие, которому должны удовлетворять данные, вносимые в таблицу.

• Ограничение первичного ключа: на практике рекомендуется для каждой таблицы создавать первичный ключ, особенностью которого является не допуск null значения.

• Ограничение уникальных ключей: необходимость ввода различных уникальных) данных.

В данном курсовом проекте используются следующие ограничения данных в таблицах:

1. Таблица Клиенты

Поля Код, Имя, Фамилия, Адрес, Номер телефона, Электронная почта являются обязательными.

В поле **Имя** используется ограничение по кол-ву символов — 50.

В поле **Фамилия** используется ограничение по кол-ву символов — 50.

В поле **Отчество** используется ограничение по кол-ву символов — 50.

В поле **Адрес** используется ограничение по кол-ву символов — 50.

В поле **Номер телефона** используется ограничение по кол-ву символов — 50.

В поле **Электронная почта** используется ограничение по кол-ву символов — 50.

1. Таблица Устройства

Поля Код, Модель, Производитель, Тип устройства, Год выпуска, Код клиента являются обязательными.

В поле **Модель** используется ограничение по кол-ву символов — 50.

В поле **Производитель** используется ограничение по кол-ву символов — 50.

В поле **Тип устройства** используется ограничение по кол-ву символов — 50.

1. Таблица Ремонты

Поля Код, Код устройства, Код сотрудника, Дата приёма, Дата окончания, Стоимость ремонта, Описание проблемы, Описание проделанной работы являются обязательными.

В поле **Описание проблемы** используется ограничение по кол-ву символов — 500.

В поле **Описание проделанной работы** используется ограничение по кол-ву символов — 500.

1. Таблица Сотрудники

Поля Код, Имя, Фамилия, Должность, Зарплата, Дата приёма на работу являются обязательными.

В поле **Имя** используется ограничение по кол-ву символов — 50.

В поле **Фамилия** используется ограничение по кол-ву символов — 50.

В поле **Отчество** используется ограничение по кол-ву символов — 50.

В поле **Должность** используется ограничение по кол-ву символов — 50.

1. Таблица Заказы комплектующих

Поля Код, Код клиента, Код комплектующего, Количество, Стоимость, Дата заказа, Статус заказа являются обязательными.

В поле **Количество** используется ограничение по кол-ву символов — 50.

В поле **Статус заказа** используется ограничение по кол-ву символов — 50.

1. Таблица Комплектующие/Зачасти

Поля Код, Название, Описание, Производитель, Стоимость, Код поставщика являются обязательными.

В поле **Название** используется ограничение по кол-ву символов — 50.

В поле **Описание** используется ограничение по кол-ву символов — 250.

В поле **Производитель** используется ограничение по кол-ву символов — 50.

1. Таблица Поставщики

В поле **Название компании** используется ограничение по кол-ву символов — 50.

В поле **Контактное лицо** используется ограничение по кол-ву символов — 50.

В поле **Номер телефона** используется ограничение по кол-ву символов — 50.

В поле **Адрес** используется ограничение по кол-ву символов — 50.

В поле **Электронная почта** используется ограничение по кол-ву символов — 50.

# **3.5 Разработка инфологической модели предметной области**

**Инфологическая модель** описывает предметную область на содержательном уровне.

Результатом этого анализа являются списки объектов предметной области, перечни свойств, или атрибутов, определение связей между объектами и описание структуры предметной области в виде диаграммы.

Определим связи данной предметной области на этапе разработки инфологической модели.

Связь между сущностями можно охарактеризовать степенью связи и классом принадлежности сущности к связи. Где степень связи показывает, сколько экземпляров одной сущности могут быть связано с каждым экземпляром другой сущности, и может иметь три значения:

— Один к одному (1:1) - означает. Что каждый экземпляр первой сущности может быть связан только с одним экземпляром второй сущности и наоборот.

— Один ко многим (1:М или М:1) означает, что каждый экземпляр первой сущности может быть связан с несколькими экземплярами второй сущности, а каждый экземпляр второй сущности может быть связан только с одним экземпляром первой сущности.

— Многие ко многим (M:N) означает, что каждый экземпляр первой сущности может быть связан с несколькими экземплярами второй сущности и наоборот.

Класс принадлежности сущности к связи может быть обязательным (каждый экземпляр сущности обязательно должен быть связан с другой сущностью) и необязательным (каждый экземпляр сущности не требует связи с экземпляром другой сущности).

В данном курсовом проекте используются односторонние связи.

1. Для создания отношений по односторонней связи “один – ко – многим” требуется сформировать 2 отношения: по одному для каждой сущности (таблицы «Клиенты» и «Устройства») с первичным ключом, соответствующим сущности. Сущности «Клиенты» и «Устройства» являются обязательными. Данное отношение отображено на рисунке 1.

я

**Рисунок 1** — Связи сущностей «Устройства» и «Клиенты»

Устройства М

1 Клиенты

Имеет

2. Для создания отношений по односторонней связи “один – ко – многим” требуется сформировать 2 отношения: по одному для каждой сущности (таблицы «Устройства» и «Ремонты») с первичным ключом, соответствующим сущности. Сущности «Устройства» и «Ремонты» являются обязательными. Данное отношение отображено на рисунке 2.

**Рисунок 2** — Связи сущностей «Ремонты» и «Устройства»

Ремонты М

1 Устройства

Имеет

3. Для создания отношений по односторонней связи “многие – ко – многим” требуется сформировать 2 отношения: по одному для каждой сущности (таблицы «Сотрудники» и «Ремонты») с первичным ключом, соответствующим сущности. Сущности «Ремонты» и «Сотрудники» являются обязательными. Данное отношение отображено на рисунке 3.

**Рисунок 3** — Связи сущностей «Ремонты» и «Сотрудники»

Ремонты М

1 Сотрудники

Имеет

4. Для создания отношений по односторонней связи “многие – ко – многим” требуется сформировать 2 отношения: по одному для каждой сущности (таблицы «Клиенты» и «Заказы комплектующих») с первичным ключом, соответствующим сущности. Сущности «Клиенты» и «Заказы комплектующих» являются обязательными. Данное отношение отображено на рисунке 4.

**Рисунок 4** — Связи сущностей «Заказы комплектующих» и «Клиенты»

Заказы комплектующих М

1 Клиенты

Имеет

5. Для создания отношений по односторонней связи “многие – ко – многим” требуется сформировать 2 отношения: по одному для каждой сущности (таблицы «Заказы комплектующих» и «Комплектующие/Запчасти») с первичным ключом, соответствующим сущности. Сущности «Заказы комплектующих» и «Комплектующие/Запчасти» являются обязательными. Данное отношение отображено на рисунке 5.

Заказы комплектующих М

1 Комплектую-

щие/Запчасти

Имеет

**Рисунок 5** — Связи сущностей «Комплектующие/Запчасти» и «Заказы комплектующих»

6. Для создания отношений по односторонней связи “многие – ко – многим” требуется сформировать 2 отношения: по одному для каждой сущности (таблицы «Комплектующие/Запчасти» и «Поставщики») с первичным ключом, соответствующим сущности. Сущности «Комплектующие/Запчасти» и «Поставщики» являются обязательными. Данное отношение отображено на рисунке 6.

**Рисунок 6** — Связи сущностей «Поставки» и «Поставщики»

Комплектую- М

щие/Запчасти

1 Поставщики

Имеет



**Рисунок 7**— Расширенная ER-диаграмма

# **3.6 Доказательство того, что все отношения (таблицы) находятся в 1-ой, 2-ой, 3-ей нормальных формах**

Для того чтобы таблица считалась нормализованной к первой нормальной форме, каждое из ее полей должно быть неделимым (атомарным), и таблица не должна содержать никаких повторяющихся групп полей.

Для того чтобы привести таблицу ко второй нормальной форме, нужно, чтобы она удовлетворяла первой нормальной форме и, чтобы все не ключевые поля полностью зависели от первичного ключа таблицы и от каждого поля в первичном ключе, если последний состоит из нескольких полей. Это значит, что каждое не ключевое поле должно уникально определяться первичным 14 ключом и полями, его составляющими. Второй нормальной форме удовлетворяют все таблицы.

Для того чтобы таблица была приведена к третьей нормальной форме, нужно, чтобы она удовлетворяла второй нормальной форме и все не ключевые поля полностью зависели от первичного ключа таблицы и не зависели друг от друга. Таким образом, к квалификации второй нормальной Формы добавляется требование независимости каждого не ключевого поля таблицы от других не ключевых полей. Нужно исключить из таблицы также поля, которые можно вычислить по другим не ключевым полям таблицы. В нашем случае третьей нормальной форме удовлетворяют все таблицы.

# **4 Проектирование даталогической модели БД**

# **4.1 Выбор СУБД**

Для создания, введения и осуществления возможности коллективного пользования базами данных средства, используются программные системами управления базами данных (СУБД). Система называемые управления базами данных представляет собой совокупность языковых и программных средств, с помощью которых база данных создается и поддерживается. На данный момент существует множество языков, с помощью которых можно создавать различные структуры и вводить в них необходимые элементы управления. Из всего множества СУБД была выбрана SQL Server 2022.

# **4.2 Отображение инфологической модели на даталогическую модель**

**Даталогическая модель** описывает объекты и связи предметной области на формальном уровне. Ее разработка основывается на инфологической модели. В процессе разработки осуществляется выбор модели данных, и определяются ее элементы.

Учитывая выбранную СУБД и разработанную инфологическую модель предметной области, была разработана следующая даталогическая модель:

— **Клиенты**:

Код клиента (Код), Имя клиента (Имя), Фамилия клиента (Фамилия), Отчество клиента (Отчество), Адрес, Номер телефона, Электронная почта.

— **Устройства**:

Код устройства (Код), Модель устройства (Модель), Производитель, Тип устройства, Год выпуска, Серийный номер, Код клиента.

— **Ремонты**:

Код ремонта (Код), Код устройства, Код сотрудника, Дата приёма устройства на ремонт (Дата приёма), Дата окончания ремонта (Дата окончания), Стоимость ремонта, Описание проблемы, Описание проделанной работы.

— **Сотрудники**:

Код сотрудника (Код), Имя сотрудника (Имя), Фамилия сотрудника (Фамилия), Отчество сотрудника (Отчество), Стаж, Должность, Зарплата, Дата приёма на работу.

— **Заказы комплектующих**:

Код заказа (Код), Код клиента, Код комплектующего, Количество, Стоимость, Дата заказа, Статус заказа.

— **Комплектующие/Запчасти**:

Код, Название, Описание, Производитель, Стоимость, Код поставщика.

— **Поставщики**:

Код, Название компании, Контактное лицо, Номер телефона, Адрес, Электронная почта.

Таблица 1 — Клиенты

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Имя столбца | Содержательное описание | Тип данных | Размерность | Область допустимых значений | Возможность значения Null | Роль | Пример |
| Код | Код клиента | Целый | 4 | 0001-9999 | нет | PK | 1 |
| Имя | Имя клиента | Символьный | 50 | «А-Я» 0001-9999 | нет |  | Егор |
| Фамилия | Фамилия клиента | Символьный | 50 | «А-Я» 0001-9999 | нет |  | Давыдов |
| Отчество | Отчество клиента | Символьный | 50 | «А-Я» 0001-9999 | да |  | Тимофеевич |
| Адрес | Адрес клиента | Символьный | 50 | «А-Я» 0001-9999 | нет |  | г. Рязань ул. Крупской д. 11 |
| Номер телефона | Номер телефона клиента | Символьный | 50 | «А-Я» 0001-9999 | нет |  | 8(927)973-61-56 |
| Электронная почта | Электронная почта клиента | Символьный | 50 | «А-Я» 0001-9999 | нет |  | Davidov@mail.ru |

Таблица 2 — Устройства

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Имя столбца | Содержательное описание | Тип данных | Размерность | Область допустимых значений | Возможность значения Null | Роль | Пример |
| Код | Код устройства | Целый | 4 | 0001-9999 | нет | PK | 1 |
| Модель | Модель устройства | Символьный | 50 | «А-Я» 0001-9999 | нет |  | P2500 |
| Производитель | Производитель устройства | Символьный | 50 | «А-Я» 0001-9999 | нет |  | Pantum |
| Тип устройства | Тип устройства | Символьный | 50 | «А-Я» 0001-9999 | нет |  | Принтер лазерный |
| Год выпуска | Год выпуска устройства | Цельный | 4 | 0001-9999 | нет |  | 2015 |
| Серийный номер | Серийный номер устройства | Целый | 4 | 0001-9999 | да |  | 3254295 |
| Код клиента | Код клиента | Целый | 4 | 0001-9999 | нет | FK | 1 |

Таблица 3 — Ремонты

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Имя столбца | Содержательное описание | Тип данных | Размерность | Область допустимых значений | Возможность значения Null | Роль | Пример |
| Код | Код ремонта | Целый | 4 | 0001-9999 | нет | PK | 1 |
| Код устройства | Код устройства | Целый | 4 | 0001-9999 | нет | FK | 1 |
| Код сотрудника | Код сотрудника | Целый | 4 | 0001-9999 | нет | FK | 1 |
| Дата приёма | Дата приёма устройства на ремонт | Дата | 10 |  | нет |  | 2023-05-01 |
| Дата окончания | Дата окончания ремонта | Дата | 10 |  | нет |  | 2023-05-04 |
| Стоимость ремонта | Стоимость ремонта | Деньги | 24 | -922,337,203,685,477.5808  -922,337,203,685,477.5807 | нет |  | 2000,0000 |
| Описание проблемы | Описание проблемы | Символьный | 500 | «А-Я» 0001-9999 | нет |  | Принтер не включается. |
| Описание проделанной работы | Описание проделанной работы | Символьный | 500 | «А-Я» 0001-9999 | нет |  | Замена и перепайка внутренних микросхем. |

Таблица 4 — Сотрудники

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Имя столбца | Содержательное описание | Тип данных | Размерность | Область допустимых значений | Возможность значения Null | Роль | Пример |
| Код | Код сотрудника | Целый | 4 | 0001-9999 | нет | PK | 1 |
| Имя | Имя сотрудника | Символьный | 50 | «А-Я» 0001-9999 | нет |  | Иван |
| Фамилия | Фамилия сотрудника | Символьный | 50 | «А-Я» 0001-9999 | нет |  | Калинин |
| Отчество | Отчество сотрудника | Символьный | 50 | «А-Я» 0001-9999 | да |  | Сергеевич |
| Стаж | Стаж сотрудника | Целый | 4 | 0001-9999 | да |  | 8 |
| Должность | Должность сотрудника | Символьный | 50 | «А-Я» 0001-9999 | нет |  | Электрик |
| Зарплата | Зарплата сотрудника | Деньги | 24 | -922,337,203,685,477.5808  -922,337,203,685,477.5807 | нет |  | 34000,0000 |
| Дата приёма на работу | Дата приёма на работу | Дата | 10 |  | нет |  | 2021-04-01 |

Таблица 5 — Заказы комплектующих

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Имя столбца | Содержательное описание | Тип данных | Размерность | Область допустимых значений | Возможность значения Null | Роль | Пример |
| Код | Код заказа | Целый | 4 | 0001-9999 | нет | PK | 1 |
| Код клиента | Код клиента | Целый | 4 | 0001-9999 | нет | FK | 1 |
| Код комплектующего | Код комплектующего | Целый | 4 | 0001-9999 | нет | FK | 1 |
| Количество | Количество комплектующих | Символьный | 50 | «А-Я» 0001-9999 | нет |  | 1 штука |
| Стоимость | Стоимость заказа | Деньги | 24 | -922,337,203,685,477.5808  -922,337,203,685,477.5807 | нет |  | 1000,0000 |
| Дата заказа | Дата заказа | Дата | 10 |  | нет |  | 2023-03-01 |
| Статус заказа | Статус заказа | Символьный | 50 | «А-Я» 0001-9999 | нет |  | Получен |

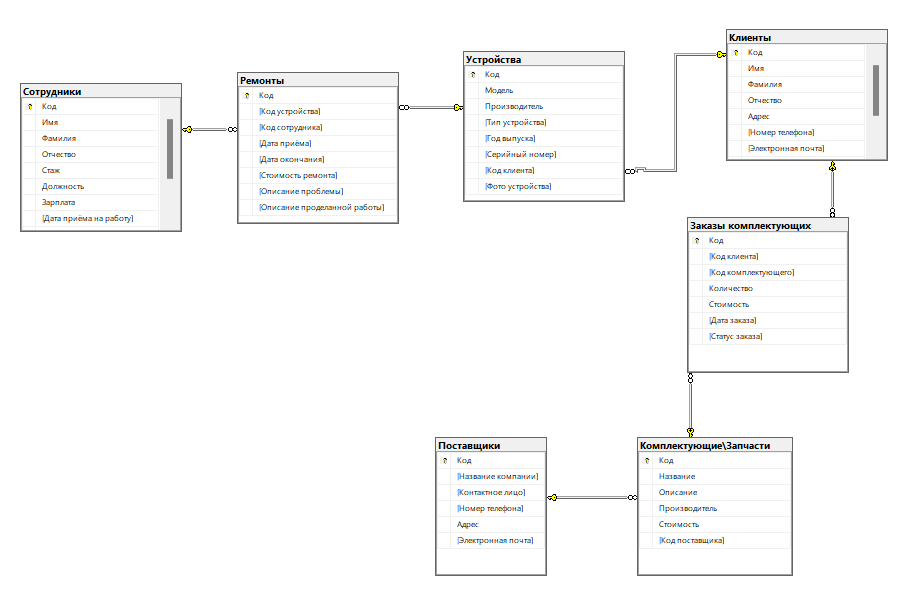
Таблица 6 — Комплектующие/Запчасти

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Имя столбца | Содержательное описание | Тип данных | Размерность | Область допустимых значений | Возможность значения Null | Роль | Пример |
| Код | Код комплектующего | Целый | 4 | 0001-9999 | нет | PK | 1 |
| Название | Название комплектующего | Символьный | 50 | «А-Я» 0001-9999 | нет |  | Аккумулятор для ноутбуков HP |
| Описание | Описание комплектующего | Символьный | 250 | «А-Я» 0001-9999 | да |  | Литиевый аккумулятор, подходящий для большинства моделей ноутбуков HP |
| Производитель | Производитель | Символьный | 50 | «А-Я» 0001-9999 | нет |  | HP |
| Стоимость | Стоимость | Деньги | 24 | -922,337,203,685,477.5808  -922,337,203,685,477.5807 | нет |  | 1000,0000 |
| Код поставщика | Код поставщика | Целый | 4 | 0001-9999 | нет | FK | 1 |

Таблица 7 — Поставщики

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Имя столбца | Содержательное описание | Тип данных | Размерность | Область допустимых значений | Возможность значения Null | Роль | Пример |
| Код | Код поставщика | Целый | 4 | 0001-9999 | нет | PK | 1 |
| Название компании | Название компании | Символьный | 50 | «А-Я» 0001-9999 | нет |  | MARVEL Дистрибуция |
| Контактное лицо | Контактное лицо | Символьный | 50 | «А-Я» 0001-9999 | да |  | Рузаев Андрей Викторович |
| Номер телефона | Номер телефона | Символьный | 50 | «А-Я» 0001-9999 | да |  | 8(978)876-23-41 |
| Адрес | Адрес | Символьный | 50 | «А-Я» 0001-9999 | нет |  | г. Москва, ул. Краснобогатырская, д. 89 стр. 1 |
| Электронная почта | Электронная почта | Символьный | 50 | «А-Я» 0001-9999 | нет |  | info@marvel.ru |

# **4.3 Схема даталогической модели базы данных (схема данных)**



**Рисунок 8** — Схема данных

# **5 Реализация БД**

# **5.1 Разработка средств реализации ограничений целостности**

В любой момент времени БД имеет некоторую определенную конфигурацию значений данных, которые отражают действительность, т.е. являются частью реального мира. Просто определить конфигурацию значений не имеет смысла без связи с внешним миром. Поэтому требуется уточнить определение БД, включив в него правила целостности, которые необходимы для информирования СУБД о, различного рода, ограничениях реального мира с целью не допустить «абсурдных» значений данных;

Для любого отношения можно создать ряд правил. ограничений. Каждая конкретная БД должна иметь свои ограничения, связанные с предметной областью, которые накладываются на хранящиеся в ней данные. К таким ограничениям целостности относятся:

1. Ограничения на атрибуты (тип атрибута, диапазон допустимых значений).

2. Число кортежей отношения должно быть равно числу первичных ключей (наличие кортежей дубликатов не допускается).

Первое ограничение накладывается на атрибуты всех отношений на этапе определения типа атрибута.

Второе ограничение накладывается на отношения на этапе заполнения таблиц данными о БД

Существует также два общих правила целостности. Они касаются потенциальных и внешних ключей:

1. Первичный ключ является уникальным идентификатором отношения. Не допускается, чтобы какой-либо атрибут, участвующий в первичном ключе, принимал неопределенное значение. В отношении не может быть несколько кортежей с одинаковыми значениями первичного ключа.

Потенциальный ключ отношения не может иметь пустого значения (NULL.). Так как объект, не имеющий идентичности, не существует.

2. Если 12 - некоторое отношение с внешним ключом х, то должно существовать такое базовое отношение rl с первичным ключом К, что каждое значение X в 12 совпадает со значением К в каком-либо кортеже отношения г1.

В процессе создания БД сначала осуществляется конструирование таблиц, далее создается схема данных, в которой фиксируются связи между таблицами. В этой схеме могут быть заданы параметры обеспечения целостности базы данных, если модель была разработана в соответствии с требованиями нормализации. Целостность данных означает, что в БД установлены и корректно поддерживаются взаимосвязи между записями разных таблиц при их загрузке, добавлении и удалении в связанных таблицах, а также при изменении значений ключевых полей.

# **5.2 Разработка процедур ведения БД**

# **5.2.1 Представления**

1. **AccessorsBySuppliers** (Комплектующие по поставщикам)

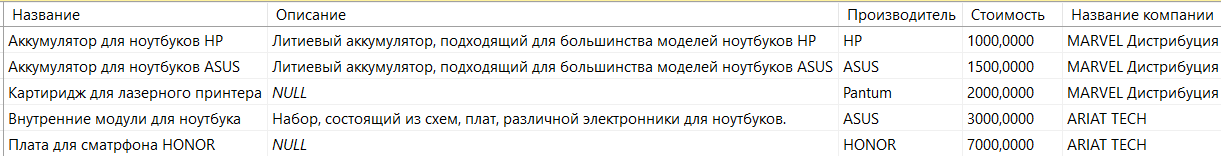
*Формирование представления:*

SELECT dbo.[Комплектующие\Запчасти].Название, dbo.[Комплектующие\Запчасти].Описание, dbo.[Комплектующие\Запчасти].Производитель, dbo.[Комплектующие\Запчасти].Стоимость, dbo.Поставщики.[Название компании]

FROM dbo.[Комплектующие\Запчасти]

INNER JOIN dbo.Поставщики

ON dbo.[Комплектующие\Запчасти].[Код поставщика] = dbo.Поставщики.Код



**Рисунок 9** — Результат выполнения AccessorsBySuppliers

2. **AllInfAboutRepair** (Полная информация о ремонтах)

*Формирование представления:*

SELECT dbo.Клиенты.Имя, dbo.Клиенты.Фамилия, dbo.Клиенты.Отчество, dbo.Устройства.[Тип устройства], dbo.Ремонты.[Дата приёма], dbo.Ремонты.[Дата окончания], dbo.Ремонты.[Стоимость ремонта],

dbo.Ремонты.[Описание проблемы], dbo.Ремонты.[Описание проделанной работы], dbo.Сотрудники.Имя AS [Имя сотрудника], dbo.Сотрудники.Фамилия AS [Фамилия сотрудника],

dbo.Сотрудники.Отчество AS [Отчество сотрудника]

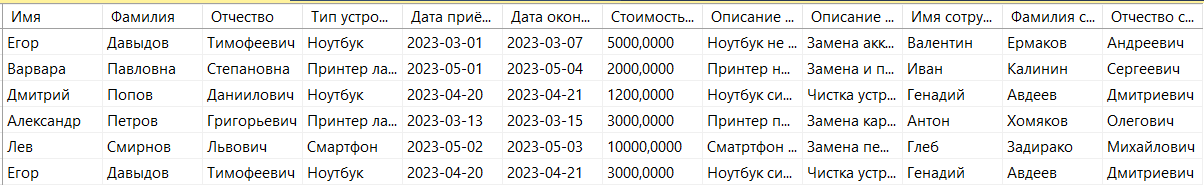
FROM dbo.Устройства

INNER JOIN dbo.Клиенты ON dbo.Устройства.[Код клиента] = dbo.Клиенты.Код

INNER JOIN dbo.Сотрудники

INNER JOIN dbo.Ремонты ON dbo.Сотрудники.Код = dbo.Ремонты.[Код сотрудника]

ON dbo.Устройства.Код = dbo.Ремонты.[Код устройства]



**Рисунок 10** — Результат выполнения AllInfAboutRepair

3. **AllInfAdoutOrdersAccessories** (Полная информация о заказах комплектующих)

*Формирование представления:*

SELECT dbo.Клиенты.Имя, dbo.Клиенты.Фамилия, dbo.Клиенты.Отчество, dbo.[Комплектующие\Запчасти].Название, dbo.[Комплектующие\Запчасти].Описание, dbo.[Комплектующие\Запчасти].Производитель,

dbo.[Комплектующие\Запчасти].Стоимость, dbo.Поставщики.[Название компании], dbo.[Заказы комплектующих].Количество, dbo.[Заказы комплектующих].Стоимость AS [Стоимость заказа],

dbo.[Заказы комплектующих].[Дата заказа], dbo.[Заказы комплектующих].[Статус заказа]

FROM dbo.[Заказы комплектующих]

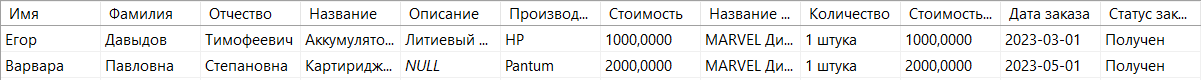
INNER JOIN dbo.Клиенты

ON dbo.[Заказы комплектующих].[Код клиента] = dbo.Клиенты.Код

INNER JOIN dbo.[Комплектующие\Запчасти] ON dbo.[Заказы комплектующих].[Код комплектующего] = dbo.[Комплектующие\Запчасти].Код

INNER JOIN dbo.Поставщики

ON dbo.[Комплектующие\Запчасти].[Код поставщика] = dbo.Поставщики.Код



**Рисунок 11** — Результат выполнения AllInfAdoutOrdersAccessories

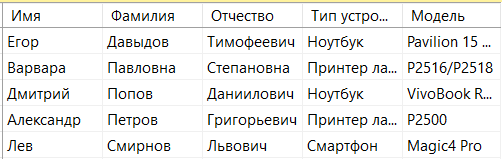
4. **DevicesByClients** (Устройства по клиентам)

*Формирование представления:*

SELECT dbo.Клиенты.Имя, dbo.Клиенты.Фамилия, dbo.Клиенты.Отчество, dbo.Устройства.[Тип устройства], dbo.Устройства.Модель

FROM dbo.Клиенты

INNER JOIN dbo.Устройства ON dbo.Клиенты.Код = dbo.Устройства.[Код клиента]



**Рисунок 12** — Результат выполнения DevicesByClients

5. **DevicesLinkedToEmployees** (Устройства по сотрудникам)

*Формирование представления:*

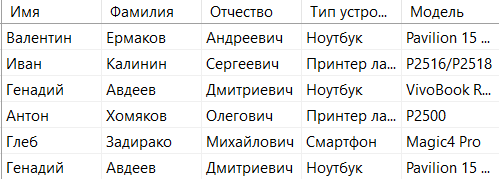
SELECT dbo.Сотрудники.Имя, dbo.Сотрудники.Фамилия, dbo.Сотрудники.Отчество, dbo.Устройства.[Тип устройства], dbo.Устройства.Модель

FROM dbo.Сотрудники

INNER JOIN dbo.Ремонты

ON dbo.Сотрудники.Код = dbo.Ремонты.[Код сотрудника]

INNER JOIN dbo.Устройства ON dbo.Ремонты.[Код устройства] = dbo.Устройства.Код



**Рисунок 13** — Результат выполнения DevicesLinkedToEmployees

# **5.2.2 Функции**

1. **GetCountOrdersClient** (Получает количество ремонтов конкретного клиента)

*Формирование функции:*

USE [Ремонт компьютерной техники]

GO

/\*\*\*\*\*\* Object: UserDefinedFunction [dbo].[GetCountOrdersClient] Script Date: 20.05.2023 15:29:18 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER FUNCTION [dbo].[GetCountOrdersClient] (@Code int)

RETURNS int

AS

BEGIN

DECLARE @Count int

SELECT @Count = COUNT(\*) FROM Ремонты

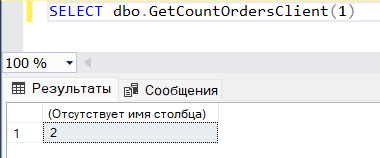
INNER JOIN Устройства ON [Код устройства] = Устройства.Код

INNER JOIN Клиенты ON [Код клиента] = Клиенты.Код

WHERE @Code = [Код устройства]

RETURN @Count

END



**Рисунок 14 -** Результат выполнения функции GetCountOrdersClient

2. **GetCountOrdersOfAccessoriesClients** (Получает количество заказов комплектующих конкретного клиента)

*Формирование функции:*

USE [Ремонт компьютерной техники]

GO

/\*\*\*\*\*\* Object: UserDefinedFunction [dbo].[GetCountOrdersOfAccessoriesClients] Script Date: 20.05.2023 15:31:47 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER FUNCTION [dbo].[GetCountOrdersOfAccessoriesClients] (@Code int)

RETURNS int

AS

BEGIN

DECLARE @Count int

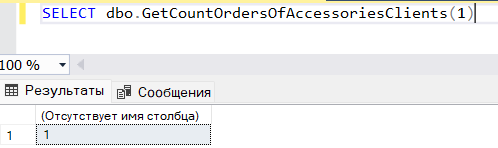
SELECT @Count = COUNT(\*) FROM [Заказы комплектующих]

INNER JOIN Клиенты ON [Код клиента] = Клиенты.Код

WHERE @Code = [Код клиента]

RETURN @Count

END



**Рисунок 15 -** Результат выполнения функции GetCountOrdersOfAccessoriesClients

3. **GetCountOrdersStaff** (Получает количество ремонтов конкретного сотрудника)

*Формирование функции:*

USE [Ремонт компьютерной техники]

GO

/\*\*\*\*\*\* Object: UserDefinedFunction [dbo].[GetCountOrdersStaff] Script Date: 20.05.2023 15:33:43 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER FUNCTION [dbo].[GetCountOrdersStaff] (@Code int)

RETURNS int

AS

BEGIN

DECLARE @Count int

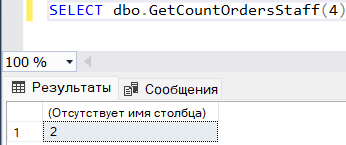
SELECT @Count = COUNT(\*) FROM Ремонты

INNER JOIN Сотрудники ON [Код сотрудника] = Сотрудники.Код

WHERE @Code = [Код сотрудника]

RETURN @Count

END



**Рисунок 16 -** Результат выполнения функции GetCountOrdersStaff

4. **GetCountTypeDevices** (Получает количество конкретного типа устройств)

*Формирование функции:*

USE [Ремонт компьютерной техники]

GO

/\*\*\*\*\*\* Object: UserDefinedFunction [dbo].[GetCountTypeDevices] Script Date: 20.05.2023 15:36:49 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER FUNCTION [dbo].[GetCountTypeDevices] (@Type nvarchar(50))

RETURNS int

AS

BEGIN

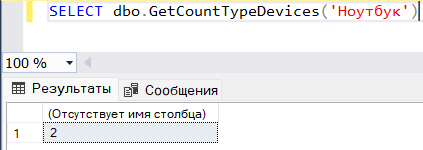
DECLARE @Count int

SELECT @Count = COUNT(\*) FROM Устройства

WHERE @Type = [Тип устройства]

RETURN @Count

END



**Рисунок 17 -** Результат выполнения функции GetCountTypeDevices

5. **GetSumOrdersClient** (Получает суммарную стоимость ремонтов конкретного клиента)

*Формирование функции:*

USE [Ремонт компьютерной техники]

GO

/\*\*\*\*\*\* Object: UserDefinedFunction [dbo].[GetSumOrdersClient] Script Date: 20.05.2023 15:39:10 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER FUNCTION [dbo].[GetSumOrdersClient] (@Code int)

RETURNS money

AS

BEGIN

DECLARE @Sum money

SELECT @Sum = SUM(Ремонты.[Стоимость ремонта]) FROM Ремонты

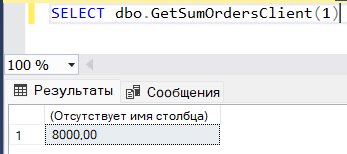
INNER JOIN Устройства ON [Код устройства] = Устройства.Код

INNER JOIN Клиенты ON [Код клиента] = Клиенты.Код

WHERE @Code = [Код клиента]

RETURN @Sum

END



**Рисунок 18 -** Результат выполнения функции GetSumOrdersClient

# **5.2.3 Триггеры**

1. **InsertedClient** (триггер, срабатывающий на добавление информации в таблицу «Клиенты», действием которого является отображения добавленной записи).

*Формирование триггера:*

USE [Ремонт компьютерной техники]

GO

/\*\*\*\*\*\* Object: Trigger [dbo].[InsertedClients] Script Date: 20.05.2023 16:14:36 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER TRIGGER [dbo].[InsertedClients]

ON [dbo].[Клиенты]

AFTER INSERT

AS

BEGIN

SET NOCOUNT ON;

SELECT \* FROM Клиенты

WHERE Клиенты.Код = (SELECT Код FROM inserted)

END

2. **InsertedAccessories** (триггер, срабатывающий на добавление информации в таблицу «Комплектующие/Запчасти», действием которого является отображения добавленной записи).

*Формирование триггера:*

USE [Ремонт компьютерной техники]

GO

/\*\*\*\*\*\* Object: Trigger [dbo].[InsertedAccessories] Script Date: 20.05.2023 16:17:01 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER TRIGGER [dbo].[InsertedAccessories]

ON [dbo].[Комплектующие\Запчасти]

AFTER INSERT

AS

BEGIN

SET NOCOUNT ON;

SELECT \* FROM [Комплектующие\Запчасти]

WHERE [Комплектующие\Запчасти].Код = (SELECT Код FROM inserted)

END

3. **InsertedSupliers** (триггер, срабатывающий на добавление информации в таблицу «Поставщики», действием которого является отображения добавленной записи).

*Формирование триггера:*

USE [Ремонт компьютерной техники]

GO

/\*\*\*\*\*\* Object: Trigger [dbo].[InsertedSuppliers] Script Date: 20.05.2023 16:18:38 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER TRIGGER [dbo].[InsertedSuppliers]

ON [dbo].[Поставщики]

AFTER INSERT

AS

BEGIN

SET NOCOUNT ON;

SELECT \* FROM Поставщики

WHERE Поставщики.Код = (SELECT Код FROM inserted)

END

4. **InsertedStaff** (триггер, срабатывающий на добавление информации в таблицу «Сотрудники», действием которого является отображения добавленной записи).

*Формирование триггера:*

USE [Ремонт компьютерной техники]

GO

/\*\*\*\*\*\* Object: Trigger [dbo].[InsertedStaff] Script Date: 20.05.2023 16:19:33 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER TRIGGER [dbo].[InsertedStaff]

ON [dbo].[Сотрудники]

AFTER INSERT

AS

BEGIN

SET NOCOUNT ON;

SELECT \* FROM Сотрудники

WHERE Сотрудники.Код = (SELECT Код FROM inserted)

END

5. **InsertedDevices** (триггер, срабатывающий на добавление информации в таблицу «Устройства», действием которого является отображения добавленной записи).

*Формирование триггера:*

USE [Ремонт компьютерной техники]

GO

/\*\*\*\*\*\* Object: Trigger [dbo].[InsertedDevices] Script Date: 20.05.2023 16:20:17 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER TRIGGER [dbo].[InsertedDevices]

ON [dbo].[Устройства]

AFTER INSERT

AS

BEGIN

SET NOCOUNT ON;

SELECT \* FROM Устройства

WHERE Устройства.Код = (SELECT Код FROM inserted)

END

# **5.2.4 Процедуры**

1. **AddAccessories** (Процедура добавления в таблицу «Комплектующие/Запчасти»)

*Формирование процедуры:*

USE [Ремонт компьютерной техники]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[AddAccessories] Script Date: 20.05.2023 16:22:38 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER PROCEDURE [dbo].[AddAccessories] (

@Name nvarchar(50), @Description nvarchar(50), @Manufacturer nvarchar(50),

@Cost money, @SupplierId int)

AS

BEGIN

SET NOCOUNT ON

INSERT INTO [Комплектующие\Запчасти](Название, Описание, Производитель, Стоимость, [Код поставщика])

VALUES (@Name, @Description, @Manufacturer, @Cost, @SupplierId)

END

2. **AddClient** (Процедура добавления в таблицу «Клиенты»)

*Формирование процедуры:*

USE [Ремонт компьютерной техники]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[AddClient] Script Date: 20.05.2023 16:23:26 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER PROCEDURE [dbo].[AddClient] (

@Name nvarchar(50), @Surname nvarchar(50), @Patronymic nvarchar(50),

@Adress nvarchar(50), @Telephonenumber nvarchar(50), @Email nvarchar(50))

AS

BEGIN

SET NOCOUNT ON

INSERT INTO Клиенты(Имя, Фамилия, Отчество, Адрес, [Номер телефона], [Электронная почта])

VALUES (@Name, @Surname, @Patronymic, @Adress, @Telephonenumber, @Email)

END

3. **AddDevice** (Процедура добавления в таблицу «Устройства»)

*Формирование процедуры:*

USE [Ремонт компьютерной техники]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[AddDevice] Script Date: 20.05.2023 16:24:07 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER PROCEDURE [dbo].[AddDevice] (

@Model nvarchar(50), @Manufacturer nvarchar(50), @Type nvarchar(50),

@YearOfRelease smallint, @SerialNumber int, @ClientId int,

@Photo nvarchar(50))

AS

BEGIN

SET NOCOUNT ON

INSERT INTO Устройства(Модель, Производитель, [Тип устройства], [Год выпуска], [Серийный номер], [Код клиента], [Фото устройства])

VALUES (@Model, @Manufacturer, @Type, @YearOfRelease, @SerialNumber, @ClientId, @Photo)

END

4. **AddStaff** (Процедура добавления в таблицу «Сотрудники»)

*Формирование процедуры:*

USE [Ремонт компьютерной техники]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[AddStaff] Script Date: 20.05.2023 16:24:57 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER PROCEDURE [dbo].[AddStaff] (

@Name nvarchar(50), @Surname nvarchar(50), @Patronymic nvarchar(50),

@Experience int, @Post nvarchar(50), @Salary money,

@DateOfEmployment date, @Photo nvarchar(50))

AS

BEGIN

SET NOCOUNT ON

INSERT INTO Сотрудники(Имя, Фамилия, Отчество, Стаж, Должность, Зарплата, [Дата приёма на работу], Фотография)

VALUES (@Name, @Surname, @Patronymic, @Experience, @Post, @Salary, @DateOfEmployment, @Photo)

END

5. **AddSupllier** (Процедура добавления в таблицу «Поставщики»)

*Формирование процедуры:*

USE [Ремонт компьютерной техники]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[AddSupllier] Script Date: 20.05.2023 16:25:34 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

ALTER PROCEDURE [dbo].[AddSupllier] (

@NameOfCompany nvarchar(50), @ContactPerson nvarchar(50), @Telephonenumber nvarchar(50),

@Adress nvarchar(50), @Email nvarchar(50))

AS

BEGIN

SET NOCOUNT ON

INSERT INTO Поставщики([Название компании], [Контактное лицо], [Номер телефона], Адрес, [Электронная почта])

VALUES (@NameOfCompany, @ContactPerson, @Telephonenumber, @Adress, @Email)

END

# **ЗАКЛЮЧЕНИЕ**

В заключение данного курсового проекта приведем краткие сведения о проделанной работе, о полученных в проекте результатах, о трудностях, возникших при выполнении поставленной цели и о перспективах развития и улучшения данного программного продукта.

Целью данного курсового проекта было проектирование базы данных предметной области «Ремонт компьютерной техники». Подсистема «Обработка заказов на ремонт». Чтобы выполнить поставленную цель, необходимо было решить следующие задачи:

• Проектирование инфологической модели данных на основе анализа предметной области;

• Проектирование даталогической модели данных на основе инфологической модели;

• Проектирование базы данных;

• Составление программной документации.

В результате выполнения курсового проекта была получена работоспособная база данных, удовлетворяющая начальному заданию. Были реализованы все поставленные задачи автоматизации обработки заказов на ремонт, а именно: добавление, удаление, изменение информации о ремонтах, комплектующих, поставщиках и клиентах, а также устройствах клиентов.

Данная база данных была разработана только для обработки заказов на ремонт, в качестве дальнейшего ее совершенствования представляется возможным расширение функционала с целью повышения автоматизации обработки информации в «Ремонте компьютерной техники».

# **СПИСОК ЛИТЕРАТУРЫ**

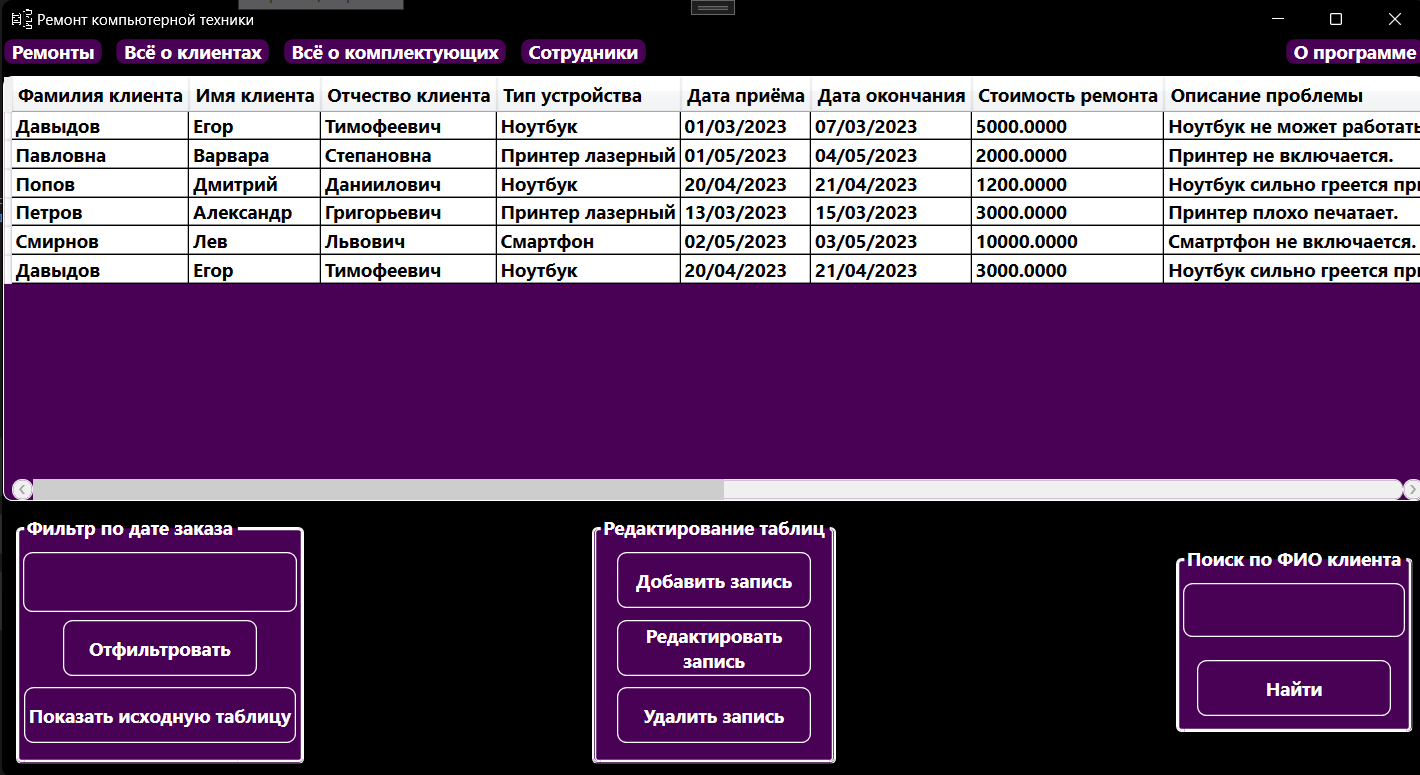
В процессе выполнения курсового проекта был использован материал следующей литературы:

1. Нестеров, С. А. Базы данных: учебник и практикум для среднего профессионального образования / С. А. Нестеров.- Москва: Издательство Юрайт, 2022. 230 с. — (Профессиональное образование). - ISBN 978-5- 534- 11629-8. - Текст: электронный // Образовательная платформа Юрайт [сайт]. - URL: https://urait.ru/bcode/495981 (дата обращения: 24.04.2022).

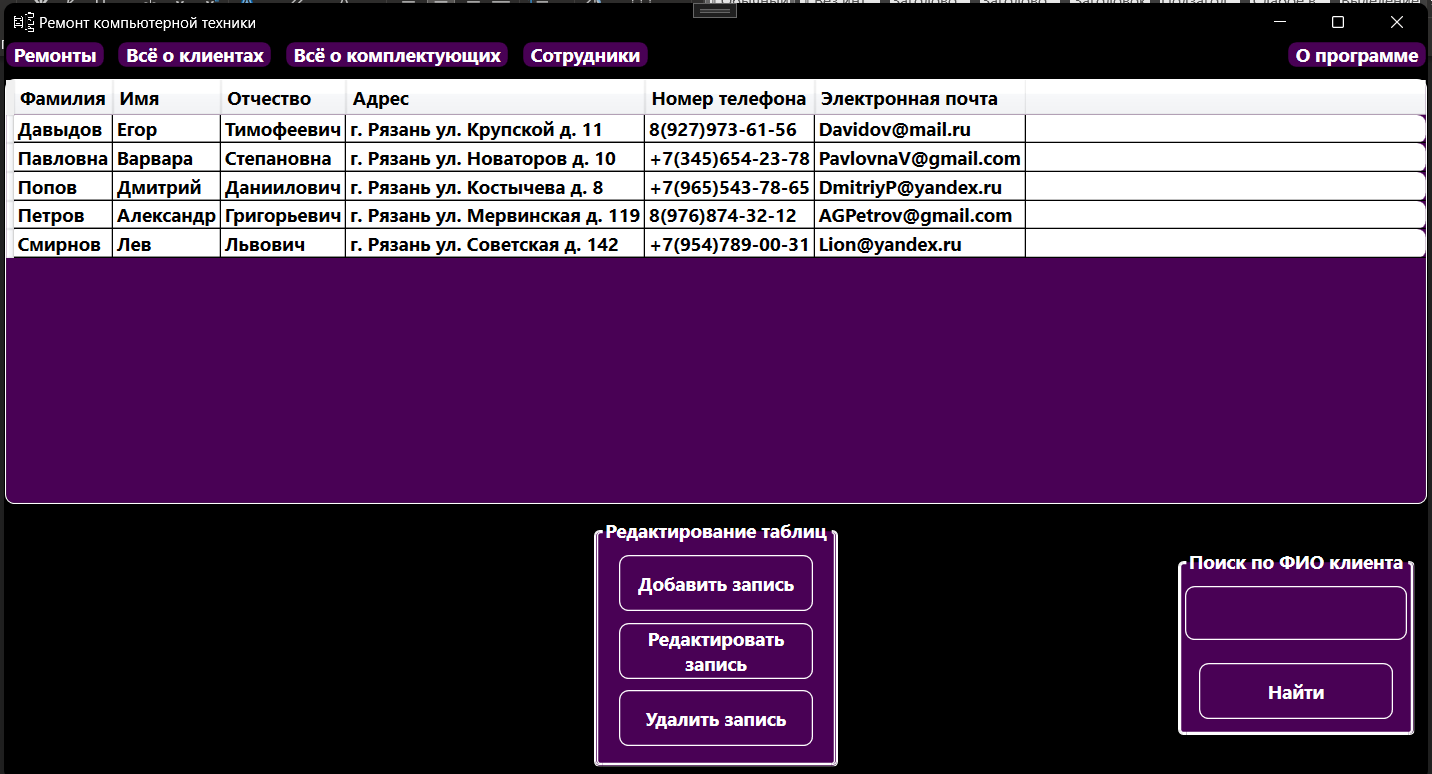
2. Маркин, А. В. Программирование на SQL : учебное пособие для среднего профессионального образования / А. В. Маркин. - Москва: Издательство Юрайт, 2022. - 435 с. — (Профессиональное образование). ISBN 978-5-534-11093-7. Текст : электронный // Образовательная платформа Юрайт [сайт]. - URL: https://urait.ru/bcode/495666 (дата обращения: 24.04.2022). –

3. Стружкин, Н. П. Базы данных: проектирование: учебник для среднего профессионального образования / Н. П. Стружкин, В. В. Годин. - Москва : Издательство Юрайт, 2022 - 477 с. — (Профессиональное образование). ISBN 978-5-534-11635-9. Текст: электронный // - Образовательная платформа Юрайт [сайт]. - URL: https://urait.ru/bcode/495973 (дата обращения: 24.04.2022)

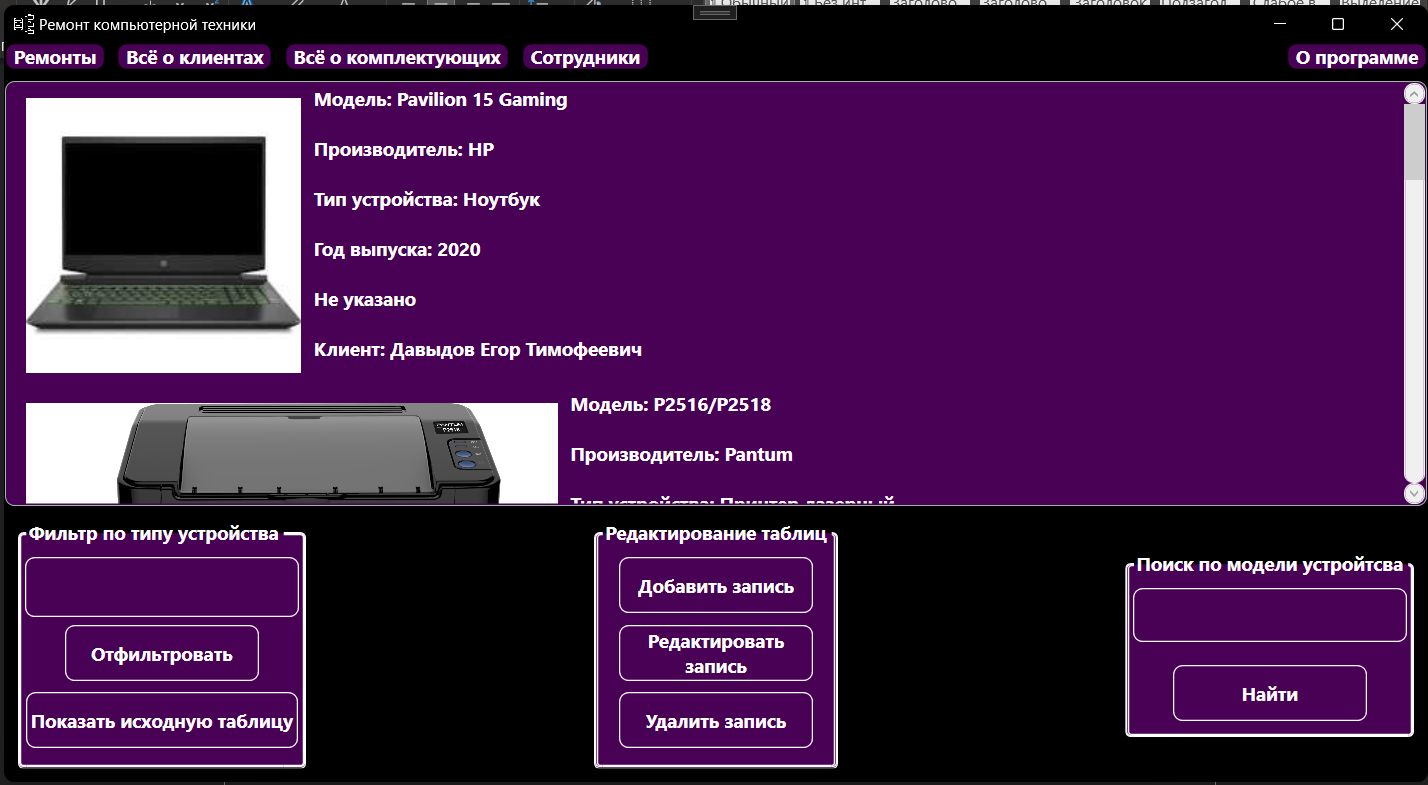
# **ПРИЛОЖЕНИЕ А**



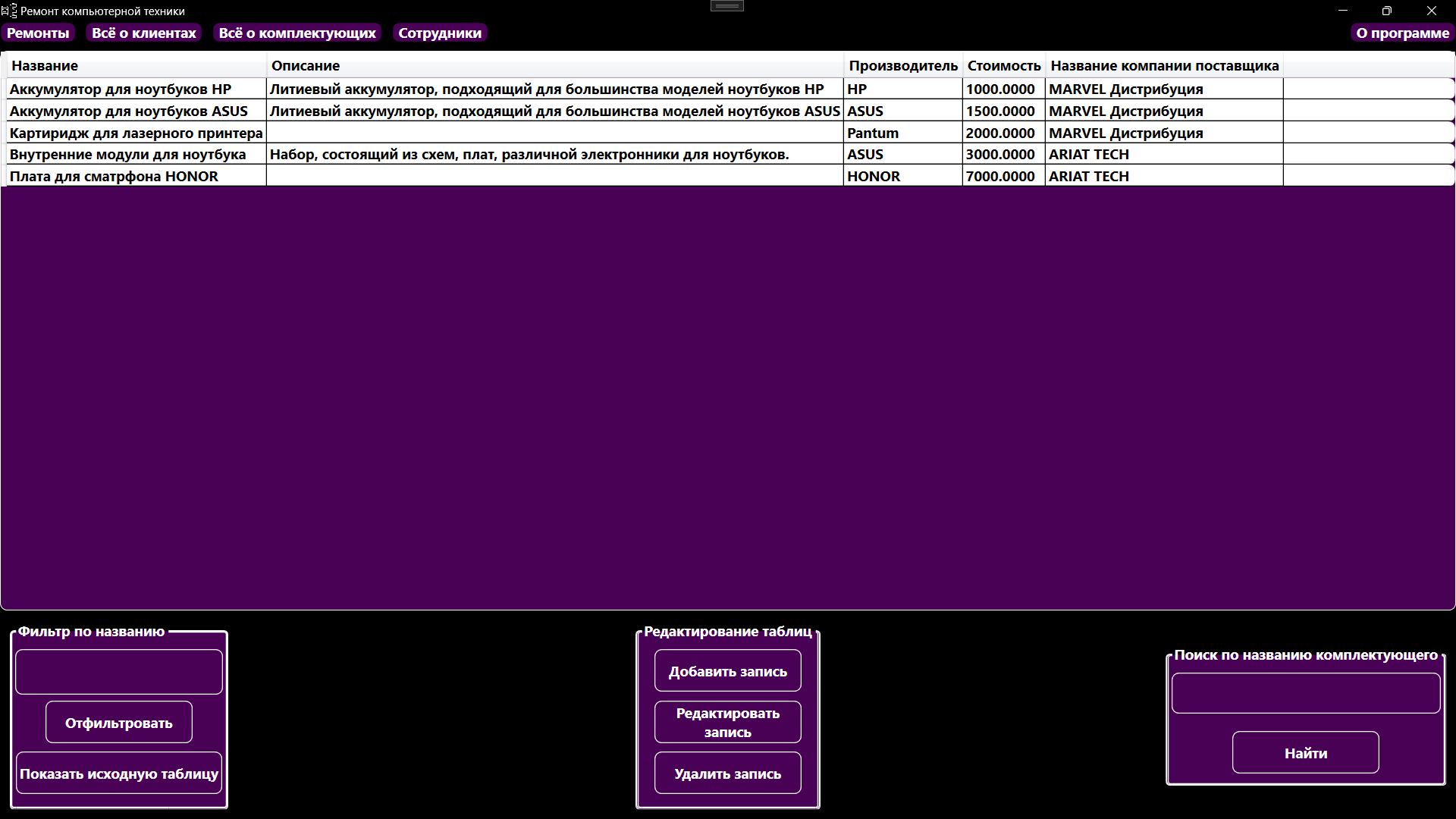
**Рисунок 19 –** Раздел «Ремонты»



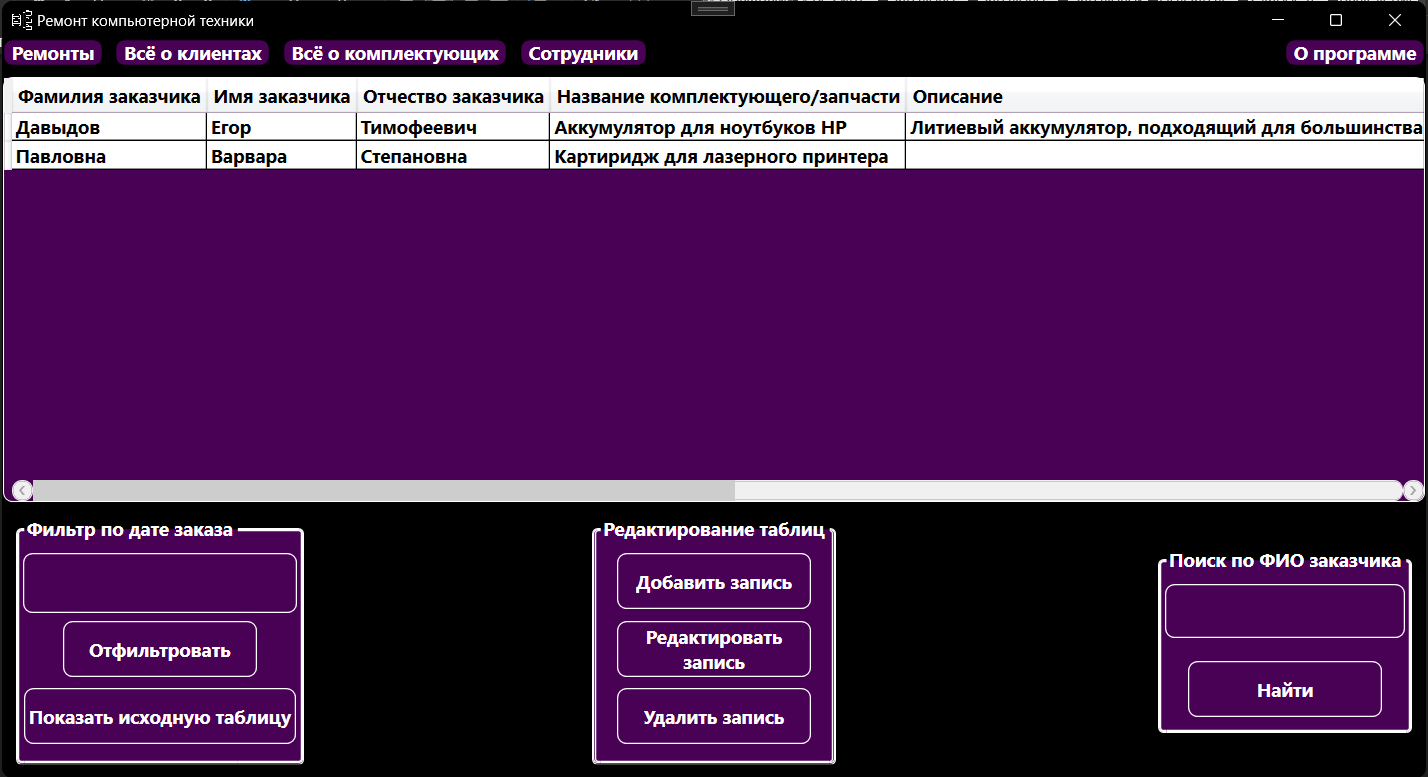
**Рисунок 20 –** Раздел «Клиенты»



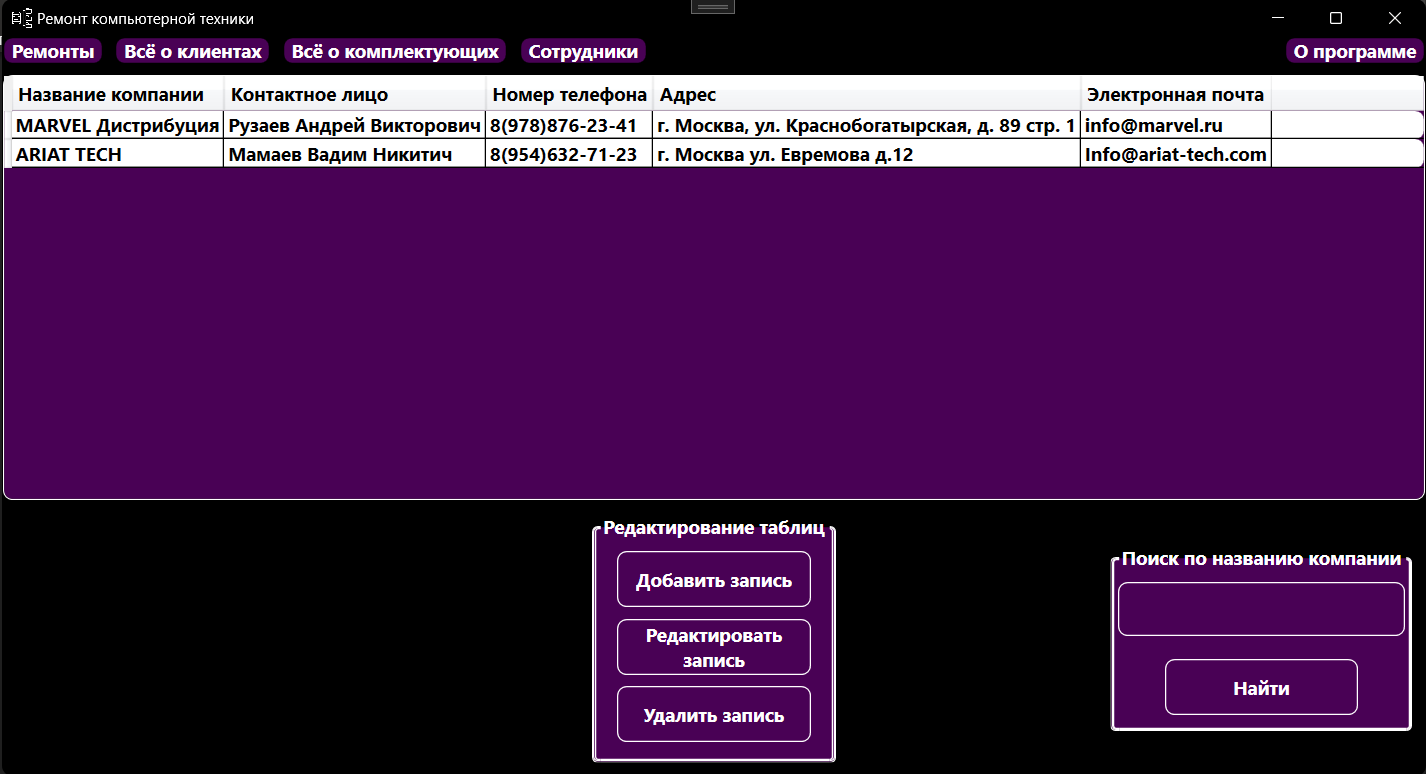
**Рисунок 21 –** Раздел «Устройства»



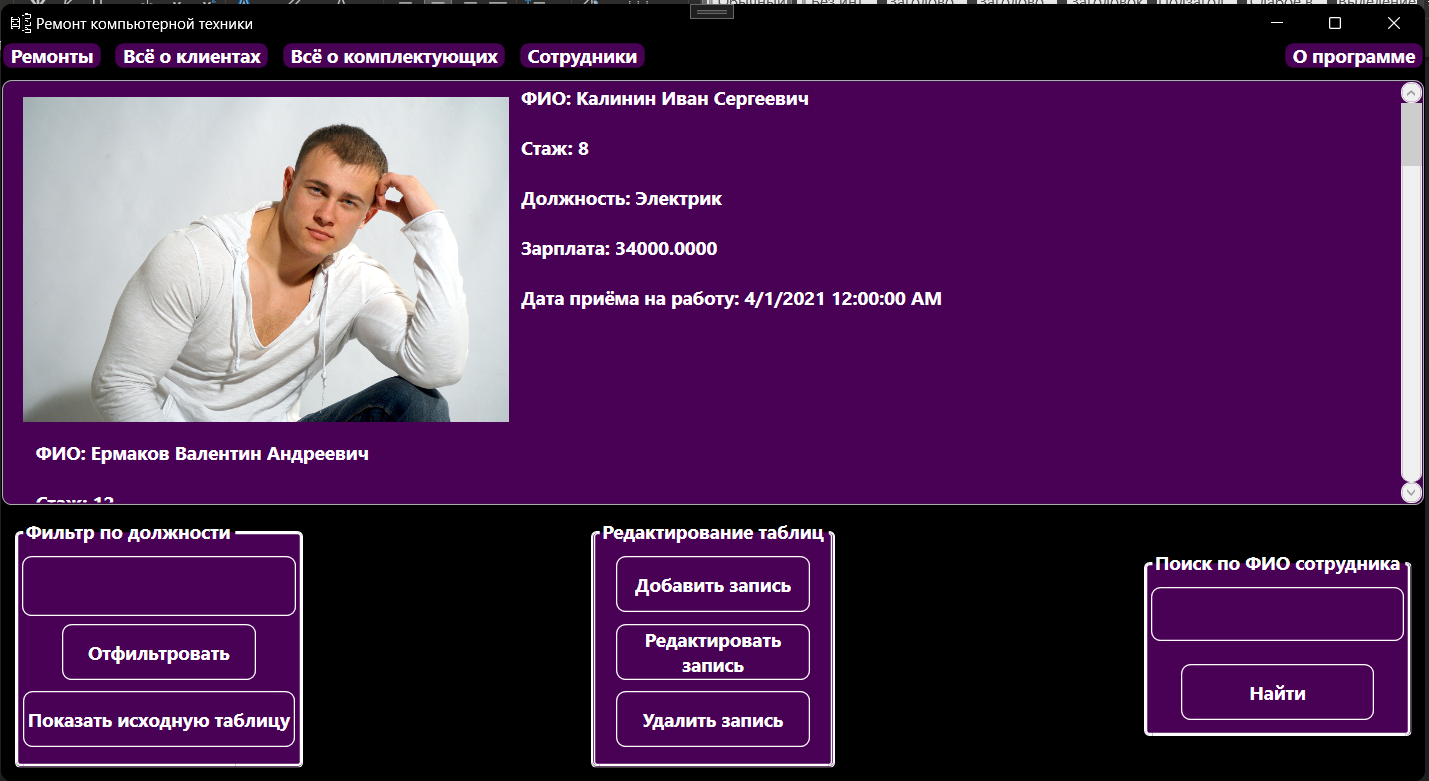
**Рисунок 22 –** Раздел «Комплектующие/Запчасти»



**Рисунок 23 –** Раздел «Заказы комплектующих»



**Рисунок 24 –** Раздел «Поставщики»



**Рисунок 25 –** Раздел «Сотрудники»

# **ПРИЛОЖЕНИЕ Б**

{Листинг события «Добавить» во вкладке «Клиенты»}

private void Add\_Click(object sender, RoutedEventArgs e)

{

AddClient add = new AddClient();

add.ShowDialog();

dgOutput.ItemsSource = db.Клиентыs.ToList();

dgOutput.Focus();

}

{Листинг события «Добавить» в окне «Добавление клиента»}

private void Add\_Click(object sender, RoutedEventArgs e)

{

client.Фамилия = Surname.Text;

client.Имя = Name.Text;

client.Отчество = Patronymic.Text;

client.Адрес = Adress.Text;

client.НомерТелефона = TelephoneNumber.Text;

client.ЭлектроннаяПочта = Email.Text;

try

{

db.Клиенты.Add(client);

db.SaveChanges();

Close();

}

catch { }

}

{Листинг события «Удалить» в окне «Клиенты»}

private void Del\_Click(object sender, RoutedEventArgs e)

{

MessageBoxResult result = MessageBox.Show("Удалить запись?", "Удаление записи", MessageBoxButton.YesNo, MessageBoxImage.Warning);

if (result == MessageBoxResult.Yes)

{

if (dgOutput.SelectedIndex != -1)

{

var item = (Клиенты) dgOutput.Items[dgOutput.SelectedIndex];

db.Клиентыs.Remove(item);

db.SaveChanges();

}

else MessageBox.Show("Выберите запись!", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

}

}

{Листинг события «Редактировать» в окне «Клиенты»}

private void Edit\_Click(object sender, RoutedEventArgs e)

{

if (dgOutput.SelectedIndex != -1)

{

var item = (Клиенты) dgOutput.Items[dgOutput.SelectedIndex];

Data.Id = item.Код;

EditClient edit = new EditClient();

edit.ShowDialog();

dgOutput.Items.Refresh();

dgOutput.Focus();

}

else MessageBox.Show("Выберите запись!", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

{Листинг события «Редактировать» в окне «Редактирование клиента»}

private void Edit\_Click(object sender, RoutedEventArgs e)

{

client.Фамилия = Surname.Text;

client.Имя = Name.Text;

client.Отчество = Patronymic.Text;

client.Адрес = Adress.Text;

client.НомерТелефона = TelephoneNumber.Text;

client.ЭлектроннаяПочта = Email.Text;

try

{

db.SaveChanges();

Close();

}

catch { }

}