**Аннотации**

Данная курсовая работа посвящена созданию базы данных и пользовательского интерфейса к ней для чтения и записи данных магазина цифровых товаров. В ходе работы в среде СУБД MySQL была спроектирована база данных магазина цифровых товаров. Для удобства работы пользователя созданы формы для ввода и редактирования данных, запросы.

**Оглавление**

[Введение 3](#_Toc104301221)

[Глава 1. Анализ предметной области 4](#_Toc104301222)

[1.1. Функциональная модель предметной области 4](#_Toc104301223)

[2. Глава 2. Проектирование БД “Магазин цифровых товаров” 5](#_Toc104301224)

[2.1. Инфологическая модель предметной области 5](#_Toc104301225)

[**2.1.1.** **Определение атрибутов объектов**: 5](#_Toc104301226)

[**2.1.2.** **Сущности:** 5](#_Toc104301227)

[**2.1.3.** **Определение отношений и мощности отношений между сущностями:** 5](#_Toc104301228)

[**2.1.4.** **Создание ER-модель “сущность-связь”** 6](#_Toc104301229)

[**2.1.5.** **Выбор СУБД** 6](#_Toc104301230)

[2.2. Даталогическая модель предметной области 8](#_Toc104301231)

[**2.2.1.** **Выбор модели данных.** 8](#_Toc104301232)

[**2.2.2.** **Преобразование ER-модели в реляционную модель.** 8](#_Toc104301233)

[**2.2.3.** **Создание глобальной схемы связей** 8](#_Toc104301234)

[**2.3.** **Физическая модель предметной области** 10](#_Toc104301235)

[**2.3.1.** **Создание таблиц.** 10](#_Toc104301236)

[**2.3.2.** **Запросы. Структура и назначение. SQL запросы.** 11](#_Toc104301237)

[**3.** **Глава 3. Интерфейс пользователя** 13](#_Toc104301238)

[**3.1. Проектирование форм. Структура и назначение существующих форм.** 13](#_Toc104301239)

[**Заключение** 16](#_Toc104301240)

[**Литература** 17](#_Toc104301241)

# Введение

В настоящее время, когда ЭВМ прочно вошли в производственную деятельность человека, нет необходимости доказывать эффективность использования вычислительной техники в системах управления технологическими процессами, проектирования, научных исследований, административного управления, в учебном процессе, банковских расчетах, здравоохранении, сфере обслуживания и т.д.

В современных условиях одной из распространенных задач для прикладного программиста является разработка и администрирование баз данных. Базы данных используются тогда, когда возникает потребность манипулировать большими массивами данных.

Разрабатываемая база данных предназначена для хранения большого объёма информации о магазине. Информационная система должна хранить данные о сотрудниках, таврах на складе, поставках новых таврах, продажах и пр.

К задачам курсового проекта относятся:

* составлении предметной области,
* разработка концептуальной, логической и физической модели базы данных,
* создание запросов,
* разработка пользовательского интерфейса.

При выполнении курсовой проекта были поставлены следующие цели:

* составлении предметной области,
* составление инфологического и даталогического проектирования баз данных,
* выбор СУБД MySQL,
* создание приложений к базам данных на языке программирования Pyrthon,
* оформление документации: описание предметной области; инфологической, даталогической моделей; структурной схемы данных; составления ER диаграмм.

Отчет оформлен по следующим главам:

1. В главе 1 разрабатывается графическая модель (графическое изображение) предметной области – основные объекты, материальные и информационные потоки и другие элементы предметной области. Приводятся все функциональные задачи для описанных пользователей системы отдельно.
2. В главе 2 описываются основные этапы проектирования базы данных: инфологическая, даталогическая и физическая модели предметной области.
3. В главе 2 «Интерфейс пользователя» описан разработанный интерфейс пользователя. Интерфейс реализован в виде экранных форм, для чтобы он был простой, понятный любому пользователю, но в тоже время функциональный**.**

# Глава 1. Анализ предметной области

Предметной областью автоматизации является магазин цифровой техники. Данный магазин должен предоставлять большой ассортимент электроники, цифровой техники, а также товаров для дома.

База данных регистрирует сами товары, лежащие на складе; новые товары поступившие на склад; персональные данные сотрудников магазина; данные о продажах и клиентах. Система позволяет оперативно добавлять, изменять, заносить данные.

Автоматизированная система позволит быстро выполнять основные должностные функции персонала, таких как администратор системы, продавцы, кассиры. С помощью программных форм это позволит ускорить и облегчить работу персонала.

К задаче продавцов относится оформление сделки путём заполнения программной формы.

Если продавец допустил ошибку вовремя заполнения этой формы, он должен заполнить форму заново и оповестить, т.е. составить заявку для администратора, который в последствии ликвидирует ошибку сотрудника.

Обязанности администратора:

* Исправление ошибок, допущенных работниками, в базе данных, сопровождение корректной работы базы данных и сервера;
* Сопровождение корректной работы вычислительной техники на территории магазина или корпорации;
* Обновление информации о данных магазина доступных только администратору;
* Регистрация новых сотрудников;
* Регистрация новых клиентов;

Обязанности сотрудника:

* Занесении в базу данных о новых продажах;
* Обновление информации, содержащейся в базе данных с доступом “Продавец”.

Также база данных предоставляет возможность клиентам магазина получить все доступные данные о товарах.

## Функциональная модель предметной области

*1. Ввод данных*

* ввод данных о новых товарах,
* ввод данных о новых поставках,
* ввод данных о новых поставщиках,
* ввод данных о новых потребителях,
* ввод данных о новых сотрудниках,
* ввод данных о новых продажах.

*2. Редактирование данных*

* редактирование данных о новых продажах,
* редактирование данных о новых товарах,
* редактирование данных о новых поставках,
* редактирование данных о новых поставщиках,
* редактирование данных о новых потребителях,
* ввод данных о новых сотрудниках.

*3. Запросы*

Запрос о наличии определенного товара по его названию:

* с указанием наличия товара,
* с указанием цены в заданном диапазоне,
* с указанием категории товаров,
* с указанием производителя,
* с указанием сортировки по цене товаров;

Запрос о наличии данных о новых клиентах;

Запрос о наличии данных о новых сотрудниках.

Запрос о наличии данных о новых поставщиках.

# Глава 2. Проектирование БД “Магазин цифровых товаров”

## Инфологическая модель предметной области

* + 1. **Определение атрибутов и их документирование**:

|  |  |  |
| --- | --- | --- |
| Объект | Атрибуты(тип данных) | Ключевой атрибут |
| Поставщик | ID поставщика(счетчик), Название фирмы(varchar), Номер телефона(varchar), Страна(varchar). | ID поставщика |
| Поставка | ID поставки(счетчик), Дата поставки(date), Количество в поставке(int), Стоимость поставки(float). | ID партии |
| Производитель | ID произв(счетчик), Фирма(varchar), Страна(varchar), Лицензия(varchar), Товарный знак(varchar) | ID произв |
| Товар | ID товара(счетчик), Название товара(varchar), Категория(varchar), Дата выпуска(date), В наличии(int), Гарантийный срок(varchar), Цена(varchar). | ID товара |
| Сотрудник | ID сотруд(счетчик), Фамилия(varchar), Имя(varchar), Должность(varchar). | ID сотруд |
| Клиент | ID клиента(счетчик), Фамилия(varchar), Имя(varchar), Номер телефона(varchar). | ID клиента |
| Продажа | ID продажи(счетчик), Дата сделки(date), Количество товара(int), Скидка(float), Сумма(float). | ID продажи |

* + 1. **Определение сущностей и их документирование:**

Поставщик (ID поставщика, Название фирмы, Номер телефона, Страна).

Поставка (ID поставки, Дата поставки, Количество в поставке, Стоимость поставки).

Производитель (ID произв, Фирма, Страна, Лицензия, Товарный знак).

Товар (ID товара, Название товара, Категория, Дата выпуска, В наличии(штук), Гарантийный срок(мес), Цена(р)).

Сотрудник (ID сотруд, Фамилия, Имя, Должность).

Клиент (ID клиента, Фамилия, Имя, Номер телефона).

Продажа (ID продажи, Дата сделки, Количество товара, Скидка, Сумма).

* + 1. **Определение отношений и мощности отношений между сущностями:**

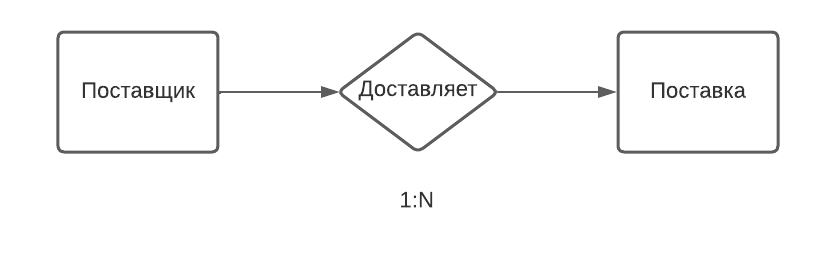


Рисунок 1. Взаимосвязь между объектами «Поставщик» и «Поставка».

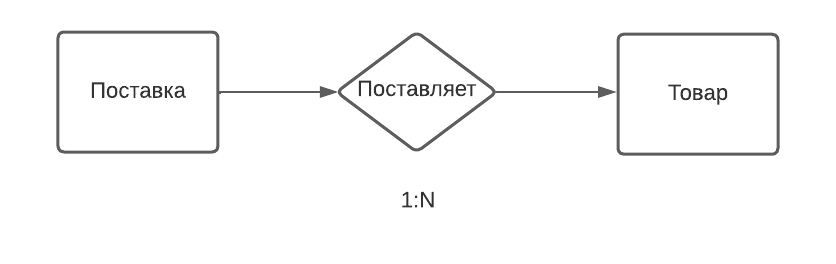


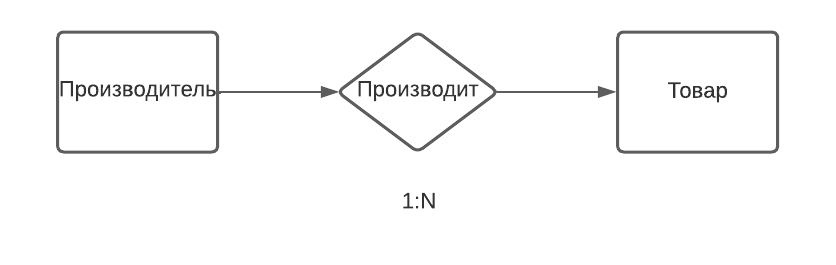
Рисунок 2. Взаимосвязь между объектами «Поставка» и «Товар».

Рисунок 3. Взаимосвязь между объектами «Производитель» и «Товар».

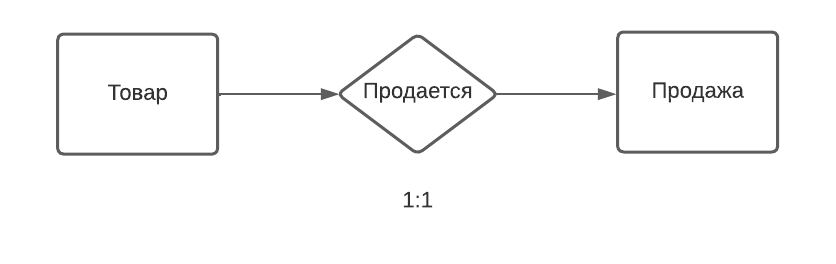


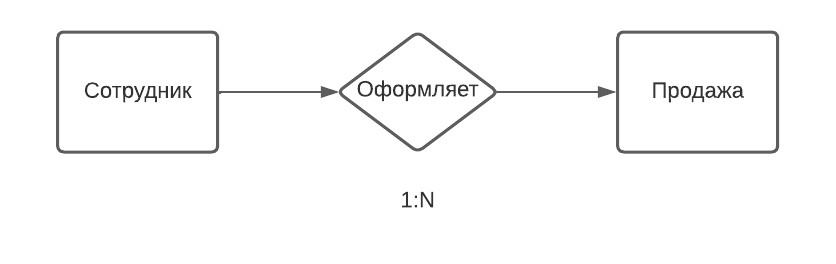
Рисунок 4. Взаимосвязь между объектами «Товар» и «Продажа».

Рисунок 5. Взаимосвязь между объектами «Сотрудник» и «Продажа».

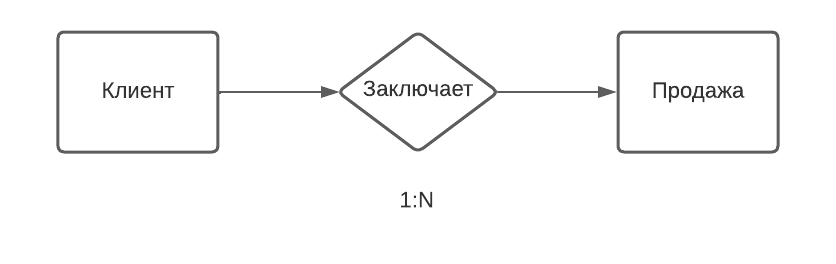


Рисунок 6. Взаимосвязь между объектами «Клиент» и «Продажа».

* + 1. **Создание ER-модель “сущность-связь”**

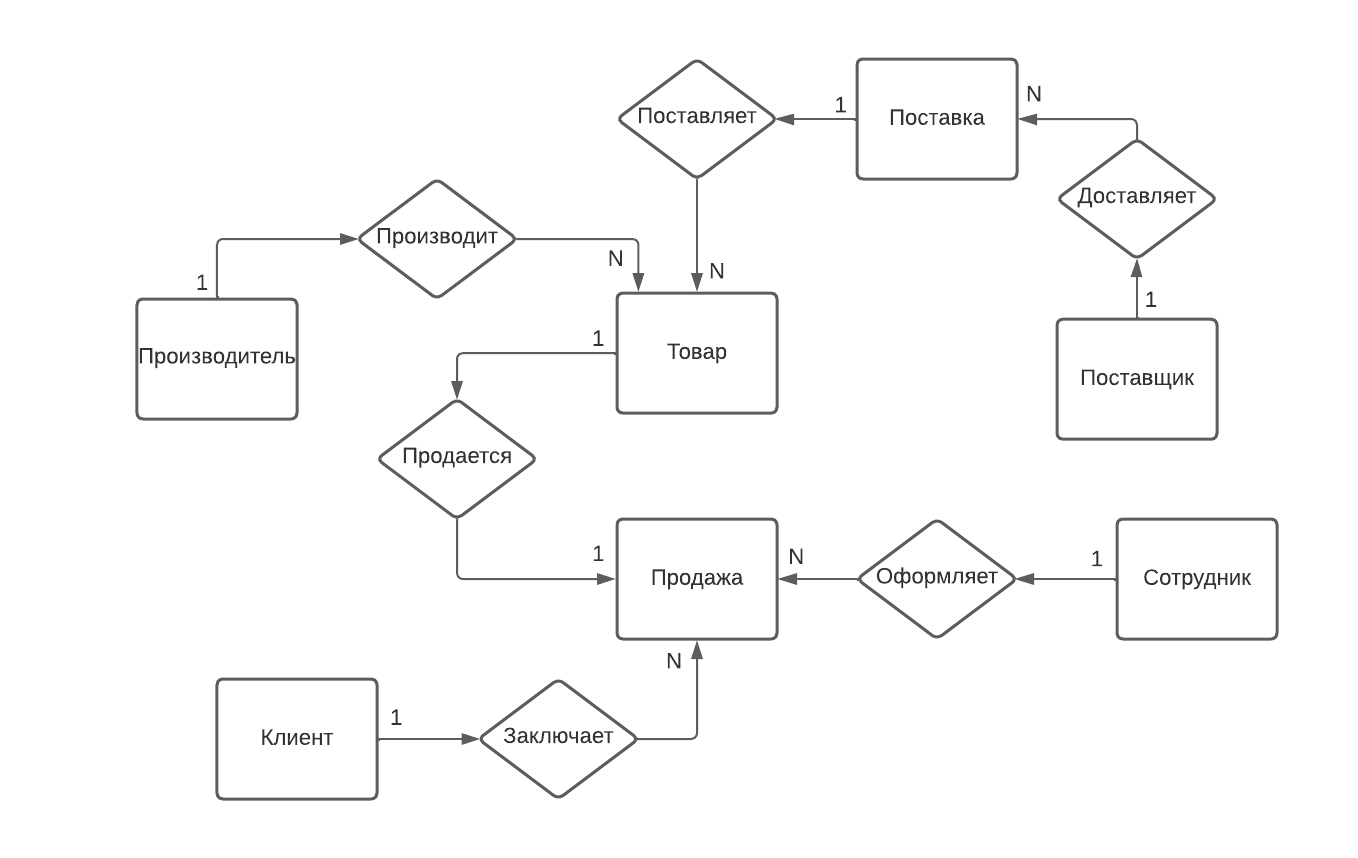


Рис 7. ER-модель предметной области “Магазин цифровых товаров”.

### **Выбор СУБД**

В качестве средства проектирования базы данных, была выбрана СУБД MySQL Workbench .MySQL Workbench — инструмент для визуального проектирования баз данных, интегрирующий проектирование, моделирование, создание и эксплуатацию БД в единое бесшовное окружение для системы баз данных MySQL.

Программа позволяет быстро накидывать схемы данных проекта, проектировать сущности и связи между ними, безболезненно внедрять изменения в схему и так же быстро и безболезненно синхронизировать её с удалённым сервером. А графический редактор EER-диаграмм, позволяет увидеть общую картину модели данных и насладиться её лёгкостью и элегантностью.

Основные сведения о MySQL.

MySQL – это система управления базами данных (СУБД) с открытым кодом. Это высокопроизводительная и масштабируемая СУБД с множеством программных интерфейсов. Она обладает огромными функциональными возможностями и подходит для решения самых разных задач.

Огромной популярностью в мире пользуются базы данных MySQL, а для управления ими - небезызвестный PHPMyAdmin. Переходя от маленьких проектов к большим, от CMS к фреймворкам, многие разработчики остаются верны MySQL. Однако для проектирования сложной базы данных с большим количеством таблиц и связей, возможностей PHPMyAdmin катастрофически не хватает.  
Ниже приведен пример2 создания базы данных с использованием MySQL Workbench - замечательной бесплатной десктопной программы для работы с MySQL.

Основные преимущества MySQL:

* многопоточность, поддержка нескольких одновременных запросов;
* оптимизация связей с присоединением многих данных за один проход;
* записи фиксированной и переменной длины;
* ODBC драйвер;
* гибкая система привилегий и паролей;
* гибкая поддержка форматов чисел, строк переменной длины и меток времени;
* интерфейс с языками C и [Perl](https://www.methodlab.ru/technology/perl.shtml), PHP;
* быстрая работа, масштабируемость;
* совместимость с ANSI SQL;
* бесплатна в большинстве случаев;
* хорошая поддержка со стороны провайдеров услуг хостинга;
* быстрая поддержка транзакций через механизм InnoDB.

## Даталогическая модель предметной области

* + 1. **Выбор модели данных.**

В данной курсовой работе используется реляционная модель данных. Реляционная модель данных представляет информацию в виде совокупности связанных таблиц, которые называются отношениями или реляциями.

Реляционная модель требует, чтобы типы используемых данных были простыми. Тип данных – эквивалентно понятию типа данных в алгоритмических языках.

Существуют:

* целочисленные типы,
* вещественные типы,
* строковые типы,
* типы данных для денежных величин,
* типы данных для временных величин;  
  + 1. **Преобразование ER-модели в реляционную модель.**

Реляционная модель проектируемой базы данных:

Поставщик (ID поставщика, Название фирмы, Номер телефона, Страна).

Поставка (ID поставки, ID поставщика, Дата поставки, Количество в поставке, Стоимость поставки).

Производитель (ID произв, Фирма, Страна, Лицензия, Товарный знак).

Товар (ID товара, ID произв, ID поставки, Название товара, Категория, Дата выпуска, В наличии(штук), Гарантийный срок(мес), Цена(р)).

Сотрудник (ID сотруд, Фамилия, Имя, Должность).

Клиент (ID клиента, Фамилия, Имя, Номер телефона).

Продажа (ID продажи, ID товара, ID клиента, ID сотруд, Дата сделки, Количество товара, Скидка, Сумма).

* + 1. **Создание глобальной схемы связей**

Схема будет построена в программе MySQL Workbench.

Графическая диаграмма даталогической модели предметной области имеет следующий вид:

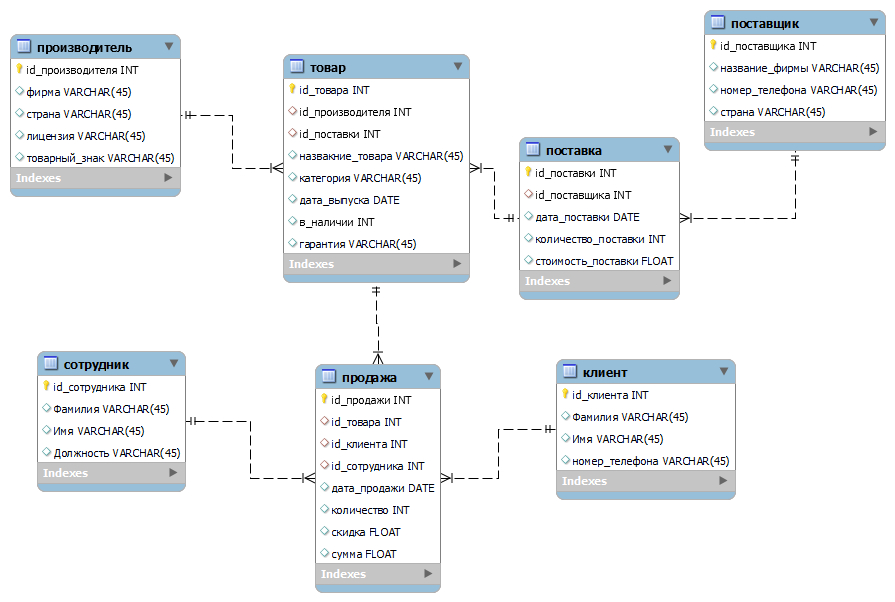


Рис 8. Глобальная схема данных БД “Магазин цифровых товаров”.

* 1. **Физическая модель предметной области**
     1. **Создание таблиц.**

Таблица производитель:

CREATE TABLE IF NOT EXISTS `магазин`.`производитель` (

`id\_производителя` INT NOT NULL AUTO\_INCREMENT,

`фирма` VARCHAR(45) NULL,

`страна` VARCHAR(45) NULL,

`лицензия` VARCHAR(45) NULL,

`товарный\_знак` VARCHAR(45) NULL,

PRIMARY KEY (`id\_производителя`));

Таблица поставщик:

CREATE TABLE IF NOT EXISTS `магазин`.`поставщик` (

`id\_поставщика` INT NOT NULL AUTO\_INCREMENT,

`название\_фирмы` VARCHAR(45) NULL,

`номер\_телефона` VARCHAR(45) NULL,

`страна` VARCHAR(45) NULL,

PRIMARY KEY (`id\_поставщика`));

Таблица поставка:

CREATE TABLE IF NOT EXISTS `магазин`.`поставка` (

`id\_поставки` INT NOT NULL AUTO\_INCREMENT,

`id\_поставщика` INT NULL,

`дата\_поставки` DATE NULL,

`количество\_поставки` INT NULL,

`стоимость\_поставки` FLOAT NULL,

PRIMARY KEY (`id\_поставки`),

CONSTRAINT `поставщик`

FOREIGN KEY (`id\_поставщика`)

REFERENCES `магазин`.`поставщик` (`id\_поставщика`));

Таблица товар:

CREATE TABLE IF NOT EXISTS `магазин`.`товар` (

`id\_товара` INT NOT NULL AUTO\_INCREMENT,

`id\_производителя` INT NULL,

`id\_поставки` INT NULL,

`назвакние\_товара` VARCHAR(45) NULL,

`категория` VARCHAR(45) NULL,

`дата\_выпуска` DATE NULL,

`в\_наличии` INT NULL,

`гарантия` VARCHAR(45) NULL,

PRIMARY KEY (`id\_товара`),

CONSTRAINT `производитель`

FOREIGN KEY (`id\_производителя`)

REFERENCES `магазин`.`производитель` (`id\_производителя`),

CONSTRAINT `поставка`

FOREIGN KEY (`id\_поставки`)

REFERENCES `магазин`.`поставка` (`id\_поставки`));

Таблица клиент:

CREATE TABLE IF NOT EXISTS `магазин`.`клиент` (

`id\_клиента` INT NOT NULL AUTO\_INCREMENT,

`Фамилия` VARCHAR(45) NULL,

`Имя` VARCHAR(45) NULL,

`номер\_телефона` VARCHAR(45) NULL,

PRIMARY KEY (`id\_клиента`));

Таблица сотрудник:

CREATE TABLE IF NOT EXISTS `магазин`.`сотрудник` (

`id\_сотрудника` INT NOT NULL AUTO\_INCREMENT,

`Фамилия` VARCHAR(45) NULL,

`Имя` VARCHAR(45) NULL,

`Должность` VARCHAR(45) NULL,

PRIMARY KEY (`id\_сотрудника`));

Таблица продажа:

CREATE TABLE IF NOT EXISTS `магазин`.`продажа` (

`id\_продажи` INT NOT NULL AUTO\_INCREMENT,

`id\_товара` INT NULL,

`id\_клиента` INT NULL,

`id\_сотрудника` INT NULL,

`дата\_продажи` DATE NULL,

`количество` INT NULL,

`скидка` FLOAT NULL,

`сумма` FLOAT NULL,

PRIMARY KEY (`id\_продажи`),

CONSTRAINT `товар`

FOREIGN KEY (`id\_товара`)

REFERENCES `магазин`.`товар` (`id\_товара`),

CONSTRAINT `клиент`

FOREIGN KEY (`id\_клиента`)

REFERENCES `магазин`.`клиент` (`id\_клиента`),

CONSTRAINT `сотрудник`

FOREIGN KEY (`id\_сотрудника`)

REFERENCES `магазин`.`сотрудник` (`id\_сотрудника`));

* + 1. **Заполнение таблиц данными.**

Таблица производитель:

INSERT INTO `магазин`.`производитель` (`фирма`, `страна`) VALUES ('Samsung', 'Южная Корея');

INSERT INTO `магазин`.`производитель` (`фирма`, `страна`) VALUES ('Xiaomi', 'Китай');

INSERT INTO `магазин`.`производитель` (`фирма`, `страна`) VALUES ('Apple', 'США');

INSERT INTO `магазин`.`производитель` (`фирма`, `страна`) VALUES ('Acer', 'Тайвань');

INSERT INTO `магазин`.`производитель` (`фирма`, `страна`) VALUES ('ASUS', 'Тайвань');

INSERT INTO `магазин`.`производитель` (`фирма`, `страна`) VALUES ('LG', 'Южная Корея');

Таблица поставщик:

INSERT INTO `магазин`.`поставщик` (`название\_фирмы`, `номер\_телефона`, `страна`) VALUES ('Биртранс', '8 (123) 453 2343', 'Китай');

INSERT INTO `магазин`.`поставщик` (`название\_фирмы`, `номер\_телефона`, `страна`) VALUES ('Егор Авто', '8 (423) 553 2523', 'Германия');

INSERT INTO `магазин`.`поставщик` (`название\_фирмы`, `номер\_телефона`, `страна`) VALUES ('Инжелтранс', '8 (213) 123 8453', 'Китай');

Таблица поставка:

INSERT INTO `магазин`.`поставка` (`id\_поставщика`, `дата\_поставки`, `количество\_поставки`, `стоимость\_поставки`) VALUES ('2', '22.02.2021', '12', '5000');

INSERT INTO `магазин`.`поставка` (`id\_поставщика`, `дата\_поставки`, `количество\_поставки`, `стоимость\_поставки`) VALUES ('1', '22.02.2021', '2', '2500');

INSERT INTO `магазин`.`поставка` (`id\_поставщика`, `дата\_поставки`, `количество\_поставки`, `стоимость\_поставки`) VALUES ('3', '21.01.2021', '34', '7600');

INSERT INTO `магазин`.`поставка` (`id\_поставщика`, `дата\_поставки`, `количество\_поставки`, `стоимость\_поставки`) VALUES ('2', '12.02.2021', '5', '2000');

Таблица товар:

INSERT INTO `магазин`.`товар` (`id\_производителя`, `id\_поставки`, `назвакние\_товара`, `категория`, `дата\_выпуска`, `в\_наличии`, `гарантия`) VALUES ('5', '3', 'Ноутбук ASUS Laptop E210MA-GJ001T', 'Ноутбуки', '01.02.2020', '14', '2');

INSERT INTO `магазин`.`товар` (`id\_производителя`, `назвакние\_товара`, `категория`, `дата\_выпуска`, `в\_наличии`, `гарантия`) VALUES ('3', 'Смартфон Apple iPhone SE', 'Смартфоны', '02.02.2020', '0', '1');

INSERT INTO `магазин`.`товар` (`id\_производителя`, `id\_поставки`, `назвакние\_товара`, `категория`, `дата\_выпуска`, `в\_наличии`, `гарантия`) VALUES ('1', '2', 'Смартфон Samsung Galaxy A12', 'Смартфоны', '03.02.2020', '10', '2');

INSERT INTO `магазин`.`товар` (`id\_производителя`, `назвакние\_товара`, `категория`, `дата\_выпуска`, `в\_наличии`, `гарантия`) VALUES ('4', 'Ноутбук Acer Aspire 1 A114-33-C913', 'Ноутбуки', '04.02.2020', '27', '2');

INSERT INTO `магазин`.`товар` (`id\_производителя`, `id\_поставки`, `назвакние\_товара`, `категория`, `дата\_выпуска`, `в\_наличии`, `гарантия`) VALUES ('6', '4', 'Телевизор LED LG 22TN410V-PZ', 'ТВ', '25.02.2020', '1', '0');

INSERT INTO `магазин`.`товар` (`id\_производителя`, `назвакние\_товара`, `категория`, `дата\_выпуска`, `в\_наличии`, `гарантия`) VALUES ('1', 'Смартфон Samsung Galaxy A03', 'Смартфоны', '21.02.2020', '0', '1');

INSERT INTO `магазин`.`товар` (`id\_производителя`, `id\_поставки`, `назвакние\_товара`, `категория`, `дата\_выпуска`, `в\_наличии`, `гарантия`) VALUES ('1', '1', 'Телевизор LED Samsung UE24N4500AUXRU', 'ТВ', '25.02.2020', '24', '2');

INSERT INTO `магазин`.`товар` (`id\_производителя`, `назвакние\_товара`, `категория`, `дата\_выпуска`, `в\_наличии`, `гарантия`) VALUES ('2', 'Смартфон Xiaomi Redmi 9C NF', 'Смартфоны', '08.02.2020', '0', '3');

Таблица клиент:

INSERT INTO `магазин`.`клиент` (`Фамилия`, `Имя`, `номер\_телефона`) VALUES ('Андреев', 'Андрей', '8 (932) 409 4234');

INSERT INTO `магазин`.`клиент` (`Фамилия`, `Имя`, `номер\_телефона`) VALUES ('Иванов', 'Иван', '8 (932) 221 9382');

INSERT INTO `магазин`.`клиент` (`Фамилия`, `Имя`, `номер\_телефона`) VALUES ('Алексеев', 'Алексей', '8 (932) 569 8392');

INSERT INTO `магазин`.`клиент` (`Фамилия`, `Имя`, `номер\_телефона`) VALUES ('Шамилев', 'Шамиль', '8 (932) 429 3232');

INSERT INTO `магазин`.`клиент` (`Фамилия`, `Имя`, `номер\_телефона`) VALUES ('Абакаров', 'Абакар', '8 (932) 239 4544');

INSERT INTO `магазин`.`клиент` (`Фамилия`, `Имя`, `номер\_телефона`) VALUES ('Денисов', 'Денис', '8 (932) 329 6422');

INSERT INTO `магазин`.`клиент` (`Фамилия`, `Имя`, `номер\_телефона`) VALUES ('Михаилов', 'Михаил', '8 (952) 409 1534');

Таблица сотрудник:

INSERT INTO `магазин`.`сотрудник` (`Фамилия`, `Имя`, `Должность`) VALUES ('Артемов', 'Артем', 'Кассир');

INSERT INTO `магазин`.`сотрудник` (`Фамилия`, `Имя`, `Должность`) VALUES ('Афанасьев', 'Афанасий', 'Продавец-консультант');

INSERT INTO `магазин`.`сотрудник` (`Фамилия`, `Имя`, `Должность`) VALUES ('Ильясов', 'Ильяс', 'Кассир');

Таблица продажа:

INSERT INTO `магазин`.`продажа` (`id\_товара`, `id\_клиента`, `id\_сотрудника`, `дата\_продажи`, `количество`, `скидка`, `сумма`) VALUES ('2', '1', '2', '12.04.2021', '2', '10', '12500');

INSERT INTO `магазин`.`продажа` (`id\_товара`, `id\_клиента`, `id\_сотрудника`, `дата\_продажи`, `количество`, `скидка`, `сумма`) VALUES ('5', '2', '2', '04.05.2021', '12', '10', '7500');

INSERT INTO `магазин`.`продажа` (`id\_товара`, `id\_клиента`, `id\_сотрудника`, `дата\_продажи`, `количество`, `скидка`, `сумма`) VALUES ('1', '3', '1', '04.05.2021', '1', '15', '2000');

INSERT INTO `магазин`.`продажа` (`id\_товара`, `id\_клиента`, `id\_сотрудника`, `дата\_продажи`, `количество`, `скидка`, `сумма`) VALUES ('3', '4', '3', '04.05.2021', '7', '10', '34000');

INSERT INTO `магазин`.`продажа` (`id\_товара`, `id\_клиента`, `id\_сотрудника`, `дата\_продажи`, `количество`, `скидка`, `сумма`) VALUES ('2', '5', '2', '04.05.2021', '5', '30', '24000');

INSERT INTO `магазин`.`продажа` (`id\_товара`, `id\_клиента`, `id\_сотрудника`, `дата\_продажи`, `количество`, `скидка`, `сумма`) VALUES ('6', '6', '3', '04.05.2021', '3', '10', '17000');

INSERT INTO `магазин`.`продажа` (`id\_товара`, `id\_клиента`, `id\_сотрудника`, `дата\_продажи`, `количество`, `скидка`, `сумма`) VALUES ('3', '7', '2', '04.05.2021', '1', '10', '2500');

* + 1. **Запросы. Структура и назначение. SQL запросы.**

Запросы для добавления новой информации в базу данных имеют следующий вид:

INSERT INTO table (columns) VALUES (text);

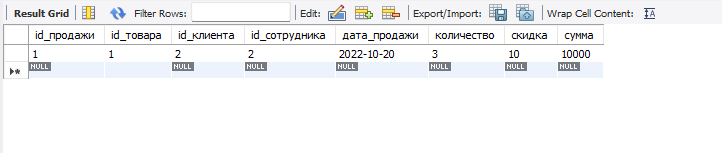
, где table – это называние таблицы, columns – название колонок, text – значения, привязанные к компонентам формы.

SQL- запросы:

INSERT INTO `магазин`.`сотрудник` (`Фамилия`, `Имя`, `Отчество`, `Должность`) VALUES ('Иванов', 'Иван', 'Иванович', 'Продавец');



INSERT INTO `магазин`.`продажа` (`id\_товара`, `id\_клиента`, `id\_сотрудника`, `дата\_продажи`, `количество`, `скидка`, `сумма`) VALUES ('1', '2', '2', '22.10.2021', '3', '10', '10000');



Запросы для изменения данных по первичному ключу имеют следующий вид:

UPDATE table SET columns = "text"

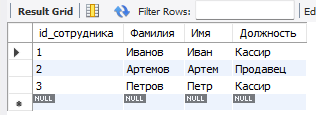
WHERE PrimaryKey =idPK;

, где table – это называние таблицы, column – название колонки, text – значение, привязанное к компонентам формы, idPK – значение первичного ключа.

SQL:

UPDATE сотрудник SET Должность = "Кассир"

WHERE id\_сотрудника = 3;



Запрос для удаления данных по первичному ключу имеют следующий вид:

DELETE FROM table

WHERE PrimaryKey =idPK;

, где table – это называние таблицы, column – название колонки, idPK – значение первичного ключа.

SQL:

DELETE FROM производитель

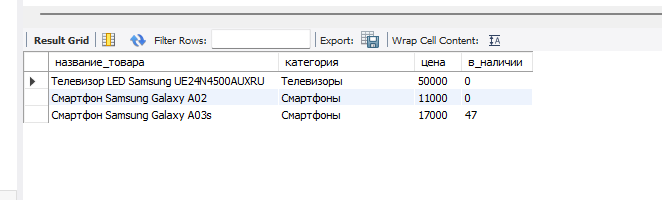
WHERE id\_производителя = 5;

Запрос на получение информации о товарах по надлежащим условиями(условия привязаны к компонентам формы):

* название\_товара соответсвует значению из поиска

SELECT название\_товара, категория, цена, в\_наличии FROM товар

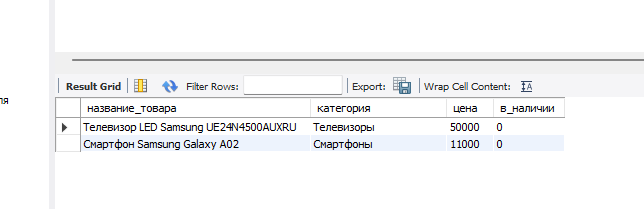
WHERE название\_товара Like '%Samsung%';



* товар отсутствует в продаже

SELECT название\_товара, категория, цена, в\_наличии FROM товар

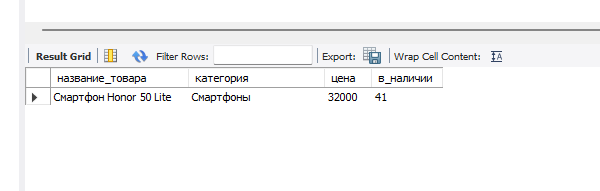
WHERE в\_наличии = 0;



* цена товара между 25000 и 40000

SELECT название\_товара, категория, цена, в\_наличии FROM товар

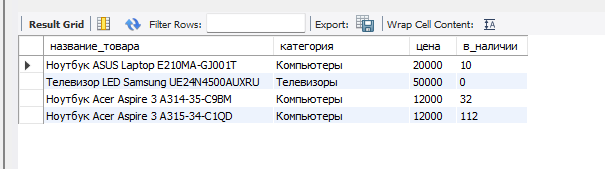
WHERE цена BETWEEN 25000 and 40000;



* товар в категории 'Телевизоры', 'Компьютеры'

SELECT название\_товара, категория, цена, в\_наличии FROM товар

WHERE категория IN ('Телевизоры', 'Компьютеры');

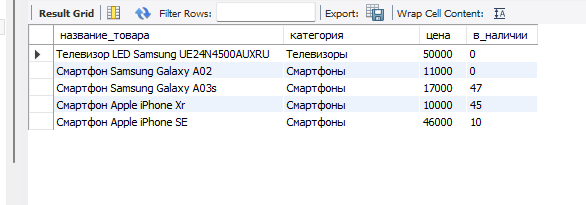


* фирма производителя товара 'Samsung', 'Apple'

SELECT название\_товара, категория, цена, в\_наличии FROM товар

WHERE id\_производителя IN (SELECT id\_производителя

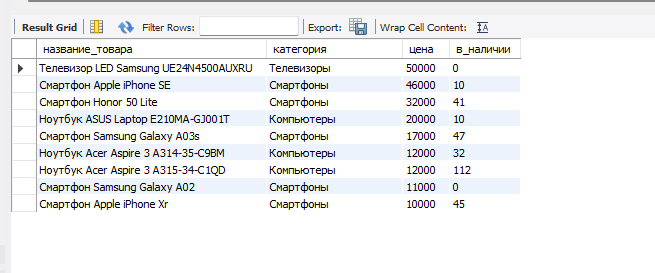
FROM производитель WHERE фирма IN ('Samsung', 'Apple'));



Сортировка товара по возростанию цены

SELECT название\_товара, категория, цена, в\_наличии FROM товар

ORDER BY цена DESC;



1. **Глава 3. Интерфейс пользователя**

**3.1. Проектирование форм. Структура и назначение существующих форм.**

В качестве среды разработки информационной системы, необходимой для курсового проекта, я выбрал программу PyCharm для работы с языком программирования Python. Для реализации графических форм была использована интегрированная библиотека Tkinter, которая содержит компоненты графического интерфейса пользователя.

С помощью предустановленной библиотеки “mysql.connector” к программе подключен хост MySQL Server, который предоставляет доступ ко всем данным СУБД.

1. **Форма “Вход”.**

Данная форма предназначена для авторизации пользователя в программе. В зависимости от выбора типа пользователя: “Администратор”, “Клиент” или “Персонал” пользователь получает соответствующие права.

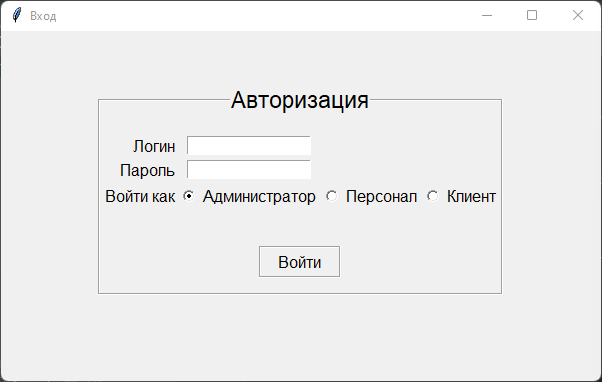


Рис 9. Главное окно формы “Вход”.

Если пользователь в разделе “Войти как” выбрал тип пользователя: “Администратор” или “Клиент”, то ему необходимо ввести пароль, иначе появиться всплывающее окно об ошибке:

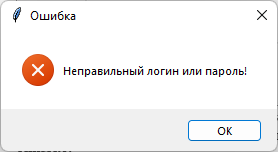


Рис 10. Всплывающее окно-уведомление об ошибке.

Если пользователь не введет логин, то программа вызовет всплывающее окно об предупреждении:

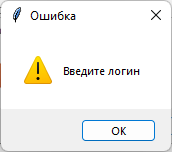


Рис 11. Всплывающее окно-уведомление об предупреждении.

Вводимый логин пользователя в дальнейшем будет использоваться в отображении в последующих формах.

1. **Форма “Рабочее поле администратора”.**

Данная форма появляется после авторизации пользователя, как администратора. Она предназначена для просмотра данных о соответствующих таблицах БД: товарах, поставщиках, поставках, продажах, сотрудниках и клиентах. А также позволяет редактировать данные. Администратору доступно редактирование всех таблиц БД.

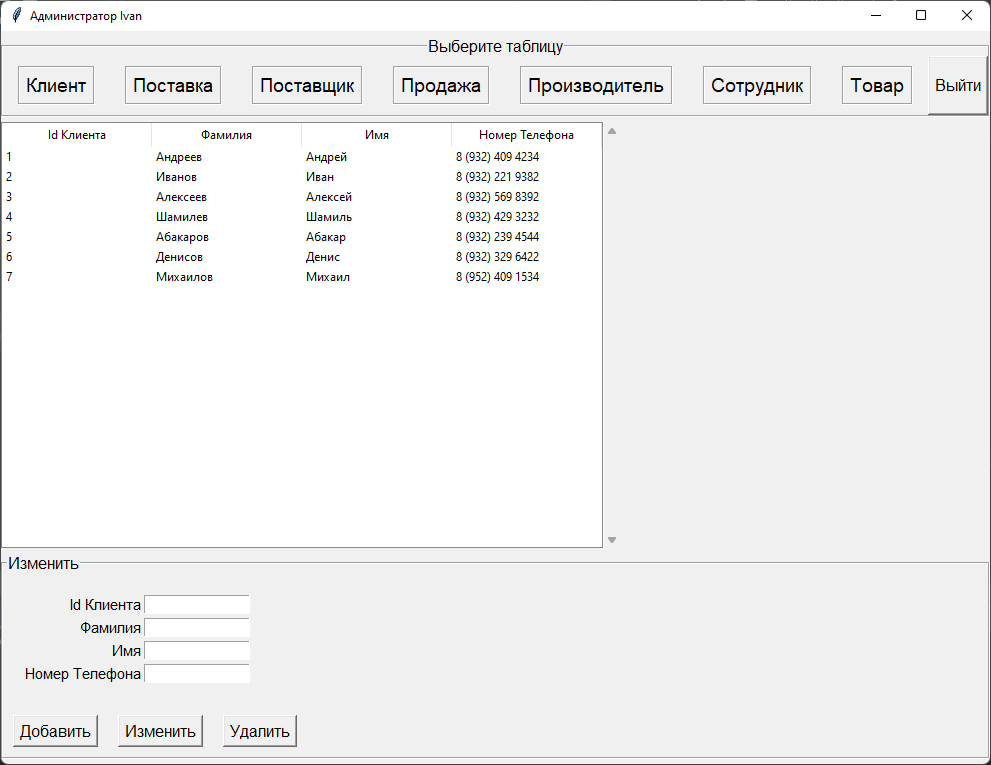


Рис 12. Главное окно формы “Рабочее поле администратора”.

В самой верхней панели пользователю доступны кнопки управления. Данные кнопки вызывают таблицы данных, меняют виджеты для других полей данного окна.

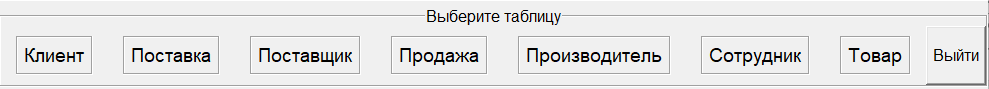


Рис 13. Верхняя панель вызова формы “Рабочее поле администратора”.

Ниже от верхней панели расположена таблица. Она отображает соответствующие данные таблиц созданных в MySQL и редактируется верхней панелью вызова (Рис 13). По умолчанию предоставлены данные из таблицы “Клиент”.

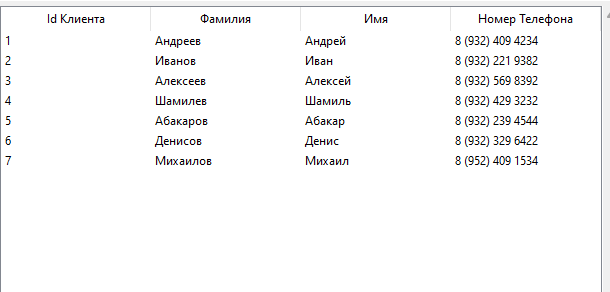


Рис 14. Таблицы формы “Рабочее поле администратора”.

Ниже таблицы расположена панель редактирования. Она отображает поля соответствующих полей таблиц БД. Для вызова панели редактирования той или иной таблицы используется верхняя панель (Рис 13).

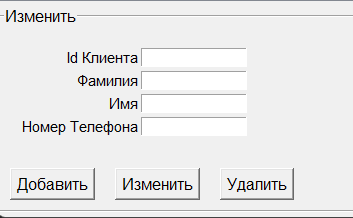


Рис 15. Панель редактирования формы “Рабочее поле администратора”.

Для добавления новых данных в таблицы достаточно ввести данные во все поля кроме первого.

Для изменения или удаления данных необходимо ввести ID в первое поле. В противном случае будет вызвано всплывающее окно с ошибкой:

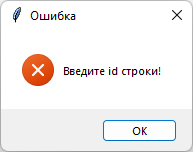


Рис 16. Уведомление об ошибке.

1. **Форма “Рабочее поле персонала”.**

Данная форма предназначена для просмотра данных о соответствующих таблицах и для оформления продаж товаров. Персоналу доступно редактирование данных только таблицы “Продажи”.

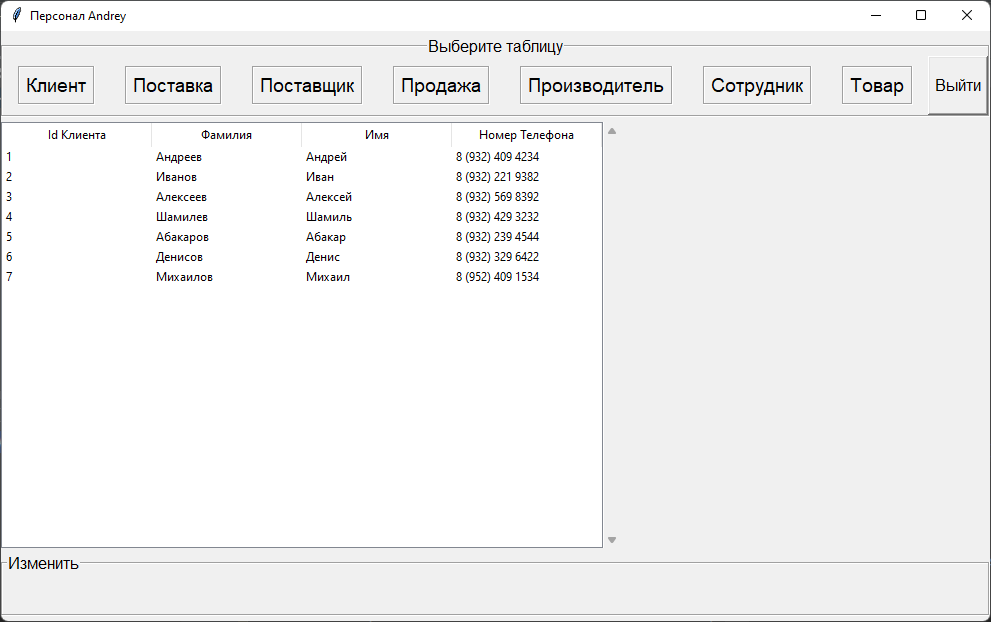


Рис 17. Главное окно формы “Рабочее поле персонала”.

Форма предоставляет те же элементы управления, что в форме “Рабочее поле администратора”.

1. **Форма “Окно клиента”.**

Данная форма предназначена для поиска информации о товарах. Окно вызывается после авторизации пользователя как клиента.

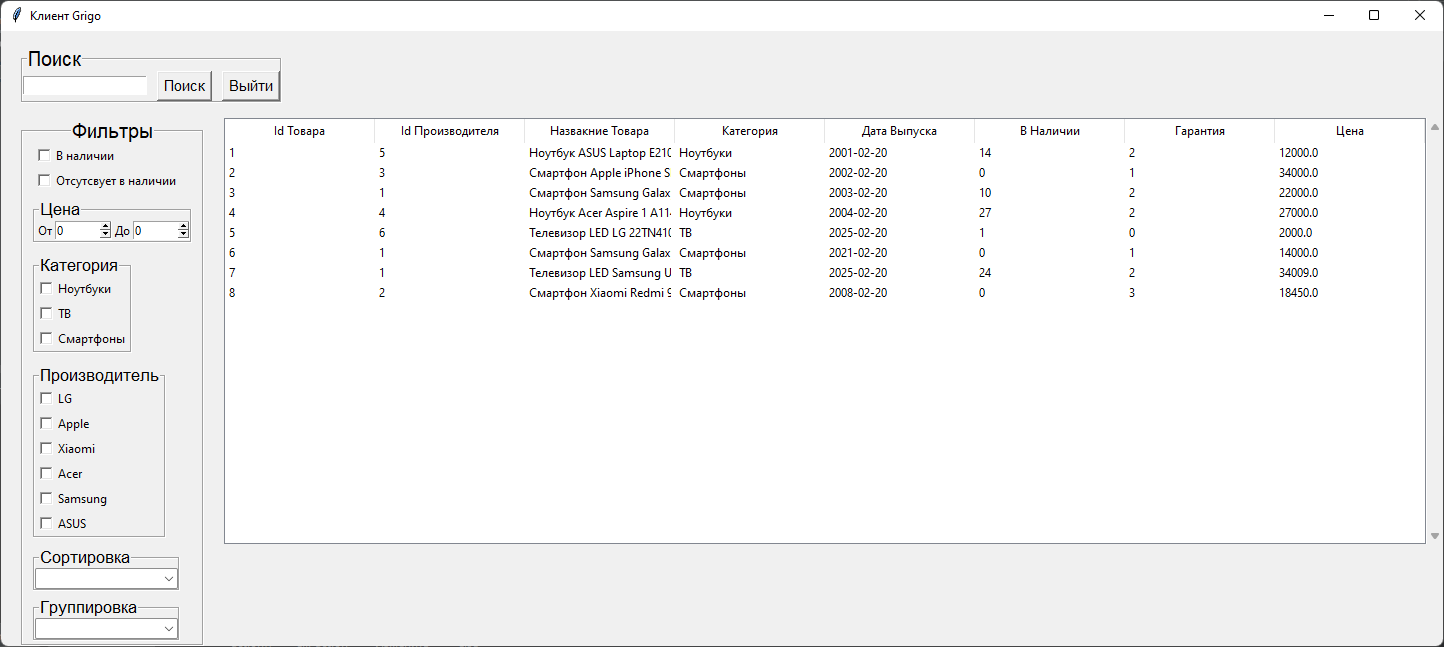
****

Рис 18. Главное окно формы “Окно клиента”.

В панели “Поиск” пользователь задает поиск с текстовыми значениями, связанными с названием товаров.

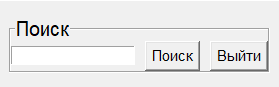


Рис 19. Панель поиска формы “Окно клиента”.

В нижней панели “Фильтры”, пользователь способен задать критерии поиска, внеся соответствующие значения в доступные поля:

* указать критерий наличия товара в продаже,
* задать диапазон цены товара,
* указать категорию товаров,
* указать производителя товара,
* выбрать сортировку товаров по цене,
* сгруппировать товары.

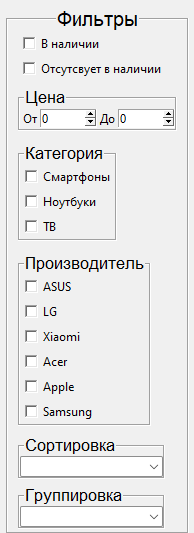


Рис 20. Панель фильтров формы “Окно клиента”.

После нажатия кнопки поиска (Рис 19) осуществляется обработка данных и поиска товаров. Вся информация о них отобразиться таблице на главном окне (Рис 18).

**Заключение**

В проделанной курсовой работе были получены навыки:

* в проектировании инфологической и даталогической моделей АИС;
* в работе с СУБД MySQL и программой MySQL workbench;
* в создании пользовательского интерфейса для управления БД
* были получены навыки грамотного оформлении и составлении документации: описана предметная область; составлена инфологическая и даталогическая модель

Проделанная работа позволяет любому пользователю хранить большие объёмы информации, обрабатывать их, сортировать, делать выборки по определённым критериям. Разработанная информационная система может быть использована в работе магазина цифровой техники.

Все исходники(код, исполняемое приложение, библиотеки) доступны по ссылке: https://github.com/10takla/Market

**Литература**

1. Бьюли, Алан Изучаем SQL / Алан Бьюли. - М.: Символ-плюс, 2018. - 325 c.
2. Гудсон, Джон Практическое руководство по доступу к данным (+ DVD-ROM) / Джон Гудсон , Роб Стюард. - М.: БХВ-Петербург, 2018. - 304 c.
3. Дунаев, В. В. Базы данных. Язык SQL для студента / В.В. Дунаев. - М.: БХВ-Петербург, 2022. - 288 c.
4. Пржиялковский, В. В. Введение в Oracle SQL / В.В. Пржиялковский. - М.: Бином. Лаборатория знаний, Интернет-университет информационных технологий, 2018. - 320 c.
5. Тейлор, Аллен SQL для чайников / Аллен Тейлор. - М.: Вильямс, 2018. - 416 c.