**Содержание**

[ОПИСАНИЕ ПРЕДПРИЯТИЯ 2](#_Toc198147096)

[СОЗДАНИЕ БАЗЫ ДАННЫХ ПО ПРЕДПРИЯТИЮ 5](#_Toc198147097)

[ОФОРМЛЕНИЕ СПЕЦИФИКАЦИИ ТРЕБОВАНИЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 8](#_Toc198147098)

[ПРОЕКТИРОВАНИЕ МОДУЛЕЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 13](#_Toc198147099)

[СОЗДАНИЕ ЗАПРОСОВ В БАЗЕ ДАННЫХ 22](#_Toc198147100)

[ИНФОРМАЦИОННАЯ СИСТЕМА ОРГАНИЗАЦИИ 22](#_Toc198147101)

[ТЕСТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ 29](#_Toc198147102)

[РАЗРАБОТКА ИПС В РАМКАХ ПРОИЗВОДСТВЕННОГО ПРОЦЕССА ПРЕДПРИЯТИЯ 37](#_Toc198147103)

[ПРИЛОЖЕНИЯ 38](#_Toc198147104)

**ОПИСАНИЕ ПРЕДПРИЯТИЯ**

В рамках прохождения практики я проводил анализ деятельности выдуманного предприятия «Красный Октябрь», специализирующегося на производстве премиальных кондитерских изделий ручной работы.

«Красный Октябрь» — это небольшое, но амбициозное предприятие, специализирующееся на выпуске эксклюзивных конфет ручной работы с использованием высококачественных натуральных ингредиентов. Фокус направлен на премиальный сегмент рынка, ориентированный на покупателей, ценящих качество, уникальность и неповторимый вкус. Предприятие располагается в пригороде Москвы, в специально оборудованном помещении, соответствующем всем санитарно-гигиеническим нормам.

**Производственный процесс**

Производственный процесс «Красного Октября» полностью ориентирован на ручную работу, что обеспечивает высокое качество и уникальность каждой конфеты. Этапы производства включают:

Отбор и обработка сырья: Используются исключительно высококачественные натуральные ингредиенты – шоколад премиум-класса, экзотические фрукты, отборные орехи и специи. Сырье тщательно отбирается и проходит многоступенчатую проверку на соответствие строгим стандартам качества.

Приготовление начинок: Начинки готовятся вручную по оригинальным рецептурам, с использованием традиционных и инновационных технологий.

Формирование конфет: Конфеты формируются вручную, что позволяет контролировать их размер, форму и внешний вид. Мастера-кондитеры обладают высоким уровнем профессионализма и опытом.

Глазировка и декорирование: Конфеты глазируются высококачественным шоколадом и декорируются вручную, что придает им неповторимый внешний вид.

Упаковка: Готовые конфеты упаковываются в элегантные подарочные коробки, подчеркивающие их премиальный статус. Упаковка также изготавливается вручную или с использованием высококачественных автоматизированных линий.

Контроль качества: На всех этапах производства осуществляется строгий контроль качества, что обеспечивает высочайшие стандарты качества готовой продукции. Для этого используется современное оборудование и высококвалифицированный персонал.

**Ассортимент продукции**

Ассортимент «Красного Октября» состоит из ограниченного, но постоянно обновляемого количества уникальных конфет. В нем представлены:

* Шоколадные конфеты с экзотическими начинками;
* Конфеты с пралине из различных орехов;
* Конфеты с фруктовыми и ягодными начинками;
* Конфеты с использованием специй и алкоголя (в ограниченном количестве);

**Маркетинг и продажи**

«Красный Октябрь» ориентируется на премиальный сегмент рынка и использует следующие маркетинговые стратегии:

* Прямые продажи через собственный магазин и онлайн-бутик;
* Сотрудничество с элитными магазинами и ресторанами;
* Участие в специализированных выставках и дегустациях;
* Активное продвижение в социальных сетях;

**Персонал**

На «Красном Октябре» работает высококвалифицированный персонал, включающий опытных мастеров-кондитеров, технологов, менеджеров по продажам и маркетингу. Компания уделяет большое внимание обучению и развитию своих сотрудников.

**Перспективы развития**

«Красный Октябрь» планирует расширить ассортимент продукции, развить дистрибуционную сеть и усилить свое присутствие на рынке премиальных кондитерских изделий. Также рассматриваются варианты франчайзинга и открытия новых точек продаж.

# **СОЗДАНИЕ БАЗЫ ДАННЫХ ПО ПРЕДПРИЯТИЮ**

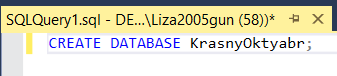


Рис.1 «Создаём базу данных»

use KrasnyOktyabr;

-- Создание таблиц

CREATE TABLE Sotrudniki (

id\_sotrudnika INT PRIMARY KEY IDENTITY(1,1),

FIO\_sotrudnika VARCHAR(255),

Telefon VARCHAR(20),

email VARCHAR(255),

id\_dolzhnosti INT

);

CREATE TABLE Dolzhnosti (

id\_dolzhnosti INT PRIMARY KEY IDENTITY(1,1),

nazvanie VARCHAR(255),

oklad DECIMAL(10, 2)

);

CREATE TABLE ProizvodstvennyeLinii (

id\_linii INT PRIMARY KEY IDENTITY(1,1),

nazvanie VARCHAR(255),

status VARCHAR(50)

);

CREATE TABLE Syrje (

id\_syrja INT PRIMARY KEY IDENTITY(1,1),

nazvanie VARCHAR(255),

ed\_izmerenija VARCHAR(50),

minimalnyj\_zapas INT

);

CREATE TABLE Klienty (

id\_klienta INT PRIMARY KEY IDENTITY(1,1),

nazvanie\_kompanii VARCHAR(255),

kontaktnoe\_lico VARCHAR(255),

telefon VARCHAR(20)

);

CREATE TABLE Produkcija (

id\_produkcii INT PRIMARY KEY IDENTITY(1,1),

nazvanie VARCHAR(255),

sebestoimost DECIMAL(10, 2),

cena\_prodazhi DECIMAL(10, 2)

);

CREATE TABLE Postavshchiki (

id\_postavshchika INT PRIMARY KEY IDENTITY(1,1),

nazvanie\_kompanii VARCHAR(255),

kontaktnoe\_lico VARCHAR(255),

adres VARCHAR(255)

);

CREATE TABLE Sklad (

id\_zapisi INT PRIMARY KEY IDENTITY(1,1),

id\_syrja INT,

id\_produkcii INT,

kolichestvo INT,

data\_postuplenija DATE

);

CREATE TABLE Zakazy (

id\_zakaz INT PRIMARY KEY IDENTITY(1,1),

id\_klienta INT,

data\_zakaz DATE,

status VARCHAR(50),

id\_transporta INT

);

CREATE TABLE Transport (

id\_transporta INT PRIMARY KEY IDENTITY(1,1),

tip VARCHAR(50),

gruzopodemnost INT,

status VARCHAR(50)

);

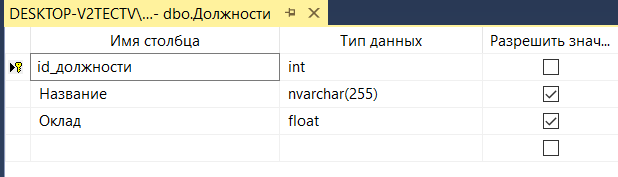


Рис. 2 Таблица «Должности»

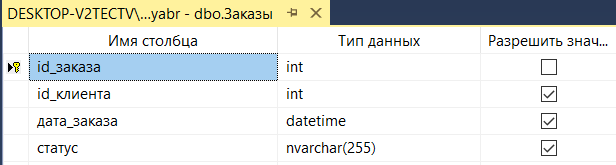


Рис. 3 Таблица «Заказы»

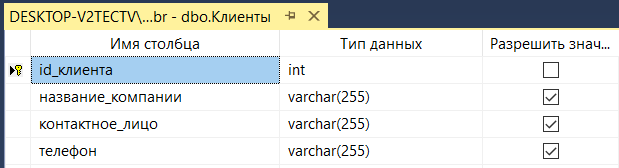


Рис. 4 Таблица «Клиенты»

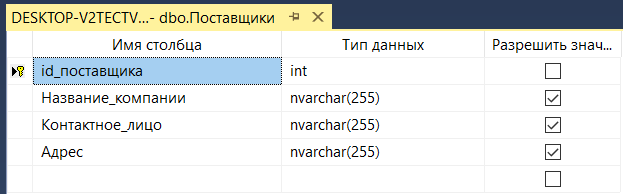


Рис. 5 Таблица «Поставщики»

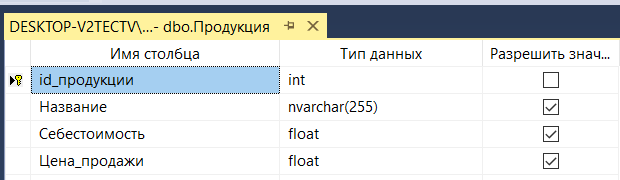


Рис. 6 Таблица «Продукция»

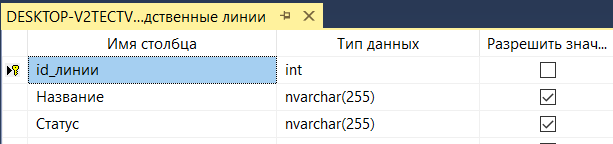


Рис. 7 Таблица «Производственные линии»

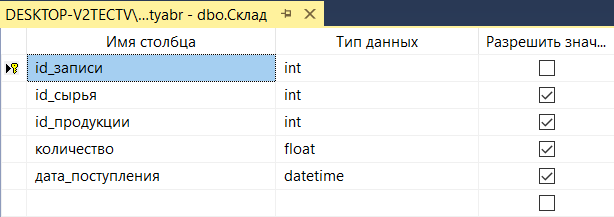


Рис. 8 Таблица «Склад»

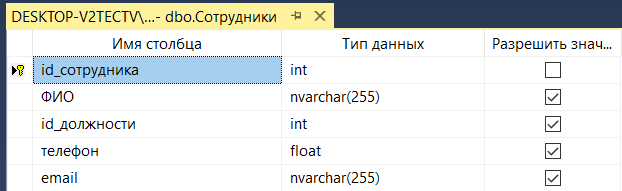


Рис. 9 Таблица «Сотрудники»

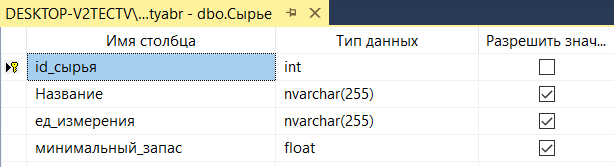


Рис. 10 Таблица «Сырье»

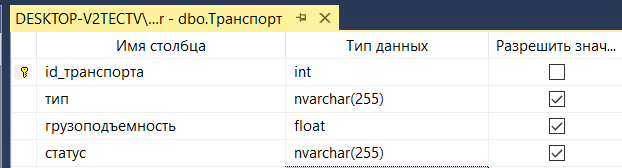


Рис. 11 Таблица «Транспорт»

# **ОФОРМЛЕНИЕ СПЕЦИФИКАЦИИ ТРЕБОВАНИЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

**1. Введение**

Название программы:

«Система учета производства кондитерских изделий»

Цель создания:

Создание централизованной базы данных для систематизации и оптимизации процессов управления информацией о производстве, сырье, готовой продукции и заказах на кондитерской фабрике «Красный Октябрь».

Разработана для:

База данных разработана для использования сотрудниками:

* Отдела снабжения
* Производственного отдела
* Отдела качества
* Склада готовой продукции
* Отдела продаж

**Предназначение для предприятия:**

Программа предназначена для эффективного учета, анализа и управления данными о сырье, технологиях производства, готовой продукции и заказах, что позволяет повысить производительность и снизить затраты.

**2. Общее описание**

**Описание по пользованию базой данных:**

База данных «Система учета производства кондитерских изделий» – это удобный инструмент для хранения и обработки информации о:

* Сырье (мука, сахар, какао и т. д.)
* Технологических процессах (рецептуры, линии производства)
* Готовой продукции (шоколад, конфеты, печенье)
* Заказах клиентов
* Складских остатках

Интерфейс интуитивно понятен и адаптирован под нужды сотрудников фабрики.

Программы, необходимые для работы:

* СУБД (MySQL, PostgreSQL или Microsoft SQL Server)
* Интерфейс (веб-приложение или клиентское ПО)

Преимущества программы:

* Централизованное хранение данных о сырье, производстве и заказах.
* Оперативный контроль качества на всех этапах изготовления.
* Автоматизация учета остатков сырья и готовой продукции.
* Снижение риска ошибок при формировании заказов.
* Интеграция между отделами (снабжение, производство, продажи).

**3. Функции системы**

**Работа пользователя в базе данных:**

1.Главное меню  
После авторизации пользователь попадает в главное меню, где доступны разделы:

* «Сырье» – учет поставок и остатков
* «Рецептуры» – технологические карты производства
* «Производство» – управление линиями выпуска
* «Готовая продукция» – контроль качества и упаковка
* «Заказы» – обработка клиентских запросов

**2. Ввод и редактирование данных**

* Автоматическая проверка корректности данных (например, запрет ввода букв в числовые поля).
* Подсказки при заполнении форм (например, выбор рецептуры из списка).

**3. Поиск и фильтрация**

* Быстрый поиск по параметрам (название продукции, артикул, дата производства).
* Формирование отчетов в реальном времени.

**Типичные ошибки и их устранение:**

Таблица 1

| Ошибка | Причина | Решение |
| --- | --- | --- |
| Некорректный формат даты | Ввод вручную с нарушением шаблона | Использовать календарь или автоматическую валидацию |
| Ошибка "Дублирование штрихкода" | Попытка добавить продукт с уже существующим кодом | Проверить уникальность кода |
| Пустые обязательные поля | Не заполнены ключевые данные | Система выделяет незаполненные поля красным |

**4. Требования к данным**

Таблица 2

| Тип данных | Формат ввода | Пример |
| --- | --- | --- |
| Текстовые поля | Кириллица/латиница (запрещены спецсимволы) | «Шоколад „Алёнка“» |
| Числовые | Целое или дробное (разделитель – точка) | 12.5, 100 |
| Дата | ДД.ММ.ГГГГ или выбор из календаря | 14.05.2025 |
| Выпадающие списки | Выбор из предустановленных значений | «Какао-бобы», «Сахарная пудра» |

**5 Пользовательский интерфейс**

Окно «Заказы» предназначено для отображения и управления информацией о заказах Сырья. Окно представляет собой таблицу, содержащую следующие столбцы:

id\_заказа: Первичный ключ, уникальный идентификатор заказа (целое число, автоматическая нумерация). Это поле не предназначено для редактирования пользователем.

id\_клиента: Внешний ключ, ссылается на идентификатор проекта, к которому относится заказ (целое число). Поле позволяет связать заказ с конкретным проектом Клиенты.

дата\_заказа: Внешний ключ, ссылается на идентификатор даты, оформившего заказ (целое число). Поле предназначено для связи заказа с информацией о дате.

статус: Статус оформления заказа (тип данных - nvarchar(255)).

Окно «Клиенты» предназначено для хранения и управления информацией о Клиентах, связанных с производством и поставкой с производством Красный октябрь. Окно представляет собой таблицу, содержащую следующие столбцы:

id\_клиента: Первичный ключ, уникальный идентификатор проекта (целое число, автоматическая нумерация). Это поле не предназначено для редактирования пользователем.

название\_компании: Название компании через которую проходит закупка сырья

контактное\_лицо: Контактное лицо (текст, varchar (255)).

телефон: Дата начала проекта (varchar(255)).

Окно «Должности» предназначено для хранения и управления информацией о складах, используемых для хранения стеллажного оборудования и комплектующих. Окно представляет собой таблицу, содержащую следующие столбцы:

id\_должности: Первичный ключ, уникальный идентификатор должности (целое число, автоматическая нумерация). Это поле не предназначено для редактирования пользователем.

Название: Название должности (текст, nvarchar(255)). Например: «Генеральный директор», «Начальник цеха».

adres:

Оклад: Оклад на сотрудника (float).

# **ПРОЕКТИРОВАНИЕ МОДУЛЕЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

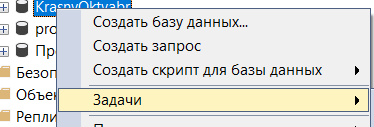


Рис.12 «Переходим в Задачи»



Рис.13 «Выбираем импорт данных»

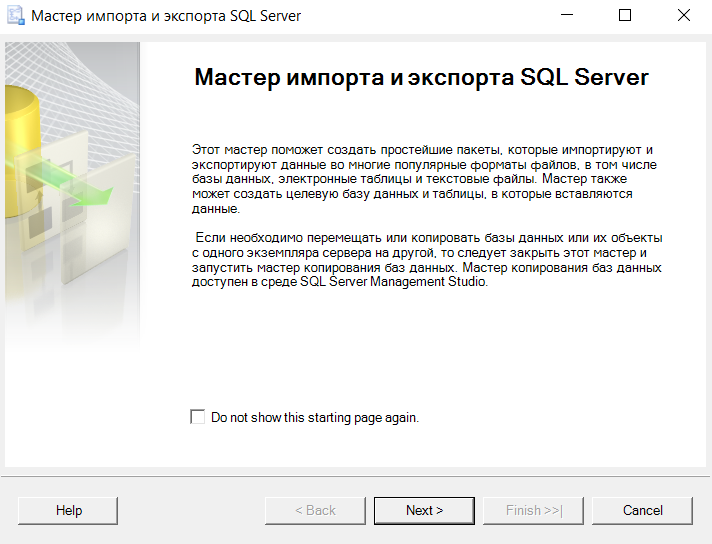


Рис.14 «Запускается мастер»

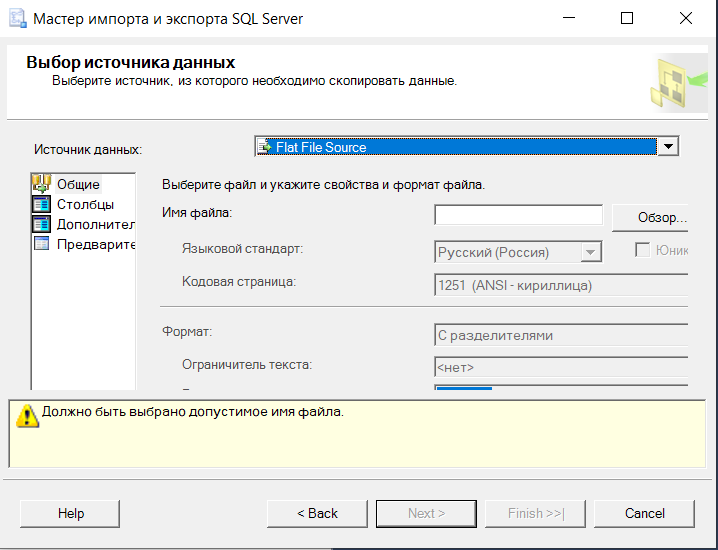


Рис.15 «Выбираем источник данных для импорта»

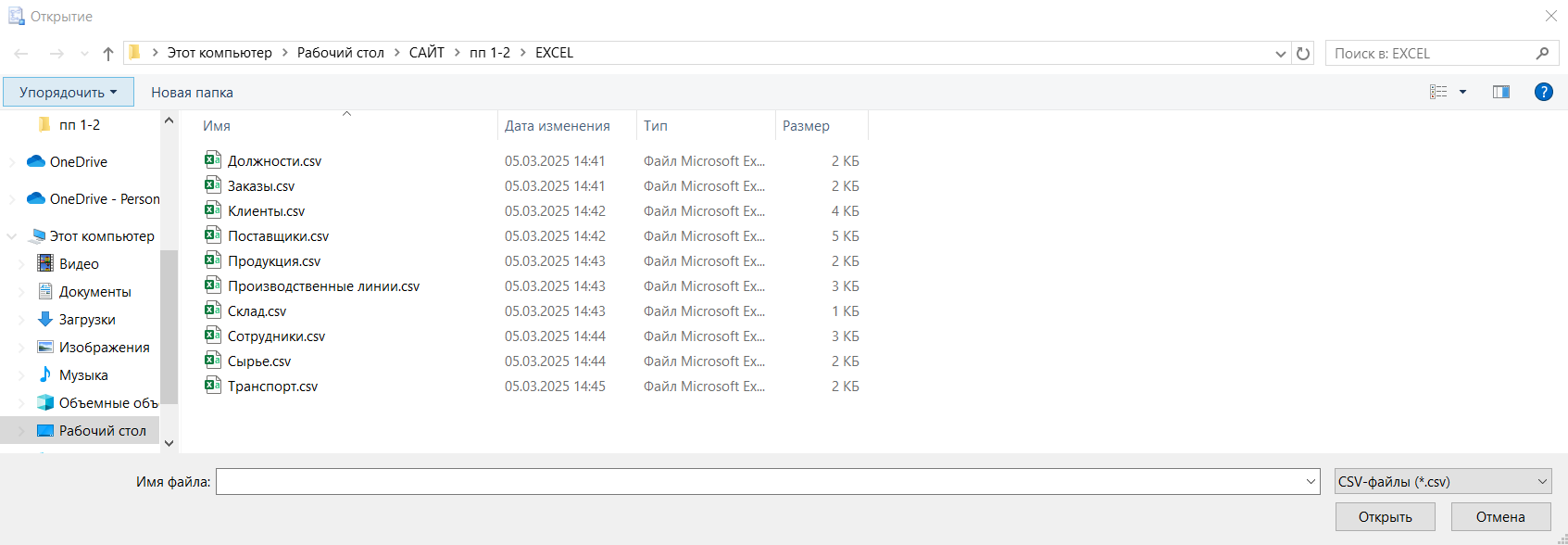


Рис.16 «Выбираем файл в формате CSV»

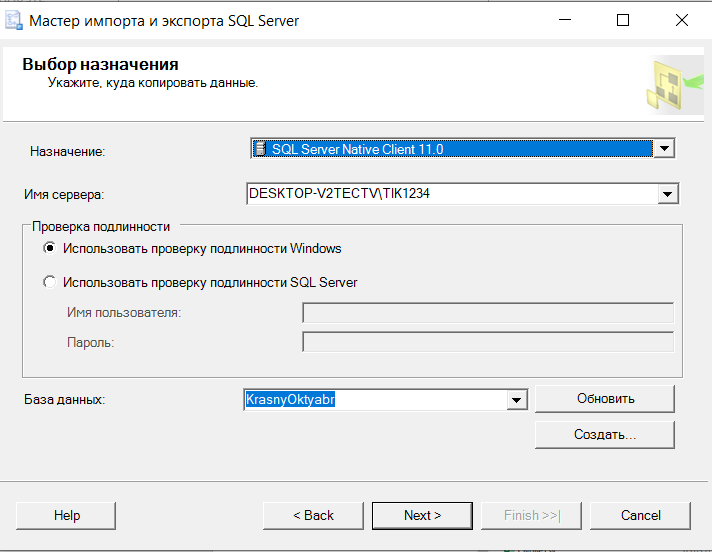


Рис.17 «Выбираем назначение копирования в базу данных»

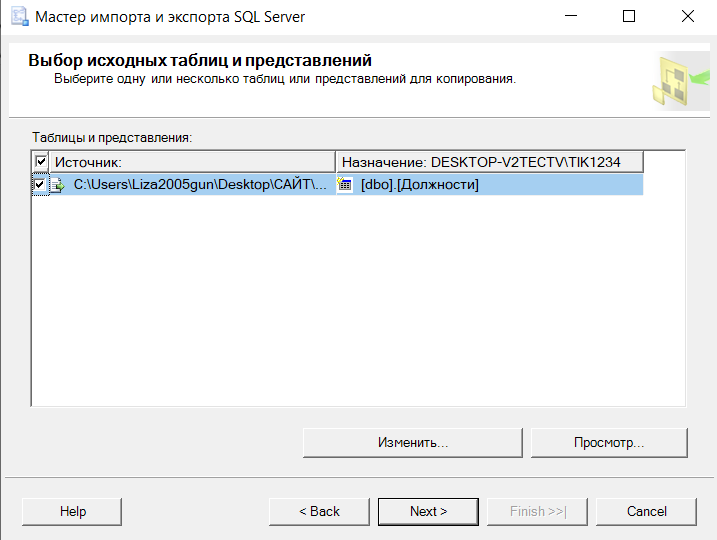


Рис.18 «Выводится таблица, в которую происходит импорт»

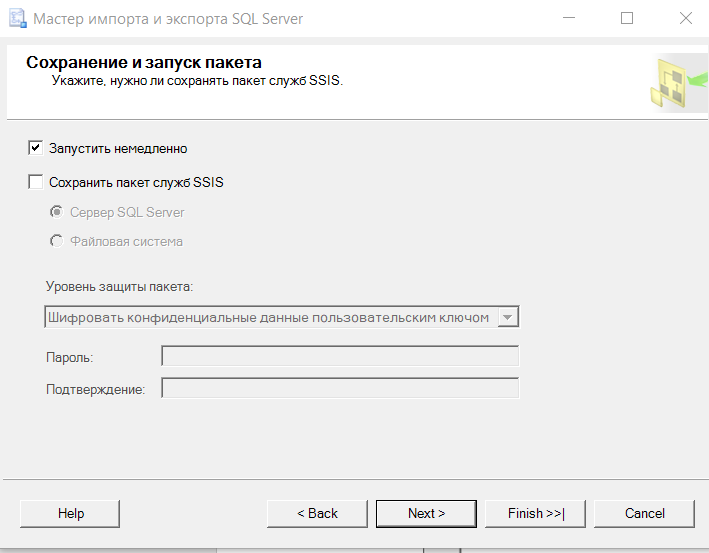


Рис.19 «Сохраняем пакет»

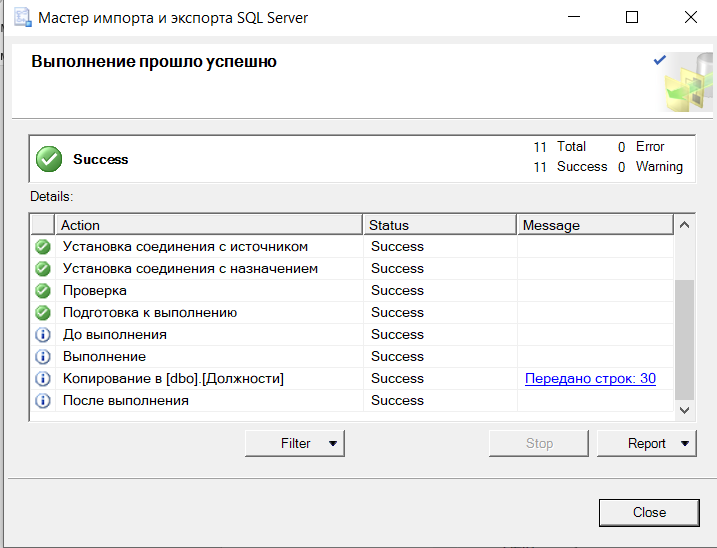


Рис.20 «Итоговый импорт»

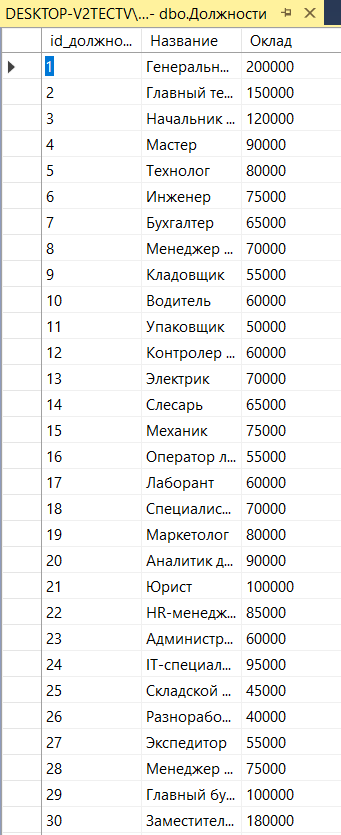


Рис.21 «Импорт таблицы Должности»

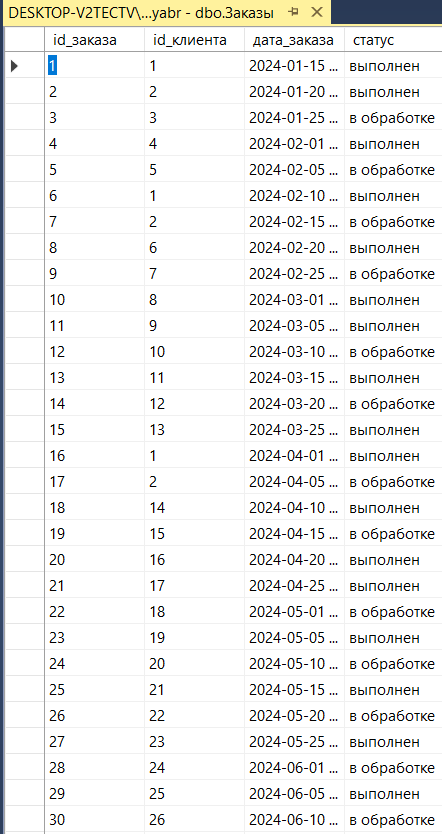


Рис.22 «Импорт таблицы Заказы»

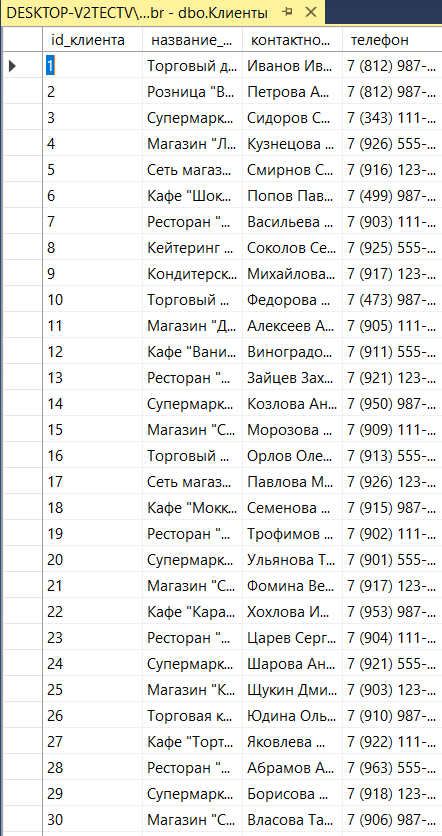


Рис.23 «Импорт таблицы Клиенты»

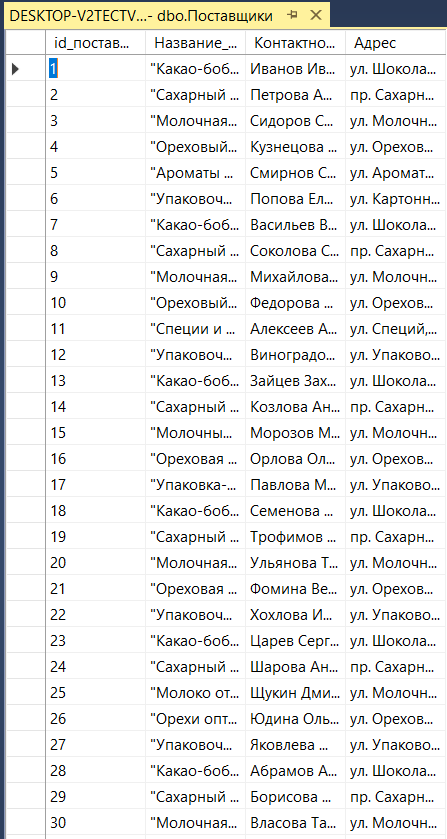


Рис.24 «Импорт таблицы Поставщики»

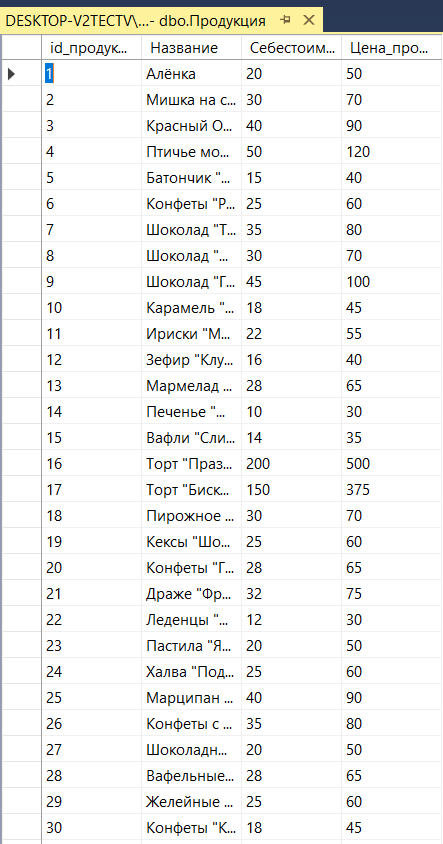


Рис.25 «Импорт таблицы Продукция»

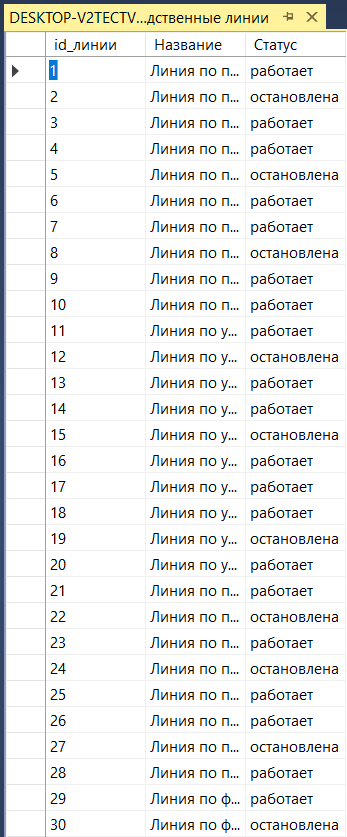


Рис.26 «Импорт таблицы Производственные линии»

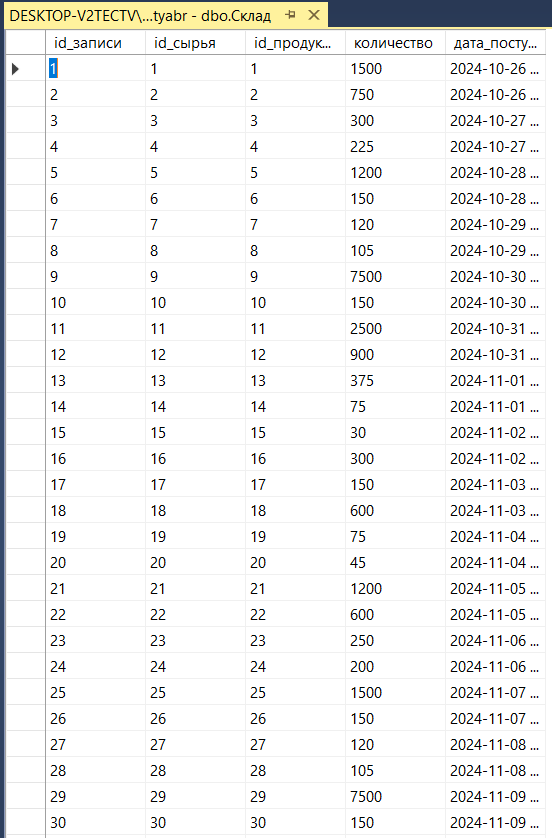


Рис.27 «Импорт таблицы Склад»

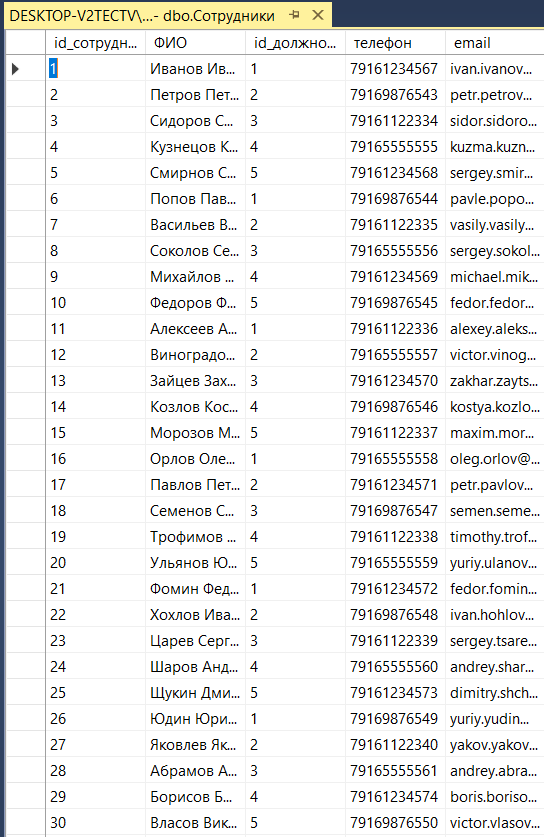


Рис.28 «Импорт таблицы Сотрудники»



Рис.29 «Импорт таблицы Сырье»

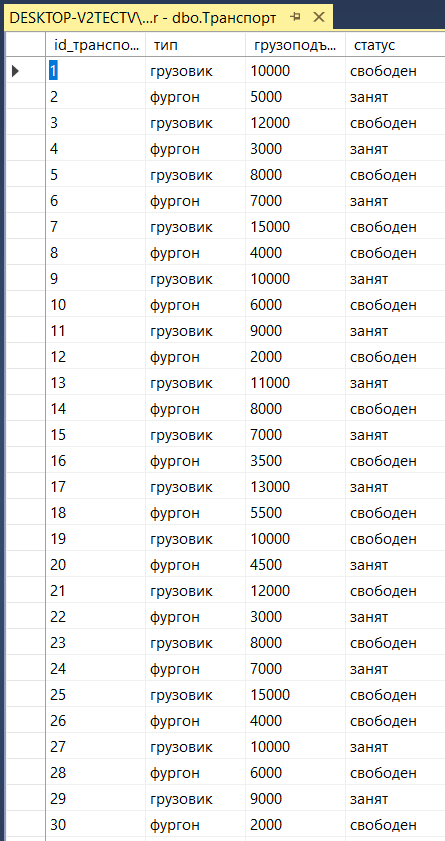


Рис.30 «Импорт таблицы Транспорт»

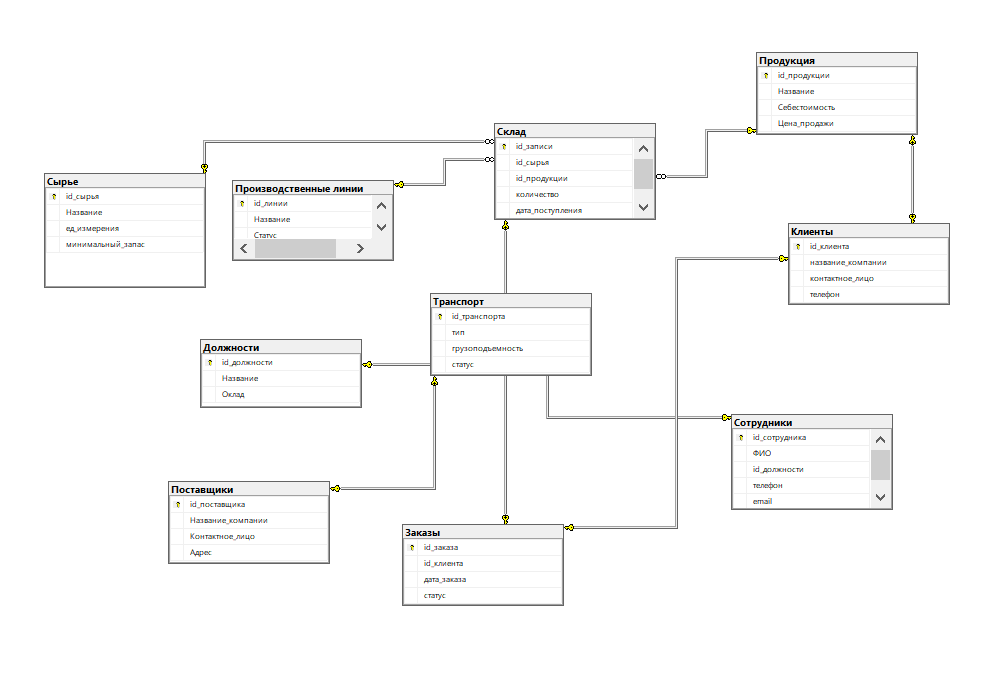


Рис. 31 «Диаграмма базы данных»

# **СОЗДАНИЕ ЗАПРОСОВ В БАЗЕ ДАННЫХ**

Запрос на выбор должностей с окладом выше 70000:

SELECT Название, Оклад FROM должности WHERE Оклад > 70000;

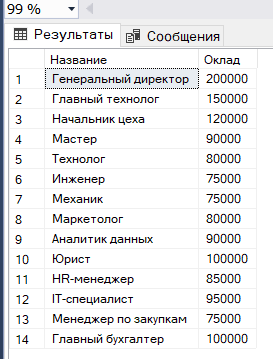


Рис. 32 «Результат выполнения запроса»

Запрос на подсчет общего числа должностей:

SELECT COUNT(\*) FROM должности;

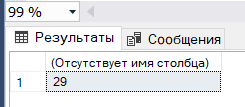


Рис. 33 «Результат выполнения запроса»

Запрос на удаление должности "Разнорабочий":

DELETE FROM должности WHERE Название = 'Разнорабочий';

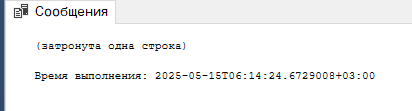


Рис. 34 «Результат выполнения запроса»

Запрос на выбор заказов со статусом "выполнен":

SELECT \* FROM Заказы WHERE статус = 'выполнен';

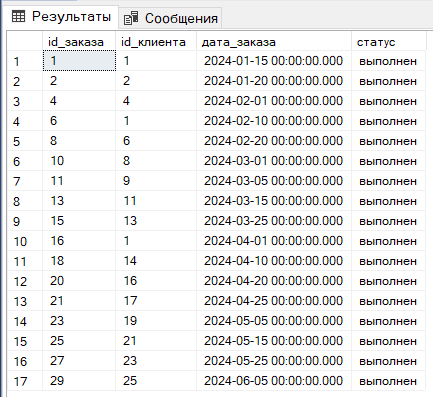


Рис. 35 «Результат выполнения запроса»

Запрос на выбор заказов, отсортированных по дате заказа:

SELECT id\_заказа, id\_клиента, дата\_заказа FROM Заказы ORDER BY дата\_заказа;

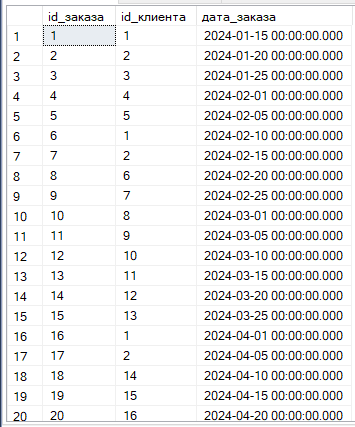


Рис. 36 «Результат выполнения запроса»

Запрос на обновление статуса заказа с id\_заказа = 3 на "выполнен":

UPDATE Заказы SET статус = 'выполнен' WHERE id\_заказа = 3;

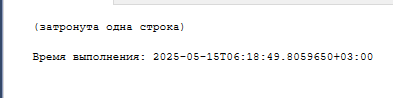


Рис. 37 «Результат выполнения запроса»

Запрос на выбор свободного транспорта:

SELECT \* FROM Транспорт WHERE статус = 'свободен';

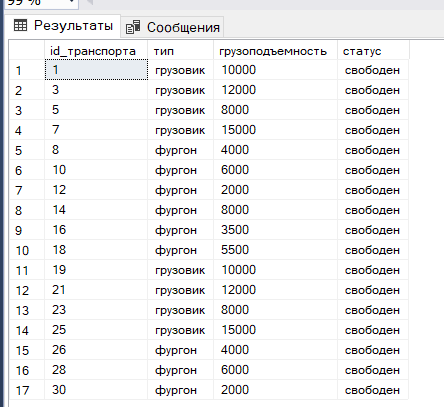


Рис. 38 «Результат выполнения запроса»

Запрос на выбор транспортных средств, отсортированных по грузоподъемности:

SELECT id\_транспорта, тип, грузоподъемность FROM Транспорт ORDER BY грузоподъемность DESC;

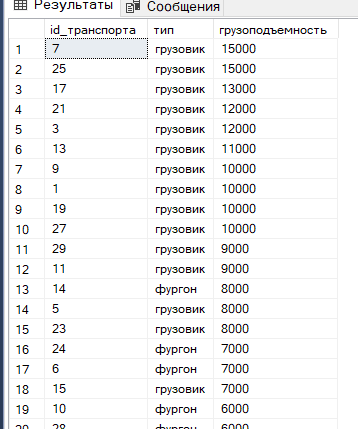
****

Рис. 39 «Результат выполнения запроса»

Запрос на выбор всех грузовиков:

SELECT \* FROM Транспорт WHERE тип = 'грузовик';

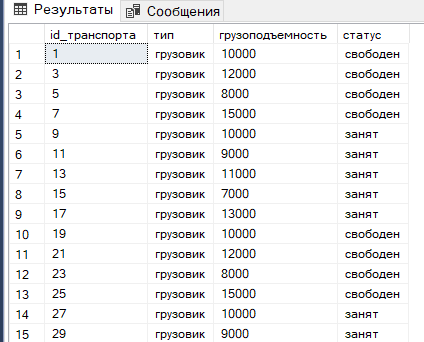
****

Рис. 40 «Результат выполнения запроса»

Запрос на удаление транспорта по id\_транспорта (например, с id\_транспорта = 10):

DELETE FROM Транспорт WHERE id\_транспорта = 10;

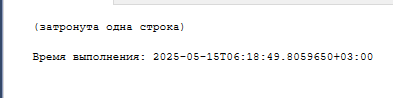


Рис. 41 «Результат выполнения запроса»

# **ИНФОРМАЦИОННАЯ СИСТЕМА ОРГАНИЗАЦИИ**

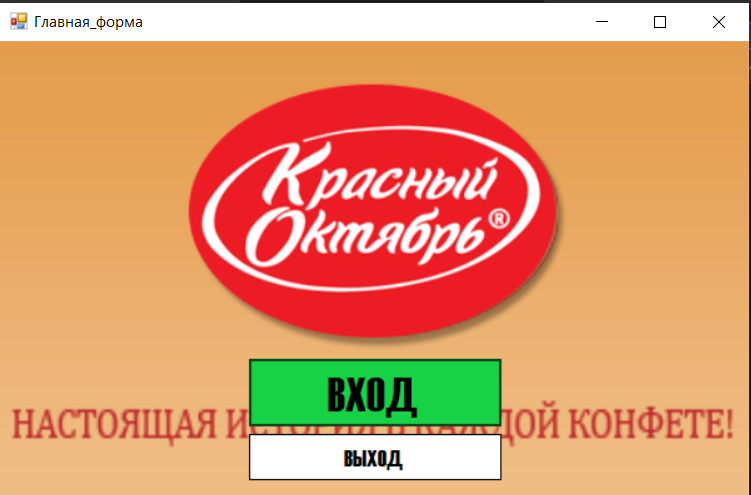


Рис. 42 «Главная форма»

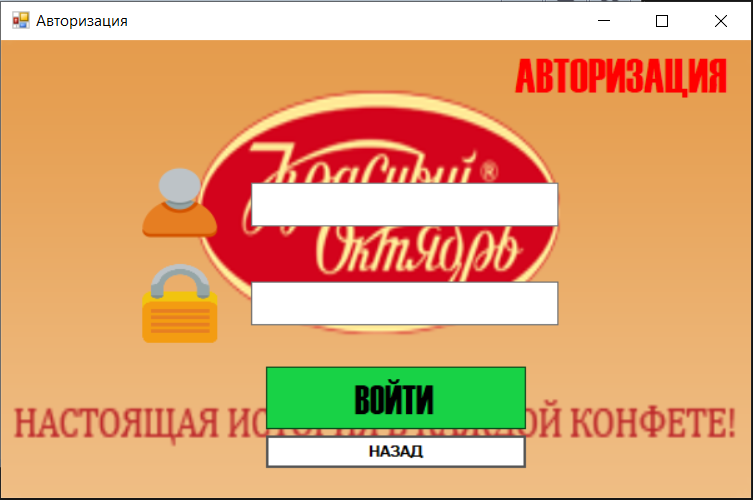


Рис. 43 «Авторизация»

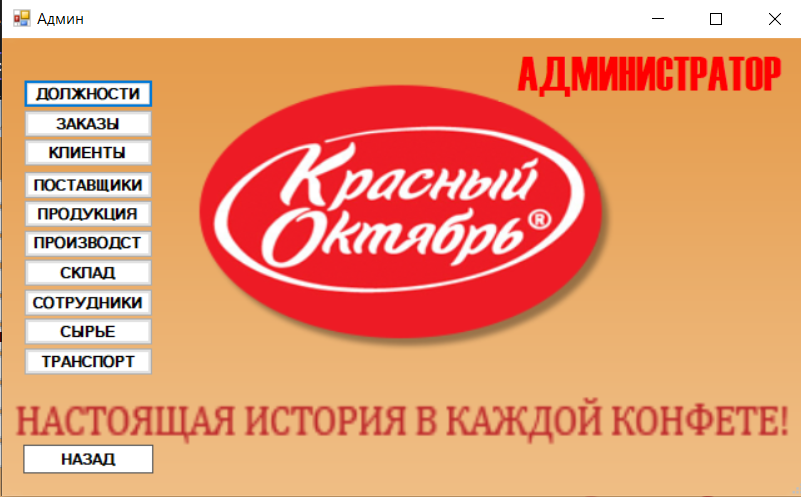


Рис. 44 «Админ»

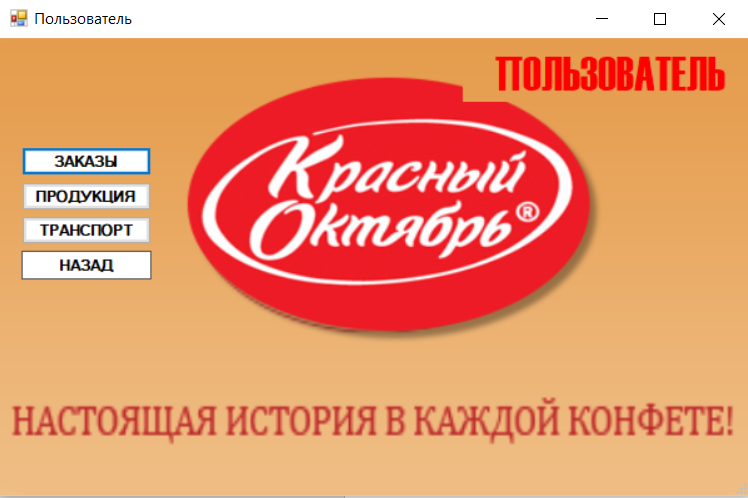


Рис. 45 «Пользователь»

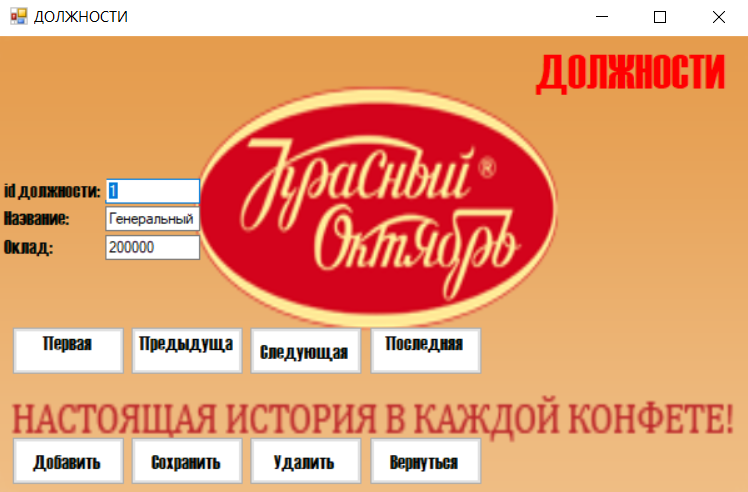
****

Рис. 46 «Форма Должности»

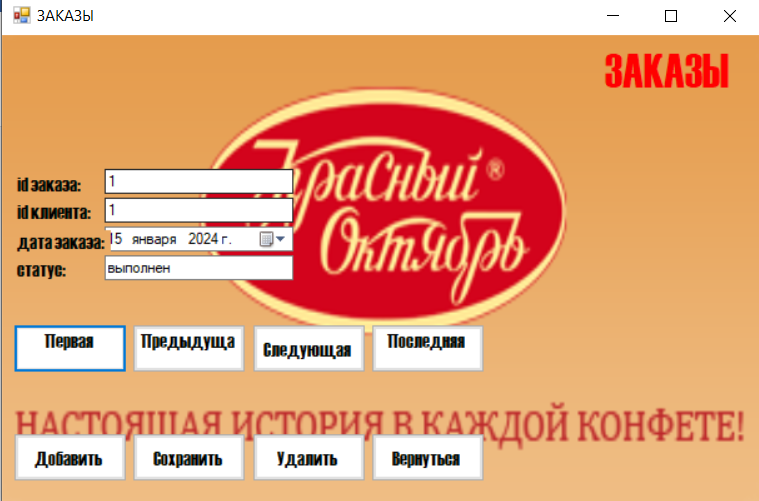
****

Рис. 47 «Форма Заказы»

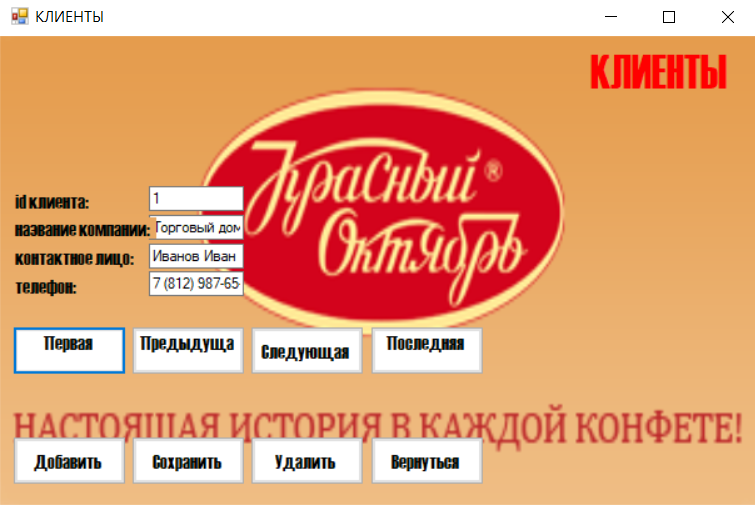
****

Рис. 48 «Форма Клиенты»

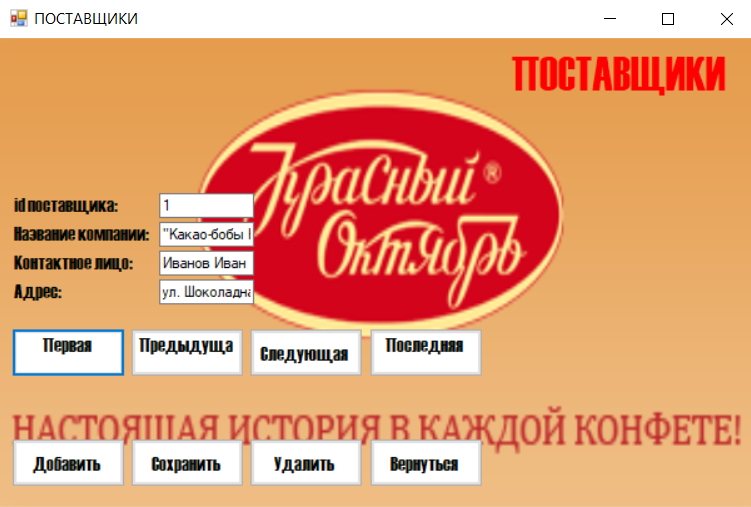
****

Рис. 49 «Форма Поставщики»

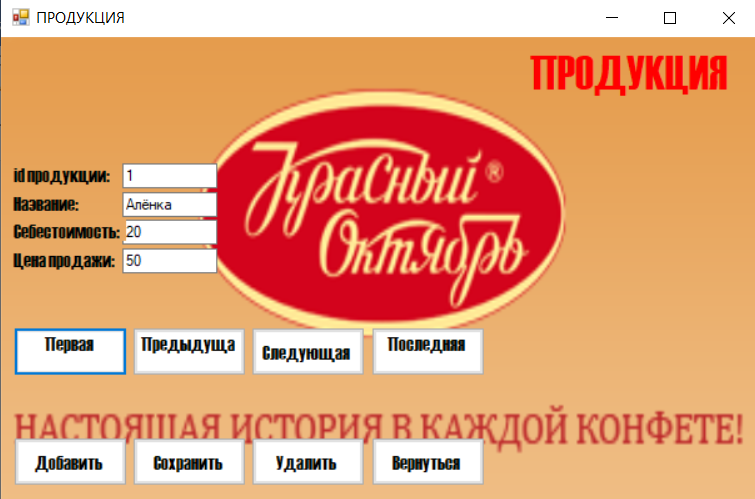


Рис. 50 «Форма Продукция»

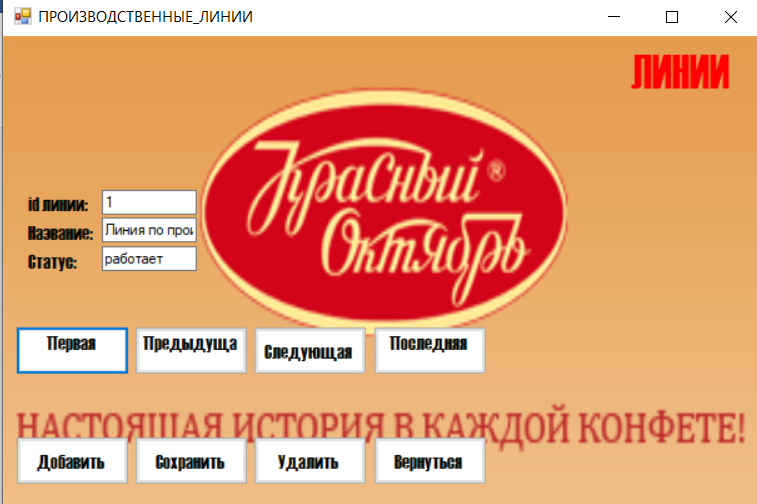


Рис. 51 «Форма Производственные линии»

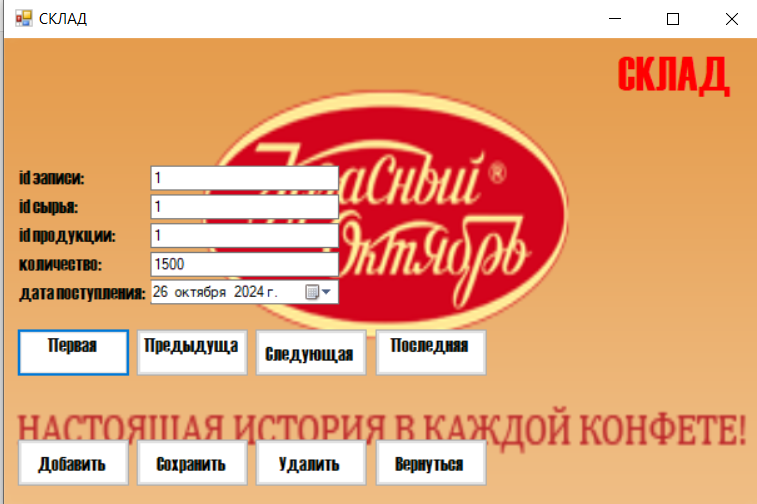


Рис. 52 «Форма Склад»

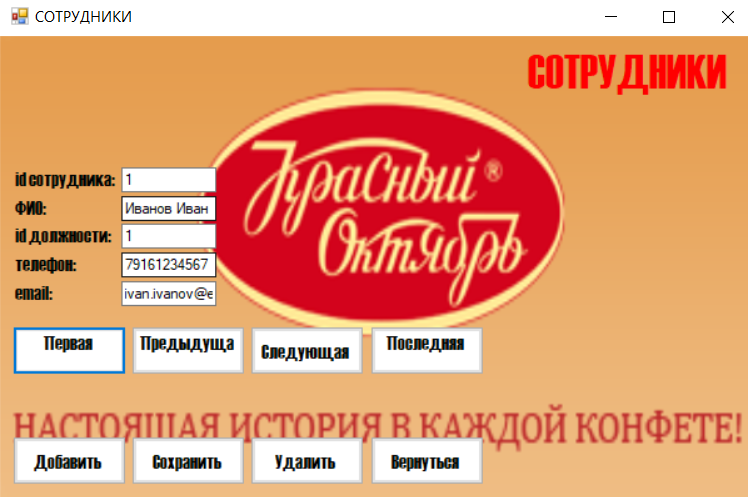


Рис. 53 «Форма Сотрудники»

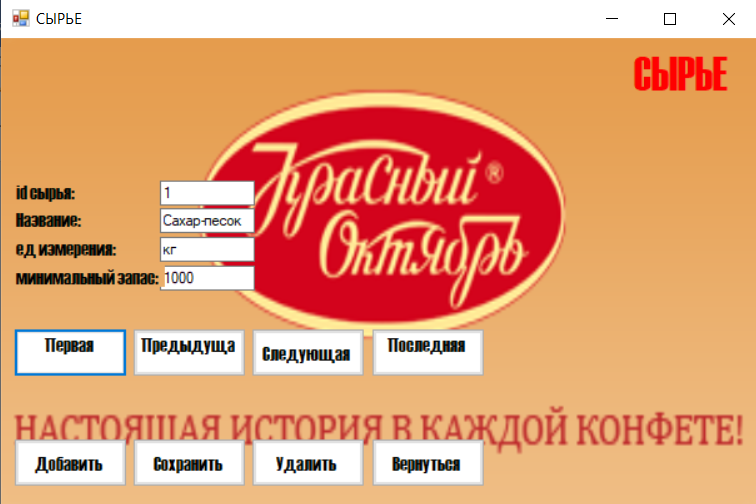


Рис. 54 «Форма Сырье»

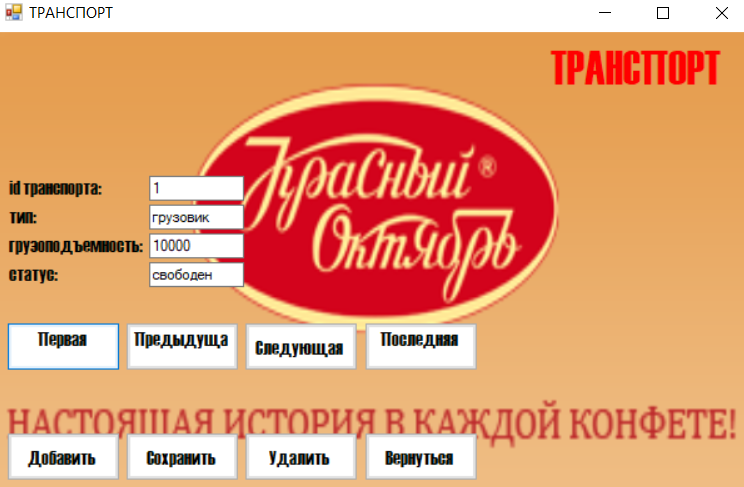


Рис. 55 «Форма Транспорт»

**ТЕСТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ**

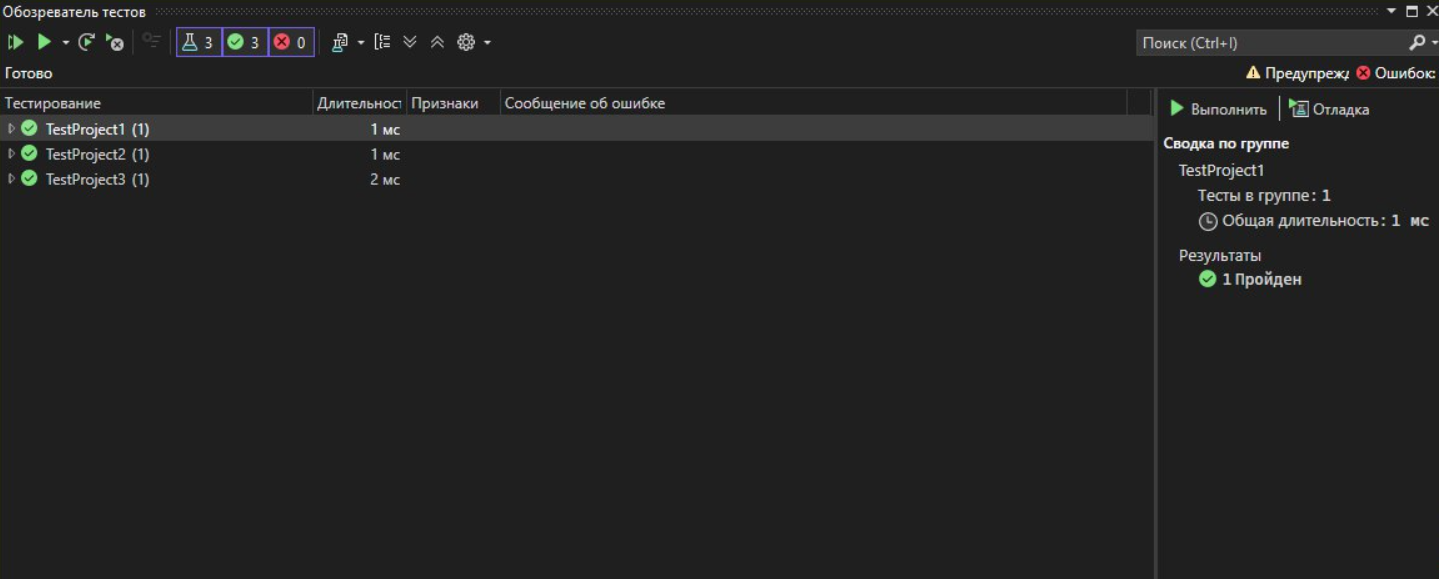


Рис. 56 «Обозреватель Unit тестов»

Тест-кейс №1 Авторизация

Таблица №3

«Авторизация»

|  |  |
| --- | --- |
| **Номер** | **1** |
| **Заголовок** | Тестирование формы «Авторизация» приложения «Красный октябрь» |
| **Предусловие** | Открыта форма «Авторизация» приложения «Красный октябрь» в программе Visual Studio |
|  | |
| **Шаг** | **Ожидаемый результат** |
| В центре формы расположен логин, в который нужно ввести свою роль «Администратор» | Введенная роль отображается в строке рядом с текстовым полем |
| В центре формы ввести корректный пароль для администратора | Введенный пароль отображается в текстовом полем |
| Посередине формы ввести роль «Пользователь» | Введенная роль отображается в строке рядом с текстовым полем |
| Посередине формы ввести корректный пароль для пользователя | Введенный пароль отображается рядом с текстовым полем |

Продолжение таблицы №3

|  |  |
| --- | --- |
| В левом нижнем углу нажать на кнопку «Выйти» | При нажатии данная кнопка позволяет выйти из приложения «Красный октябрь» |
| В верхней строчке формы нажать на кнопку «свернуть» | Свернулась форма |
| В верхней строчке формы нажать на кнопку «развернуть» | Форма открылась на весь экран |
| В верхней строчке формы нажать на кнопку «закрыть» | Форма закрылась |

Тест-кейс №2 Форма «Администратора»

Таблица №4

«Администратор»

|  |  |  |
| --- | --- | --- |
| **Номер** | **2** | |
| **Заголовок** | Тестирование формы «Администратор» приложения «Красный октябрь» | |
| **Предусловие** | | Открыта форма «Администратор» приложения «Красный октябрь» в программе Visual Studio |
| **Шаг** | | **Ожидаемый результат** |
| В верхней строчке формы нажать на кнопку «свернуть» | | Свернулась форма |
| В верхней строчке формы нажать на кнопку «развернуть» | | Форма открылась на весь экран |
| В верхней строчке формы нажать на кнопку «закрыть» | | Форма закрылась |
| Под текстовым полем «Администратор» располагается кнопка «Должности» | | Кнопка осуществляет переход на форму, на которой располагается таблица в ленточном виде «Должности» |
| Под текстовым полем «Администратор» располагается кнопка «Заказы» | | Кнопка осуществляет переход на форму, на которой располагается таблица в ленточном виде «Заказы» |

Продолжение таблицы №4

|  |  |
| --- | --- |
| Под текстовым полем «Администратор» располагается кнопка «Клиенты» | Кнопка осуществляет переход на форму, на которой располагается таблица в ленточном виде «Клиенты» |
| Под текстовым полем «Администратор» располагается кнопка «Поставщики» | Кнопка осуществляет переход на форму, на которой располагается таблица в ленточном виде «Поставщики» |
| Под текстовым полем «Администратор» располагается кнопка «Продукция» | Кнопка осуществляет переход на форму, на которой располагается таблица в ленточном виде «Продукция» |
| Под текстовым полем «Администратор» располагается кнопка «Производственные линии» | Кнопка осуществляет переход на форму, на которой располагается таблица в ленточном виде «Производственные линии» |
| Под текстовым полем «Администратор» находится кнопка «Заказы» | Кнопка осуществляет переход на форму, на которой располагается таблица в ленточном виде «Заказы» |

Тест-кейс №3 Форма «Пользователя»

Таблица №5

«Пользователь»

|  |  |  |
| --- | --- | --- |
| **Номер** | **2** | |
| **Заголовок** | Тестирование формы «Пользователь» приложения «Красный октябрь» | |
| **Предусловие** | | Открыта форма «Пользователь» приложения «Производство стеллажного оборудования» в программе Visual Studio |
| **Шаг** | | **Ожидаемый результат** |
| В верхней строчке формы нажать на кнопку «свернуть» | | Свернулась форма |
| В верхней строчке формы нажать на кнопку «развернуть» | | Форма открылась на весь экран |

Продолжение таблицы №5

|  |  |
| --- | --- |
| В верхней строчке формы нажать на кнопку «закрыть» | Форма закрылась |
| Под текстовым полем «Пользователь» располагается кнопка «Заказы» | Кнопка осуществляет переход на форму, на которой располагается таблица в ленточном виде «Заказы» |
| Под текстовым полем «Пользователь» располагается кнопка «Транспорт» | Кнопка осуществляет переход на форму, на которой располагается таблица в ленточном виде «Транспорт» |
| Под текстовым полем «Пользователь» располагается кнопка «Продукция» | Кнопка осуществляет переход на форму, на которой располагается таблица в ленточном виде «Продукция» |
| Под текстовым полем «Пользователь» располагается кнопка «Назад» | Кнопка осуществляет переход на предыдущую форму – форму авторизации |

Тест-кейс №4 Форма «Клиенты»

Таблица №6

Клиенты

|  |  |  |
| --- | --- | --- |
| **Номер** | **2** | |
| **Заголовок** | Тестирование формы «Клиенты» приложения «Красный октябрь» | |
| **Предусловие** | | Открыта форма «Клиенты» приложения «Красный октябрь» в программе Visual Studio |
| **Шаг** | | **Ожидаемый результат** |
| В верхней строчке формы нажать на кнопку «свернуть» | | Свернулась форма |
| В верхней строчке формы нажать на кнопку «развернуть» | | Форма открылась на весь экран |
| В верхней строчке формы нажать на кнопку «закрыть» | | Форма закрылась |

Продолжение таблицы №6

|  |  |
| --- | --- |
| Под текстовым полем «Клиенты» располагается кнопка «Первая» | Данная кнопка осуществляет переход на первую запись |
| Под текстовым полем «Клиенты» располагается кнопка «Следующая» | Данная кнопка осуществляет переход на следующую запись |
| Под текстовым полем «Клиенты» располагается кнопка «Предыдущая» | Данная кнопка осуществляет переход на предыдущую запись |
| Под текстовым полем «Клиенты» располагается кнопка «Последняя» | Данная кнопка осуществляет переход на последнюю запись |
| Под текстовым полем «Клиенты» располагается кнопка «Добавить» | Данная кнопка осуществляет добавление новой записи |
| Под текстовым полем «Клиенты» располагается кнопка «Сохранить» | Данная кнопка осуществляет сохранение новой записи |
| Под текстовым полем «Клиенты» располагается кнопка «Удалить» | Данная кнопка осуществляет удаление записи |

Тест-кейс №5 Форма «Заказы»

Таблица №7

Заказы

|  |  |  |  |
| --- | --- | --- | --- |
| **Номер** | **2** | | |
| **Заголовок** | Тестирование формы «Заказы» приложения «Красный октябрь» | | |
| **Предусловие** | | Открыта форма «Заказы» приложения «Красный октябрь» в программе Visual Studio | |
| **Шаг** | | **Ожидаемый результат** | |
| В верхней строчке формы нажать на кнопку «свернуть» | | Свернулась форма | |
| В верхней строчке формы нажать на кнопку «развернуть» | | Форма открылась на весь экран | |
| В верхней строчке формы нажать на кнопку «закрыть» | | Форма закрылась | |
| Под текстовым полем «Проекты» располагается кнопка «Первая» | | | Данная кнопка осуществляет переход на первую запись |
| Под текстовым полем «Клиенты» располагается кнопка «Следующая» | | | Данная кнопка осуществляет переход на следующую запись |

Продолжение таблицы №7

|  |  |
| --- | --- |
| Под текстовым полем «Проекты» располагается кнопка «Предыдущая» | Данная кнопка осуществляет переход на предыдущую запись |
| Под текстовым полем «Проекты» располагается кнопка «Последняя» | Данная кнопка осуществляет переход на последнюю запись |
| Под текстовым полем «Проекты» располагается кнопка «Добавить» | Данная кнопка осуществляет добавление новой записи |
| Под текстовым полем «Проекты» располагается кнопка «Сохранить» | Данная кнопка осуществляет сохранение новой записи |
| Под текстовым полем «Проекты» располагается кнопка «Удалить» | Данная кнопка осуществляет удаление записи |

Чек лист №1

Таблица №8

Проверка «Главная форма»

|  |  |
| --- | --- |
| **Проверки** | **Работает / не работает** |
| **Титульная** |  |
| Проверка работы кнопки «Свернуть приложение» | Успешно |
| Проверка работы функционала разворачивания приложения из панели задач | Успешно |
| Проверка работы кнопки «Развернуть приложение» | Успешно |
| Проверка работы кнопки «Закрыть приложение» | Успешно |
| Проверка кнопки «Войти» | Успешно |
| Проверка кнопки «Выйти» | Успешно |

Чек лист №2

Таблица №9

Проверка формы «Авторизация»

|  |  |
| --- | --- |
| **Проверки** | **Работает / не работает** |
| **Авторизация** |  |
| Проверка ввода текста в поле ввода пароля | Успешно |
| Проверка возможности выбора пользователя | Успешно |
| Проверка работы кнопки «Вход» от имени администратора | Успешно |

Продолжение таблицы №9

|  |  |
| --- | --- |
| Проверка работы кнопки «Выход» от имени администратора | Успешно |
| Проверка работы кнопки «Вход» от имени пользователя | Успешно |
| Проверка работы кнопки «Выход» от имени пользователя | Успешно |
| Проверка работы кнопки «Назад» | Успешно |

Чек лист №3

Таблица №10

Проверка формы «Админ»

|  |  |
| --- | --- |
| **Проверки** | **Работает / не работает** |
| **Администратор** |  |
| Проверка работы кнопки «Свернуть приложение» | Успешно |
| Проверка работы кнопки «Развернуть приложение» | Успешно |
| Проверка работы кнопки «Закрыть приложение» | Успешно |
| Проверка кнопки «Клиенты» | Успешно |
| Проверка кнопки «Документы» | Успешно |
| Проверка кнопки «Материалы» | Успешно |
| Проверка кнопки «Поставщики» | Успешно |
| Проверка кнопки «Поставки» | Успешно |
| Проверка кнопки «Проекты» | Успешно |
| Проверка кнопки «Рабочие» | Успешно |
| Проверка кнопки «Склад» | Успешно |
| Проверка кнопки «Заказанные материалы» | Успешно |
| Проверка кнопки «Заказы» | Успешно |
| Проверка кнопки «Назад» | Успешно |

Чек лист №4

Таблица №11

Проверка формы «Пользователь»

|  |  |
| --- | --- |
| **Проверки** | **Работает / не работает** |
| **Пользователь** |  |
| Проверка работы кнопки «Свернуть приложение» | Успешно |

Продолжение таблицы №11

|  |  |
| --- | --- |
| Проверка работы кнопки «Развернуть приложение» | Успешно |
| Проверка работы кнопки «Закрыть приложение» | Успешно |
| Проверка кнопки «Заказы» | Успешно |
| Проверка кнопки «Документы» | Успешно |
| Проверка кнопки «Склад» | Успешно |
| Проверка кнопки «Назад» | Успешно |

Чек лист №5

Таблица №12

Проверка формы «Заказы»

|  |  |
| --- | --- |
| **Проверки** | **Работает / не работает** |
| **Документы** |  |
| Проверка работы кнопки «Свернуть приложение» | Успешно |
| Проверка работы функционала разворачивания приложения из панели задач | Успешно |
| Проверка работы кнопки «Развернуть приложение» | Успешно |
| Проверка работы кнопки «Закрыть приложение» | Успешно |
| Проверка кнопки «Первая» | Успешно |
| Проверка кнопки «Следующая» | Успешно |
| Проверка кнопки «Предыдущая» | Успешно |
| Проверка кнопки «Последняя» | Успешно |
| Проверка кнопки «Добавить» | Успешно |
| Проверка кнопки «Сохранить» | Успешно |
| Проверка кнопки «Удалить» | Успешно |
| Проверка кнопки «Назад» | Успешно |

**РАЗРАБОТКА ИПС В РАМКАХ ПРОИЗВОДСТВЕННОГО ПРОЦЕССА ПРЕДПРИЯТИЯ**

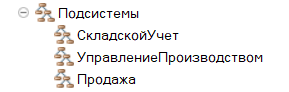


Рис. 57 «Подсистемы»

Для ООО «Авето», занимающейся производством стеллажного оборудования, эти справочники необходимы для:

1. Клиенты: Управление информацией о заказчиках, ведение истории взаимоотношений, сегментация для маркетинга.
2. Материалы: Учет используемого сырья и комплектующих, контроль складских запасов, расчет себестоимости продукции.
3. Каталог стеллажей: Систематизация информации о производимой продукции, формирование предложений для клиентов, оперативный расчет стоимости.
4. Поставщики: Ведение базы данных поставщиков, отслеживание условий сотрудничества, выбор оптимальных поставщиков.
5. Сотрудники: Учет персонала, распределение задач, контроль выполнения, расчет заработной платы.
6. Статусы заказов: Отслеживание этапов выполнения заказов, оперативное информирование клиентов, контроль сроков.
7. Типы платежей: Учет финансовых операций, контроль поступления средств, анализ дебиторской задолженности.
8. Цена на материалы: Учет стоимости материалов, оперативный пересчет цен на продукцию, анализ рентабельности.

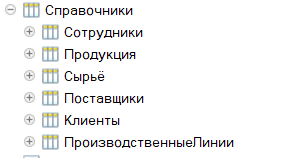


Рис. 58 «Справочники»

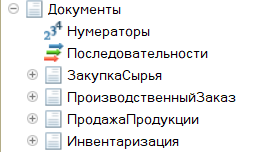


Рис. 59 «Документы»

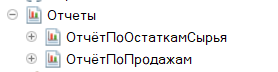


Рис. 60 «Отчеты»

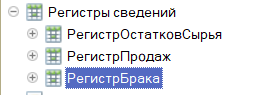


Рис. 61 «Регистры сведений»

**ПРОЦЕСС ВНЕДРЕНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

Оборудование и его характеристики

| № | Назначение на производстве | Устройства | ПО/Версия | Системная плата | Процессор | Оперативная память | Жёсткий диск |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | АРМ технолога | Мышь: Genius (NetScroll 100); Клавиатура: Defender (K5890); Монитор: Samsung (SyncMaster 933) | Windows 7 Professional | GIGABYTE GA-945GCM-S2L | Intel Pentium Dual CPU E2160 1.80GHz | DDR2 (2ГБ.) | WDC WD1600AAJS-00WAA0 (148ГБ.) |
| 2 | АРМ логиста | Мышь: Defender (Optimum MM); Клавиатура: Logitech (Y-SAP76); Монитор: LG (Flatron E1942) | Windows 7 Ultimate | P8H61-MX R2.0 | Intel Pentium G2020 2.90GHz | DDR3 (2ГБ.) | WDC WD2500AAJS-08L7A0 (232ГБ.) |
| 3 | АРМ мастера производства | Мышь: Defender (DF2330B); Клавиатура: A4Tech (KB-720); Монитор: ViewSonic (VA2013wm) | Windows 7 Ultimate | 915GV-M10 | Intel Celeron 2.66GHz | DDR2 (2ГБ.) | Samsung SP0842N (73ГБ.) |
| 4 | АРМ кладовщика сырья | Мышь: Genius NetScroll+Eye Ruby PS/2; Клавиатура: Defender (K5890); Монитор: Samsung (943n) | Windows 7 Ultimate | ASUS M2A-MX | AMD Athlon 64 X2 Dual 3800+ 2.00GHz | DDR2 (2ГБ.) | WDC WD1600AAJS-00WAA0 (148ГБ.) |
| 5 | АРМ кладовщика готовой продукции | Мышь: Genius NetScroll 110; Клавиатура: Genius (K627); Монитор: ViewSonic (VA2013wm) | Windows 7 Ultimate | ASUS 86M0ACF82425 | AMD Athlon 64 X2 Dual 4200+ 2.20GHz | DDR2 (2.5ГБ.) | Hitachi HDT721016SLA380 (147ГБ.) |

Продолжение таблицы 2

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 6 | АРМ бухгалтера | Мышь: A4Tech OP-620D; Клавиатура: Perfeo (6106); Монитор: ViewSonic (VA2013wm) | Windows 10 PRO | MSI H310M Prove PLUS | Intel Pentium Gold G5400 3.70 GHz | DDR4 (8ГБ.) | WDC WD10EZEX-08WN4A0 (920ГБ.) |
| 7 | АРМ отдела качества | Мышь: M Sniper 7630; Клавиатура: Defender (Element HB-520); Монитор: ViewSonic (VA2013wm) | Windows 10 | ASUS PSKPL-CM | Intel Core i3-7100 3.90GHz | DDR2 (2ГБ.) | MAXTOR STM380815AS (80ГБ.) |
| 8 | АРМ инженера по оборудованию | Мышь: smartbuy SBM-214-K; Клавиатура: 4Tech (KB-820); Монитор: LG (Flatron W1943SS) | Windows 10 Pro | MSI H110M PRO-VD PLUS | Intel Pentium G4400 3.30GHz | DDR4 (4ГБ.) | ST500DM002-1BD142 (358ГБ.) |
| 9 | АРМ диспетчера снабжения | Мышь: Logitech PS/2; Клавиатура: Defender (K5890); Монитор: Samsung (943n) | Windows 7 Ultimate | ASUS P5KPL-AM IN/ROEM/SI | Intel Celeron 2.93GHz | DDR2 (2ГБ.) | ST250DM001 HD253GJ (232ГБ.) |
| 10 | АРМ экономиста | Мышь: A4Tech Op-620D; Клавиатура: Defender (K5890); Монитор: LG (Flatron E1042) | Windows 7 Professional | ASRock H61M-HVS | Intel Celeron G530 2.40GHz | DDR3 (4ГБ.) | ST320DM000-1BD14C (297ГБ.) |

**ПРИЛОЖЕНИЯ**

Приложение №1

Листинг программы

Главная форма:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace КРАСНЫЙ\_ОКТЯБРЬ

{

public partial class Главная\_форма : Form

{

public Главная\_форма()

{

InitializeComponent();

}

private void button2Vixod\_Click\_1(object sender, EventArgs e)

{

Close();

}

private void button1Vxod\_Click(object sender, EventArgs e)

{

Form Главная = new Авторизация();

Главная.Show();

this.Hide();

}

private void Главная\_форма\_Load(object sender, EventArgs e)

{

}

}

}

Форма авторизации:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace КРАСНЫЙ\_ОКТЯБРЬ

{

public partial class Авторизация : Form

{

public Авторизация()

{

InitializeComponent();

}

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

{

}

private void label1\_Click(object sender, EventArgs e)

{

}

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

{

string connectionString = @"Data Source=DESKTOP-V2TECTV\TIK1234;Initial Catalog=AVTORIZACIA;Integrated Security=True;Encrypt=False";

using (SqlConnection sqlCon = new SqlConnection(connectionString))

{

try

{

sqlCon.Open();

string query = @" SELECT Username, Password FROM Avtorizacia WHERE Username = @Username AND Password = @Password";

using (SqlCommand sqlCmd = new SqlCommand(query, sqlCon))

{

sqlCmd.Parameters.AddWithValue("@Username", PasWord.Text.Trim());

sqlCmd.Parameters.AddWithValue("@Password", PasField.Text.Trim());

SqlDataAdapter sda = new SqlDataAdapter(sqlCmd);

DataTable dt = new DataTable();

sda.Fill(dt);

if (dt.Rows.Count == 1)

{

this.Hide();

string role = dt.Rows[0]["Username"].ToString();

if (role == "Admin")

{

Form ADMIN = new Админ();

ADMIN.ShowDialog();

}

else if (role == "Polzovatel")

{

Form POLZOVATEL = new Пользователи();

POLZOVATEL.ShowDialog();

}

}

else

{

MessageBox.Show("Неверное имя пользователя или пароль.");

PasField.Clear();

PasWord.Clear();

}

}

}

catch (Exception ex)

{

MessageBox.Show("Ошибка при подключении к базе данных: " + ex.Message);

}

}

}

private void button2Vixod\_Click(object sender, EventArgs e)

{

Form Авторизация = new Главная\_форма();

Авторизация.Show();

this.Hide();

}

private void PasField\_TextChanged(object sender, EventArgs e)

{

}

private void pictureBox2\_Click(object sender, EventArgs e)

{

}

private void PasWord\_TextChanged(object sender, EventArgs e)

{

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

}

}

}

Форма Админа:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace КРАСНЫЙ\_ОКТЯБРЬ

{

public partial class Админ : Form

{

public Админ()

{

InitializeComponent();

}

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

{

Form Админ = new ПРОИЗВОДСТВЕННЫЕ\_ЛИНИИ();

Админ.Show();

this.Hide();

}

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

{

Form Админ = new ТРАНСПОРТ();

Админ.Show();

this.Hide();

}

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

{

Form Админ = new СЫРЬЕ();

Админ.Show();

this.Hide();

}

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

{

Form Админ = new СОТРУДНИКИ();

Админ.Show();

this.Hide();

}

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

{

Form Админ = new СКЛАД();

Админ.Show();

this.Hide();

}

private void button2Vixod\_Click(object sender, EventArgs e)

{

Form Админ = new Авторизация();

Админ.Show();

this.Hide();

}

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

{

Form Админ = new ПРОДУКЦИЯ();

Админ.Show();

this.Hide();

}

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

{

Form Админ = new ПОСТАВЩИКИ();

Админ.Show();

this.Hide();

}

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

{

Form Админ = new КЛИЕНТЫ();

Админ.Show();

this.Hide();

}

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

{

Form Админ = new ЗАКАЗЫ();

Админ.Show();

this.Hide();

}

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

{

Form Админ = new ДОЛЖНОСТИ();

Админ.Show();

this.Hide();

}

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

{

}

}

}

Форма Пользователя:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace КРАСНЫЙ\_ОКТЯБРЬ

{

public partial class Пользователи : Form

{

public Пользователи()

{

InitializeComponent();

}

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

{

}

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

{

Form Пользователь = new ЗАКАЗЫ();

Пользователь.Show();

this.Hide();

}

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

{

Form Пользователь = new ТРАНСПОРТ();

Пользователь.Show();

this.Hide();

}

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

{

Form Пользователь = new ПРОДУКЦИЯ();

Пользователь.Show();

this.Hide();

}

private void button2Vixod\_Click(object sender, EventArgs e)

{

}

}

}

Форма «Должности» для формы Администратора:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace КРАСНЫЙ\_ОКТЯБРЬ

{

public partial class ДОЛЖНОСТИ : Form

{

public ДОЛЖНОСТИ()

{

InitializeComponent();

}

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

{

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

this.должностиTableAdapter.Fill(this.krasnyOktyabrDataSet.Должности);

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void должностиBindingSource\_CurrentChanged(object sender, EventArgs e)

{

}

private void id\_должностиLabel\_Click(object sender, EventArgs e)

{

}

private void id\_должностиTextBox\_TextChanged(object sender, EventArgs e)

{

}

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

{

}

private void названиеTextBox\_TextChanged(object sender, EventArgs e)

{

}

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

{

}

private void окладTextBox\_TextChanged(object sender, EventArgs e)

{

}

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

{

try

{

должностиBindingSource.RemoveCurrent();

должностиBindingSource.EndEdit();

tableAdapterManager.UpdateAll(krasnyOktyabrDataSet);

MessageBox.Show("Билет удалён", "Успех", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception ex)

{

MessageBox.Show("Ошибка при удалении билета: " + ex.Message, "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

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

{

try

{

должностиBindingSource.EndEdit();

tableAdapterManager.UpdateAll(krasnyOktyabrDataSet);

MessageBox.Show("Изменения сохранены", "Успех", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

catch (Exception ex)

{

MessageBox.Show("Ошибка при сохранении изменений: " + ex.Message, "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

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

{

должностиBindingSource.AddNew();

}

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

{

this.должностиBindingSource.MoveNext(); // Переход на первую запись

this.tableAdapterManager.UpdateAll(this.krasnyOktyabrDataSet);

}

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

{

Form Должности = new Админ();

Должности.Show();

this.Hide();

}

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

{

this.должностиBindingSource.MovePrevious(); // Переход на первую запись

this.tableAdapterManager.UpdateAll(this.krasnyOktyabrDataSet);

}

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

{

this.должностиBindingSource.MoveLast(); // Переход на первую запись

this.tableAdapterManager.UpdateAll(this.krasnyOktyabrDataSet);

}

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

{

this.должностиBindingSource.MoveFirst(); // Переход на первую запись

this.tableAdapterManager.UpdateAll(this.krasnyOktyabrDataSet);

}

}

}

Форма «Заказы» для формы Пользователя:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace КРАСНЫЙ\_ОКТЯБРЬ

{

public partial class ЗАКАЗЫ : Form

{

public ЗАКАЗЫ()

{

InitializeComponent();

}

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

{

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

this.заказыTableAdapter.Fill(this.krasnyOktyabrDataSet.Заказы);

}

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

{

this.заказыBindingSource.MoveFirst(); // Переход на первую запись

this.tableAdapterManager.UpdateAll(this.krasnyOktyabrDataSet);

}

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

{

this.заказыBindingSource.MoveNext(); // Переход на первую запись

this.tableAdapterManager.UpdateAll(this.krasnyOktyabrDataSet);

}

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

{

Form Заказы = new Админ();

Заказы.Show();

this.Hide();

}

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

{

this.заказыBindingSource.MovePrevious(); // Переход на первую запись

this.tableAdapterManager.UpdateAll(this.krasnyOktyabrDataSet);

}

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

{

this.заказыBindingSource.MoveLast(); // Переход на первую запись

this.tableAdapterManager.UpdateAll(this.krasnyOktyabrDataSet);

}

}

}