|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| voenmeh | МИНОБРНАУКИ РОССИИ  федеральное государственное бюджетное образовательное учреждение  высшего образования  **«Балтийский государственный технический университет «ВОЕНМЕХ» им. Д.Ф. Устинова»**  **(БГТУ «ВОЕНМЕХ» им. Д.Ф. Устинова»)** | | | | |
|  | | | | |
| Факультет | |  | И |  | Информационные системы |
|  | |  | Шифр |  | наименование |
| Кафедра | |  | И5 |  | Информационные системы и программная инженерия |
|  | |  | шифр |  | наименование |
| Дисциплина | |  | Базы данных | | |

КУРСОВОЙ ПРОЕКТ

на тему

|  |
| --- |
| «Информационно-программный комплекс для обеспечения работы |
| директора магазина одежды» |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Выполнил студент группы | | | |  | И584 |
| Труфанов Г.Е. | | | | | |
| Фамилия И.О. | | | | | |
| **ПРЕПОДАВАТЕЛЬ** | | | | | |
| Верхолат А.М. | |  |  | | |
| Фамилия И.О. Подпись | | | | | |
| Оценка |  | | | |  |
| «\_\_\_\_\_» |  | | | | 2021 г. |

Санкт-Петербург

2021

РЕФЕРАТ

В пояснительной записке к КП 50 страниц, 27 иллюстрации, 8 таблиц, 4 использованных источника, 3 приложения.

ИНФОРМАЦИОННО-ПРОГРАМНЫЙ КОМПЛЕКС, ИНФОРМАЦИОННАЯ СИСТЕМА, МАГАЗИН ОДЕЖДЫ, ИНФОЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ, ДАТАЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ, БАЗА ДАННЫХ, АВТОМАТИЗАЦИЯ, POSTGRE SQL.

Объектом исследования является магазин одежды.

Цель работы – разработка ИПК для магазина одежды, обеспечивающего работу директора по закупкам одежды у поставщиков, а также работу продавца по продажам одежды клиентам магазина.

В ходе выполнения курсового проекта был исследован сложившийся ручной процесс обработки данных. Было проведено инфологическое, даталогическое моделирование, спроектирован интерфейс пользователя. Внедрение реализованного ИПК приведёт к автоматизации и упрощению работы магазина одежды.

**СОДЕРЖАНИЕ**

[РЕФЕРАТ 2](#_Toc75391147)

[ПОСТАНОВКА ЗАДАЧИ 5](#_Toc75391148)

[2 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ 6](#_Toc75391149)

[2.1 Описание предметной области автоматизации 6](#_Toc75391150)

[2.2 Бизнес правила 6](#_Toc75391151)

[2.3 Состав сущностей и атрибутов предметной области 7](#_Toc75391152)

[2.4 Постановка задачи автоматизированной обработки 7](#_Toc75391153)

[3 ИНФОЛОГИЧЕСКОЕ МОДЕЛИРОВАНИЕ 9](#_Toc75391154)

[3.1 Спецификация сущностей 9](#_Toc75391155)

[3.2 Спецификация атрибутов 11](#_Toc75391156)

[3.3 Спецификация связей 14](#_Toc75391157)

[3.4 Ссылочная целостность по связям 16](#_Toc75391158)

[3.5 Инфологическая схема 17](#_Toc75391159)

[4 ДАТАЛОГИЧЕСКОЕ МОДЕЛИРОВАНИЕ 19](#_Toc75391160)

[4.1 Спецификация отношений 19](#_Toc75391161)

[4.2 Спецификация полей 20](#_Toc75391162)

[4.3 Спецификация связей 22](#_Toc75391163)

[4.4 Спецификация ключей 23](#_Toc75391164)

[4.5 Список ключей и ссылочная целостность по связям 24](#_Toc75391165)

[4.6 Даталогическая схема 26](#_Toc75391166)

[4.7 Триггерные процедуры 27](#_Toc75391167)

[5 ПРОЕКТИРОВАНИЕ ИНТЕРФЕЙСА ПОЛЬЗОВАТЕЛЯ 28](#_Toc75391168)

[6 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 33](#_Toc75391169)

[ЗАКЛЮЧЕНИЕ 43](#_Toc75391170)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 44](#_Toc75391171)

[ПРИЛОЖЕНИЕ А – SQL-скрипт создания БД 45](#_Toc75391172)

[ПРИЛОЖЕНИЕ Б – Справочные запросы 46](#_Toc75391173)

[Типовые SQL-запросы 46](#_Toc75391174)

[Сложные SELECT запросы 46](#_Toc75391175)

[Создание представления VIEW 47](#_Toc75391176)

[Создание хранимой процедуры 48](#_Toc75391177)

[ПРИЛОЖЕНИЕ В – Исходный текст программы 50](#_Toc75391178)

ПОСТАНОВКА ЗАДАЧИ

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

ИПК должен обеспечивать работу с данными о продаваемых товарах, о поставщиках товаров, о товарах, находящихся на складе, о договорах и заказах на поставку товара, о продаже товаров, ведение расчетов и выписку счетов за приобретенный товар, ведение расчетов проданного товара, документальный вывод договоров, счетов и счет-фактур.

Диалог с пользователем, работа с базами данных комплекса (ввод, корректировка, удаление и просмотр записей), выполнение ряда стандартных запросов к базе данных, выпуск стандартных отчетов должен осуществляться с помощью разработанного интерфейса.

2 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

## 2.1 Описание предметной области автоматизации

Предметной областью является магазин одежды, занимающийся розничной торговлей. Магазин работает с поставщиками одежды и клиентами-покупателями, являющимися физическими лицами. Магазин заключает с поставщиками долгосрочные договора, в рамках которых выполняются заказы на поставку конкретных товаров. После оформления заказа поставщик выставляет счет, который должен быть оплачен магазином. Отгрузка товаров поставщиком на склад магазина производится по накладной только после оплаты счета магазином.

Заказы на продажу товара клиенту при наличии необходимого количества товара на складе оплачиваются сразу. После оплаты заказа клиенту выписывается кассовый чек с указанием приобретенного товара.

## 2.2 Бизнес правила

1. Магазин может заключать несколько договоров с одним поставщиком, но с тем условием, что срок действия предыдущего договора закончился (договор переведен в архив).
2. Два поставщика не могут поставлять один и тот же товар.
3. Заказы на поставку товара выполняются только на основании действующего договора на поставку.
4. Товар можно включать в заказ только один раз.
5. Заказ на поставку считается выполненным, если он оплачен по счёту, выставленному поставщиком, и товар, входящий в данный заказ, отгружен на склад.
6. Счёт-фактуру от поставщика можно посмотреть и распечатать только для оплаченного заказа, а накладную только для выполненного заказа.
7. Количество товара проданного магазином не превышает количества данного товара на складе.
8. Налог НДС с каждого проданного товара составляет 20%.
9. Наценка на закупленный у поставщика товар составляет 45%. Однако директор магазина вправе устанавливать свое ценообразование.
10. После появления в заказе на поставку товара, ранее не закупаемого магазином, наименование этого товара появляется на складе и в каталоге продаваемых товаров.
11. Покупатели могут по желанию быть зарегистрированы в системе ИПК. Указывается полное имя клиента и его номер телефона. Клиенту выдается скидочная карта, которая имеет уникальный номер.
12. При покупке товара клиент может воспользоваться скидочной картой, если он зарегистрирован в базе ИПК. Скидка рассчитывается в зависимости от суммы покупок совершенных ранее и не может составлять более 60% от текущей покупки.

## 2.3 Состав сущностей и атрибутов предметной области

В договоре на поставку указывается: номер договора, дата заключения и дата окончания действия договора, условия поставки товара и оплаты, реквизиты и подписи сторон.

В заказе на поставку указывается: номер заказа, наименование товара и его количество, номер договора, поставщик, дата заказа, отметка об отгрузке, отметка об оплате.

В счете об оплате заказа указывается: номер счета, реквизиты поставщика и покупателя, сумма без НДС, НДС, сумма с НДС, дата, отметка об оплате.

В данных о поставщиках указывается: ИНН, название, адрес, директор, главный бухгалтер, телефоны, факсы, поставляемая продукция, банковские реквизиты.

В данные о товаре входят: категория товара, наименование, артикул, цена и его количество.

В чеке указывается: номер чека, название и ИНН организации (магазина), дата и время пробития, наименования товаров, их количество, цена, полная сумма без скидки, со скидкой и номер карты клиента, в случае если покупка была проведена по скидочной карте.

## 2.4 Постановка задачи автоматизированной обработки

Разрабатывается ИПК для работы магазина одежды по закупкам и продажам товаров по каталогу. ИПК должен обеспечивать работу с данными о продаваемых товарах, о поставщиках товаров, о товарах, находящихся на складе, о договорах и заказах на поставку товара, ведение счетов проданного товара, документальный вывод договоров, счетов и счет-фактур.

**Требуемые запросы к БД**

* Просмотр данных о поставщике (реквизиты, каталог поставляемых товаров);
* Поставщики по типу поставляемого товара;
* Данные об оплате счетов на поставку товара;
* Просмотр действительных и архивных договоров по поставкам;
* О договорах с истекающим сроком действия;
* Перечень товаров на складе;
* Список клиентов, участвующих в скидочной системе;
* О продажах за период (номенклатура, количество, стоимость)
* О поставках за период (номенклатура, количество, стоимость)

**Выпускаемые отчеты**

Оформленные в документальном виде договоры, чеки, накладные и счета-фактуры, а также типовые запросы.

3 ИНФОЛОГИЧЕСКОЕ МОДЕЛИРОВАНИЕ

Первый этап процесса проектирования базы данных определяется как этап концептуального или инфологического проектирования БД. Он заключается в создании концептуальной инфологической модели данных анализируемой ПО на основе информации, записанной в спецификациях требований пользователей. Инфологическое проектирование БД абсолютно не зависит от подробностей ее физической реализации.

Основные цели инфологического моделирования данных – углубить понимание семантики данных и упростить процедуры анализа предъявляемых к ним требований [1].

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

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

## 3.1 Спецификация сущностей

**Типы сущностей:**

1. Магазин.

2. Поставщик.

3. Договор.

4. Поставляемый товар.

5. Заказанный товар на поставку.

6. Заказ на поставку.

7. Счет.

8. Склад.

9. Каталог

10. Покупаемый товар.

11. Заказ на продажу.

12. Чек.

13. Клиент.

**Описание сущностей:**

1. Сущность Магазин.

Идентификатор – Код магазина.

Описательные атрибуты: Название, Адрес, Телефон, Факс, Директор, Главный бухгалтер, Банковские реквизиты, ИНН.

2. Сущность Поставщик.

Идентификатор – Код поставщика.

Описательные атрибуты: Название, Адрес, Телефон, Факс, Директор, Главный бухгалтер, Банковские реквизиты, ИНН.

3. Сущность Договор.

Идентификатор – Номер договора.

Описательные атрибуты: Код поставщика, Код магазина, Дата заключения, Дата окончания действия, Текст договора.

4. Сущность Поставляемый товар.

Идентификатор – Код товара.

Описательные атрибуты: Категория, Наименование, Цена поставщика, Код поставщика.

5. Сущность Счет.

Идентификатор – Номер счета.

Описательные атрибуты: Сумма без НДС, Сумма с НДС, Дата оплаты, Номер заказа.

6. Сущность Склад.

Идентификатор – Код товара.

Описательные атрибуты – Количество.

7. Сущность Заказ на поставку.

Идентификатор – Номер заказа.

Описательные атрибуты: Номер договора, Дата оформления заказа, Дата выполнения заказа, Отметка об оплате, Отметка об отгрузке.

8. Сущность Заказанный товар на поставку.

Идентификатор – Номер заказа, Код товара.

Описательные атрибуты: Количество, Стоимость.

9. Сущность Каталог.

Идентификатор – Код товара.

Описательные атрибуты: Цена.

10. Сущность Покупаемый товар.

Идентификатор – Код товара, Номер заказа.

Описательные атрибуты: Количество, Стоимость.

11. Сущность Заказ на продажу.

Идентификатор – Номер заказа.

Описательные атрибуты: Код клиента.

12. Сущность Чек.

Идентификатор – Номер чека.

Описательные атрибуты: Номер заказа, Дата и время покупки, Сумма без скидки, Сумма со скидкой.

13. Сущность Клиент.

Идентификатор – Код клиента.

Описательные атрибуты: ФИО, Телефон, Общая сумма покупок, Скидка, Номер карты.

## 3.2 Спецификация атрибутов

Спецификация атрибутов сущностей с указанием их наименований, типов, размеров и примерами значений из доменов представлена в таблице 1.

Таблица 1 - Спецификация атрибутов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Сущность** | **Наименование** | **Тип** | **Размер** | **Пример значения из домена** |
| Магазин | Код магазина | Алфавитноцифровой | 6 символов | МАГ-88 |
| Название | Алфавитноцифровой | 30 символов | «Мир одежды» |

Продолжение таблицы 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Адрес | Алфавитноцифровой | 50 символов | 198001, г. Новосибирск, ул. Высоцкого, д.10 |
| Телефон | Алфавитноцифровой | 18 символов | (812)777-77-89 |
| Факс | Алфавитноцифровой | 18 символов | (812)777-77-89 |
| Директор | Алфавитный | 30 символов | Иванов Андрей Артемович |
| Главный бухгалтер | Алфавитный | 30 символов | Смирнова Надежда Сергеевна |
| Банковские реквизиты | Алфавитноцифровой | 70 символов | Р/с 407028100555308 в Промышленном банке Санкт-Петербурга |
| ИНН | Цифровой | 10 символов | 8939305638 |
| Поставщик | Код поставщика | Алфавитноцифровой | 6 символов | ПОС-99 |
| Название | Алфавитноцифровой | 20 символов | «Фабрика Пальто» |
| Адрес | Алфавитноцифровой | 50 символов | - |
| Телефон | Алфавитноцифровой | 18 символов | - |
| Факс | Алфавитноцифровой | 18 символов | - |
| Директор | Алфавитный | 30 символов | - |
| Главный бухгалтер | Алфавитный | 30 символов | - |
| Банковские реквизиты | Алфавитноцифровой | 70 символов | - |
| ИНН | Цифровой | 10 символов | - |
| Договор | Номер договора | Алфавитноцифровой | 8 символов | ДОГ-8890 |
| Текст | Алфавитноцифровой | Длина переменная | - |
| Дата заключения | Дата | - | - |

Продолжение таблицы 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Дата окончания действия | Дата | - | - |
| Код поставщика | Алфавитноцифровой | 6 символов | ПОС-99 |
| Код магазина | Алфавитноцифровой | 6 символов | МАГ-88 |
| Заказ на поставку | Номер заказа | Алфавитноцифровой | 8 символов | ЗАК-999 |
| Номер договора | Алфавитноцифровой | 8 символов | ДОГ-8890 |
| Дата оформления заказа | Дата | - | - |
| Дата выполнения заказа | Дата | - | - |
| Отметка об оплате | Логический | - | - |
| Отметка об отгрузке | Логический | - | - |
| Поставляемый товар | Код товара | Алфавитноцифровой | 10 символов | ТОВ-12578 |
| Категория | Алфавитный | 20 символов | Куртка |
| Наименование | Алфавитноцифровой | 50 символов | Джинсовая куртка, жен., XS |
| Цена поставщика (за единицу) | Числовой | - | - |
| Код поставщика | Алфавитноцифровой | 6 символов | ПОС-99 |
| Заказанный товар на поставку | Код товара | Алфавитноцифровой | 10 символов | ТОВ-12578 |
| Номер заказа | Алфавитноцифровой | 8 символов | - |
| Количество | Числовой | 999999.99 | - |
| Стоимость | Числовой | 999999.99 | - |
| Счет | Номер счета | Алфавитноцифровой | 8 символов | СЧ-88995 |
| Сумма без НДС | Числовой | 999999.99 | - |
| Сумма с НДС | Числовой | 999999.99 | - |
| Дата оплаты | Дата | - | - |
| Номер заказа | Алфавитноцифровой | 8 символов | ЗАК-999 |

Продолжение таблицы 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Склад | Код товара | Алфавитноцифровой | 10 символов | ТОВ-12578 |
| Количество | Числовой | 9999.99 | - |
| Покупаемый товар | Код товара | Алфавитноцифровой | 10 символов | ТОВ-12578 |
| Номер заказа | Алфавитноцифровой | 9 символов | Зак-12578 |
| Количество | Числовой | 999999.99 | - |
| Стоимость | Числовой | 999999.99 | - |
| Чек | Номер чека | Алфавитноцифровой | 10 символов | ЧЕК-12578 |
| Номер заказа | Алфавитноцифровой | 9 символов | Зак-12578 |
| Дата и время покупки | Время | - | - |
| Сумма без скидки | Числовой | 999999.99 | - |
| Сумма со скидкой | Числовой | 999999.99 | - |
| Каталог | Код товара | Алфавитноцифровой | 10 символов | ТОВ-12578 |
| Цена | Числовой | 999999.99 | - |
| Заказ на продажу | Номер заказа | Алфавитноцифровой | 9 символов | Зак-12578 |
| Код клиента | Алфавитноцифровой | 6 символов | КЛ-148 |
| Клиент | Код клиента | Алфавитноцифровой | 6 символов | КЛ-148 |
| Номер карты клиента | Цифровой | 10 символов | 1278504789 |
| ФИО | Алфавитный | 30 символов | - |
| Телефон | Алфавитноцифровой | 18 символов | (812)777-77-89 |
| Общая сумма покупок | Числовой | 999999.99 | - |
| Скидка | Числовой | 999999.99 | - |

## 3.3 Спецификация связей

Связи выступают в модели в качестве средства, с помощью которого представляются отношения между сущностями, имеющими место в ПО. Тип связи рассматривается между типами сущностей.

Различают связи типа “один к одному” (1:1), “один ко многим” (1:М), “многие ко многим” (M:N).

* 1:1 - Каждому экземпляру сущности А соответствует один и только один экземпляр сущности В и наоборот. Идентификация экземпляров сущностей уникальна в обоих направлениях для отображения 1:1, т.е. один экземпляр сущности от которого направлена связь, идентифицирует один и только один экземпляр другой сущности и наоборот.
* 1:М - Одному экземпляру А соответствует несколько экземпляров В. Однако каждому экземпляру В соответствует только один экземпляр А, т.е. идентификация экземпляров при связи 1:М уникальна только в направлении от В к А.
* M:N - Каждому экземпляру А соответствует несколько экземпляров В и наоборот.

Спецификация связей для разрабатываемой модели представлена в таблице 2.

Типы связей: *Заключает, Оформляется, Поставляет, Выбран из, Оплачивается, Поступает, Забирается, Включает, Оформляет.*

Таблица 2 - Спецификация связей

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Номер** | **Связь** | **Тип** | **От Сущности** | **К Сущности** | **По Атрибуту** |
| 1 | Заключает | 1:M | Магазин | Договор | Код магазина |
| 2 | Заключает | 1:M | Поставщик | Договор | Код поставщика |
| 3 | Оформляется | 1:M | Договор | Заказ на поставку | Номер договора |
| 4 | Поставляет | 1:M | Поставщик | Поставляемый товар | Код поставщика |
| 5 | Выбран из | 1:M | Поставляемый товар | Заказанный товар на поставку | Код товара |
| 6 | Оплачивается | 1:1 | Заказ на поставку | Счет | Номер заказа |
| 7 | Поступает | 1:M | Склад | Заказанный товар на поставку | Код товара |
| 8 | Включает | 1:M | Заказ на поставку | Заказанный товар на поставку | Номер заказа |
| 9 | Выбран из | 1:M | Каталог | Покупаемый товар | Код товара |
| 10 | Включает | 1:M | Заказ на продажу | Покупаемый товар | Номер заказа |
| 11 | Оплачивается | 1:1 | Заказ на продажу | Чек | Номер заказа |
| 12 | Оформляет | 1:M | Клиент | Заказ на продажу | Код клиента |
| 13 | Забирается | 1: M | Склад | Покупаемый товар | Код товара |

## 3.4 Ссылочная целостность по связям

Правила по ссылочной целостности данных базы действуют следующим образом.

Правило по модификации идентификатора экземпляра родительской сущности:

* **каскадное** – определяет каскадное изменение атрибутов связи экземпляров дочерней сущности при изменении идентификатора экземпляра родительской сущности;
* **проверка** – определяет невозможность изменения идентификатора экземпляра родительской сущности в случае наличия хотя бы одного экземпляра подчиненной дочерней сущности.

Правило по удалению родительской сущности:

* **каскадное** – определяет каскадное удаление экземпляров подчиненных дочерних сущностей при удалении экземпляра родительской сущности;
* **проверка** – определяет невозможность удаления экземпляра родительской сущности в случае наличия хотя бы одного экземпляра подчиненной дочерней сущности.

Правило по включению дочерних сущностей:

* **проверка** – определяет невозможность включения экземпляра дочерней сущности с атрибутом связи, значение которого не совпадает ни с одним значением идентификатора экземпляров родительской сущности.

Ограничения ссылочной целостности по связям для разрабатываемой модели представлены в таблице 3.

Таблица 3 - Ссылочная целостность по связям

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Номер** | **Связь** | **Тип** | **Целостность связи** | | |
| **Правило по модификации** | **Правило по удалению** | **Правило по включению** |
| 1 | Заключает | 1:M | Каскадное | Проверка | Проверка |
| 2 | Заключает | 1:M | Каскадное | Проверка | Проверка |
| 3 | Оформляется по | 1:M | Каскадное | Проверка | Проверка |
| 4 | Поставляет | 1:M | Каскадное | Каскадное | Проверка |
| 5 | Выбран из | 1:M | Каскадное | Проверка | Проверка |
| 6 | Оплачивается по | 1:1 | Каскадное | Каскадное | Проверка |
| 7 | Поступает | 1:M | Каскадное | Проверка | Проверка |
| 8 | Включает | 1:M | Каскадное | Каскадное | Проверка |
| 9 | Выбран из | 1:M | Каскадное | Проверка | Проверка |
| 10 | Включает | 1:M | Каскадное | Каскадное | Проверка |
| 11 | Оплачивается по | 1:1 | Каскадное | Каскадное | Проверка |
| 12 | Оформляет | 1:M | Каскадное | Проверка | Проверка |
| 13 | Забирается | 1: M | Каскадное | Проверка | Проверка |

## 3.5 Инфологическая схема

На основании описания предметной области были выделены основные сущности, которые имеют набор атрибутов. На инфологической схеме, которая представлена на рисунке 1:

* сущности отображаются прямоугольниками, где указывается тип сущности в единственном числе в именительном падеже;
* ключ сущности (идентифицирующий атрибут) отображается овалом, соединённым ненаправленным ребром с сущностью, и подчеркивается;
* связи отображаются в виде ромбов, соединяющихся с советующими типами сущностей направленными ребрами. Вид стрелки определяется отношением между сущностями (двойная стрелка – «многие», одинарная стрелка – «один»).

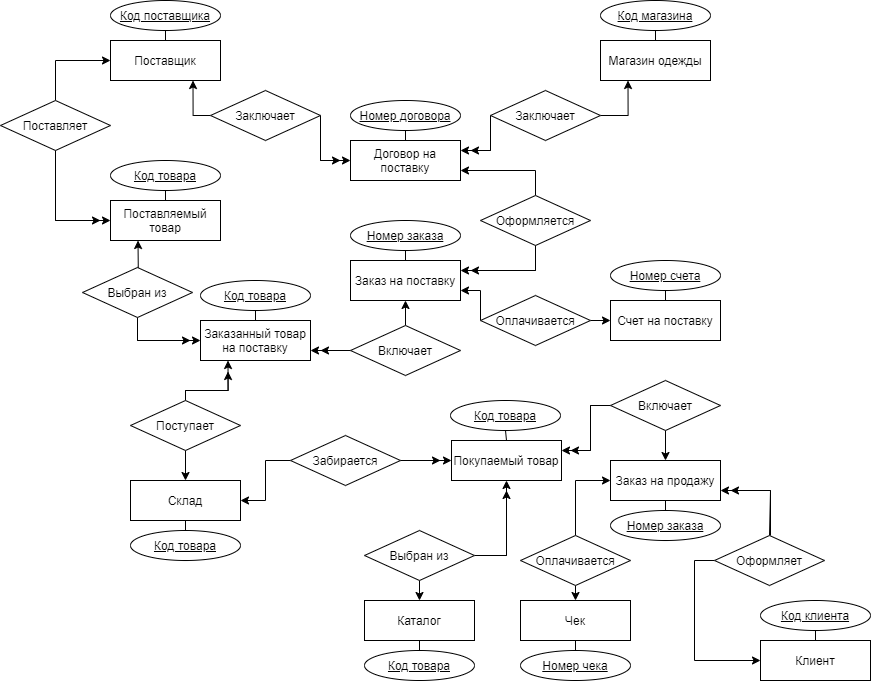


Рисунок 1 - Инфологическая схема

4 ДАТАЛОГИЧЕСКОЕ МОДЕЛИРОВАНИЕ

Разработка даталогической схемы БД представляет собой процесс создания модели ПО с учетом выбранной модели даталогического уровня организации данных, но независимо от физических аспектов реализации БД. Этот этап называется также логическим проектированием БД. Цель его состоит в создании логической модели данных для выбранной ПО. При этом логическая модель учитывает особенности выбранной модели организации данных в целевой СУБД, которая может быть либо реляционной, либо иерархической, либо сетевой, либо объектно-ориентированной моделью.

В качестве СУБД была выбрана PostgresSQL — свободная объектно-реляционная система управления базами данных [2].

Выбранный подход логического проектирования БД основан на концептуальной модели данных, создаваемой на этапе инфологического моделирования. После предварительного упрощения эта модель преобразуется в реляционную с использованием определенных правил преобразования. Однако данный процесс не гарантирует получение нормализованной реляционной модели. Поэтому к полученной таким путем реляционной схеме БД для проверки ее корректности применяется процесс нормализации. Нормализация гарантирует, что выведенные из существующей модели отношения не будут обладать избыточностью данных, способной вызвать аномалии обновления и удаления после их физической реализации [3].

## 4.1 Спецификация отношений

Для создания логической модели на основе концептуальной модели последняя была модифицирована. Сущности «Каталог» и «Склад» были преобразованы в единую сущность «Склад» для упрощения работы с данными.

Спецификация отношений разработанной даталогической модели системы представлена в таблице 4.

Таблица 4 - Спецификация отношений

|  |  |
| --- | --- |
| **Типы отношений** | **Описание** |
| stores | магазины |
| suppliers | поставщики |
| contracts | договоры |
| orders\_sup | заказы на поставку |
| catalog\_sup | поставляемые товары |
| ordered\_sup\_goods | заказанные товары на поставку |
| stock | склад |
| receipts\_sup | счета на поставку |
| ordered\_client\_goods | покупаемые товары |
| orders\_client | заказы на продажу |
| receipts\_client | чеки |
| clients | клиенты |

## 4.2 Спецификация полей

Таблица 5 – Спецификация полей

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Отношение** | **Поле** | | **Тип** | **Размер** | **Описание** |
| **Имя** | **Описание** |
| stores  (магазины) | code | Код магазина | text |  | Первичный ключ |
| name | Название | text |  |  |
| address | Адрес | text |  |
| phone | Телефон | text |  |
| fax | Факс | text |  |
| director | Директор | text |  |
| accountant | Главный бухгалтер | text |  |
| bank | Банковские реквизиты | text |  |
| inn | ИНН | numeric | 10, 0 |
| suppliers (поставщики) | code | Код поставщика | text |  | Первичный ключ |
| name | Название | text |  |  |
| address | Адрес | text |  |
| phone | Телефон | text |  |
| fax | Факс | text |  |
| director | Директор | text |  |
| accountant | Главный бухгалтер | text |  |
| bank | Банковские реквизиты | text |  |
| inn | ИНН | numeric | 10, 0 |
| contracts  (договоры) | num | Номер договора | character | 10 | Первичный ключ |
| text | Краткий текст | text |  |  |
| conclusion\_date | Дата заключения | date |  |
| completion\_date | Дата окончания действия | date |  |
| supplier\_code | Код поставщика | text |  | Вторичный ключ |

Продолжение таблицы 5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | store\_code | Код магазина | text |  | Вторичный ключ |
| orders\_sup  (заказы на поставку) | num | Номер заказа | text |  | Первичный ключ |
| contract\_num | Номер договора | text |  | Вторичный ключ |
| registration\_date | Дата оформления заказа | date |  |  |
| execution\_date | Дата выполнения заказа | date |  |
| paid | Отметка об оплате | boolean |  |
| sent | Отметка об отгрузке | boolean |  |
| catalog\_sup  (каталог поставщика – поставляемый товар) | code | Код товара | text |  | Первичный ключ |
| name | Категория | text |  |  |
| description | Полное наименование | text |  |
| price | Цена поставщика (за единицу) | money |  |
| supplier\_code | Код поставщика | text |  | Вторичный ключ |
| ordered\_sup\_goods (заказанные товары на поставку) | n | Номер (искусственный ключ) | integer |  | Первичный ключ |
| good\_code | Код товара | text |  | Вторичный ключ |
| order\_sup­­­\_num | Номер заказа | text |  | Вторичный ключ |
| number | Количество | integer |  |  |
| cost | Стоимость | money |  |
| stock  (склад) | code | Код товара | text |  | Первичный ключ, вторичный ключ |
| number | Количество | integer |  |  |
| price | Цена магазина (за единицу) | money |  |
| receipts\_sup  (счета на поставку) | num | Номер счета | text |  | Первичный ключ |
| price | Сумма без НДС | money |  |  |
| price\_nds | Сумма с НДС | money |  |
| date | Дата оплаты | date |  |
| order\_sup\_num | Номер заказа | text |  | Вторичный ключ |

Продолжение таблицы 5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ordered\_client\_goods  (покупаемые товары) | n | Номер (искусственный ключ) | integer |  | Первичный ключ |
| order\_client\_num | Номер заказа | text |  | Вторичный ключ |
| good\_code | Код товара | text |  | Вторичный ключ |
| number | Количество | integer |  |  |
| cost | Стоимость | money |  |
| orders\_client  (заказы на продажу) | num | Номер заказа | text |  | Первичный ключ |
| client\_code | Код клиента | text |  | Вторичный ключ |
| receipts\_client  (чеки) | num | Номер чека | text |  | Первичный ключ |
| order\_client\_num | Номер заказа | text |  | Вторичный ключ |
| time | Дата и время покупки | timestamp |  |  |
| price | Сумма без скидки | money |  |
| discount\_price | Сумма со скидкой | money |  |
| clients  (клиенты) | code | Код клиента | text |  | Первичный ключ |
| name | ФИО | text |  |  |
| phone | Телефон | text |  |
| purchase\_amount | Общая сумма покупок | money |  |
| discount | Скидка | numeric | 4,2 |
| card | Номер карты | integer |  |

## 4.3 Спецификация связей

Атрибут или группа атрибутов, которая в рассматриваемой таблице не является ключом, а в другой таблице ключом является, называется внешним ключом (вторичным ключом). Связь “ключ” - “внешний ключ” определяет наличие связи 1:М между записями соответствующих отношений.

Первичный ключ является специальным типом индекса. Индексы в свою очередь представляют собой наиболее эффективное средство, которое позволяет значительно ускорить поиск данных в таблицах по сравнению с таблицами, не содержащими индексов.

Таблица 6 - Спецификация связей

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **№** | **Связь** | | | | | | **Тип** |
| **от** | | | **к** | | |
| **отношение** | **поле** | **индекс** | **отношение** | **поле** | **индекс** |
| 1 | stores | code | pk\_st\_code | contracts | store\_code | fk\_st\_code | 1:М |
| 2 | suppliers | code | pk\_sup\_code | contracts | supplier\_code | fk\_sup\_code1 | 1:М |
| 3 | contracts | num | pk\_contr\_num | orders\_sup | contract\_num | fk\_contr\_num | 1:М |
| 4 | suppliers | code | pk\_sup\_code | catalog\_sup | supplier\_code | fk\_sup\_code2 | 1:М |
| 5 | catalog\_sup | code | pk\_cat\_sup\_code | ordered\_sup\_goods | good\_code | fk\_g\_code1 | 1:М |
| 6 | orders\_sup | num | pk\_ord\_num | ordered\_sup\_goods | order\_sup\_num | fk\_order\_s\_num | 1:М |
| 7 | orders\_sup | num | pk\_ord\_num | receipts\_sup | order\_sup\_num | fk\_order\_s\_num1 | 1:1 |
| 8 | stock | code | pk\_stock\_g | ordered\_client\_goods | good\_code | fk\_g\_code2 | 1:М |
| 9 | catalog\_sup | code | pk\_cat\_sup\_code | stock | code | fk\_good | 1:М |
| 10 | orders\_client | num | pk\_ord\_num2 | receipts\_client | order\_client\_num | fk\_order\_c\_num1 | 1:1 |
| 11 | clients | code | pk\_cl\_code | orders\_client | client\_code | fk\_c\_code | 1:М |

Связь 1:1 между таблицами заказов и счетов (чеков) обеспечивается за счет наложения на вторичный ключ ограничения уникальности (Unique constraint).

## 4.4 Спецификация ключей

Таблица 7 – Спецификация ключей

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Отношение** | **Первичный ключ** | | | | **Вторичный ключ** | | | |
| **Поле** | | **Индекс** | | **Поле** | | **Индекс** | |
| **Имя** | **Описание** | **Имя** | **Тип** | **Имя** | **Описание** | **Имя** | **Тип** |
| stores | code | Код магазина | pk\_st\_code | Primary | - | - | - | - |
| suppliers | code | Код поставщика | pk\_sup\_code | Primary | - | - | - | - |
| contracts | num | Номер договора | pk\_contr\_num | Primary | store\_code | Код магазина | fk\_st\_code | Foreign |
|  | | | | supplier\_code | Код поставщика | fk\_sup\_code1 | Foreign |
| orders\_sup | num | Номер заказа на поставку | pk\_ord\_num | Primary | contract\_num | Номер договора | fk\_contr\_num | Foreign |
| catalog\_sup | code | Код товара | pk\_cat\_sup\_code | Primary | supplier\_code | Код поставщика | fk\_sup\_code2 | Foreign |

Продолжение таблицы 7

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ordered\_sup\_goods | n | Номер | pk\_n2 | Primary | order\_sup\_num | Номер заказа на поставку | fk\_order\_s\_num | Foreign |
|  | | | | good\_code | Код товара | fk\_g\_code1 | Foreign |
| stock | code | Код товара | pk\_stock\_g | Primary | code | Код товара | fk\_good | Foreign |
| receipts\_sup | num | Номер счета | pk\_receipt\_sup | Primary | order\_sup\_num | Номер заказа на поставку | fk\_order\_s\_num1 | Foreign |
| ordered\_client\_goods | n | Номер | pk\_n1 | Primary | order\_client\_num | Номер заказа на продажу | fk\_order\_c\_num | Foreign |
|  | | | | good\_code | Код товара | fk\_g\_code2 | Foreign |
| orders\_client | num | Номер заказа на продажу | pk\_ord\_num2 | Primary | client\_code | Код клиента | fk\_c\_code | Foreign |
|  | | | |  |  |  |  |
| receipts\_client | num | Номер чека | pk\_receipt\_c | Primary | order\_client\_num | Номер заказа на продажу | fk\_order\_c\_num1 | Foreign |
| clients | code | Код клиента | pk\_cl\_code | Primary | - | - | - | - |

## 4.5 Список ключей и ссылочная целостность по связям

Для обеспечения ссылочной целостности по связям применяют следующие меры:

1. При изменении значения первичного ключа или ключа связи в родительской таблице:

* Cascade - каскадное изменение всех соответствующих значений в дочерней таблице.
* Restrict - не позволяет изменять значения полей первичного ключа или ключа связи в родительской таблице, если в дочерней таблице имеется хотя бы одна запись, содержащая ссылку на изменяемую запись.

2. При удалении записи в родительской таблице:

* Cascade - каскадное удаление всех записей из дочерней таблице, связанных с удаляемой записью.
* Restrict - не позволяет удалять запись в родительской таблице при наличии хотя бы одной подчиненной записи в дочерней таблице.

3. При добавлении новой записи в дочернюю таблицу или редактировании в ней существующей записи не позволяет вводить запись (Restrict), если значение индексного выражения дочерней таблицы не соответствует одной из записей в родительской таблице.

Таблица 8 – Список ключей и ссылочная целостность по связям

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Parent Table** | **Child Table** | **Update** | **Delete** | **Insert** | **Parent Tag** | **Child Tag** |
| stores | contracts | Cascade | Restrict | Restrict | code | store\_code |
| suppliers | contracts | Cascade | Restrict | Restrict | code | supplier\_code |
| contracts | orders\_sup | Cascade | Restrict | Restrict | num | contract\_num |
| suppliers | catalog\_sup | Cascade | Restrict | Restrict | code | supplier\_code |
| catalog\_sup | ordered\_sup\_goods | Cascade | Restrict | Restrict | code | good\_code |
| orders\_sup | ordered\_sup\_goods | Cascade | Restrict | Restrict | num | order\_sup\_num |
| orders\_sup | receipts\_sup | Cascade | Restrict | Restrict | num | order\_sup\_num |
| catalog\_sup | stock | Cascade | Restrict | Restrict | code | code |
| stock | ordered\_client\_goods | Cascade | Restrict | Ignore | code | good\_code |
| orders\_client | ordered\_client\_goods | Cascade | Restrict | Restrict | num | order\_client\_num |
| orders\_client | receipts\_client | Cascade | Restrict | Restrict | num | order\_client\_num |
| clients | orders\_client | Cascade | Restrict | Restrict | code | client\_code |

Вариант ограничения Ignore для вставки записей в таблицу «stock» был выбран по причине того, что вставка товаров на склад будет осуществляться автоматически посредством использования триггерной процедуры.

Вариант Restrict для удаления записи из родительской таблицы был выбран с целью сохранения сведений о произведенных покупках, продажах и договорах, так как в случае выбора иного варианта произойдет потеря информации.

## 4.6 Даталогическая схема

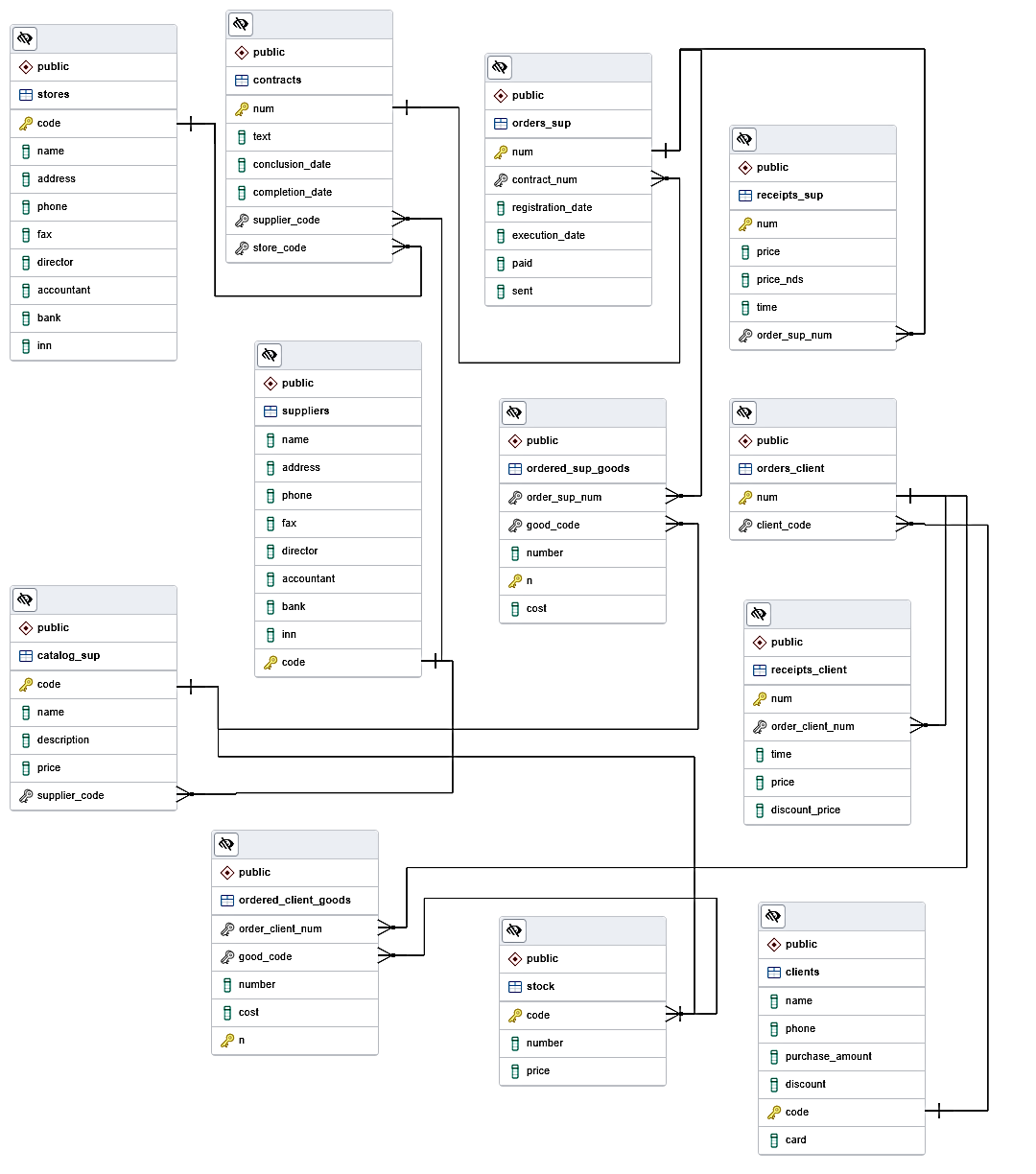
Полученная даталогическая схема представлена на рисунке 2. Желтым цветом обозначены первичные ключи отношений, серым – вторичные.

Рисунок 2 – Даталогическая схема

## 4.7 Триггерные процедуры

Вставка товара на склад осуществляется автоматически после того как заказ становится выполненным. Это осуществляется за счет механизма триггера, установленного на обновление поля sent в таблице orders\_sup.

CREATE TRIGGER AddToStock\_aft\_upd AFTER UPDATE OF sent ON orders\_sup FOR EACH ROW WHEN (new.sent = true) EXECUTE FUNCTION order\_sup\_AddToStock();

CREATE FUNCTION order\_sup\_AddToStock() RETURNS trigger

LANGUAGE plpgsql

AS $$

DECLARE

good RECORD;

curs CURSOR FOR SELECT good\_code, number, cost

FROM ordered\_sup\_goods where order\_sup\_num = NEW.num;

BEGIN

FOR good IN curs LOOP

IF EXISTS(SELECT FROM stock WHERE code = good.good\_code)

THEN

UPDATE stock SET number = number + good.number WHERE code = good.good\_code;

ELSE

NSERT INTO stock (code, number, price)

VALUES (good.good\_code, good.number, good.cost/good.number\*1.4);

END IF;

END LOOP;

RETURN NEW;

END;

Для вставки товара применяется технология курсора – создания временной таблицы для заказанных товаров в оперативной памяти для ее позаписной обработки. Количество заказанного товара позаписно прибавляется к количеству товара на складе, который имеет тот же код.

Те же механизмы реализованы для того, чтобы удалить купленный товар со склада. Полный SQL-скрипт создания БД можно найти в приложении А к данному курсовому проекту.

5 ПРОЕКТИРОВАНИЕ ИНТЕРФЕЙСА ПОЛЬЗОВАТЕЛЯ

Интерфейс пользователя был разработан с помощью свободной кроссплатформенной IDE для разработки интерфейсов на C++ - «Qt Creator» версии 5.15.

Qt обеспечивает поддержку баз данных SQL с помощью модуля Qt SQL. Основные классы для удобной работы с запросами и таблицами – это «QSqlQuery», предоставляющий средства для выполнения запросов, «QSqlTableModel», предоставляющий редактируемую модель данных для одной таблицы базы данных, и «QSqlQueryModel», предоставляющий доступную только для чтения модель данных для наборов результатов SQL запросов [4].

Рассмотрим работу с данными классами на примере создания заказа. Окно заказа содержит две таблицы. Верхняя таблица – таблица каталога выбранного поставщика. Нижняя – таблица с товарами, выбранными для покупки.

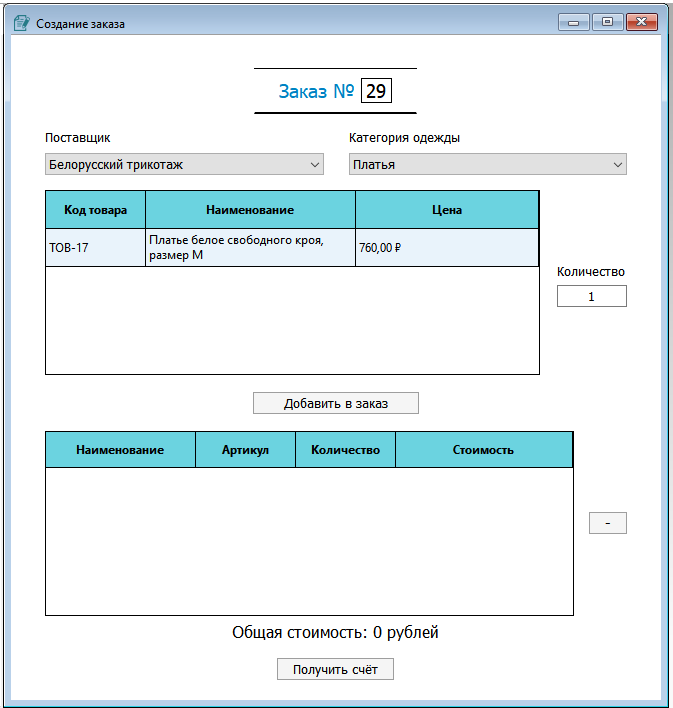


Рисунок 3 – Окно создания заказа

В первую очередь в выпадающий список нужно загрузить поставщиков, с которыми заключен действующий договор на поставку. А затем загрузить категории товаров, которые поставляет выбранный поставщик.

\_comboSupplierModel.setQuery ("SELECT DISTINCT ON (name) name, code FROM suppliers WHERE EXISTS (SELECT num FROM contracts WHERE completion\_date >= NOW() AND contracts.supplier\_code = suppliers.code)");

ui->suppliers->setModel(*&\_comboSupplierModel*);

\_supplierCode = "'" + \_comboSupplierModel.*index*(0, 1).data().toString() + "'";

\_comboSupplyCategoriesModel.setQuery("SELECT DISTINCT name FROM catalog\_sup WHERE supplier\_code = " + \_supplierCode);

ui->supplyCategories->setModel(*&\_comboSupplyCategoriesModel*);

\_comboSupplierModel и \_comboSupplyCategoriesModel являются объектами класса QSqlQueryModel. На выпадающие списки с поставщиками (ui->suppliers) и поставляемыми товарами (ui->supplyCategories) устанавливаются модели с вышеописанными запросами. Затем, необходимо загрузить каталог выбранного поставщика в таблицу.

\_sqlCatalogModel.*setTable*("catalog\_sup");

\_sqlCatalogModel.*setFilter*("supplier\_code = " + \_supplierCode);

\_sqlCatalogModel.*select*();

\_sqlCatalogModel.*setHeaderData*(0, Qt::Horizontal, "Код товара");

\_sqlCatalogModel.*setHeaderData*(2, Qt::Horizontal, "Наименование");

\_sqlCatalogModel.*setHeaderData*(3, Qt::Horizontal, "Цена");

\_sqlCatalogModel – это объект класса QSqlTableModel. Таким образом, мы устанавливаем таблицу каталога поставщика из базы данных в качестве модели для таблицы интерфейса. Модель осуществляет соединение с источником данных, предоставляя необходимый интерфейс таблице. Установка фильтра для QSqlTableModel – это применение условия «where» SQL запроса.

Фильтрация на уровне категории поставляемых товаров осуществляется на уровне интерфейса (без обращения к БД) с помощью QSortFilterProxyModel.

\_catalogFilterModel.*setSourceModel*(*&\_sqlCatalogModel*);

\_catalogFilterModel.SetCategoryFilter(ui->supplyCategories->currentText());

ui->supplyGoods->*setModel*(*&\_catalogFilterModel*);

Нижняя таблица – таблица выбранных товаров была создана с помощью своей разработанной модели GoodsInOrderModel, наследуемой от класса QAbstractTableModel.

При нажатии на кнопку «Добавить в заказ»:

const auto quantity = ui->quantity->text().toInt();

const auto supplyOrderIndex = ui->supplyGoods->currentIndex();

const auto supplyOrderName = \_catalogFilterModel.*index*(supplyOrderIndex.row(), 1).data().toString();

const auto goodPriceStr = \_catalogFilterModel.*index*(supplyOrderIndex.row(), 3).data().toString().simplified().remove(' ');

const auto totalGoodPrice = \_numExpression.match(goodPriceStr).captured().replace(',', '.').toDouble() \* quantity;

\_totalCost += totalGoodPrice;

ui->totalCost->setText("Общая стоимость: " + QString::number(\_totalCost, 'f', 2).append(" рублей"));

auto row = \_goodsInOrderModel.FindAndUpdateGoodInfo(supplyOrderName, quantity, totalGoodPrice);

if (row >= 0) { // Если товар уже добавлен, к нему просто прибавится количество и цена

ui->goodsInOrder->update(\_goodsInOrderModel.*index*(row, 2));

ui->goodsInOrder->update(\_goodsInOrderModel.*index*(row, 3));

ui->goodsInOrder->selectRow(row);

}

else { // Добавление товара в заказ

const auto goodCode = \_sqlCatalogModel.*index*(supplyOrderIndex.row(), 0).data().toString();

row = \_goodsInOrderModel.*rowCount*();

\_goodsInOrderModel.*insertRows*(row, 1);

\_goodsInOrderModel.*setData*(\_goodsInOrderModel.*index*(row, 0), supplyOrderName);

\_goodsInOrderModel.*setData*(\_goodsInOrderModel.*index*(row, 1), goodCode);

\_goodsInOrderModel.*setData*(\_goodsInOrderModel.*index*(row, 2), quantity);

\_goodsInOrderModel.*setData*(\_goodsInOrderModel.*index*(row, 3), totalGoodPrice);

ui->goodsInOrder->selectRow(\_goodsInOrderModel.*rowCount*() - 1);

}

После нажатия на кнопку «Получить счет» в приложении будет открыто новое окно «Счет», а также будет вызван ряд следующих подготовленных запросов:

QSqlQuery contractNumber("SELECT num FROM contracts WHERE supplier\_code = " + \_supplierCode);

contractNumber.next();

QSqlQuery addOrder;

addOrder.prepare("INSERT INTO orders\_sup (contract\_num, paid) "

"VALUES (:contract\_num, :paid) RETURNING num");

addOrder.bindValue(":contract\_num", contractNumber.value(0));

addOrder.bindValue(":paid", iw->IsPaid());

addOrder.exec();

addOrder.next();

QSqlQuery addOrderedSupGoods;

addOrderedSupGoods.prepare("INSERT INTO ordered\_sup\_goods (order\_sup\_num, good\_code, number, cost) VALUES (:order\_sup\_num, :good\_code, :number, :cost)");

addOrderedSupGoods.bindValue(":order\_sup\_num", addOrder.value(0));

for (int i = 0; i < \_goodsInOrderModel.*rowCount*(); i++) {

addOrderedSupGoods.bindValue(":good\_code", \_goodsInOrderModel.*index*(i, 1).data());

addOrderedSupGoods.bindValue(":number", \_goodsInOrderModel.*index*(i, 2).data());

addOrderedSupGoods.bindValue(":cost", \_goodsInOrderModel.*index*(i, 3).data().toDouble());

addOrderedSupGoods.exec();

}

QSqlQuery addSupReceipt;

addSupReceipt.prepare("INSERT INTO receipts\_sup (price, price\_nds, time, order\_sup\_num) VALUES (:price, :price\_nds, :time, :order\_sup\_num)");

addSupReceipt.bindValue(":price", iw->GetCost());

addSupReceipt.bindValue(":price\_nds", iw->GetCostVat());

addSupReceipt.bindValue(":time", iw->IsPaid() ? "NOW()" : nullptr);

addSupReceipt.bindValue(":order\_sup\_num", addOrder.value(0));

addSupReceipt.exec();

Интерфейс предоставляет пользователю возможность ввода, вывода и просмотра информации при работе с базой данных магазина одежды. Карта разработанного интерфейса представлена на рисунке 4.

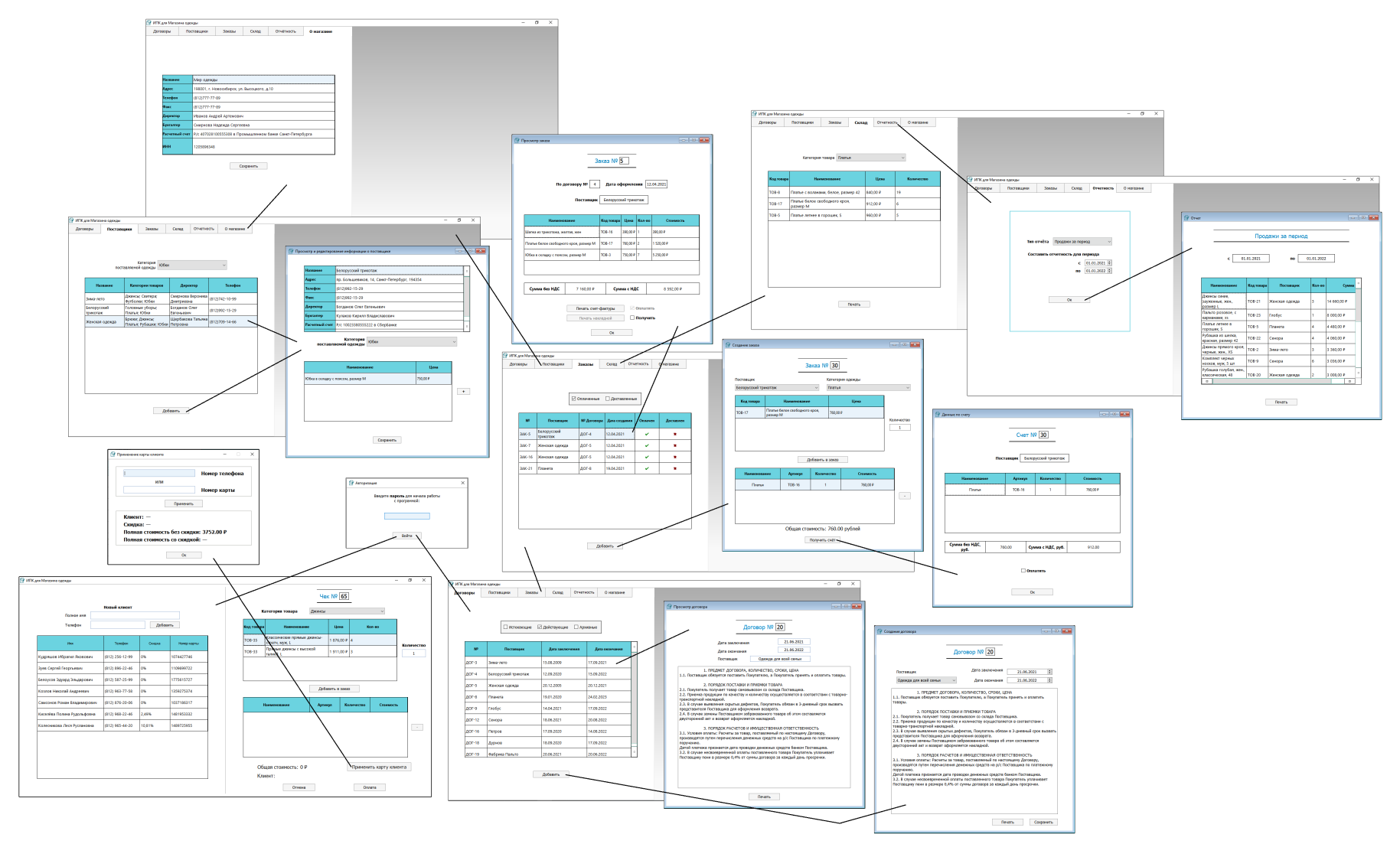


Рисунок 4 - Карта разработанного интерфейса

6 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Работа в программе начинается с авторизации пользователя, из которой с помощью ввода пароля можно попасть в меню директора или же в меню кассира. Окно авторизации представлено на рисунке 5.

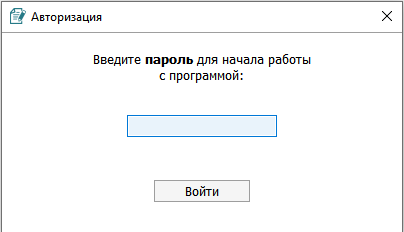


Рисунок 5 – Окно авторизации пользователя

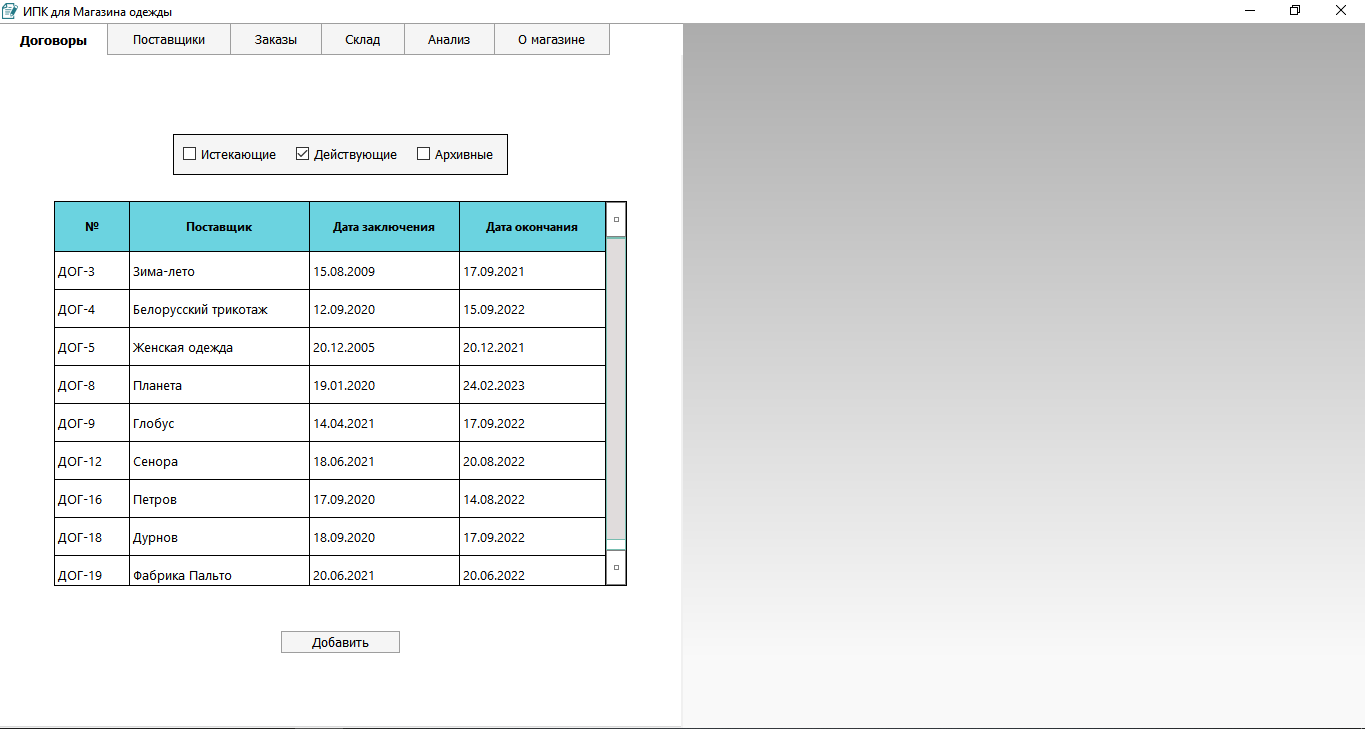
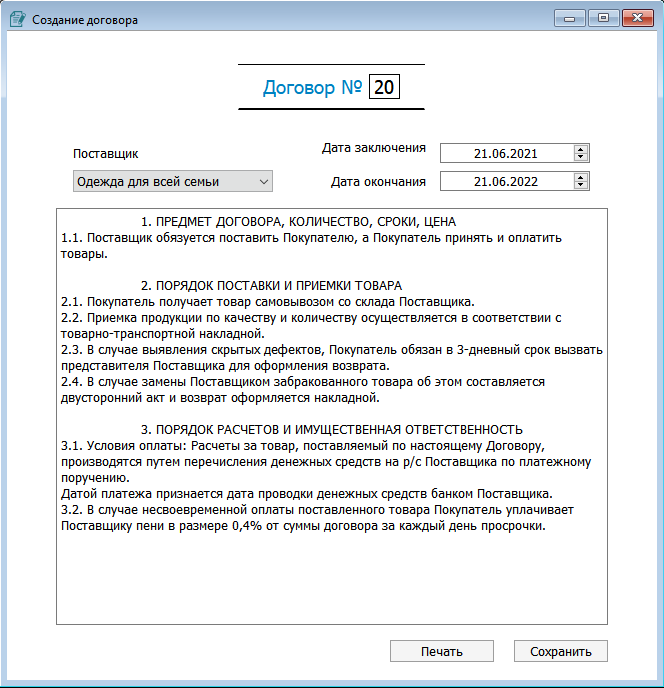
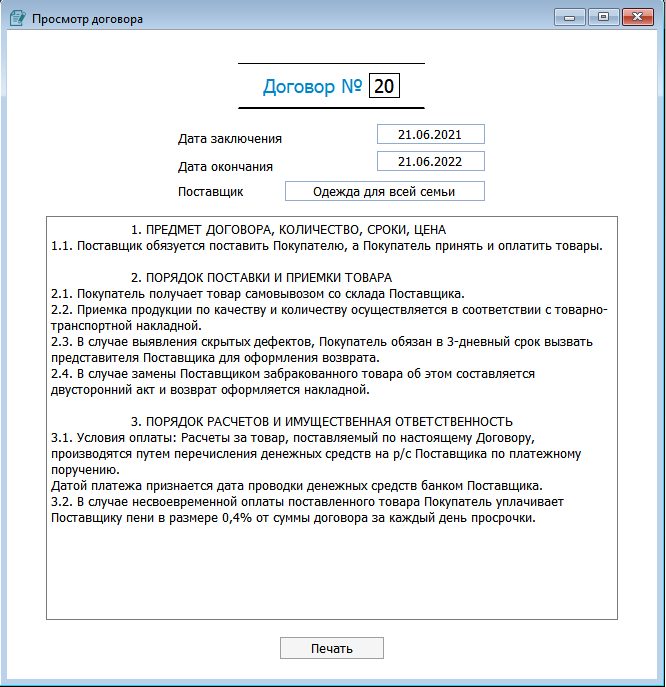
Введя пароль директора, пользователь попадает в главное меню директора. В верхней левой части окна расположены вкладки для навигации по основным сущностям. Первая открытая вкладка – «Договоры» представлена на рисунке 6.

Рисунок 6 – Вкладка «Договоры»

Над таблицей всех составленных договоров расположены параметры фильтрации. Двойным щелчком по строке в таблице можно открыть окно просмотра договора. А при нажатии на кнопку «Добавить» будет открыто окно создания договора. Скриншоты окон представлены на рисунке 7.

Рисунок 7 – Окна создания и просмотра договора



Нажав на кнопку «Печать» пользователь увидит документ, готовый к печати.

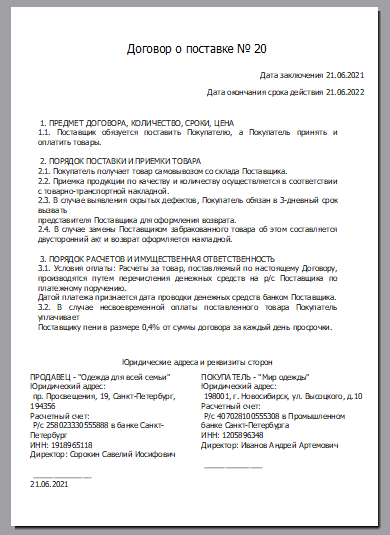
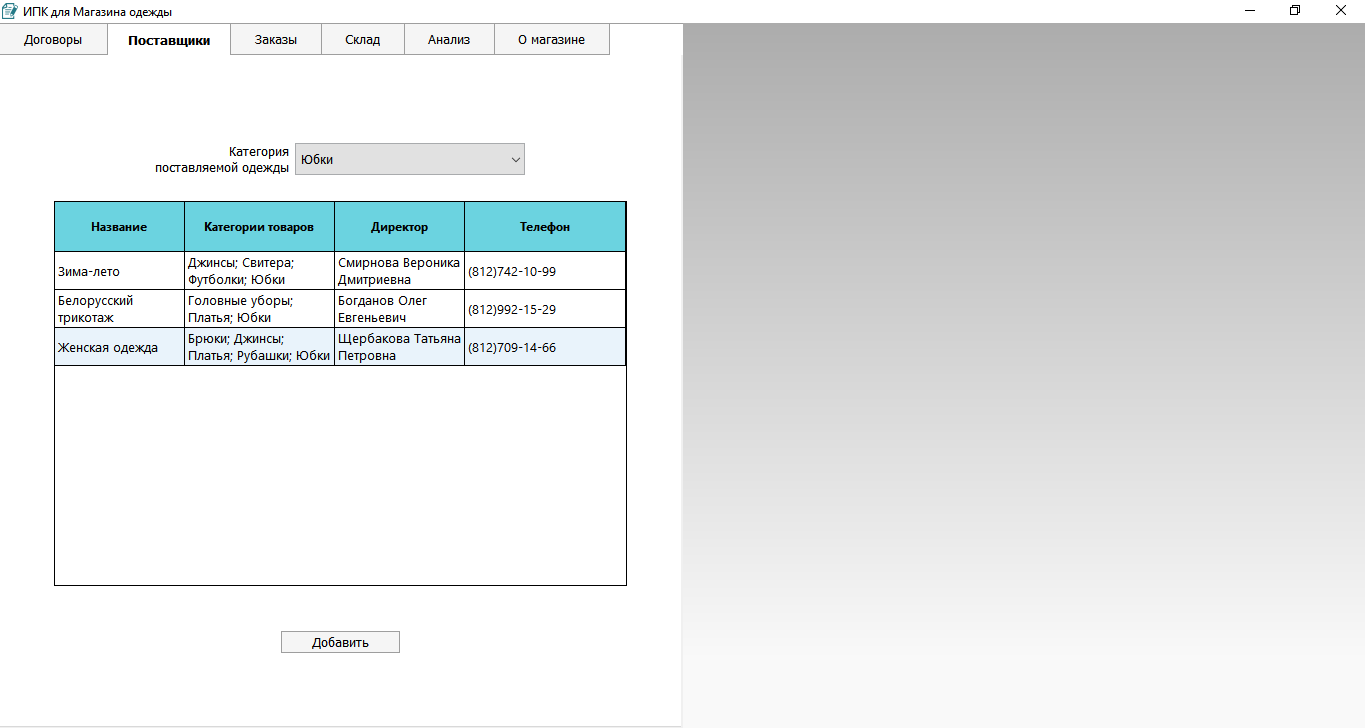


Рисунок 8 - Печать договора

 Открыв вкладку «Поставщики» в главном меню директора можно увидеть таблицу с поставщиками. Окно представлено на рисунке 9.

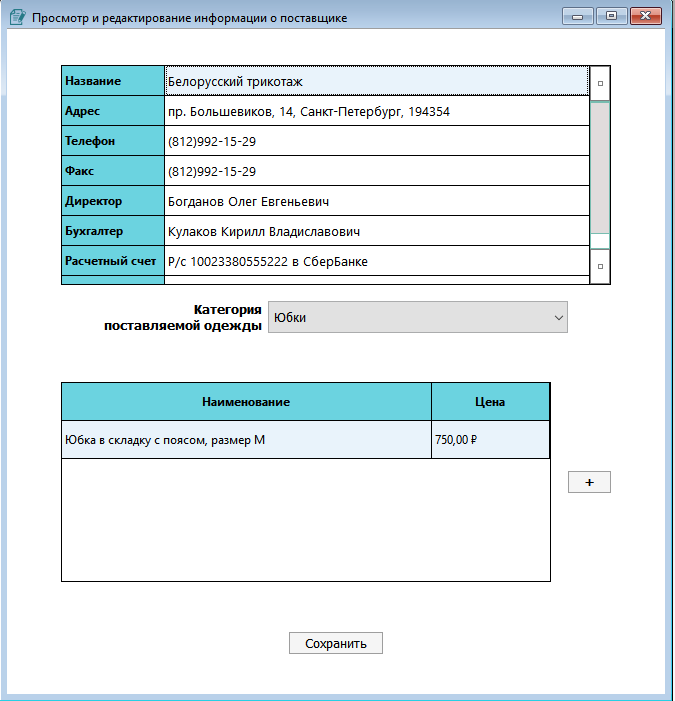
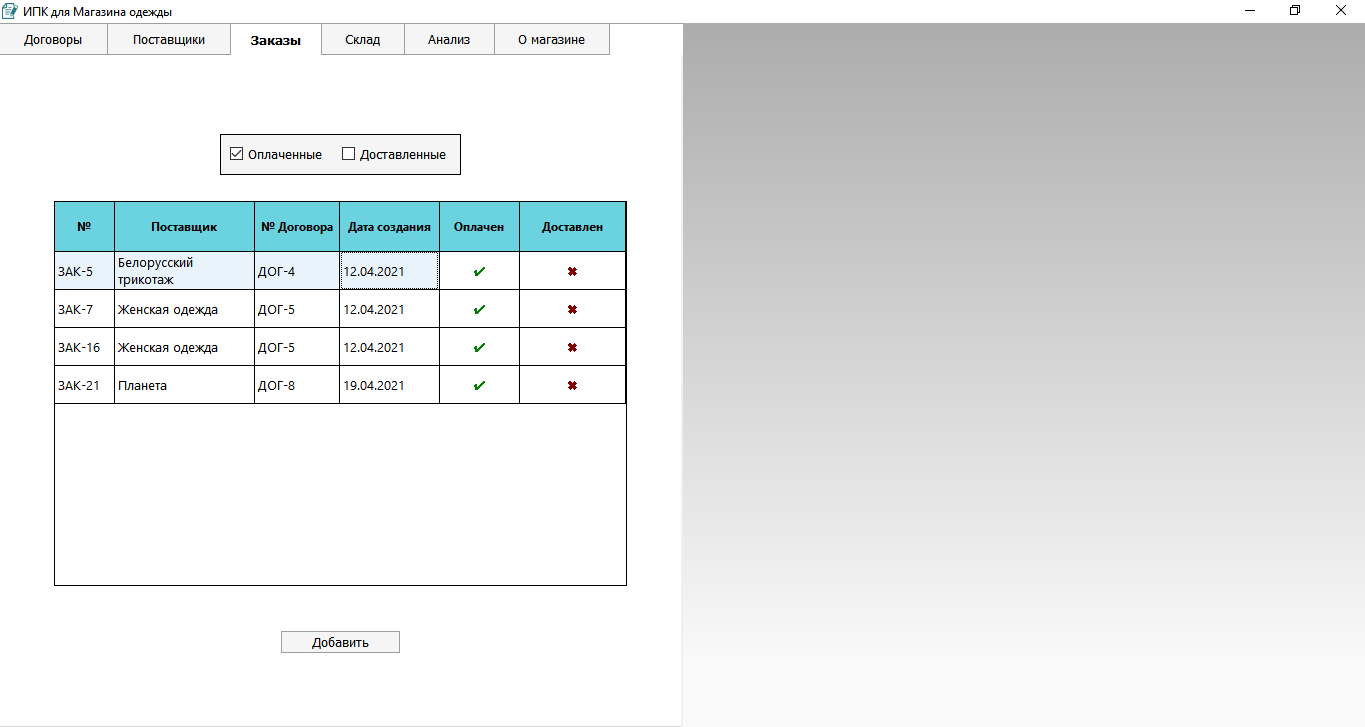
 Окно просмотра и редактирования поставщика, представленное на рисунке 10, можно открыть двойным щелчком по строке в таблице или же, нажав на кнопку «Добавить», тогда будет открыта новая пустая форма.

Рисунок 9 – Вкладка «Поставщики»

Рисунок 10 - Окно редактирования информации о поставщике

Верхняя таблица является редактируемой. Для заполнения каталога поставщика в первую очередь необходимо выбрать из выпадающего списка необходимую категорию одежды, затем нажать на кнопку «+» справа от таблицы каталога. В появившуюся пустую строку следует ввести требуемую информацию о товаре.

Открытая вкладка «Заказы» представлена на рисунке 11.

Создать новый заказ можно нажав на кнопку «Добавить». В открывшейся форме создания заказа следует выбрать желаемого поставщика, затем категорию поставляемого им товара. В таблице каталога поставщика загрузятся товары. Товар следует выбрать одним щелчком, затем ввести необходимое его количество в поле для ввода справа и нажать на кнопку «Добавить в заказ». Выбранный товар отобразится в таблице корзины заказа. Закончив выбор, следует нажать на кнопку «Получить счет». Будет открыто окно счета. Счет может быть оплачен сразу, нажатием на чекбокс «Оплатить».

Рисунок 11 – Вкладка «Заказы»

Окно создания заказа и окно счета представлены на рисунке 12.

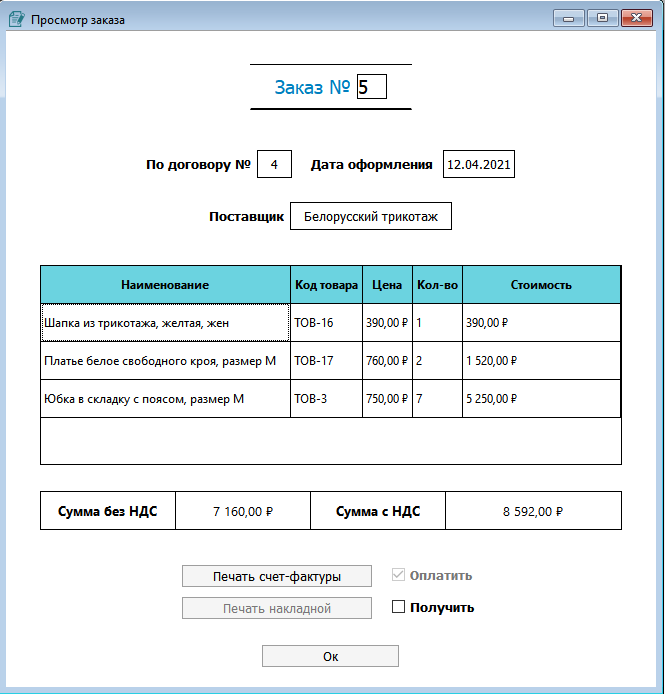
Для просмотра уже совершенного заказа следует щелкнуть по нему двойным щелчком в таблице заказов. Будет открыто окно, представленное на рисунке 13.

Рисунок 12 – Окна создания заказа и получения счета

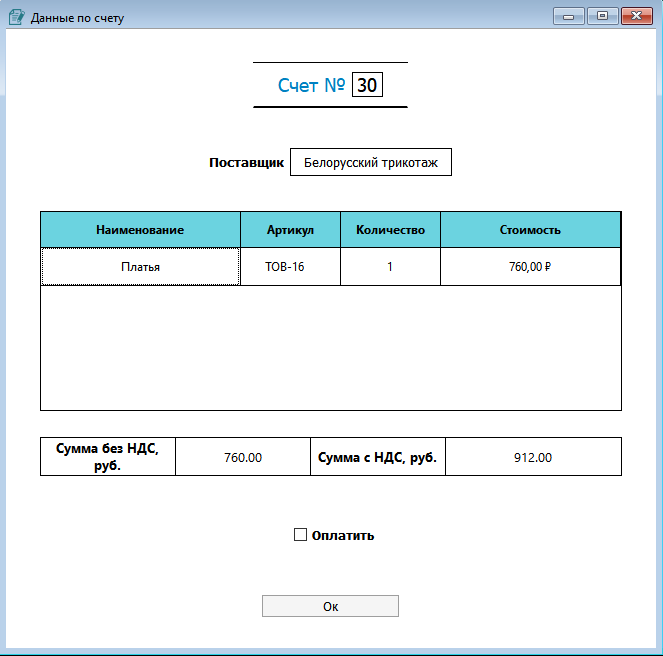
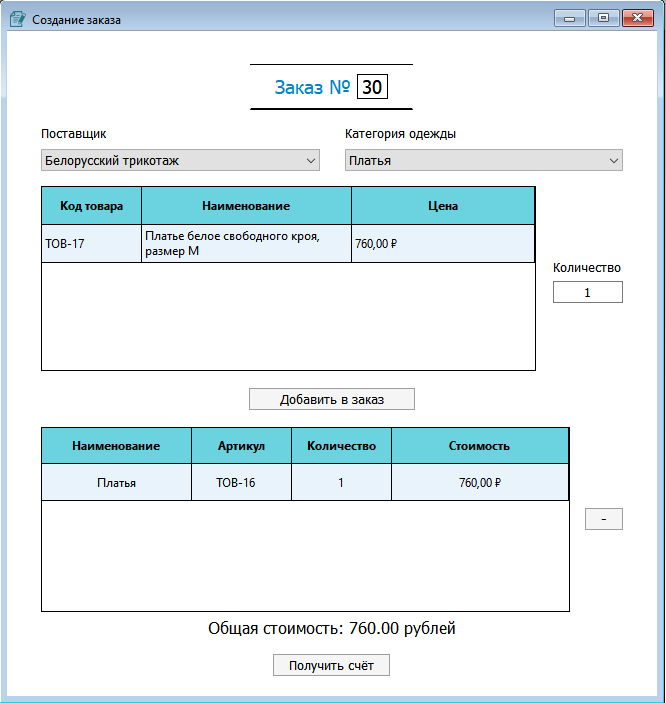


Рисунок 13 – Окно просмотра заказа

Для оплаченного заказа можно распечатать счет-фактуру, а для отправленного – накладную. Отображение печати для данных документов представлено на рисунке 14.

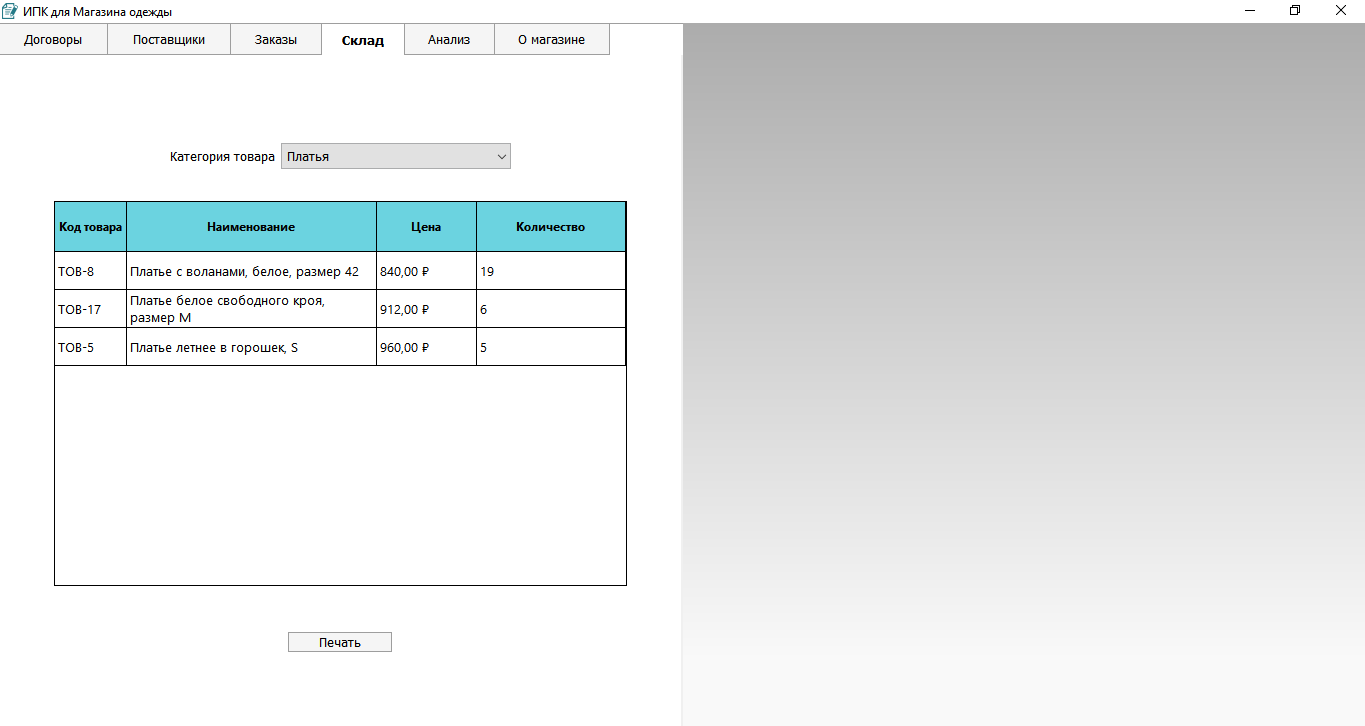
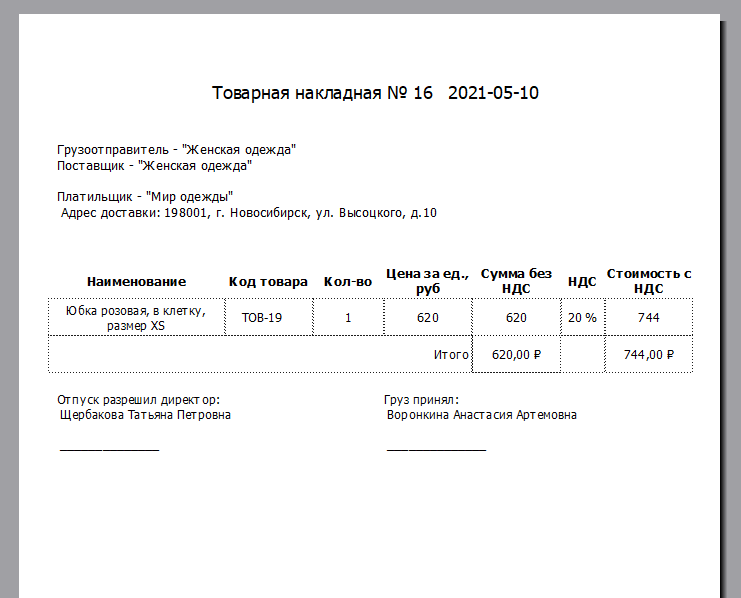
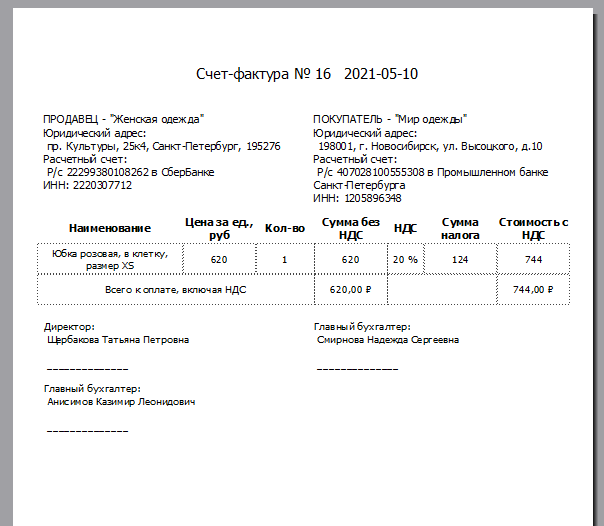
Открытая вкладка «Склад» представлена на рисунке 15. Перечень товаров на складе также может быть напечатан.

Рисунок 15 – Вкладка «Склад»

Рисунок 14 - Печать счет-фактуры и накладной



Вкладка «Отчетность» («Анализ») необходима для составления отчетов по поставкам и продажам за период. Окно представлено на рисунке 16.

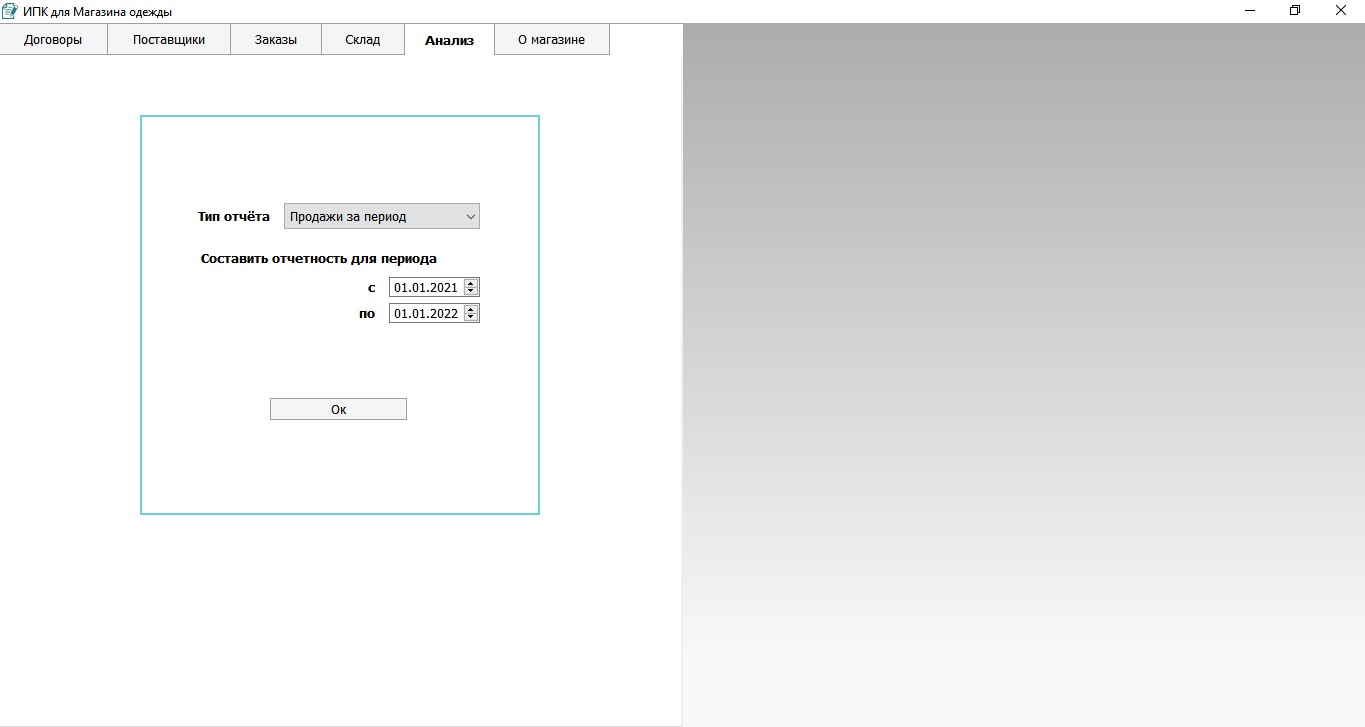
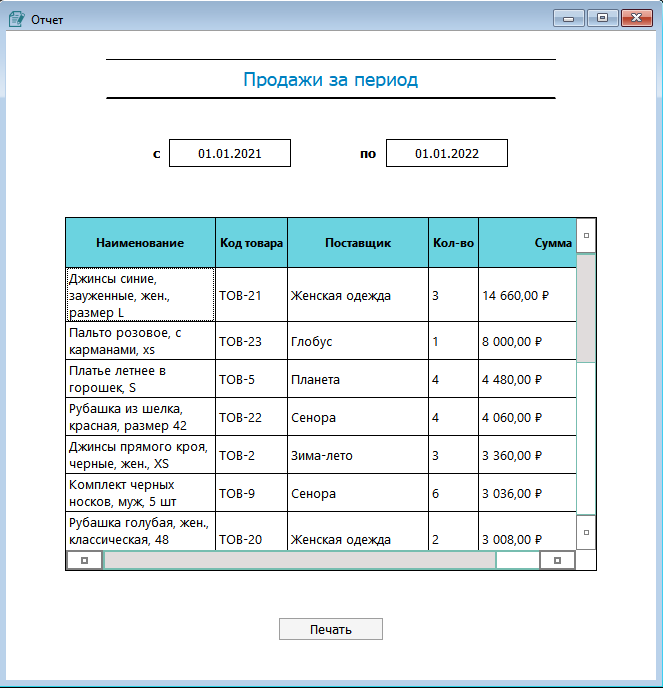
Для составления отчета необходимо выбрать тип отчета и указать даты отчетного периода. После нажатия на кнопку «Ок» будет открыто окно отчета, представленное на рисунке 17.

Рисунок 17 – Окно отчета

Рисунок 16 – Вкладка «Анализ»

Полученный отчет можно вывести на печать. Пример печати отчета представлен на рисунке 18.



Рисунок 18 - Печать отчета

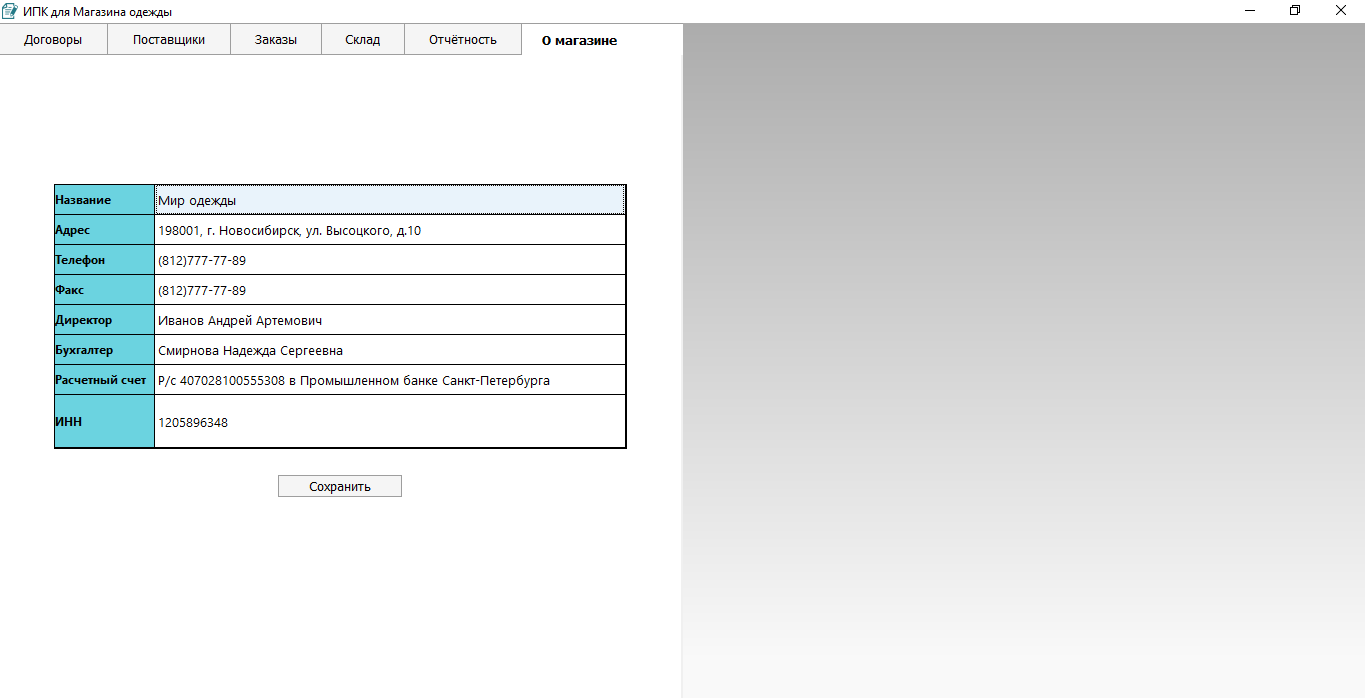
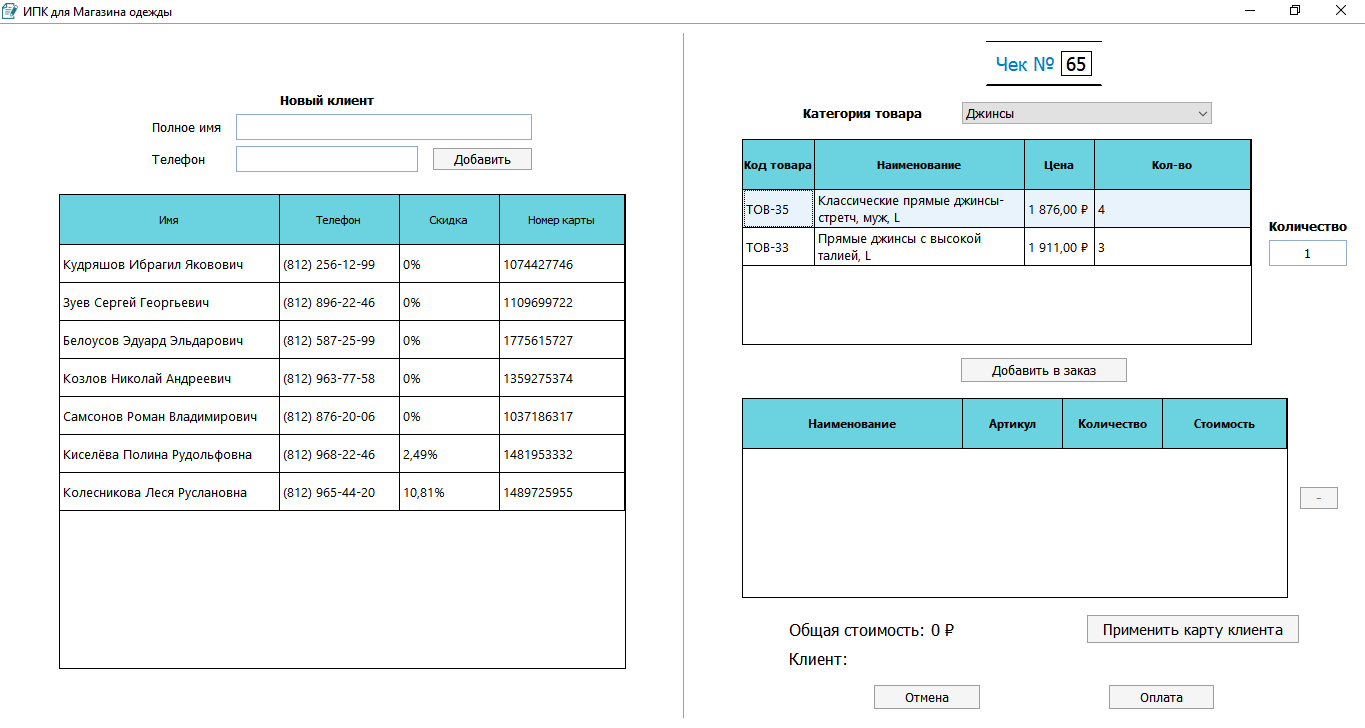
Вкладка «О магазине» предназначена для просмотра и редактирования основной информации о магазине одежды. Окно представлено на рисунке 19.

Рисунок 19 – Вкладка «О магазине»

При вводе в окне авторизации пароля кассира пользователь попадает в главное меню кассира. Окно представлено на рисунке 20.

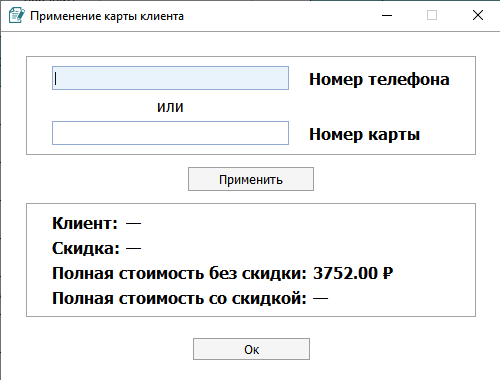
В левой части окна происходит регистрация нового клиента в базе данных. В правой части – оформление покупки. После того, как пользователь заполнит корзину товаров аналогично оформлению заказа на поставку он может нажать на кнопку «Применить карту клиента». Будет выведено окно «Применение карты клиента», представленное на рисунке 21.

Рисунок 20 – Главное меню кассира

Рисунок 21 – Применение карты клиента

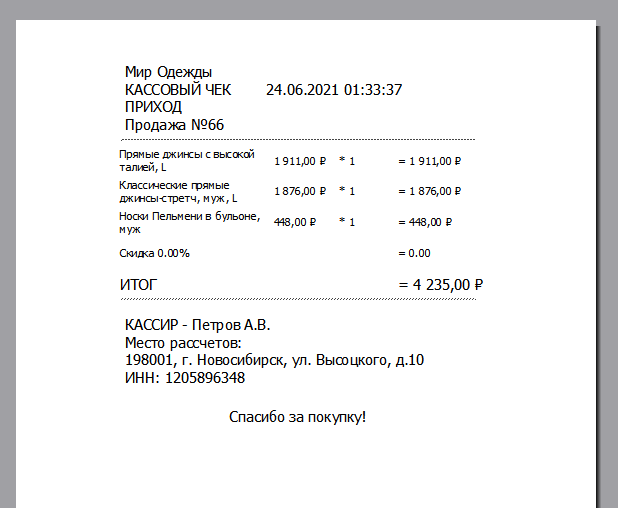
После внесения требуемых данных в форме отобразится стоимость со скидкой. Далее заказ необходимо будет оплатить, нажав на кнопку «Оплатить». На печать будет выведен чек, изображенный на рисунке 22.

Рисунок 22 – Печать чека

ЗАКЛЮЧЕНИЕ

В ходе выполнения курсового проекта поставленная цель – разработка ИПК для магазина одежды, обеспечивающего работу директора по закупкам одежды у поставщиков, а также работу продавца по продажам одежды клиентам магазина – была достигнута.

Для этого пришлось решить следующие задачи:

* описать предметную область для разрабатываемого ИПК;
* провести анализ предметной области, выявить объекты, их свойства и связи, которые являются существенными для пользователей системы;
* определить потребности пользователей системы ИПК, описать запросы к БД;
* описать сущности и их атрибуты, задать первичные и альтернативные ключи, определить ограничения целостности;
* привести модель к требуемому уровню нормализации данных;
* преобразовать концептуальную модель к модели даталогического уровня;
* задать параметры хранения данных (перейти к физическому уровню БД);
* реализовать приложение;

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Верхолат, А.М. Проектирование структуры базы данных: пособие по курсовому проектированию. Изд. 2-е, испр. и доп. / А.М. Верхолат, В.П. Суслов; Балт. гос. техн. ун-т. – СПб., 2018. – 65 с.
2. Документация к PostgreSQL 13.3. Перевод на русский язык, 2015-2021 гг. - Компания «Постгрес Профессиональный». Официальная страница русскоязычной документации - URL: https://postgrespro.ru/docs/ (Дата обращения 17.06.2021).
3. Диго С.М. БАЗЫ ДАННЫХ. ПРОЕКТИРОВАНИЕ И СОЗДАНИЕ: Учебно-методический комплекс. – М.: Изд. центр ЕАОИ. 2008. – 171 с.
4. Документация к Qt 5.15. Официальная страница – URL: https://doc.qt.io/qt-5.15/ (Дата обращения 17.06.2021).

ПРИЛОЖЕНИЕ А – SQL-скрипт создания БД

SQL-cкрипт создания БД, выгруженный из БД и включающий только SQL-операторы генерации структуры БД, находится в архиве «Курсовой проект». Файл называется «ПРИЛОЖЕНИЕ А - SQLsqript.sql».

ПРИЛОЖЕНИЕ Б – Справочные запросы

## Типовые SQL-запросы

1. Добавление нового поставщика

INSERT INTO suppliers (name, address, phone, fax, director, accountant, bank, inn) VALUES ('Глобус', 'пр. Луначарского, 54, Санкт-Петербург, 194356', '(812)425-97-85', '(812)425-97-85', 'Петров Андрей Васильевич', 'Иванова Мария Максимовна', 'Р/с 40817810099910004312 в СберБанке ', 1289647682);

2. Добавление нового товара в каталог поставщика

INSERT INTO catalog\_sup(name, description, price, supplier\_code) VALUES ('Пальто', 'Пальто розовое, с карманами, xs', '3100', 'ПОС-9');

3. Выполнение оплаты заказа

UPDATE orders\_sup SET paid = true WHERE num = 'ЗАК-23';

4. Удаление пустого неоплаченного заказа

DELETE FROM orders\_sup as o WHERE NOT EXISTS (SELECT \* FROM ordered\_sup\_goods WHERE order\_sup\_num = o.num) AND paid = false

## Сложные SELECT запросы

1. **Договоры с истекшим сроком (архивные)**

SELECT con.num, sup.name, st.name FROM contracts AS con

JOIN suppliers AS sup ON con.supplier\_code = sup.code

JOIN stores AS st ON con.store\_code = st.code

WHERE con.completion\_date < NOW() ;

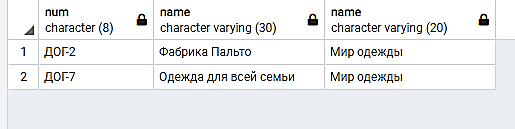


Рисунок 23 - Договоры с истекшим сроком (архивные)

1. **Самые продаваемые товары (n – количество проданных шт)**

SELECT SUM(cl\_g.number) AS n, cl\_g.good\_code, cat.name, cat.description, st.price, sup.name

FROM ordered\_client\_goods AS cl\_g

JOIN catalog\_sup AS cat ON good\_code = cat.code

JOIN stock AS st ON good\_code = st.code

JOIN suppliers AS sup ON cat.supplier\_code = sup.code

GROUP BY cl\_g.good\_code, cat.name, cat.description, st.price, sup.name ORDER BY n DESC;

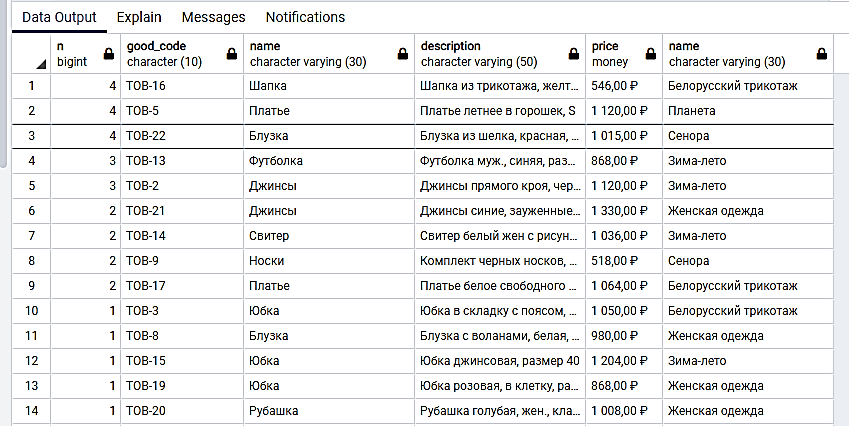


Рисунок 24 - Самые продаваемые товары

## Создание представления VIEW

1. **Информация по заказам на поставку**

CREATE OR REPLACE VIEW orders\_from\_supplier AS SELECT

ord.num AS "Номер заказа",

contr.num AS "Номер договора",

sup.name AS "Поставщик",

rec.num AS "Номер счета",

rec.price\_nds AS "Сумма",

ord.paid AS "Оплата",

ord.sent AS "Получение"

FROM orders\_sup ord

JOIN receipts\_sup rec ON rec.order\_sup\_num = ord.num

 JOIN contracts contr ON contr.num = ord.contract\_num::bpchar

JOIN suppliers sup ON sup.code = contr.supplier\_code;

Все заказы у поставщика «Сенора»: SELECT \* FROM public.orders\_from\_supplier where "Поставщик" = 'Сенора';

Рисунок 25 - Информация по заказам на поставку

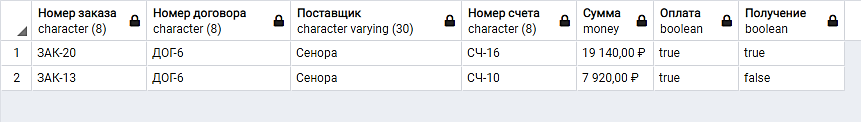


Рисунок 26 - Все заказы у поставщика «Сенора»

## Создание хранимой процедуры

**Добавление, изменение и удаление клиента**

CREATE OR REPLACE PROCEDURE client\_processing(

choose\_ integer,

name\_ character varying DEFAULT NULL::character varying,

phone\_ character varying DEFAULT NULL::character varying,

code\_ character varying DEFAULT NULL::character varying)

LANGUAGE 'plpgsql'

AS $BODY$

begin

if (choose\_= 1) then

INSERT INTO clients (name, phone)

VALUES (name\_, phone\_);

elsif (choose\_ = 2) then

UPDATE clients SET

name = name\_, phone = phone\_

WHERE code = code\_;

elsif (choose\_ = 3) then

DELETE FROM clients

WHERE code = code\_;

end if;

end

$BODY$;

**Пример использования:**

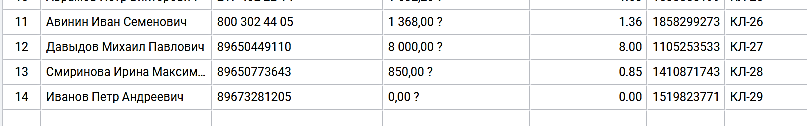
CALL client\_processing (1, 'Иванов Петр Андреевич', '89673281205')

Рисунок 27 - Пример использования

Полный список разработанных типовых запросов, представлений и хранимых процедур можно найти в архиве «Курсовой проект» в файле «ПРИЛОЖЕНИЕ Б – Справочные запросы.docx».

ПРИЛОЖЕНИЕ В – Исходный текст программы

Исходный текст программы находится в архиве «Курсовой проект». Папка с исполняемыми файлами называется «clothing-store».