Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Сибирский государственный автомобильно-дорожный университет (СибАДИ)»

Институт Информационные системы, экономика и управление

Кафедра Информационная безопасность

Направление Информационная безопасность

Профиль Информационная безопасность автоматизированных систем

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**к курсовому проекту**

по дисциплине: «Безопасность систем баз данных»

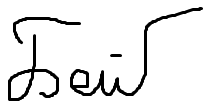
Тема проекта: «Обеспечение безопасности данных в автоматизированной системе «Учет товара на складе» с использованием СУБД PostgreSQL 16 и ЯП Python»

Выполнил:

Студент группы БИб-21Э1 Бейсенов Руслан Асылханович

Курсовой проект сдан на проверку:

«4» июня 2024 г.

СтудентБейсенов Руслан Асылханович

(подпись студента)

Курсовой проект допущен к защите:

«5» июня 2024 г.

Руководитель проекта

доцент\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Т.М. Опарина

Консультант проекта

д.п.н., проф. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_З.В. Семенова

Курсовой проект защищен с оценкой:

«6» июня 2024 г.

Руководитель проекта

доцент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Т.М. Опарина

Консультант проекта

д.п.н., проф. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_З.В. Семенова

Омск 2024

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Сибирский государственный автомобильно-дорожный университет (СибАДИ)»

Кафедра «Информационная безопасность»

**ЗАДАНИЕ**

**к курсовому проекту**

студенту группы БИб-21ЭБейсенов Руслан Асылханович

1. Тема курсового проекта: «Обеспечение безопасности данных в автоматизированной системе «Учет товара на складе» с использованием СУБД PostgreSQL 16 и ЯП Python»
2. Исходные данные к курсовому проекту:

* предметная область: склад на базе магазина, в котором ведется учёт товаров, покупателей, поставщиков и всего остального для полного функционирования;
* минимальное количество таблиц – 8, минимальное количество информационных запросов – 8, минимальное количество групп ролей – 3;
* механизмы защиты: триггеры, резервное копирование, шифрование данных, роли;
* используемые инструментарий: СУБД, PostgreSQL 16 – серверная часть, IDE PyCharm Community Edition 2023 – инструмент разработки клиентской части, PyQt – клиентская часть;
* официальная документация ЯП Python;
* официальная документация PyQt;
* официальная документация PostgreSQL 16.

1. Содержание пояснительной записки:

1.ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ ПРЕДМЕТНОЙ ОБЛАСТИ И МЕТОДОВ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ

1.1. Описание предметной области и бизнес-процессов магазина

1.2. Описание информационных запросов, реализуемых в ИС

1.3. Механизмы обеспечения безопасности данных в СУБД PostgreSQL

2. РЕАЛИЗАЦИЯ ИС «Учет товара на складе» И ОБЕСПЕЧЕНИЕ БЕЗОПАСНОСТИ С ИСПОЛЬЗОВАНИЕМ СУБД POSTGRESQL И ЯП PYTHON

2.1. Логическое проектирование БД ИС «Учет товара на складе»

2.2. Физическое проектирование БД ИС «Учет товара на складе»

2.2.1. Реализация таблиц БД ИС «Учет товара на складе» в СУБД PostgreSQL

2.2.2. Реализация запросов БД ИС «Учет товара на складе» в СУБД PostgreSQL

2.2.3. Реализация триггеров БД ИС «Учет товара на складе» в СУБД PostgreSQL

2.3. Реализация клиентской части

2.4. Реализация основных методов и средств обеспечения безопасности данных

3. ДОКУМЕНТАЦИЯ НА ИС «Учет товара на складе»

3.1. Руководство по использованию ИС «Учет товара на складе» для покупателя

3.2. Руководство по использованию ИС «Учет товара на складе» для продавца

3.3. Руководство по использованию ИС «Учет товара на складе» для администратора

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Перечень демонстрационного материала для сопровождения доклада   
   при защите курсового проекта:

* Презентация Microsoft PowerPoint.

1. Срок сдачи работы: «05 июня» 2024 г.

Задание выдано «12» февраля 2024 г.

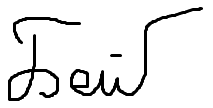
Руководитель проекта: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Т.М. Опарина

(подпись преподавателя)

Консультант проекта: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ З.В. Семенова

(подпись преподавателя)

Задание к исполнению принял «» 2024 г.

Студент  Бейсенов Руслан Асылханович

(подпись студента)

**Содержание**

[**Введение** 5](#_Toc169758074)

[**1.** **ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ ПРЕДМЕТНОЙ ОБЛАСТИ И МЕТОДОВ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ** 6](#_Toc169758075)

[**1.1. Описание предметной области и бизнес-процессов магазина** 6](#_Toc169758076)

[**1.2. Описание информационных запросов, реализуемых в ИС** 9](#_Toc169758077)

[**1.3. Механизмы обеспечения безопасности данных** 12](#_Toc169758078)

[**2.** **РЕАЛИЗАЦИЯ ИС «Учет товара на складе»** 16](#_Toc169758079)

[**2.1. Логическое проектирование БД ИС «Учет товара на складе»** 16](#_Toc169758080)

[**2.2. Физическое проектирование БД ИС «Учет товара на складе»** 19](#_Toc169758081)

[**2.2.1. Реализация таблиц БД ИС «Учет товара на складе» в СУБД PostgreSQL** 20](#_Toc169758082)

[**2.2.2. Реализация запросов БД ИС «Учет товара на складе» в СУБД PostgreSQL** 24](#_Toc169758083)

[**2.2.3. Реализация триггеров БД ИС «Учет товара на складе» в СУБД PostgreSQL** 28](#_Toc169758084)

[**2.3. Реализация клиентской части АС** 32](#_Toc169758085)

[**2.4. Реализация основных методов и средств обеспечения безопасности данных** 35](#_Toc169758086)

[**3. ДОКУМЕНТАЦИЯ НА ИС «Учет товара на складе»** 39](#_Toc169758087)

[**3.1**.**Руководство по использованию ИС «Учет товара на складе» для покупателя** 39](#_Toc169758088)

[**3.2.Руководство по использованию ИС «Учет товара на складе» для продавца** 40](#_Toc169758089)

[**3.3.Руководство по использованию ИС «Учет товара на складе» для администратора** 44](#_Toc169758090)

[**Заключение** 49](#_Toc169758091)

[**Список литературы.** 50](#_Toc169758092)

# **Введение**

В современном мире автоматизированные системы играют важную роль в бизнесе, позволяя оптимизировать и упростить многие процессы. Одной из ключевых задач при разработке и эксплуатации таких систем является обеспечение безопасности данных. В случае с автоматизированной системой «Учет товара на складе» это особенно важно, так как потеря или несанкционированный доступ к данным может привести к серьезным финансовым и репутационным потерям.

По данным анализа Positive Technologies кибератаки в I квартале 2024 года приводили к разным последствиям: злоумышленникам удавалось успешно атаковать как небольшие предприятия, так и отраслевых гигантов, не обошлось без сбоев в работе целых городов. Основной фокус преступников был направлен на получение конфиденциальной информации (доля таких атак составила 54%) и нарушение основной деятельности организаций (доля таких атак составила 33%). Поэтому так важно обеспечить защиту информационным системам [21].

Цель проекта: обеспечение безопасности данных в информационной системе «Учет товара на складе» с использованием СУБД PostgreSQL и ЯП Python.

Для достижения цели необходимо выполнить задачи.

1. Выявить предметную область, описать бизнес-процессы и определить информационные запросы для ИС «Учет товара на складе».
2. Охарактеризовать возможности обеспечения безопасности СУБД PostgreSQL 16.
3. Спроектировать и разработать ИС «Учет товара на складе».
4. Реализовать основные механизмы защиты данных ИС «Учёт товара на складе».
5. Подготовить руководство по использованию ИС.
6. **ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ ПРЕДМЕТНОЙ ОБЛАСТИ И МЕТОДОВ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ**
   1. **Описание предметной области и бизнес-процессов магазина**

Магазин — это физическое виртуальное пространство, где продаются товары. Основная цель магазина — удовлетворить потребности и желания клиентов, предлагая широкий ассортимент товаров.

У магазина есть различные функции.

1. Закупка и хранение товаров или запасов.
2. Выкладка и демонстрация товаров для привлечения клиентов.
3. Оказание помощи клиентам в выборе и покупке товаров.
4. Обработка платежей и выдача сдачи.
5. Управление запасами и контроль над потерями.
6. Обеспечение обслуживания клиентов, включая отклик на отзывы.
7. Ключевые группы сотрудников магазина.

В магазине работают различные группы сотрудников, каждая из которых играет важную роль в обеспечении бесперебойной работы магазина и удовлетворенности клиентов. Основными группами пользователями ИС магазина являются:

* покупатели – покупают товар;
* продавцы - взаимодействуют с клиентами, помогают им с выбором и покупкой товаров, а также обрабатывают платежи;
* администраторы - отвечают за общее управление магазином, управление запасами и обеспечение обслуживания клиентов.

Рассмотрим пример функционирования информационной системы «Учет товара на складе» на примере основной функции этой системы.

К функциям системы можно отнести.

1. Регистрация поступления товара: при поступлении товара на склад система регистрирует его идентификатор, количество и цену.
2. Учет текущих запасов: система поддерживает актуальную информацию о количестве товаров в режиме реального времени.
3. Регистрация продажи товара: при продаже товара со склада система регистрирует идентификатор, количество и стоимость. А также система регистрирует новую продажу, куда уже заносит информацию о товаре, покупателе и дату продажи.
4. Управление заказами на пополнение запасов: система автоматически генерирует заказы на пополнение запасов на основе установленных пороговых значений запасов. Но также продавцы могут вручную регулировать заказ на пополнение.

Таким образом, на основе описаний предметной области и процесса учета товаров на складе в информационной системе «Учет товара на складе» можно выделить следующие основные бизнес-процессы.

1. Бизнес-процесс «Авторизация пользователя».
2. Бизнес-процесс «Регистрация новых пользователей».
3. Бизнес-процесс «Учет поставок».
4. Бизнес-процесс «Учет продаж».

Бизнес-процессы спроектированы на сайте bpmn.io [18]. Бизнес-процесс «Авторизация пользователя», представленный на рисунке 1 предусматривает ведение базы данных пользователей, аутентификацию и авторизацию пользователей, а также контроль доступа к данным пользователя ИС. Пользователями данной ИС являются: покупатель, продавец и администратор.

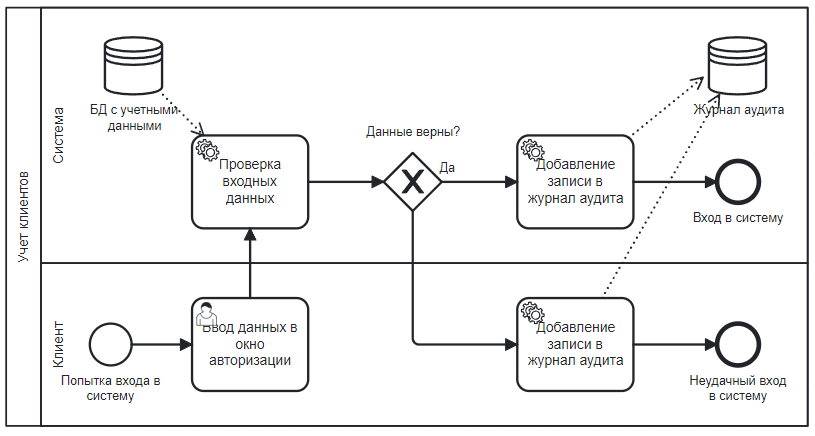


Рисунок 1 – Бизнес-процесс «Авторизация пользователя»

Бизнес-процесс «Регистрация новых пользователей» представлен на рисунке 2. Он предусматривает сбор информации о новом пользователе, проверку ее достоверности, создание учетной записи пользователя, шифрование данных и назначение прав доступа.

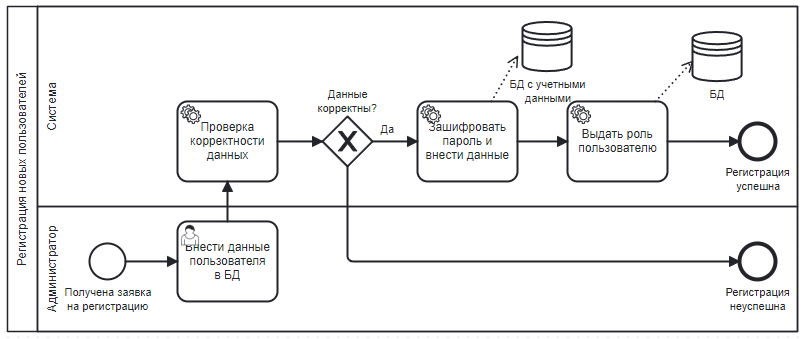


Рисунок 2 – Бизнес-процесс «Регистрация новых пользователей»

Бизнес-процесс «Учет поставок» представлен на рисунке 3. Он начинается с приемки товара от поставщика и проверки его количества и качества. Далее администратор вносит информацию о поставке в систему учета, включая наименование товара, количество, цену и дату поставки. После проверки происходит обновление данных о количестве товара на складе.

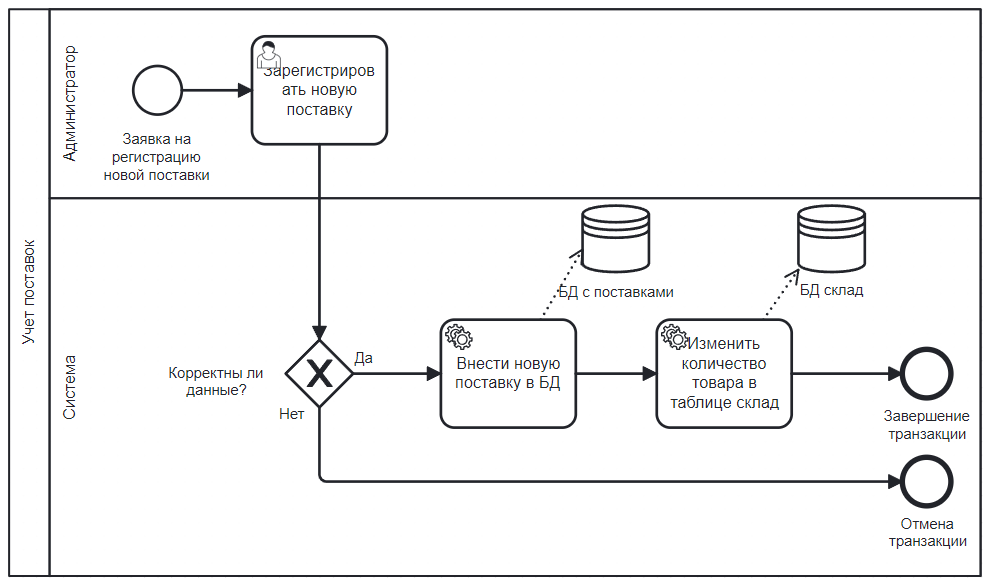


Рисунок 3 – Бизнес-процесс «Учет поставок»

Бизнес-процесс «Учет продаж» представлен на рисунке 4. Он предусматривает сбор информации о новой продаже, проверку ее достоверности, создание новой записи и обновление количества товара на складе.

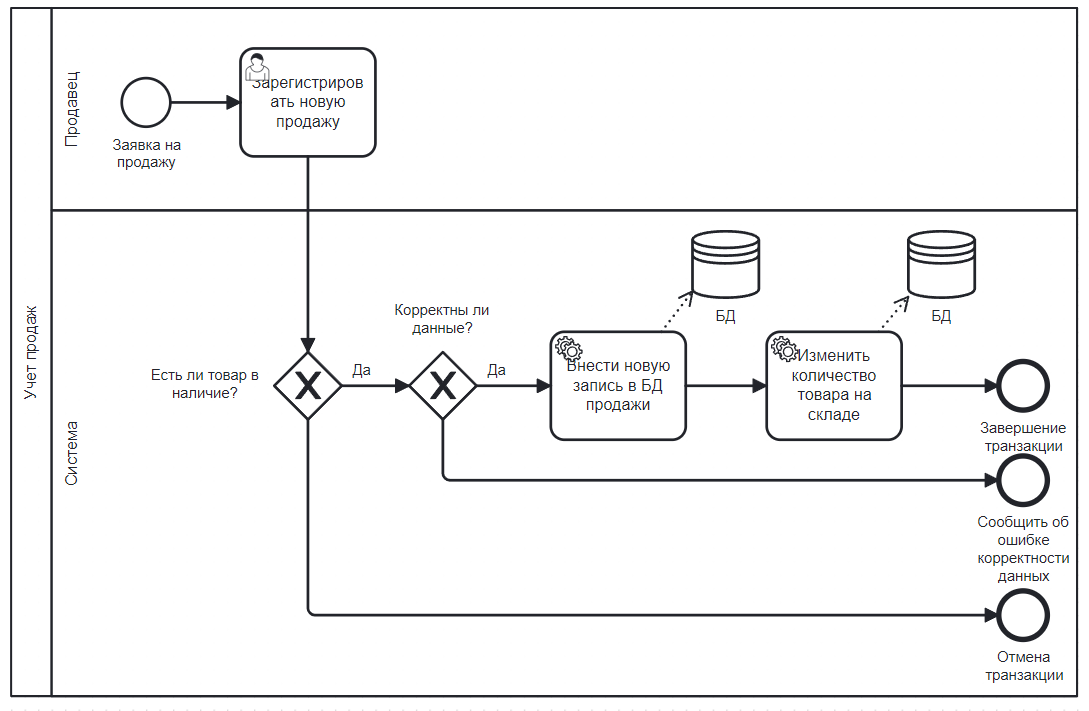


Рисунок 4 – Бизнес-процесс «Учет продаж»

После формирования бизнес-процессов необходимо описать информационный запросы, которые должны выполняться в ИС.

**1.2. Описание информационных запросов, реализуемых в ИС**

Для эффективного удовлетворения информационных потребностей пользователей информационной системы необходимо сформулировать информационные запросы, определяя при этом группу пользователей, имеющих право их выполнять. Запросы представлены в таблице 1.

Таблица 1 – Запросы

| Группа пользователей | Запрос | Описание |
| --- | --- | --- |
| Администратор, продавец | Вывести информацию о товарах конкретного отдела(id товара, название товара и idотдела) | Нужен для более эффективного сбора статистики о товарах. |
| Администратор, продавец | Создать запрос, в результате которого выводится список  просроченных товаров (id товара, количество товара, дата изготовления, срок годности) | Нужен для более эффективной работы магазина. |
| Администратор, продавец | Найти количество товаров конкретного отдела (id товара, название, id отдела) | Нужен для более эффективного сбора статистики о товарах. |
| Администратор, продавец | Вывести информацию о товарах по цене меньше 30 руб. (id товара, название товара) | Нужен для более эффективного сбора статистики о товарах. |
| Администратор, продавец | Вывести информацию о покупателях, посетивших магазин за последние 2 дня.(id покупателя, фамилия, имя, номер телефона, email покупателей) | Нужен для более эффективного сбора статистики о покупателях. |
| Администратор, продавец | Вывести информацию о товарах, которых осталось на складе меньше 10% от товара с максимальным количеством. (id товара, название товаров) | Нужен для более эффективного сбора статистики о товарах. |
| Администратор, продавец | Вывести информацию о товарах, поставленных за последний месяц(id поставки, id поставщика, id товара, дату поставки, товаров) | Нужен для более эффективного сбора статистики о товарах. |
| Администратор | Вывести информацию о покупателях, поставивших оценку выше 4(id отзыва, id продажи, оценку и id покупателя) | Нужен для улучшения обслуживания клиентов на основе анализа отзывов. |
| Покупатель | Дать право покупателям просматривать данные о своих покупках (id продажи, адрес, id покупателя, дата продажи, количество товара, id товара) | Нужен для анализа своих покупок |
| Продавец | Дать право продавцу вывести информацию о количестве товара на складе (id товара, количество товара), которых меньше 10 и добавлять информацию о поставках (id товара, количество, дата заказа), если этого товара осталось в наличие меньше 10 штук | Нужен для более эффективной работы магазина. |
| Администратор | Дать право просматривать первые 10% записей (id отзыва, id продажи, оценка) из таблицы отзывы администратору | Нужен для оптимизации обслуживания клиентов на основе анализа отзывов. |

## **1.3. Механизмы обеспечения безопасности данных**

Обеспечение безопасности баз данных подразумевает комплекс процедур, инструментов и средств управления, направленных на защиту баз данных от любых угроз. Главная цель этих мер заключается в обеспечении защиты конфиденциальной информации, поддержании конфиденциальности, доступности и целостности баз данных. Для достижения этой цели применяются средства защиты, которые обеспечивают безопасность не только данных, но и систем управления базами данных, связанных приложений и систем, физических и виртуальных серверов, а также сетевой инфраструктуры [1].

В СУБД имеются разнообразные механизмамы обеспечения безопасности данных. Кратко охарактеризуем.

1. **Аутентификация**. PostgreSQL поддерживает несколько методов аутентификации, такие как пароль, kerberos, LDAP, и другие. Это помогает предотвратить несанкционированный доступ к базе данных.
2. **Авторизация**. Пользователи и группы в PostgreSQL могут иметь различные привилегии на уровне базы данных, таблицы или столбца. Это позволяет администраторам контролировать, кто и как может взаимодействовать с данными.
3. **Шифрование данных**. PostgreSQL поддерживает шифрование данных на уровне столбца с помощью модулей pgcrypto и pgp\_sym\_enc. Кроме того, PostgreSQL поддерживает шифрование всей базы данных с помощью полного шифрования диска (FDE) [7].
4. **Роли** в PostgreSQL предоставляет возможность назначать права доступа на таблицы, столбцы и строки для различных пользователей и групп. Это позволяет контролировать, кто может просматривать, изменять или удалять данные в базе данных [2]. Пример реализации ролевой политики представлен на рисунке 5.

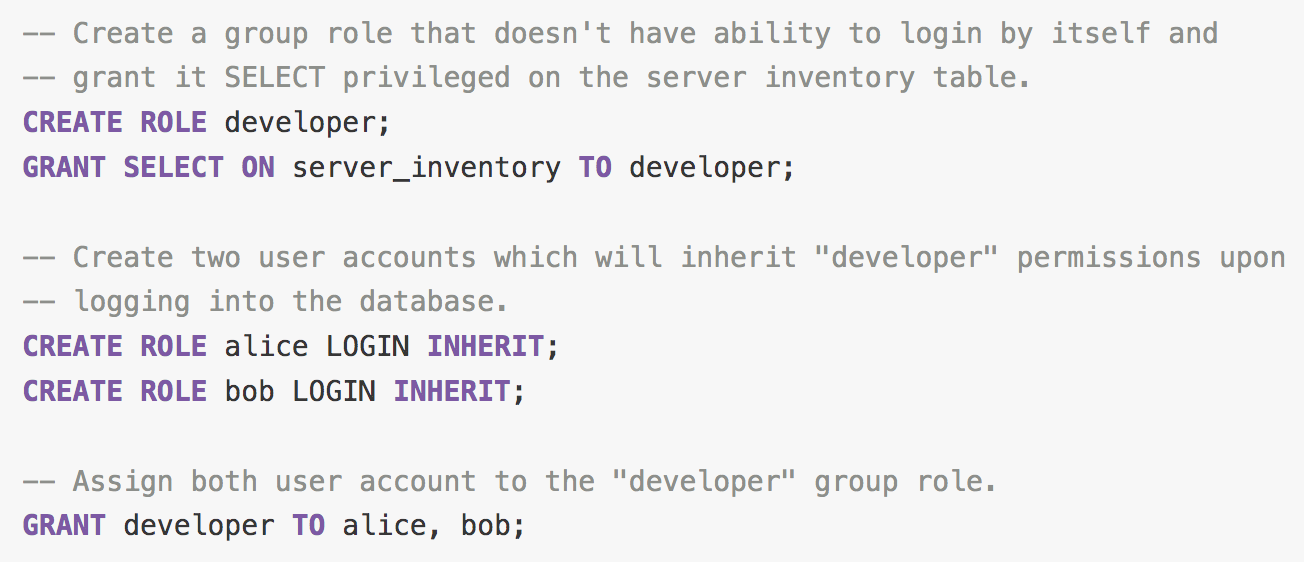


Рисунок 5 – Пример ролевой политики безопасности

1. **Безопасность на уровне строк (Row Level Security, RLS)** в PostgreSQL позволяет контролировать доступ к отдельным строкам в таблицах на основе политик безопасности. Это означает, что администратор базы данных может настроить политики, которые определяют, какие строки в таблице доступны для просмотра или изменения конкретным пользователям или группам пользователей. RLS реализуется с помощью политик, которые определяются для каждой таблицы [2]. Пример реализации политик безопасности представлен на рисунке 6.

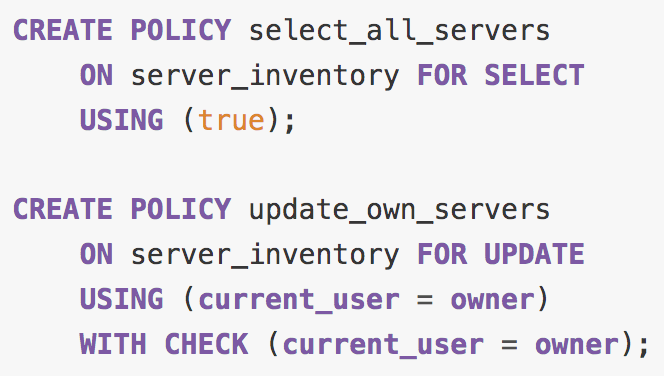
****

Рисунок 6 – Пример политик безопасности

1. **Аудит** в PostgreSQL поддерживает аудит событий, связанных с безопасностью, таких как попытки несанкционированного доступа к базе данных, изменения прав доступа и другие. Это позволяет отслеживать деятельность пользователей и обнаруживать потенциальные угрозы безопасности [2].
2. **Резервное копирование и восстановление**. PostgreSQL предоставляет возможность создавать резервные копии баз данных и восстанавливать их в случае потери данных или повреждения базы данных. Это позволяет обеспечить сохранность данных и минимизировать риски потери данных [3].
3. PostgreSQL обеспечивает **шифрование** на разных уровнях и даёт гибкость в выборе средств защиты данных в случае кражи сервера, от недобросовестных администраторов или в небезопасных сетях. Шифрование может также требоваться для защиты конфиденциальных данных, например, медицинских сведений или финансовых транзакций [4].

Подводя итог, можно сказать, что предметная область включает в себя различные виды деятельности, такие как закупка и хранение товаров, управление запасами и контроль над потерями, учет покупателей и поставщиков. Основными группами пользователей системы являются покупатели, продавцы и администраторы.

Для обеспечения безопасности данных необходимо применять комплекс процедур, инструментов и средств управления, направленных на защиту баз данных от любых угроз. Главная цель этих мер заключается в обеспечении защиты конфиденциальной информации, поддержании конфиденциальности, доступности и целостности баз данных. Таким образом, для обеспечения безопасности данных в магазине необходимо использовать информационную систему с надежными механизмами защиты данных и применять комплекс мер по защите баз данных от любых угроз.

# **РЕАЛИЗАЦИЯ ИС «Учет товара на складе»**

* 1. **Логическое проектирование БД ИС «Учет товара на складе»**

Атрибуты данных – это необходимый элемент базы, который представляет собой какую-либо характерную и уникальную информацию [17].

ИС включает в себя такие наименование таблиц и атрибуты, которые представленные в таблице 2.

Таблица 2 – Атрибуты и типы данных таблиц.

| **Наименование таблиц.** | **Атрибуты и типы данных.** |
| --- | --- |
| Склад | id\_товара (serial, primary key, references Товары (id Товара)),  Количество товара (integer),  Стоимость (integer) |
| Заказы | id\_товара (serial, primary key, references Товары (id Товара)),  Количество товара (integer),  Дата заказа (date) |
| Товары | id\_товара (serial, primary key),  Название товара (text),  ID отдела (text, referencesОтделы(id\_отдела)) |
| Поставка | Idпоставки (serial, primary key),  id товара (integer, references Товары (id товара)),  Количество товара (integer),  Дата поставки (date),  Дата изготовления (date),  Срок годности (integer) |
| Отзывы | Id отзыва (serial, Primary key),  Id продажи (integer, references Продажи (id продажи),  Оценка (integer) |
| Отделы | Id отдела (serial, Primary key),  Название (text) |
| Покупатели | Id покупателя (serial, Primary key),  Фамилия (text),  Имя (text),  Номер телефона (integer),  Email (text) |
| Адреса | Id адреса (serial, Primary key),  Адрес (text) |
| Продажи | Id продажи (serial, Primary key),  Idтовара (integer, references Товары (id товара)),  Idпокупателя (integer, references Покупатели (id покупателя)),  Дата продажи (date),  Idпоставки (integer, references Поставки (id поставки)),  Количества товара (integer),  Idадреса (integer, references Адреса(id адреса)) |

Склад. Эта таблица хранит информацию о товарах на складе. Здесь указывается id товара и его количество на складе.

Заказы. В этой таблице хранится информация о заказах. Здесь указывается id товара, количество заказанного товара и дата заказа.

Товары. Эта таблица содержит информацию о товарах. Здесь указывается id товара, его название и id отдела, к которому относится товар.

Отделы. В этой таблице хранится информация об отделах. Здесь указывается id отдела и его название.

Поставка. В этой таблице хранится информация о поставках. Здесь указывается id поставки, id поставщика, id товара, количество поставленного товара, дата поставки, дата изготовления и срок годности товара.

Продажи. Эта таблица содержит информацию о продажах. Здесь указывается id продажи, id товара, id покупателя, дата продажи, id поставки, количество проданного товара и id адреса доставки.

Покупатели. В этой таблице хранится информация о покупателях. Здесь указывается id покупателя, его фамилия, имя, номер телефона и email.

Адреса. В этой таблице хранится информация об адресах. Здесь указывается id адреса, id покупателя, которому принадлежит адрес, и сам адрес.

Таблица «Отзывы» хранит информацию об отзывах о проданных товарах. Здесь указывается id отзыва, id продажи, к которой относится отзыв, и оценка, данная покупателем товару.

Для дальнейшей работы необходимо понимать структуру информационной системы, для чего была разработана схема базы данных на основе таблицы 1 при помощи сайта draw.io [8], которая представлена на рисунке 7.

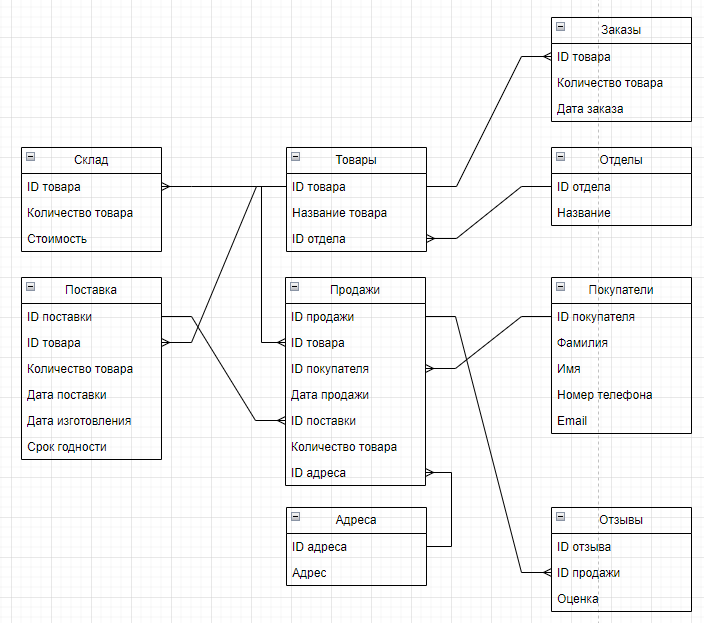


Рисунок 7 – Схема БД

На рисунке выше представлена схема БД, на которой описаны таблицы, позволяющие осуществить задуманный функционал ИС. К примеру, в таблице «Склад» хранятся данные о текущем количестве товаров. Значения изменяются при каждой новой записи в таблицу «Поставки» или «Продажи», что позволяет отслеживать текущее количество товара. И в случае понижения количества товара ниже установленной нормы добавляется новая запись в таблицу «Заказы», чтобы поддерживать уровень товара на складе.

Таким образом, выполнено логическое проектирование базы данных информационной системы «Учет товара на складе». В ходе проектирования были определены таблицы и атрибуты, необходимые для хранения информации о товарах, заказах, поставках, продажах, отделах, покупателях и адресах.

## **2.2. Физическое проектирование БД ИС «Учет товара на складе»**

Для реализации серверной части выбраны следующие элементы:

PostgreSQL 16 - система управления объектно-реляционными базами данных [16].

PgAdmin 4 – графический инструмент для управления базой данных.

ApplicationStackBuilder - система для установки дополнительных инструментов и драйверов для PostgreSQL.

CommandLineTools – инструменты командной строки

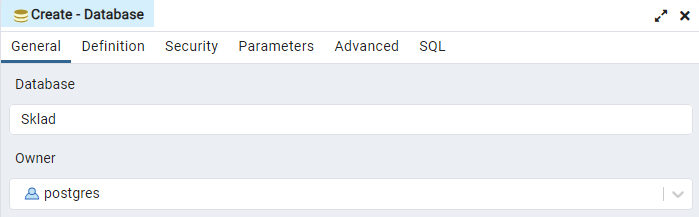


Рисунок 8 – Создание базы данных

Для создания базы данных, необходимо прописать команду create database <name> или в интерфейсе создать как показано на рисунке 8.

### **2.2.1. Реализация таблиц БД ИС «Учет товара на складе» в СУБД PostgreSQL**

Таблица — основной объект базы данных, предназначенный для хранения данных в структурированном виде [15].

На рисунках 9, 10 и 11 изображен процесс создания различных таблиц в базе данных с использованием языка SQL.

На рисунке 9 создаются таблицы «отдел», «покупатели» и «поставка». Таблица «отдел» содержит информацию о различных отделах товаров, таких как мясной отдел, молочный отдел и так далее. Таблица «покупатели» содержит информацию о покупателях, такую как имя, адрес, контактные данные и историю покупок. Таблица «поставка» содержит информацию о поставках товаров от поставщиков, такую как дата поставки, количество товара и информацию о поставщике.

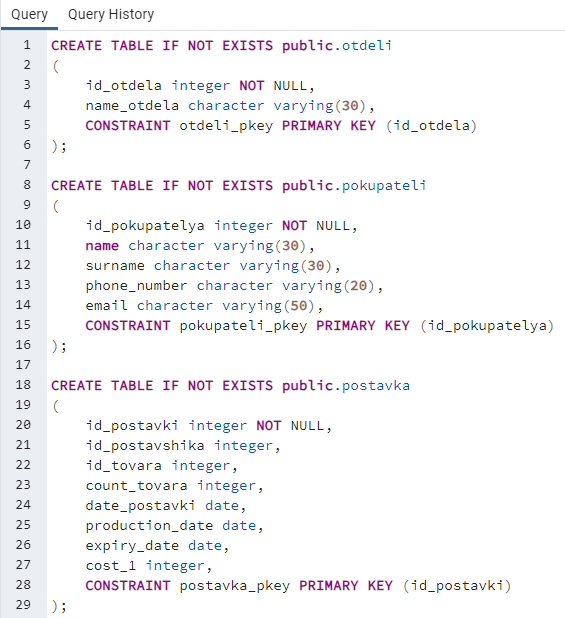


Рисунок 9 – Создание таблиц «отдел», «покупатели», «поставка»

На рисунке 10 создаются таблицы «поставщик», «продажи» и «товары». Таблица «поставщик» содержит информацию о поставщиках товаров, такую как имя, адрес, контактные данные и историю поставок. Таблица «продажи» содержит информацию о продажах товаров, такую как дата продажи, количество товара, информацию о покупателе и цену. Таблица «товары» содержит информацию о товарах, такую как название, описание, цену и количество на складе.

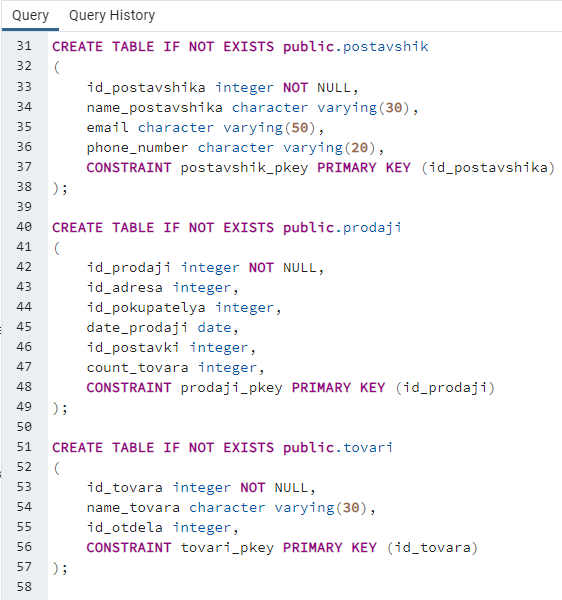


Рисунок 10 – Создание таблиц «поставщик», «продажи», «товары»

На рисунке 11 создаются таблицы «адреса», «отзывы», «склад» и «заказы». Таблица «адреса» содержит информацию об адресах покупателей. Таблица «отзывы» содержит отзывы покупателей о товарах. Таблица «склад» содержит информацию о товарах на складе, такую как количество товара и стоимость его. Таблица «заказы» содержит информацию о новых заказах на пополнение товаров, такую как дата заказа и количество.

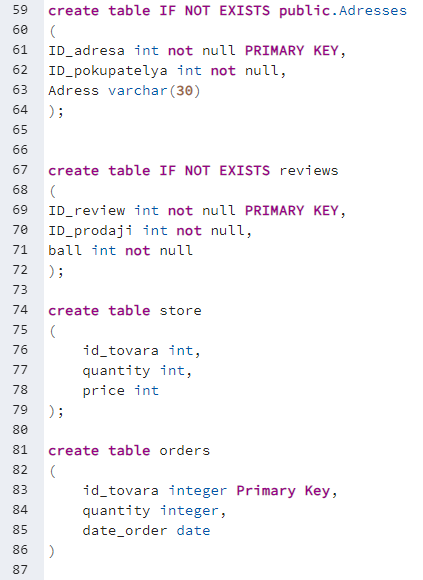


Рисунок 11 – Создание таблиц «адресы», «отзывы», «склад», «заказы»

На рисунке 12 изображен процесс создания связей между таблицами. Связи между таблицами позволяют объединять информацию из разных таблиц и создавать более сложные запросы к базе данных. Например, можно создать связь между таблицами «покупатели» и «продажи», чтобы получить информацию о продажах конкретного покупателя.



Рисунок 12 – Создание связей между таблицами

Таким образом, создана БД, которая описана на рисунке 7.

### **2.2.2. Реализация запросов БД ИС «Учет товара на складе» в СУБД PostgreSQL**

SQL-запросы - это наборы команд для работы с реляционными (табличными) базами данных [14].

Вывести все товары из отдела «Консервы». Запрос на вывод всех товаров из отдела «Консервы» представлен на рисунке 13 и нужен для получения информации о всех товарах, относящихся к данному отделу. Это может быть актуально при планировании ассортимента, подсчете остатков или для формирования специальных предложений для клиентов.

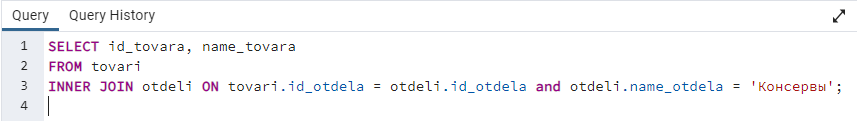


Рисунок 13 – Запрос на вывод товара

Создать запрос, в результате которого выводится список просроченных товаров. Запрос на вывод просроченных товаров нужен для контроля качества товаров и своевременного их утилизации. Это позволяет избежать продажи просроченных товаров и соответственно сохранить репутацию магазина. Реализация запроса представлена на рисунке 14.

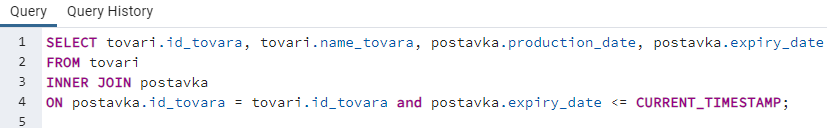


Рисунок 14 – Запрос на вывод просроченных товаров

Найти количество товаров молочного ассортимента. Запрос на вывод количества товаров молочного ассортимента представлен на рисунке 15 и может быть актуален для подсчета остатков товаров в данной категории или для планирования закупок.

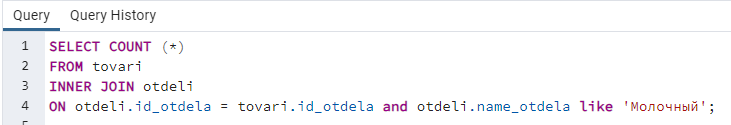


Рисунок 15 – Запрос на вывод товара молочного отдела

Создать запрос о товарах по цене меньше 30 руб. Запрос на вывод товаров со стоимостью меньше 30 рублей может быть актуален для формирования специальных предложений для клиентов, например, для продажи товаров по акции «Все до 30 рублей». Реализация данного запроса представлена на рисунке 16.

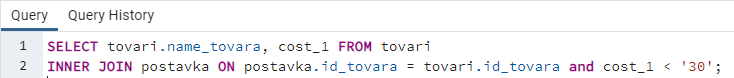


Рисунок 16 – Запрос на вывод товара со стоимостью меньше 30

Создать запрос, в котором содержится количество покупателей, посетивших магазин за последние 2 дня. Данный запрос актуален для анализа посещаемости магазина и прогнозирования объема продаж, а также он представлен на рисунке 17.



Рисунок 17 – Запрос на вывод последних покупателей

Определить список товаров, которых осталось на прилавках меньше 10% от товара с максимальным количеством. Запрос на выдачу списка товаров, которых осталось на прилавках меньше 10% от товара с максимальным количеством. Данный запрос необходим для своевременного пополнения товаров на прилавках и предотвращения ситуаций, когда товар заканчивается. Его реализация представлена на рисунке 18.

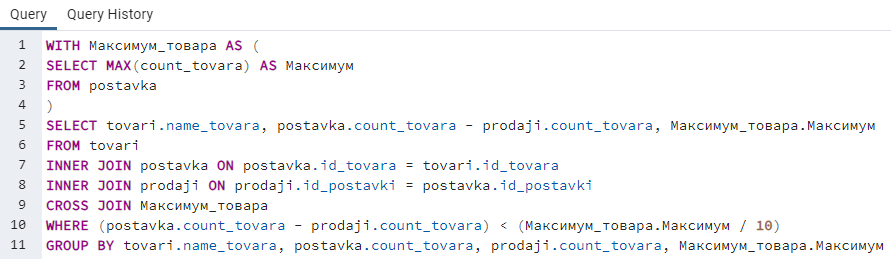


Рисунок 18 – Запрос на выдачу списка товаров, которых меньше 10%

Выдать список продуктов поставленные за последний месяц. Этот запрос нужен для контроля качества товаров и своевременного обнаружения бракованных партий. SQL код представлен на рисунке 19.

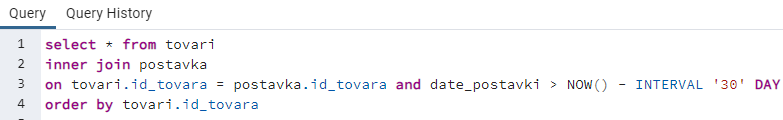


Рисунок 19 – Запрос на выдачу товаров поставленных в течении месяца

Выдать список покупателей, поставивших оценку выше 4. Запрос на вывод покупателей, оставивших оценку выше 4, может быть актуален для анализа удовлетворенности клиентов и формирования лояльности к магазину. Пример реализации представлен на рисунке 20.

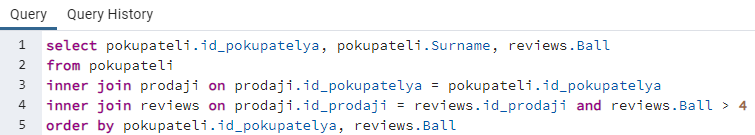


Рисунок 20 – Запрос на вывод покупателей

Все эти запросы могут быть актуальны для решения различных задач, связанных с управлением магазином, контролем качества товаров и анализом поведения клиентов.

### **2.2.3. Реализация триггеров БД ИС «Учет товара на складе» в СУБД PostgreSQL 16**

Триггер является указанием, что база данных должна автоматически выполнить заданную функцию, всякий раз, когда выполнен определённый тип операции. Триггеры можно использовать с таблицами, с представлениями и с внешними таблицами [13]. Ниже представлена реализация нескольких триггеров для безопасности информации и автоматизации работы БД ИС «Учет товара на складе». На рисунке 20 представлен триггер, который позволяет автоматически обновлять информацию о количестве товара в таблице store при добавлении данных в таблице postavka. Это может быть полезно для учета поставок, отслеживания количества конкретного товара на складе. Что и требуется для бизнес-процесса «Учет поставок», который представлен на рисунке 3.

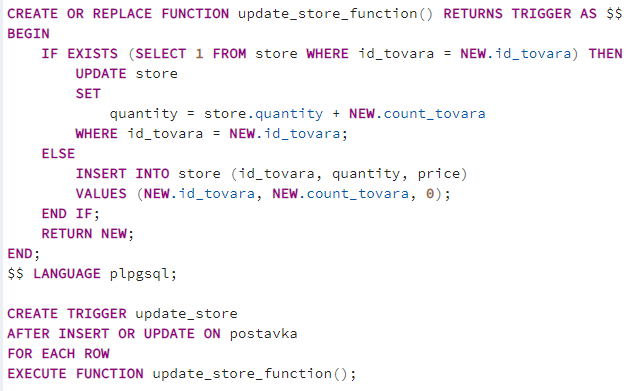


Рисунок 21 – Триггер на обновление после поставки

Триггер представленный на рисунке 22 используется для обновления информации о товаре в таблице store после вставки или обновления строки в таблице prodaji. Таким образом, этот триггер позволяет автоматически обновлять информацию о количестве и цене товара в таблице store при изменении данных в другой таблице. Это может быть полезно для отслеживания товаров на складе при учете продаж, который описан в бизнес-процессе «Учет продаж», который представлен на рисунке 4.

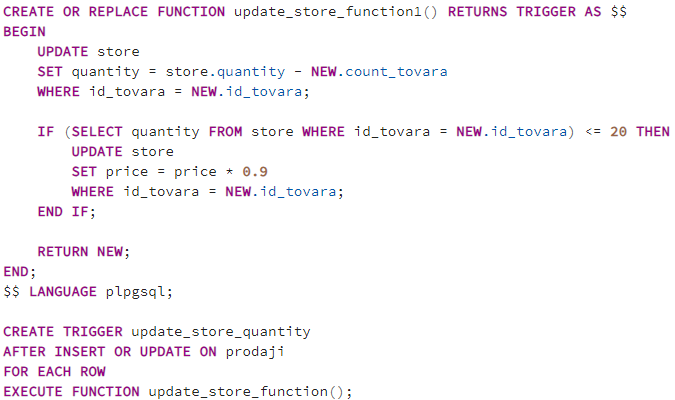
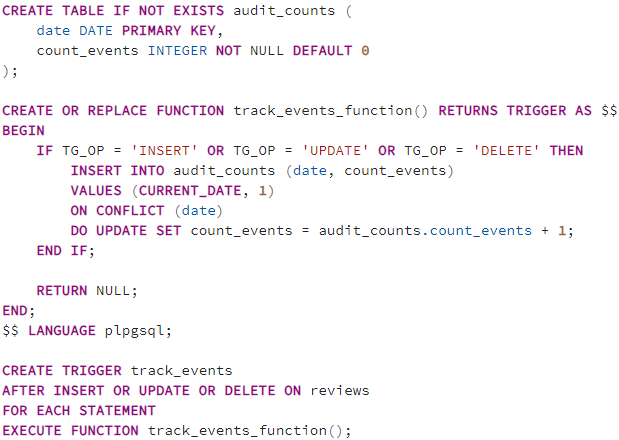


Рисунок 22 – Триггер на обновление после продажи

На рисунке 23 представлен пример триггера, который позволяет автоматически отслеживать количество событий вставки, обновления и удаления из таблицы reviews для каждой даты. Это может быть полезно, например, для мониторинга активности в базе данных или для отслеживания изменений в определенной таблице.

Рисунок 23 – Триггер на аудит событий вставки, обновления и удаления

Триггер на рисунке 24 позволяет автоматически отслеживать удаленные строки из таблицы prodaji в формате JSONB. Это может быть полезно, например, для отслеживания изменений в базе данных или для восстановления случайно удаленных данных.

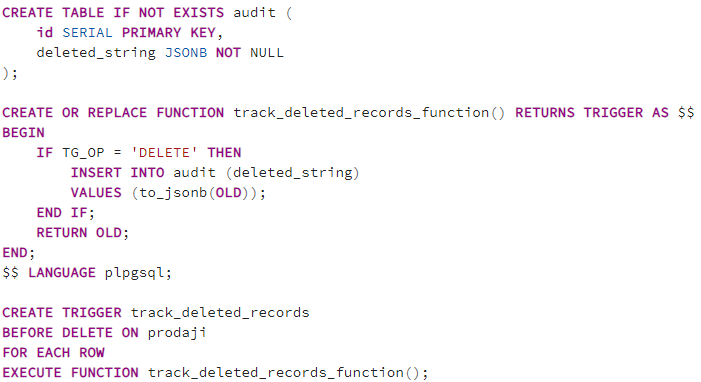


Рисунок 24 – Триггер на сохранение удаленных строк

Триггер на рисунке 25 позволяет автоматически шифровать поле email при вставке новой строки в таблицу pokupateli. Это может быть полезно для защиты конфиденциальных данных, таких как адреса электронной почты, при хранении их в базе данных. Данный триггер позволяет реализовать функции шифрования, которое требуется при регистрации нового пользователя, как представлено на рисунке 1.

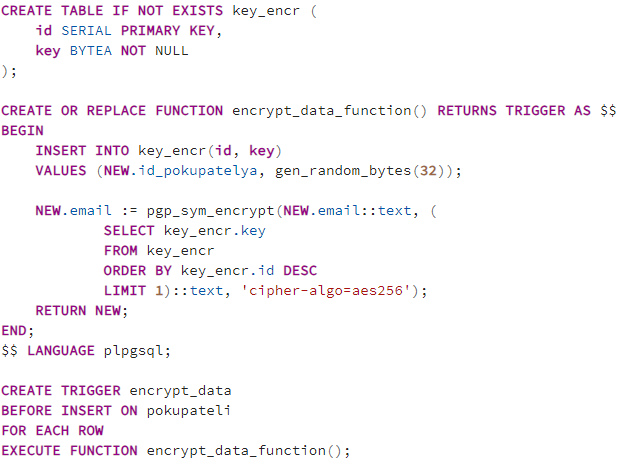


Рисунок 25 – Триггер на шифрование данных

Триггер на рисунке 26 выполняет проверку данных столбца «quantity» при новой продаже, чтобы количество проданных товаров не было больше, чем количество текущих товаров на складе. Данный триггер необходим для проверке целостности данных при учете поставок, как на рисунке 3.

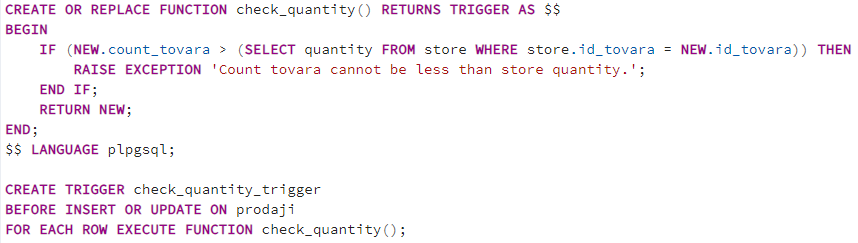


Рисунок 26 – Триггер на проверку количества поставленного товара

Рисунок 27 представляет собой триггер, который проверяет, что значение столбца count\_tovara в таблице postavka является положительным целым числом. Функция триггера check\_quantity\_fn() определена с использованием языка PL/pgSQL. В этой функции условие IF NEW.count\_tovara <= 0 проверяет, что значение count\_tovara не отрицательное и не равно нулю. Если это условие выполняется, то генерируется исключение с сообщением об ошибке «Количество товара должно быть больше 0». Если условие не выполняется, то функция возвращает новое значение строки с помощью команды RETURN NEW. Триггер check\_quantity\_trg вызывает функцию check\_quantity\_fn() перед вставкой или обновлением строки в таблице postavka.

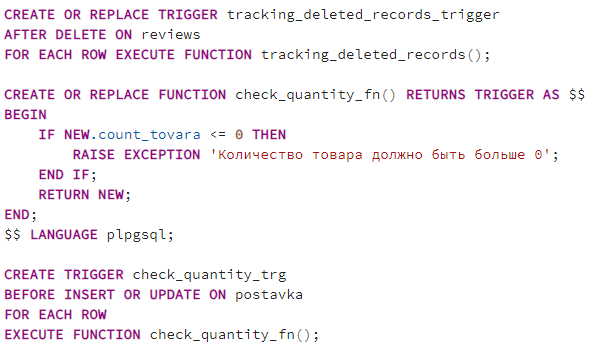


Рисунок 27 – Триггер на проверку целого числа

Рисунок 28 представляет собой триггер, который проверяет, что значение столбца date\_postavki в таблице postavka не превышает текущую дату и время. Функция триггера check\_date\_postavki\_fn() определена с использованием языка PL/pgSQL. В этой функции условие IF NEW.date\_postavki > CURRENT\_TIMESTAMP проверяет, что значение date\_postavki не больше текущей даты и времени. Если это условие выполняется, то генерируется исключение с сообщением об ошибке «Дата поставки не может быть больше текущей даты». Если условие не выполняется, то функция возвращает новое значение строки с помощью команды RETURN NEW. Триггер check\_date\_postavki\_trg вызывает функцию check\_date\_postavki\_fn() перед вставкой или обновлением строки в таблице postavka. Данный триггер необходим для целостности данных при реализации бизнес-процесса «Учет поставок», который представлен на рисунке 3.

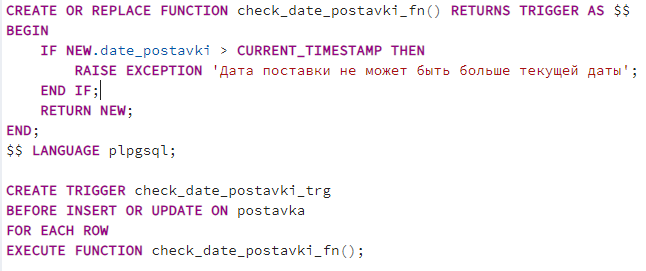


Рисунок 28 – Триггер на проверку даты поставки

На рисунке 29 представлен триггер, который проверяет, что значение столбца production\_date в таблице postavka не больше текущей даты и времени и не больше даты поставки. Функция триггера check\_date\_production\_fn() определена с использованием языка PL/pgSQL. В этой функции условие IF NEW.production\_date > CURRENT\_TIMESTAMP OR NEW.production\_date > NEW.date\_postavki проверяет, что значение production\_date не больше текущей даты и времени и не больше даты поставки. Если это условие выполняется, то генерируется исключение с сообщением об ошибке «Дата производства не может быть больше текущей даты или даты поставки». Если условие не выполняется, то функция возвращает новое значение строки с помощью команды RETURN NEW. Триггер check\_date\_production\_trg вызывает функцию check\_date\_production\_fn() перед вставкой или обновлением строки в таблице postavka. Данный триггер необходим для целостности данных при реализации бизнес-процесса «Учет поставок», который представлен на рисунке 3.

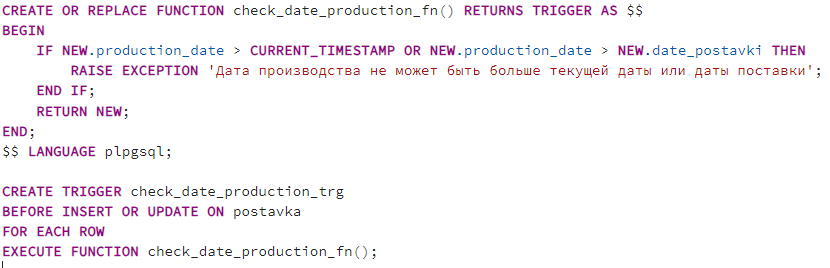


Рисунок 29 – Триггер на проверку даты производства

На рисунке 30 представлен триггер, который проверяет, что значение столбца expiry\_date в таблице postavka не меньше даты производства и не больше текущей даты и времени. Функция триггера check\_expiry\_date\_fn() определена с использованием языка PL/pgSQL. В этой функции условие IF NEW.expiry\_date < NEW.production\_date OR NEW.expiry\_date > CURRENT\_TIMESTAMP проверяет, что значение expiry\_date не меньше даты производства и не больше текущей даты и времени. Если это условие выполняется, то генерируется исключение с сообщением об ошибке "Срок годности не может быть меньше даты производства или больше текущей даты". Если условие не выполняется, то функция возвращает новое значение строки с помощью команды RETURN NEW. Триггер check\_expiry\_date\_trg вызывает функцию check\_expiry\_date\_fn() перед вставкой или обновлением строки в таблице postavka. Данный триггер необходим для целостности данных при реализации бизнес-процесса «Учет поставок», который представлен на рисунке 3.

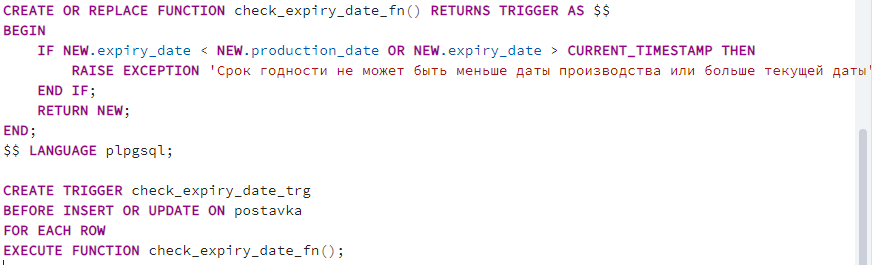


Рисунок 30 – Триггер на проверку срока годности

Использование триггеров в базе данных ИС «Учет товара на складе» позволяет автоматизировать работу с данными, улучшить безопасность информации и снизить риск ошибок при ручном вводе данных. Это в свою очередь позволяет улучшить качество работы ИС и повысить эффективность работы сотрудников склада.

* 1. **Реализация клиентской части АС**

В качестве языка программирования для разработки клиентской части был выбран язык программирования Python. Это обосновано его простотой, гибкостью и наличием большого количества библиотек для решения различных задач. Для создания графического интерфейса использовалась библиотека PyQt5 [19], которая предоставляет широкие возможности для создания интуитивно понятного и функционального интерфейса.

Для связи с базой данных в качестве инструмента выбран библиотека psycopg2. Это обосновано тем, что psycopg2 является одной из самых популярных и стабильных библиотек для работы с базами данных PostgreSQL в Python. Она предоставляет широкий функционал для выполнения запросов к базе данных, обработки результатов и управления транзакциями. Кроме того, psycopg2 имеет хорошую документацию и поддержку сообщества, что облегчает процесс разработки и отладки приложений. Существует несколько альтернативных библиотек для работы с базами данных PostgreSQL в Python, таких как pg8000, asyncpg. Однако psycopg2 остается одной из самых популярных и стабильных библиотек для этих целей. В то же время, некоторые альтернативные библиотеки могут иметь свои преимущества в определенных ситуациях. Например, asyncpg может быть более подходящим выбором для асинхронных приложений, так как он предоставляет асинхронный интерфейс для работы с базами данных PostgreSQL. pg8000 может быть более подходящим выбором для приложений, требующих высокой производительности, так как он полностью написан на C и имеет более низкие накладные расходы по сравнению с psycopg2.

Python является интерпретируемым, объектно-ориентированным языком программирования высокого уровня с динамической семантикой. Встроенные данные высокого уровня, динамическая типизация и динамическое связывание делают его очень привлекательным для быстрой разработки приложений, а также для использования в качестве скриптового или «клейкого» языка для связи существующих компонентов. Простой, легко изучаемый синтаксис Python подчеркивает читаемость и, следовательно, снижает затраты на обслуживание программ. Python поддерживает модули и пакеты, что способствует модульности программы и повторному использованию кода. Интерпретатор Python и обширная стандартная библиотека доступны в исходном или двоичном виде бесплатно для всех основных платформ и могут свободно распространяться [5].

PyQt — это библиотека Python для создания приложений с графическим интерфейсом с помощью инструментария Qt. Созданная в Riverbank Computing, PyQt является свободным ПО (по лицензии GPL) и разрабатывается с 1999 года. Последняя версия PyQt6 — на основе Qt 6 — выпущена в 2021 году, и библиотека продолжает обновляться [6].

В качестве библиотеки для подключения к базе данных выбрана psycopg2 [20], потому что библиотека предоставляет широкий набор функций и методов для работы с базой данных, включая поддержку транзакций, курсоров, запросов и других возможностей PostgreSQL. Это позволяет разработчику реализовывать сложные запросы и операции с данными.

На рисунке 31 отображается окно авторизации, через которое пользователь может войти в систему, введя свои логин и пароль.

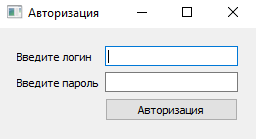


Рисунок 31 – Окно авторизации

На рисунке 32 показана неудачная попытка входа в систему.

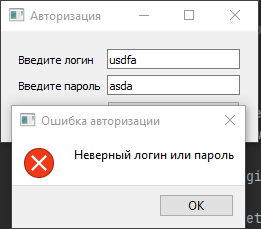
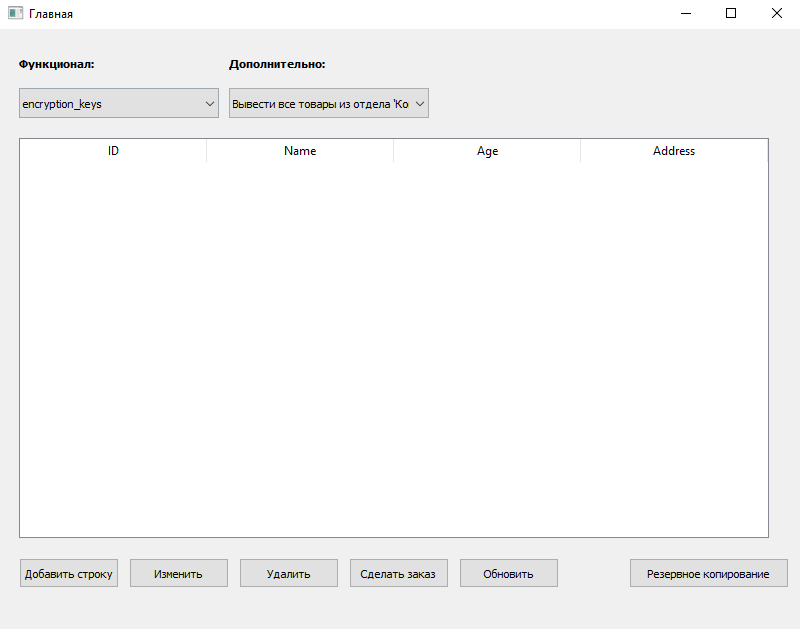


Рисунок 32 – Неудачная попытка входа

На рисунке 33 показан основной интерфейс программы, который содержит меню, панель инструментов и область просмотра данных.

  
Рисунок 33 - Основное окно

Выбор Python, PyQt5 и psycopg2 позволил реализовать все необходимые функции для работы с базой данных, включая поддержку транзакций, курсоров, запросов и других возможностей PostgreSQL.

## **2.4. Реализация основных методов и средств обеспечения безопасности данных**

Роль базы данных может иметь атрибуты, определяющие её полномочия и взаимодействие с системой аутентификации клиентов [12]. Для обеспечения безопасности при помощи ролей созданы следующие пользователи.

1. Покупатели могут просматривать данные о своих покупках, включая название, цену, количество и дату покупки.
2. Продавцы могут добавлять информацию о поставках, когда товара остается менее 10 штук, чтобы своевременно пополнять товары на прилавках и предотвращать ситуации, когда товар заканчивается.
3. Администратор может всё, удалять, добавлять и обновлять записи в любой таблице, создавать резервное копирование, просматривать первые 10% записей из таблицы отзывов, удалять функции (процедуры), чтобы управлять функционалом базы данных и оптимизировать ее работу.

Реализация ролевой политики представлена на рисунке 34, где созданы 3 группы пользователей и 3 пользователя и выданы права пользования таблицами.

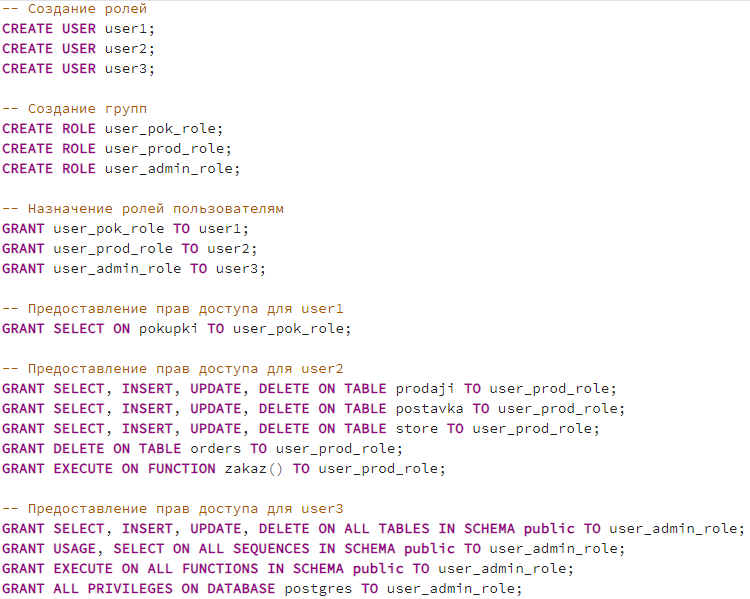


Рисунок 34 – Создание ролей и прав доступа

Как и всё, что содержит важные данные, базы данных PostgreSQL следует регулярно сохранять в резервной копии [10]. С помощью утилиты pg\_dump существует возможность создать целостный бэкап одной базы данных PostgreSQL, причем в процессе копирования можно продолжать работу с БД. В готовом дампе будет храниться только база данных — глобальные объекты (роли или табличные пространства) нужно сохранять с помощью других программ [9].

Для реализации резервного копирования была использована библиотека subprocess, которая позволяет создавать новые процессы и подключаться к ним [11]. Фрагмент кода для реализации резервного копирования представлен на рисунке 35.

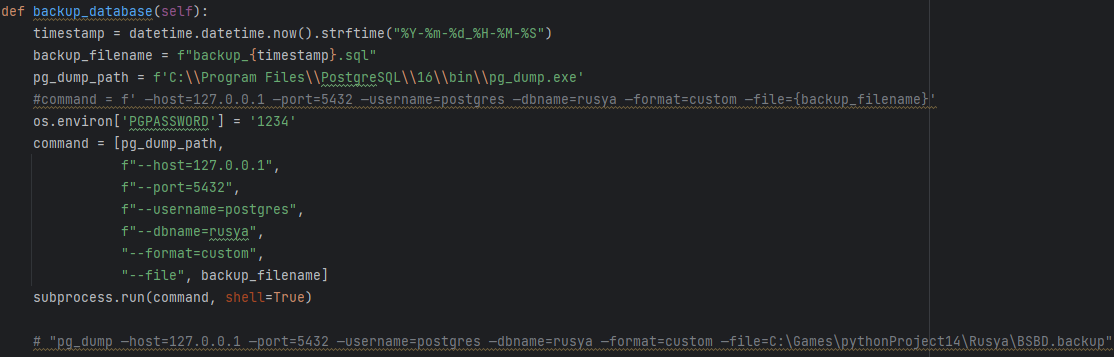


Рисунок 35 – Функция резервного копирования

На рисунке 36 показано успешное восстановление данных из резервной копии.

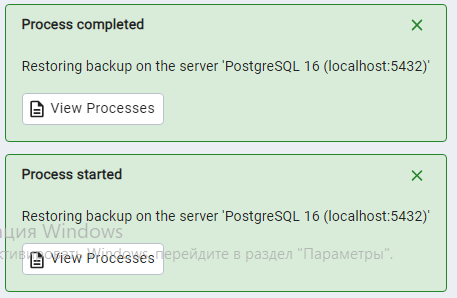


Рисунок 36 – Успешное восстановление данных

На рисунке 37 показано неудачное восстановление данных. Это может произойти в случае, если резервная копия была повреждена или удалена, или если структура исходной таблицы была изменена после создания резервной копии. В таких случаях необходимо восстанавливать данные из более ранней резервной копии или использовать другие методы восстановления данных.

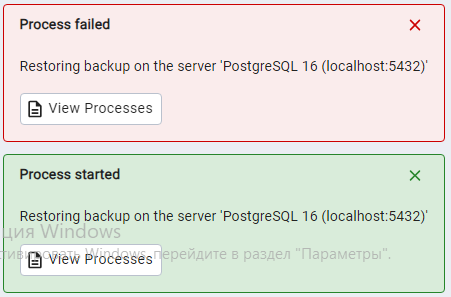


Рисунок 37 – Неудачное восстановление данных

Резервное копирование и восстановление данных являются важными задачами администрирования баз данных, и требуется тщательное планирование и тестирование для обеспечения надежности и безопасности данных. Кроме того, существует множество инструментов и методов резервного копирования и восстановления данных в PostgreSQL, и выбор подходящего метода зависит от конкретных требований и особенностей конкретной базы данных.

# **3.ДОКУМЕНТАЦИЯ НА ИС «Учет товара на складе»**

## **3.1**. **Руководство по использованию ИС «Учет товара на складе» для покупателя**

Прежде всего покупателю нужно авторизоваться в системе, для этого необходимо ввести учетные данных в окна ввода, представленные на рисунке 38.

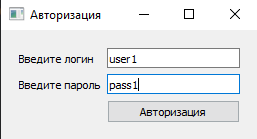


Рисунок 38 - Авторизация покупателя

После авторизации у него открывается главное окно, представленное на рисунке 39. Покупатели ничего не могут изменять в системе. Они могут только просматривать свои покупки через представление pokupki. Также можно сортировать по убыванию, возрастанию при нажатии на название столбца.

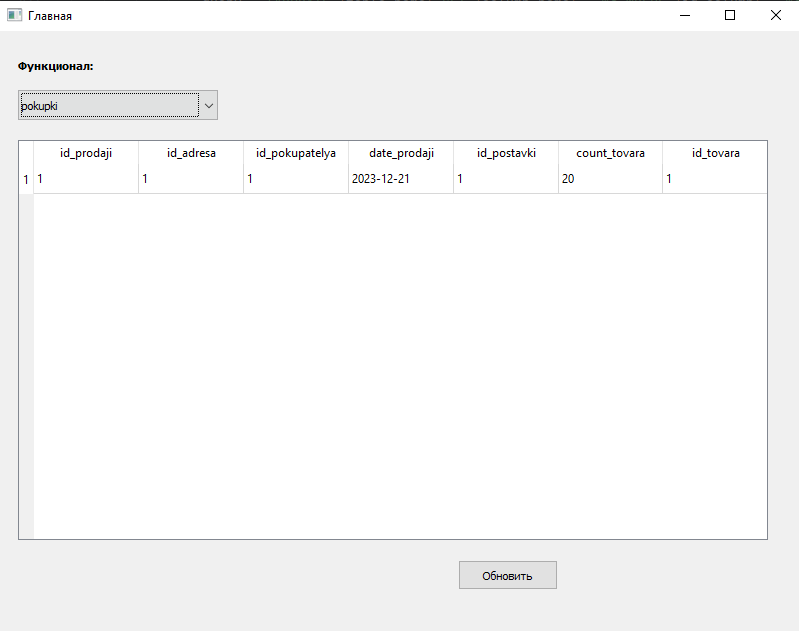


Рисунок 39 - Функционал покупателя

Функционал покупателя ограничен только просмотром и сортировкой информации, изменения в системе им производить нельзя. Это обеспечивает безопасность данных и предотвращает несанкционированные изменения.

* 1. **Руководство по использованию ИС «Учет товара на складе» для продавца**

Для начала работы с системой в качестве продавца необходимо внести учетные данные в поля ввода, как представлено на рисунке 40.

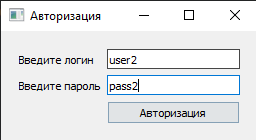
****

Рисунок 40 - Авторизация продавца

После успешной авторизации открывается доступ к таблицам «заказы», «поставка», «продажи», «склад». Он может добавлять данные, изменять, удалять, а также выполнять процедуру, которая добавляет в таблицу «заказы» записи товаров, которых на складе меньше установленной нормы. На рисунке 41 представлен просмотр таблицы «склад».

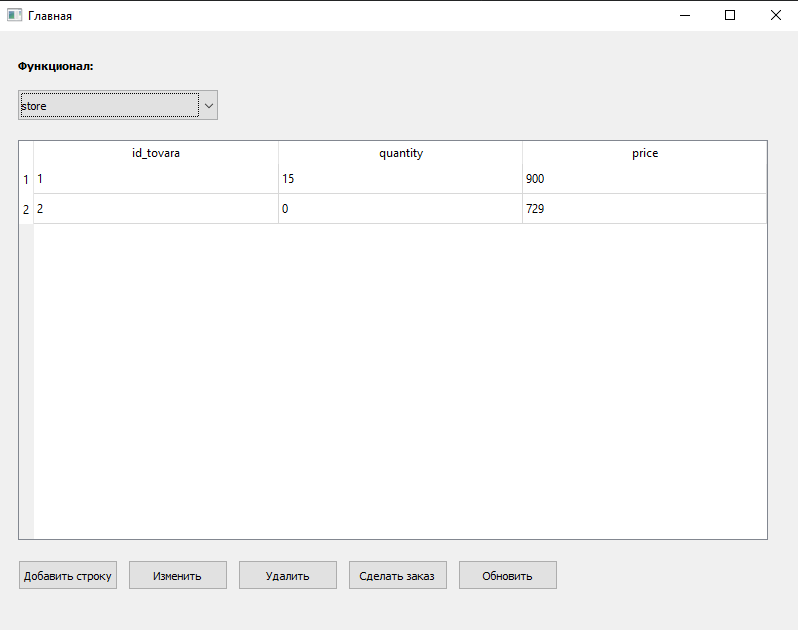
****

Рисунок 41 - Просмотр таблицы «Склад»

Для показа функционала, на рисунке 41 видно, что товар с id = 2 на складе отсутствует, соответственно после нажатия на кнопку «Сделать заказ» появляется окно как на рисунке 42 и заносится новая запись в таблицу «Заказы» с id товара = 2 и количеством 100. Результат представлен на рисунке 43. При условии, что нет нужды пополнять запасы товаров при нажатии на кнопку «Сделать заказ» появится ошибка как показано на рисунке 44.

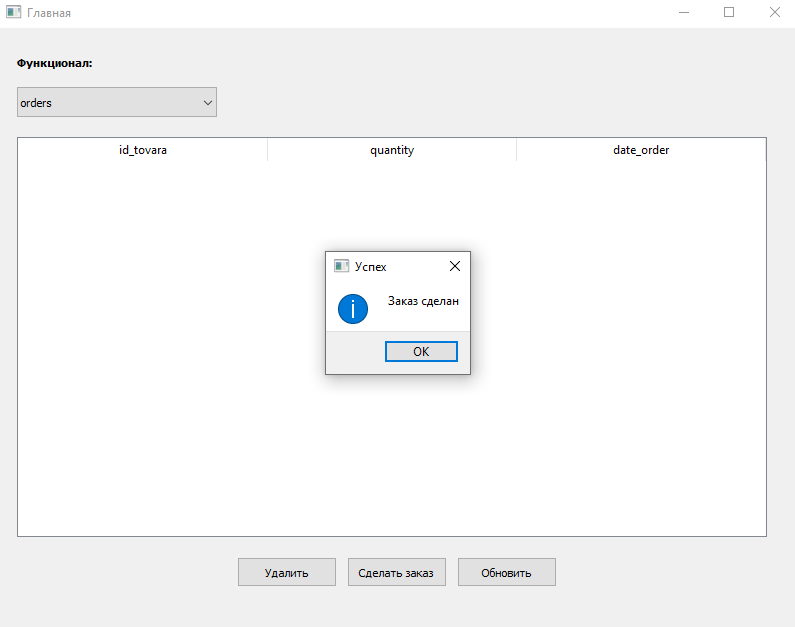


Рисунок 42 - Выполнение процедуры «Заказ на пополнение»

Рисунок 43 показывает результат выполнения процедуры «Заказ на пополнение». После нажатия на кнопку «ОК» в окне, показанном на рисунке 36, в таблицу «Заказы» добавляется новая запись с id товара = 2 и количеством 100.

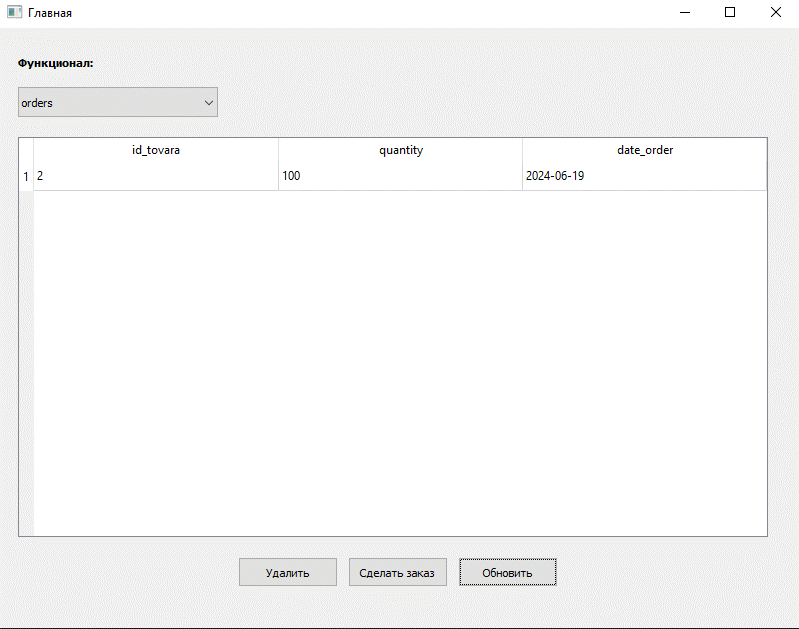


Рисунок 43–Результат выполнения процедуры

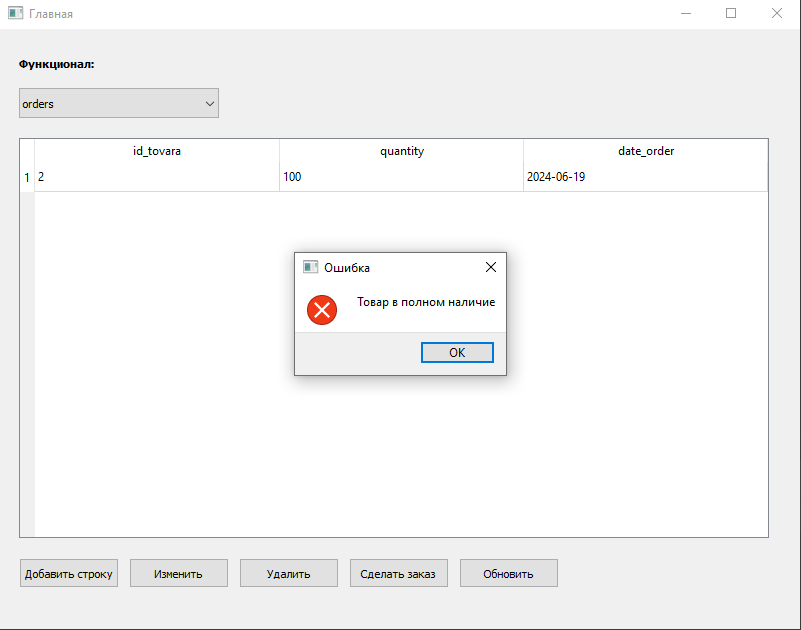


Рисунок 44 – Ошибка выполнения процедуры

Таким образом, ИС «Учет товара на складе» предоставляет продавцу необходимые инструменты для эффективного управления товарами на складе и контроля за их продажами.

## **3.3. Руководство по использованию ИС «Учет товара на складе» для администратора**

После авторизации к системе как администратор при помощи логина user3 и пароля pass3 нам открывается окно с полным функционалом, как показано на рисунке 45 и 46. В том числе он может видеть и такую важную информацию как ключи шифрования, для этого необходимо нажать на кнопку «Функционал» и выбрать таблицу «key\_encr», пример показан на рисунке 47.

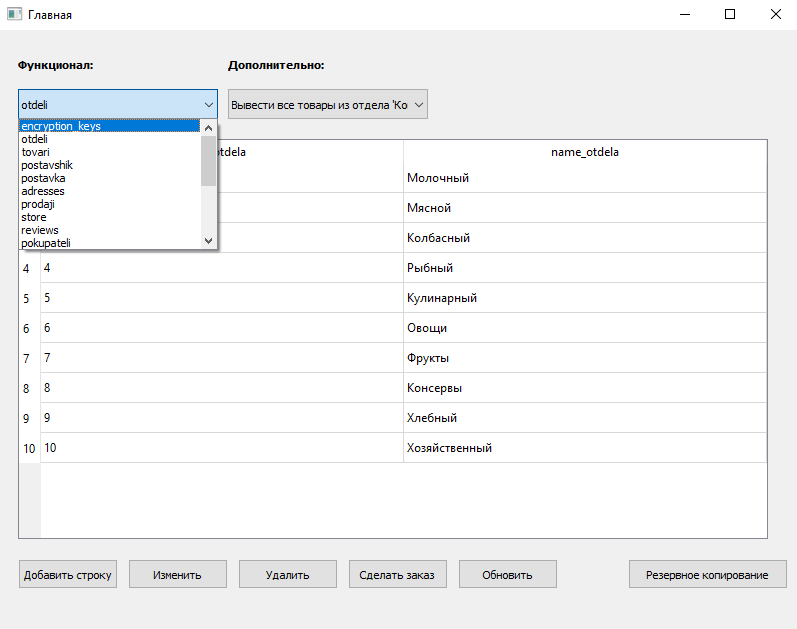


Рисунок 45 - Просмотр всех таблиц и представлений

Рисунок 46 показывает пример выполнений запросов. Чтобы выполнить запрос необходимо нажать кнопку «Дополнительно» и выбрать необходимый запрос. В итоге, администратор может увидеть результат выполнения запроса, а также отсортировать и отфильтровать данные по необходимым критериям.

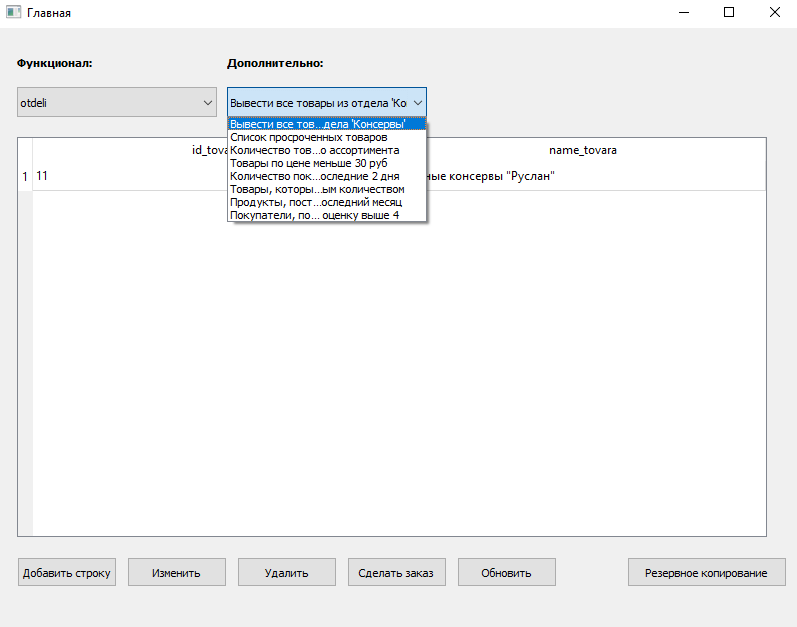


Рисунок 46 - Выполнение всех запросов на выборку

Рисунок 47 представляет собой пример таблицы с ключами шифрования, которая доступна только администратору. В таблице содержатся ключи, используемые для шифрования и дешифрования данных в базе данных. Они заполняются автоматически триггером, обеспечивающим шифрование данных при регистрации нового пользователя.

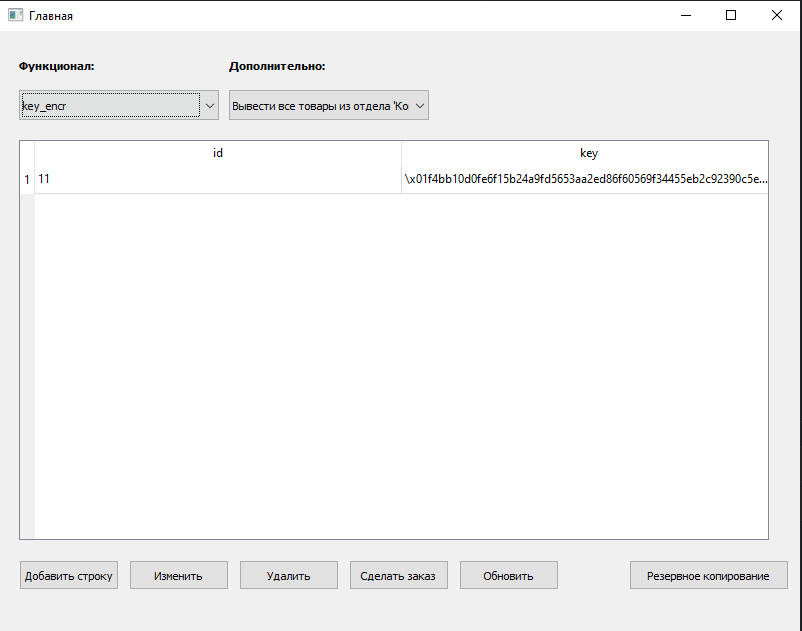


Рисунок 47 – Таблица с ключами

Также на рисунке 48 представлено то, как реализована функция добавления данных, а именно при нажатии кнопки «Добавить строку» появляется окно вставки новых данных с полями ввода под каждый столбец.

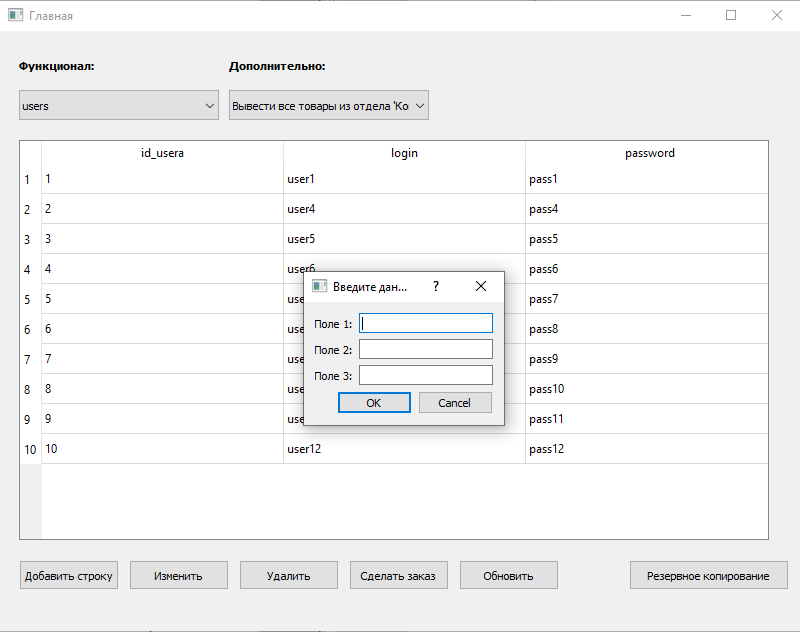


Рисунок 48 – Вставка данных

Данный функционал позволяет обеспечить работу системного администратора в БД.

# **Заключение**

Выполнено описание предметной области «Учет товара на складе», которое включает в себя бизнес-процессы, группы пользователей, а также информационные запросы, которые должны выполняться в информационной системе. Кратко охарактеризованы основные механизмы обеспечения безопасности в PostgreSQL 16. Это описание стало основой для дальнейшей разработки информационной системы.

Для реализации ИС использовались PostgreSQL 16 и ЯП Python. Спроектирована и разработана ИС «Учет товара на складе». Созданы таблицы базы данных, разработаны SQL-запросы для работы с данными, созданы формы и отчеты для пользователей. Реализована логика работы системы, предусмотренная бизнес-процессами и информационными запросами. Также разработана клиентская часть приложения, которая обеспечивает взаимодействие пользователей с базой данных. Для обеспечения безопасности данных реализованы основные методы и средства, такие как аутентификация, авторизация, шифрование, резервное копирование, аудит, политики безопасности и ролевая политика.

Для удобства использования разработанной информационной системы подготовлено руководство по ее использованию. Руководство содержит подробные инструкции по работе с приложением, а также описание функциональных возможностей системы. Разработаны руководства для различных типов пользователей: для покупателей, продавцов и администраторов.

Таким образом, задачи выполнены, цель достигнута.

# **Список литературы**

1. Безопасность баз данных: определение и объяснение // Microsoft Azure: сайт. – URL: <https://azure.microsoft.com/ru-ru/resources/cloud-computing-dictionary/what-is-database-security>(дата обращения: 31.05.2024).
2. Введение в резервное копирование баз данных PostgreSQL // Habr: сайт. – URL: <https://habr.com/ru/articles/550882/>(дата обращения: 31.05.2024).
3. Документация PostgreSQL 9.6: Резервное копирование и восстановление // PostgresPro: сайт. – URL: <https://postgrespro.ru/docs/postgresql/9.6/backup>(дата обращения: 31.05.2024).
4. Документация PostgreSQL 12: Параметры шифрования // PostgresPro: сайт. – URL: <https://postgrespro.ru/docs/postgresql/12/encryption-options>(дата обращения: 31.05.2024).
5. Что такое Python? // Python.org: сайт. – URL: <https://www.python.org/doc/essays/blurb/>(дата обращения: 31.05.2024).
6. Введение в Python для Data Science // Habr: сайт. – URL: <https://habr.com/ru/companies/skillfactory/articles/599599/>(дата обращения: 31.05.2024).
7. Документация PostgreSQL: Администрирование // PostgreSQL: сайт. – URL: <https://www.postgresql.org/docs/current/admin.html>(дата обращения: 31.05.2024).
8. Diagrams.net: онлайн-сервис для создания диаграмм // Diagrams.net: сайт. – URL: <https://app.diagrams.net/>(дата обращения: 31.05.2024).
9. Резервное копирование и восстановление баз данных PostgreSQL // REG.RU: сайт. – URL: <https://help.reg.ru/support/servery-vps/oblachnyye-servery/rabota-s-serverom/rezervnoye-kopirovaniye-i-vosstanovleniye-baz-dannykh-postgresql#1>(дата обращения: 31.05.2024).
10. Документация PostgreSQL 9.6: Резервное копирование и восстановление // PostgresPro: сайт. – URL: <https://postgrespro.ru/docs/postgresql/9.6/backup>(дата обращения: 31.05.2024).
11. Документация Python: Модуль subprocess // Python.org: сайт. – URL: <https://docs.python.org/3/library/subprocess.html>(дата обращения: 31.05.2024).
12. Документация PostgreSQL 9.6: Атрибуты ролей // PostgresPro: сайт. – URL: <https://postgrespro.ru/docs/postgresql/9.6/role-attributes>(дата обращения: 31.05.2024).
13. Документация PostgreSQL 9.4: Определение триггеров // PostgresPro: сайт. – URL: <https://postgrespro.ru/docs/postgresql/9.4/trigger-definition>(дата обращения: 31.05.2024).
14. Виды и типы SQL-запросов // Higher School of Economics: сайт. – URL: <https://hsbi.hse.ru/articles/vidy-i-tipy-sql-zaprosov/>(дата обращения: 31.05.2024).
15. Атрибуты данных // ГБ: сайт. – URL: <https://gb.ru/blog/atributy-dannyh/>(дата обращения: 31.05.2024).
16. PostgreSQL: Главная страница // PostgreSQL: сайт. – URL: <https://www.postgresql.org/>(дата обращения: 31.05.2024).
17. Атрибуты данных // GB: сайт. – URL: https://gb.ru/blog/atributy-dannyh/ (дата обращения: 31.05.2024).
18. BPMN.io: онлайн-редактор BPMN-диаграмм // BPMN.io: сайт. – URL: <https://bpmn.io/>(дата обращения: 31.05.2024).
19. PyQt5: библиотека для создания GUI на Python // PyPI: сайт. – URL: <https://pypi.org/project/PyQt5/>(дата обращения: 31.05.2024).
20. psycopg2: драйвер для работы с PostgreSQL на Python // PyPI: сайт. – URL: <https://pypi.org/project/psycopg2> (дата обращения: 31.05.2024).
21. MaxPatrolSIEM выявляет атаки на СУБД PostgreSQL // PTSecurity: сайт. – URL: <https://www.ptsecurity.com/ru-ru/about/news/maxpatrol-siem-vyyavlyaet-ataki-na-subd-postgresql/>(дата обращения: 31.05.2024).