



Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ

ИУ «ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ»

КАФЕДРА

ИУ7 «ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЭВМ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ»

КУРСОВАЯ РАБОТА

НА ТЕМУ:

*Разработка базы данных для хранения и
обработки данных цветочного магазина*

Студент

ИУ7-65Б

(группа)

(подпись, дата)

Федченко

(И.О. Фамилия)

Руководитель курсового
проекта

(подпись, дата)

Никульшина Т.А.

(И.О. Фамилия)

Консультант

(подпись, дата)

(И.О. Фамилия)

2025 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Аналитический раздел	4
1.1 Анализ предметной области. Сравнительный анализ существующих решений	4
1.2 Формулировка требований к разрабатываемой базе данных и приложению	6
1.3 Формализация и описание информации, подлежащей хранению в проектируемой базе данных	6
1.4 Проведенный анализ существующих баз данных на основе формализации данных	6
1.5 ER-диаграмма сущностей проектируемой базы данных в нотации Чена	7
1.6 Формализация и описание пользователей проектируемого приложения к базе данных	9
1.7 Диаграмма вариантов использования	9
Вывод	9
2 Конструкторский раздел	11
2.1 Диаграмма проектируемой базы данных	11
2.2 Описание сущностей проектируемой базы данных	11
2.3 Описание проектируемых ограничений целостности базы данных	15
2.4 Описание всех проектируемых процедур/функций/триггеров в формате схемы	15
2.5 Описание проектируемой ролевой модели на уровне базы данных	18
Вывод	18

3	Технологический раздел	19
3.1	Обоснование выбора средств реализации базы данных и приложения	19
3.2	Реализация	19
3.2.1	Создание таблиц	19
3.2.2	Создание триггеров	23
3.2.3	Создание ролей	26
3.3	Тестирование	27
3.4	Пример работы программы	30
	Вывод	33
4	Исследовательский раздел	34
	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	35

ВВЕДЕНИЕ

Данная работа посвящена проектированию и реализации базы данных и приложения для автоматизации работы цветочного магазина. Основное внимание уделяется разработке структуры базы данных, обеспечению целостности хранимых данных, а также созданию функционального приложения, позволяющего эффективно управлять товарными запасами, учетными записями пользователей и бизнес-процессами магазина.

Цель: создание надежной и масштабируемой базы данных и приложения для цветочного магазина, обеспечивающих автоматизацию ключевых операций.

Задачи:

- 1) сформулировать требования и ограничения к разрабатываемой базе данных и приложению;
- 2) определить роли и описание пользователей системы;
- 3) спроектировать сущности базы данных и ограничения целостности;
- 4) реализовать триггер для автоматического обновления товарных остатков при поступлении новых партий;
- 5) выбрать средства реализации базы данных и приложения;
- 6) разработать и реализовать сущности базы данных с учетом ограничений целостности;
- 7) описать методы тестирования функционала и разработать тестовые сценарии;
- 8) провести исследование производительности запросов в зависимости от объема данных с использованием индексов и без них.

1 Аналитический раздел

1.1 Анализ предметной области. Сравнительный анализ существующих решений

Общая характеристика предметной области

Цветочный бизнес — это динамично развивающаяся отрасль, включающая продажу свежих цветов, комнатных растений, сопутствующих товаров (упаковка, открытки, удобрения) и дополнительные услуги (доставка). Современные цветочные магазины стремятся автоматизировать процессы учета товаров, управления заказами и взаимодействия с клиентами, что требует эффективных IT-решений.

Ключевые процессы, подлежащие автоматизации:

- управление ассортиментом (учет цветов, растений, упаковки);
- контроль остатков на складе;
- оформление и отслеживание заказов (включая доставку);
- управление клиентской базой и маркетинговыми акциями;
- аналитика продаж и формирование отчетности;

Сравнительный анализ существующих решений

На рынке представлено несколько типовых решений для автоматизации работы цветочных магазинов:

1) Ручные системы учета

Наиболее простой вариант, включающий использование Excel и бумажных журналов. Основные недостатки: высокая вероятность ошибок при вводе данных и сложности при расширении бизнеса.

2) Универсальные CRM-платформы

Такие системы как 1С [1] и Битрикс24 предлагают широкий функционал, но их внедрение требует существенных финансовых вложений и часто включает невостребованные возможности.

3) Отраслевые программные продукты

Специализированные решения типа FloristWare и FlowerShop Pro созданы специально для флористики, но имеют ограниченную гибкость настройки

и высокую стоимость лицензий.

Как показано в таблице 1.1, сравнительный анализ существующих решений демонстрирует, что ручные методы учета неэффективны при росте бизнеса из-за высокого риска ошибок и сложности масштабирования. Универсальные CRM-системы (1С, Битрикс24), хотя и предлагают широкий функционал, оказываются избыточными для малого цветочного магазина, требуя значительных затрат на адаптацию. Специализированные решения (FloristWare, FlowerShop Pro), разработанные для флористического бизнеса, часто не покрывают все специфические потребности магазинов и отличаются высокой стоимостью лицензирования.

Таблица 1.1 — Сравнительный анализ функциональных возможностей систем управления для цветочного бизнеса

Критерий	Microsoft Excel	1С:Предприятие	FloristWare Pro
Автоматизация учета	Ручной ввод данных, высокая вероятность ошибок	Полная автоматизация всех учетных операций	Специализированная автоматизация для флористики
Управление складом	Базовый учет остатков без аналитики	Полноценный складской учет с аналитикой сроков годности	Контроль остатков с напоминаниями о пополнении
Обработка заказов	Текстовые записи в ячейках таблицы	Интегрированная CRM-система с историей заказов	Оптимизированный интерфейс для быстрого оформления
Отчетность	Ручное составление отчетов по шаблонам	Встроенные отчетные формы с аналитикой	Стандартные отчеты по специфике цветочного бизнеса
Интеграции	Ограниченные возможности	Широкие возможности интеграции	Специализированные API для флористики
Стоимость	Включена в пакет Office	Лицензия + внедрение)	Подписка

1.2 Формулировка требований к разрабатываемой базе данных и приложению

Функциональные требования:

- управление товарами (хранение информации о цветах и сопутствующих товарах, учет остатков на складе, обновление количества при поступлении);
- управление заказами (регистрация заказов, учет клиентов и истории заказов);
- управление поставщиками (ведение списка поставщиков, учет поступлений товаров).

1.3 Формализация и описание информации, подлежащей хранению в проектируемой базе данных

Требования к БД:

- Сущности:
 - Товары (id, название, цена, количество)
 - Поставщики (id, название, контакты)
 - Пользователи (id, имя, телефон)
 - Поставки товаров (id, товары, дата, ответственный)
 - Заказы (id, клиент_id, статус, ответственный)
 - Склады (id, название, адрес)
- Триггеры:
 - автообновление остатков при поставках.

1.4 Проведенный анализ существующих баз данных на основе формализации данных

В таблице 1.2 представлено сравнение трёх моделей хранения данных: **SQL** (реляционная), **NoSQL** (документная) [2] и **Ключ-Значение** [3], проведенное на основе анализа структуры проектируемой базы данных.

Таблица 1.2 — Сравнение моделей хранения данных

Критерий	SQL (PostgreSQL, MySQL)	NoSQL (MongoDB)	Ключ- Значение (Redis)
Структура	Жёсткая схема (таблицы)	Гибкая (JSON- документы)	Просто ключ + значение
Связи	JOIN, внешние ключи	Вложенные до- кументы	Нет связей
Пример для Зака- за	Отдельные таблицы (Заказ, Контрагент)	Единый доку- мент с вложен- ными данными	order:123 → "JSON-данные"
Транзакции	Полная под- держка (ACID)	Ограниченная	Нет (или про- стые операции)
Масштабируемость	Вертикальное	Горизонтальное	Горизонтальное
Скорость	Оптимизирована для сложных запросов	Быстрые встав- ки/чтения	Максимальная скорость для простых опера- ций
Проблемы	Сложность мас- штабирования	Дублирование данных	Нет сложных за- просов

Для данной базы данных SQL является оптимальным выбором, так как:

- Данные имеют чёткую структуру и связи (например, «Партия → Номенклатура»)
- Требуются транзакции (например, при списании товаров)
- Необходимы сложные запросы с JOIN (анализ продаж, остатков)

NoSQL и Ключ-Значение не подходят для основной системы, но могут использоваться как дополнение.

Вывод: реляционная СУБД лучше всего соответствует требованиям данной системы.

1.5 ER-диаграмма сущностей проектируемой базы данных в нотации Чена

На рисунке 1.1 представлена ER-диаграмма сущностей проектируемой базы данных в нотации Чена для разрабатываемого приложения.

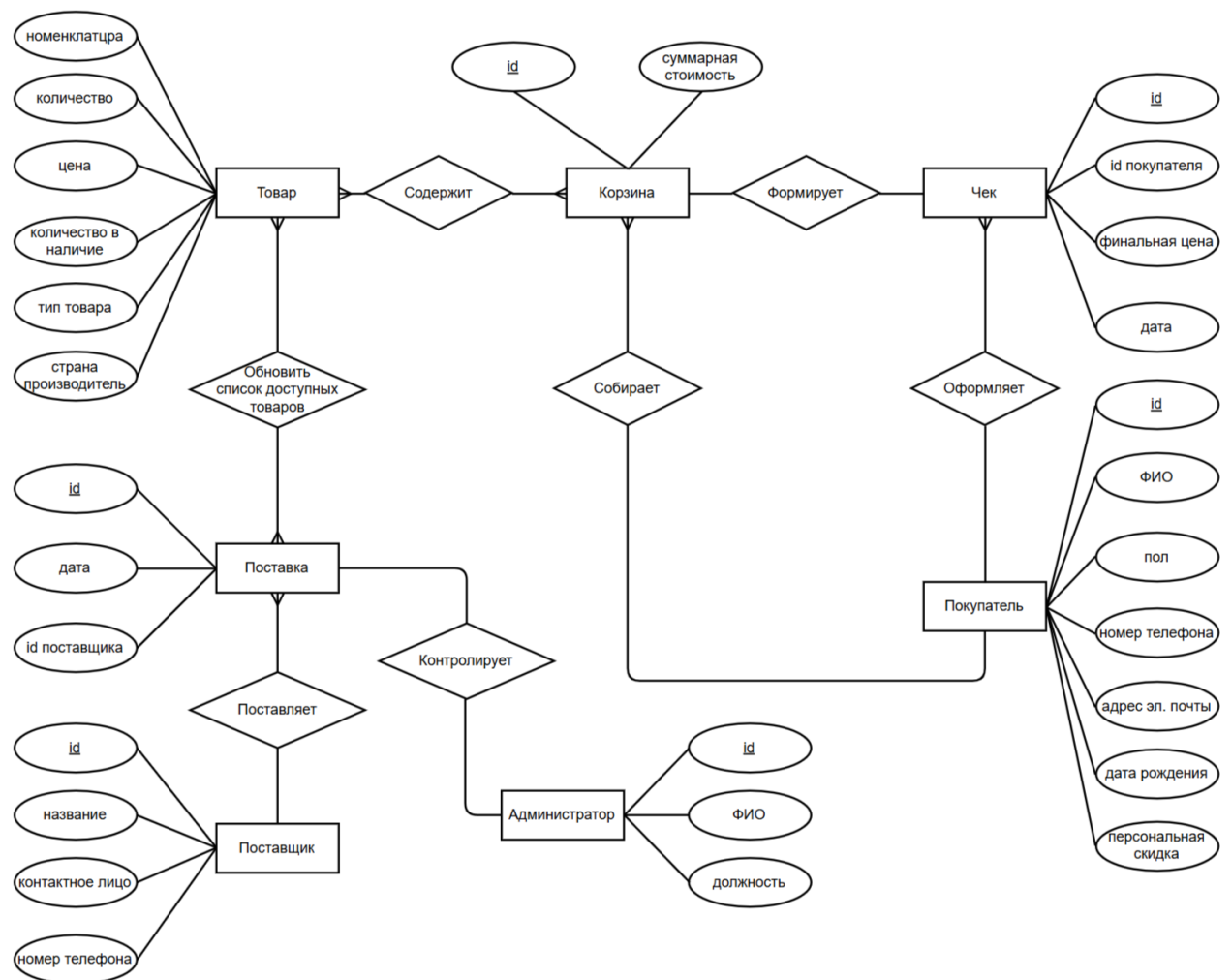


Рисунок 1.1 — ER-диаграмма сущностей проектируемой базы данных в нотации Чена

1.6 Формализация и описание пользователей проектируемого приложения к базе данных

Пользователи приложения:

- администратор (полный доступ);
- продавец (просмотр каталога товаров, работа с заказами);
- кладовщик (загрузка информации о новой партии товара, просмотр каталога товаров)

Роли БД:

- администратор;
- продавец;
- кладовщик;

1.7 Диаграмма вариантов использования

На рисунке 1.2 представлена диаграмма вариантов использования для разрабатываемого приложения.

Вывод

Проведённый анализ предметной области и сравнение существующих решений показали необходимость разработки специализированной системы автоматизации для цветочного бизнеса. На основе формализованных требований была спроектирована структура базы данных с чёткими связями между сущностями. Сравнительный анализ моделей хранения данных подтвердил преимущества реляционного подхода, обеспечивающего целостность данных и поддержку транзакций. В качестве СУБД выбрана PostgreSQL, как наиболее подходящая система благодаря её надежности, производительности и богатому функционалу для работы со сложными запросами. Полученные результаты стали основой для дальнейшего проектирования базы данных и разработки приложения.

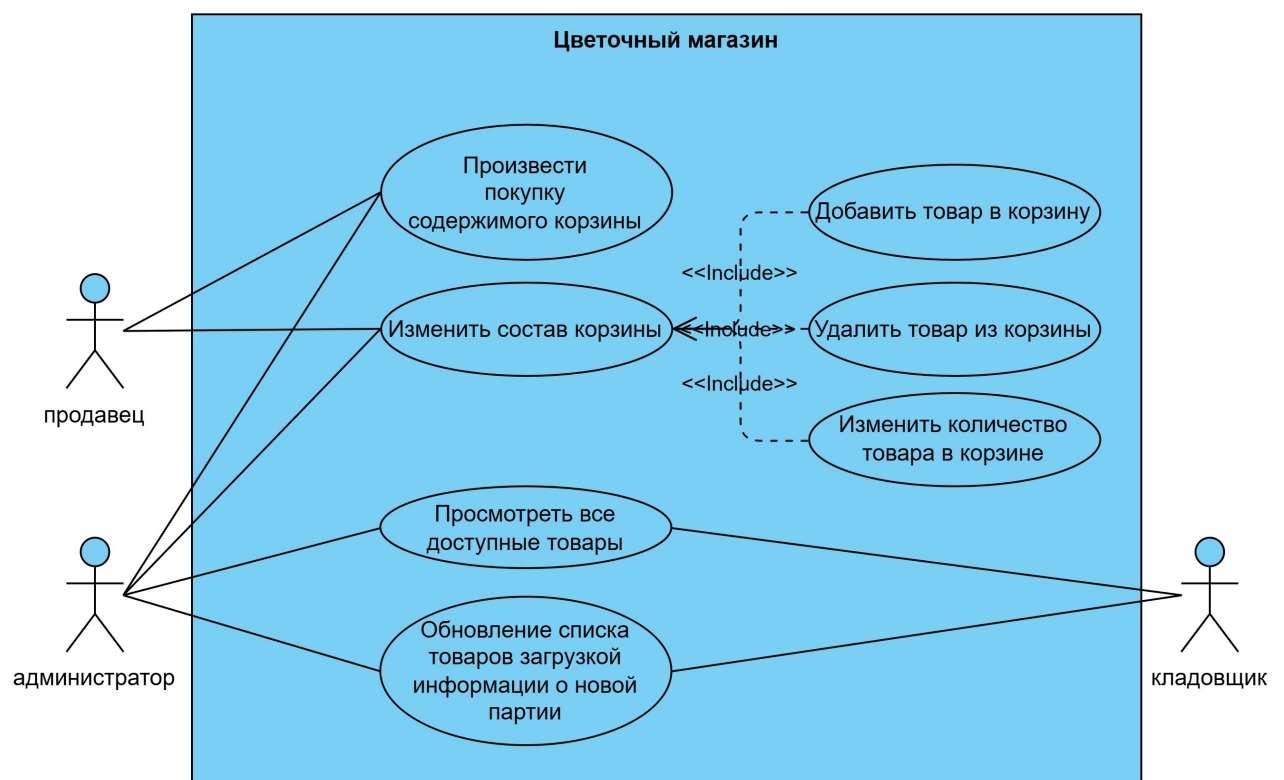


Рисунок 1.2 — Диаграмма вариантов использования

2 Конструкторский раздел

2.1 Диаграмма проектируемой базы данных

На рисунке 2.1 представлена диаграмма проектируемой базы данных.

2.2 Описание сущностей проектируемой базы данных

— Списание

- Основная сущность для учёта списания товаров
- Атрибуты:
 - * id - уникальный идентификатор
 - * номер партии
 - * номенклатура
 - * количество списываемого товара
 - * ответственный за списание (пользователь с ролью кладовщик)

— Пользователь

- Сущность для хранения данных пользователей системы
- Атрибуты:
 - * id - уникальный идентификатор
 - * название (имя пользователя)
 - * роль в системе

— Страна

- Справочник стран
- Атрибуты:
 - * id - уникальный идентификатор
 - * название страны

— Товар на складе

- Сущность учёта текущих остатков на складе
- Атрибуты:
 - * id - уникальный идентификатор
 - * номенклатура
 - * количество товара
 - * место хранения

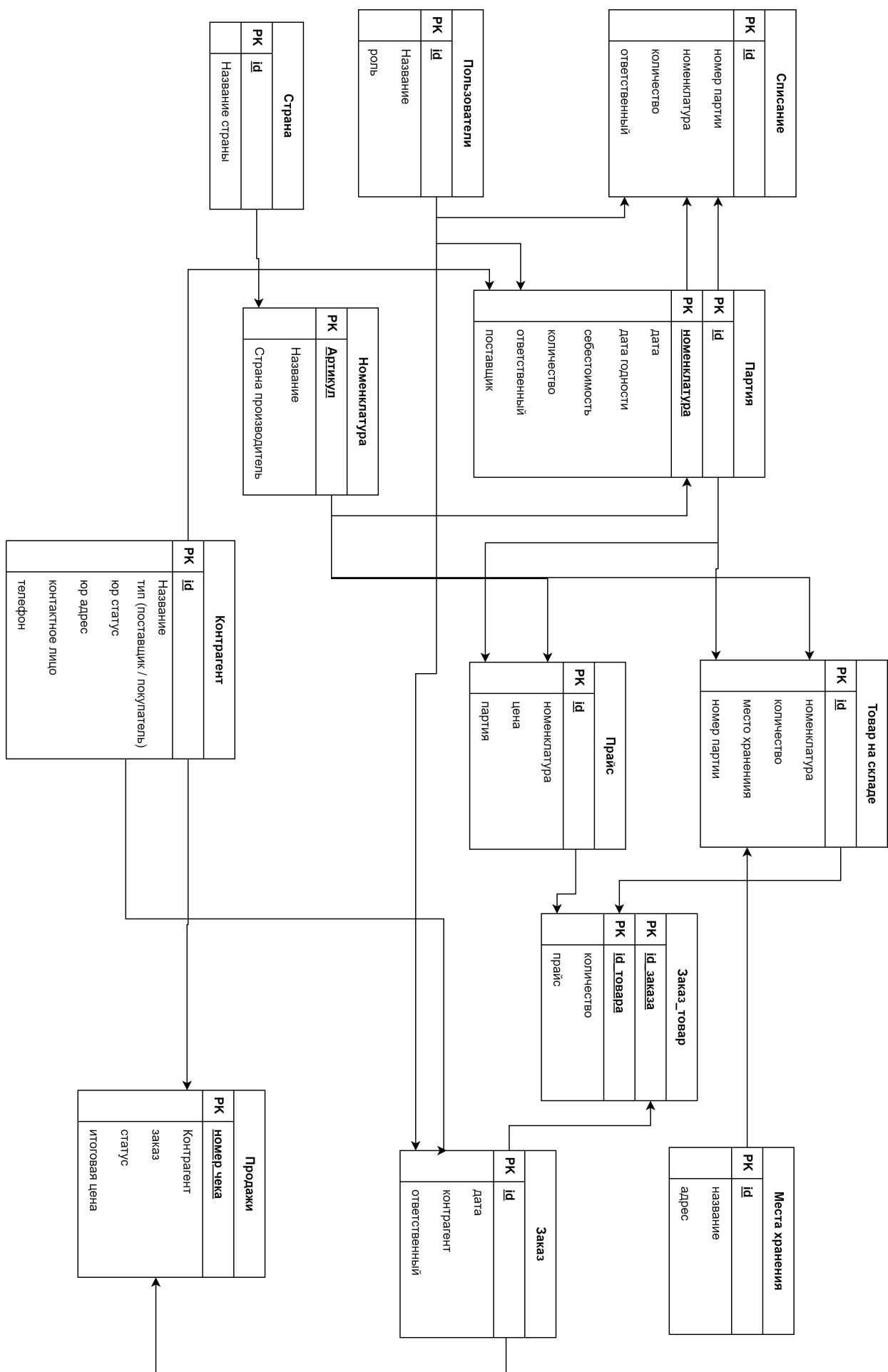


Рисунок 2.1 — Диаграмма проектируемой базы данных

- * номер партии

— **Заказ**

- Сущность для оформления заказов
- Атрибуты:
 - * id - уникальный идентификатор
 - * дата
 - * контрагент
 - * ответственный за заказ

— **Заказ товар**

- Сущность для сопоставления заказа и товаров
- Атрибуты:
 - * id заказа
 - * id товара
 - * количество
 - * прайс

— **Контрагент**

- Сущность для хранения данных контрагентов
- Атрибуты:
 - * id - уникальный идентификатор
 - * название
 - * тип (поставщик/покупатель)
 - * юридический статус
 - * юридический адрес
 - * контактное лицо
 - * телефон

— **Продажи**

- Сущность для учёта продаж
- Атрибуты:
 - * номер чека
 - * контрагент
 - * заказ
 - * статус продажи
 - * итоговая цена

— **Место хранения**

- Справочник складских помещений
- Атрибуты:
 - * id - уникальный идентификатор
 - * название места хранения
 - * адрес

— **Номенклатура**

- Основной справочник товаров/продукции
- Атрибуты:
 - * id - уникальный идентификатор
 - * название товара
 - * страна производства

— **Партия**

- Учёт поступлений товаров партиями
- Атрибуты:
 - * id - уникальный идентификатор
 - * дата
 - * дата годности
 - * себестоимость
 - * количество
 - * ответственный
 - * поставщик

— **Прайс**

- Справочник цен на товары
- Атрибуты:
 - * id - уникальный идентификатор
 - * номенклатура
 - * цена
 - * партия

2.3 Описание проектируемых ограничений целостности базы данных

В таблице 2.1 приведены ограничения целостности проектируемой базы данных.

Таблица 2.1 — Ограничения в таблицах базы данных

Таблица	Описание
Страна	Длина названия страны должна быть больше 0
Номенклатура	Длина названия номенклатуры должна быть больше 0
	Поле country_id не может быть NULL
Контрагент	Длина названия контрагента должна быть больше 0
	Длина имени контактного лица должна быть больше 0
	Телефон должен соответствовать формату (+ необязателен, затем от 10 до 15 цифр)
Пользователь	Длина имени пользователя должна быть больше 1
Партия товаров	Дата производства не может быть будущей
	Срок годности должен быть после даты производства
	Себестоимость должна быть положительной
	Количество должно быть положительным
Списание	Количество списания не может быть отрицательным
Цена	Цена продажи должна быть положительной
Место хранения	Длина названия места хранения должна быть больше 0
Товар на складе	Количество на складе не может быть отрицательным
Связь заказа и товара	Внешний ключ на таблицу product_in_stock
	Внешний ключ на таблицу order
Продажи	Итоговая цена должна быть положительной

2.4 Описание всех проектируемых процедур/функций/триггеров в формате схемы

На рисунках 2.2 и 2.3 представлены диаграммы алгоритмов триггера на обновление товара на складе при поступлении новой партии товара и триггера на обновление товара на складе при покупке, соответственно.

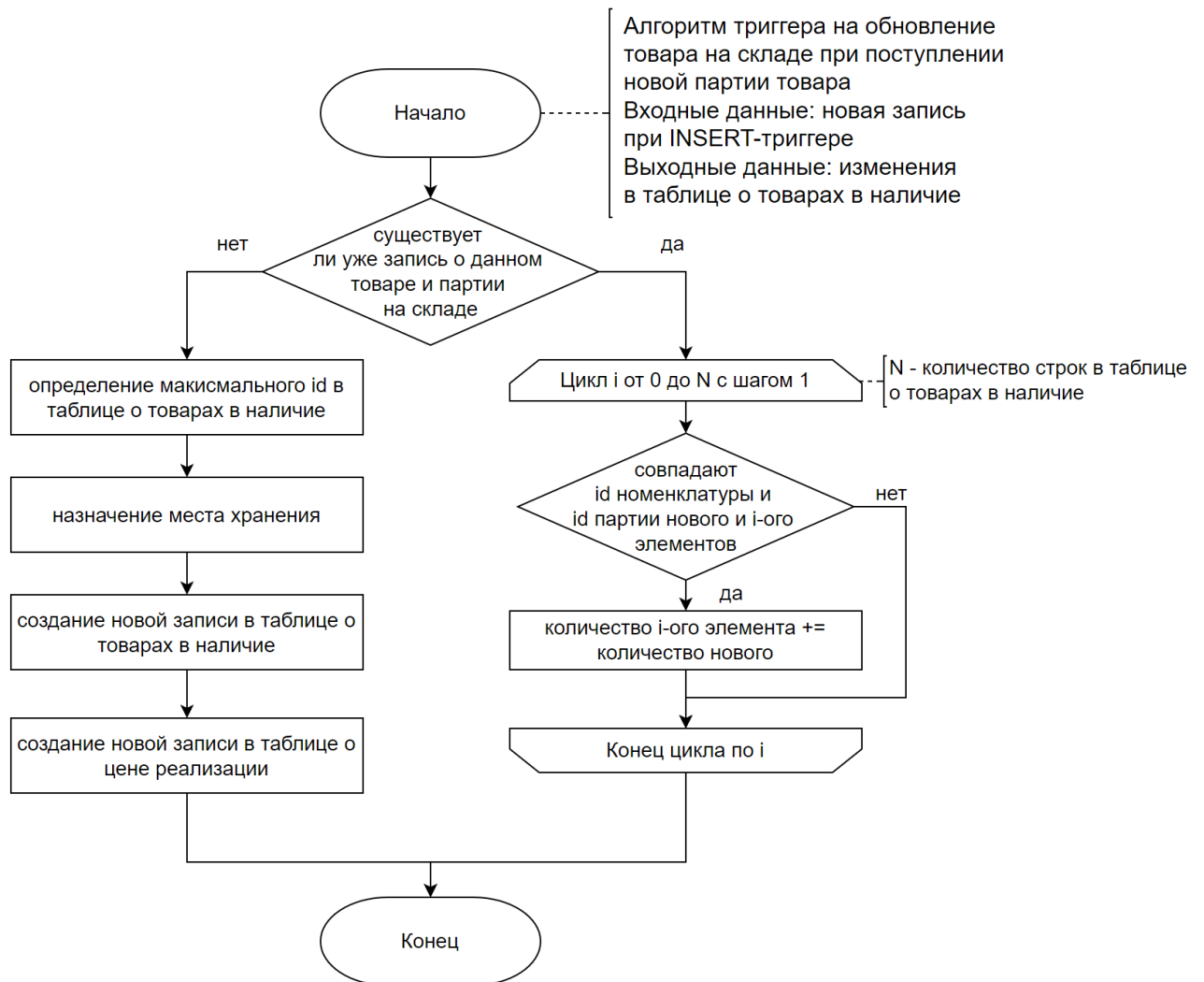


Рисунок 2.2 — Диаграмма алгоритма триггера на обновление товара на складе при поступлении новой партии товара

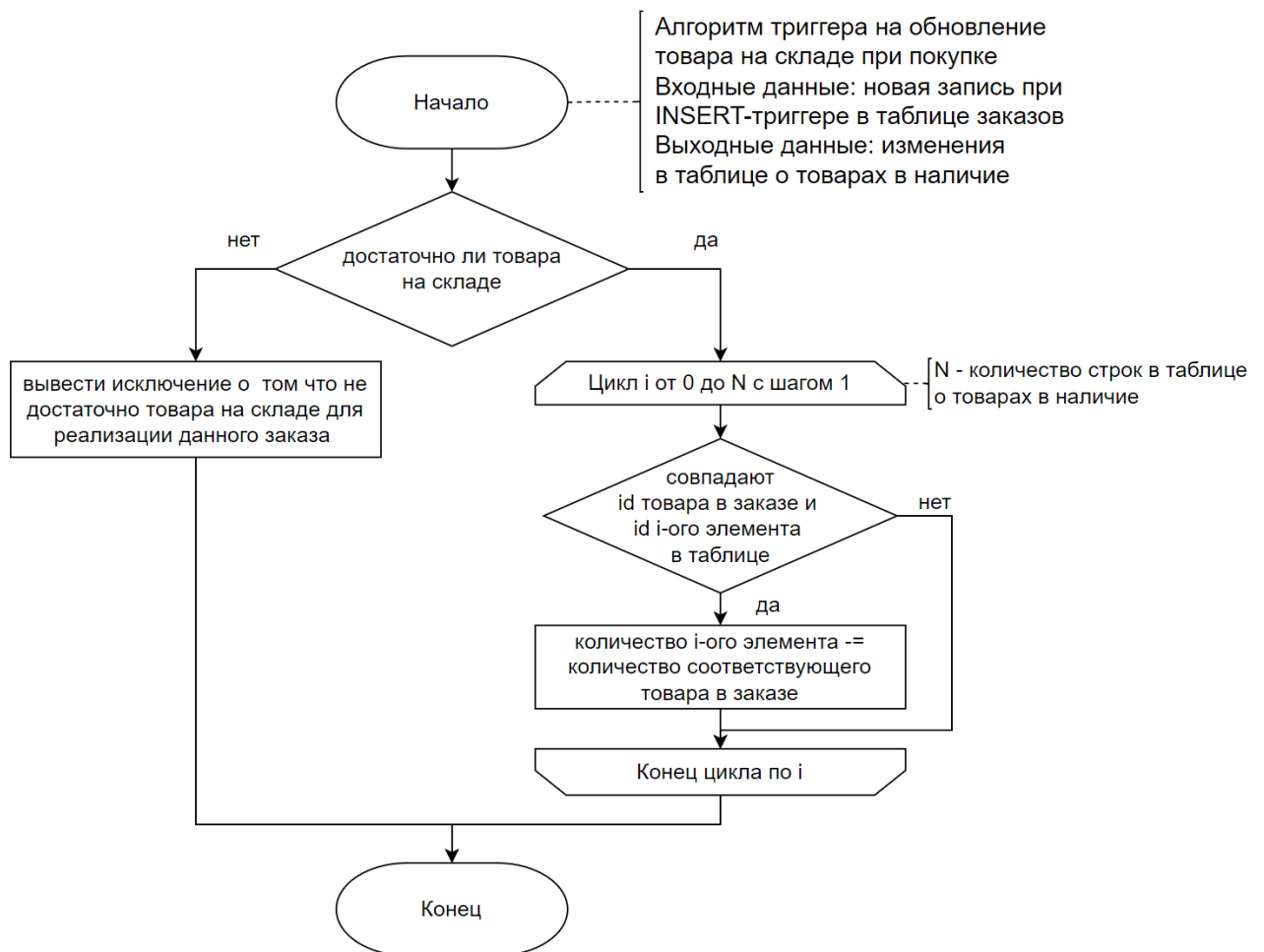


Рисунок 2.3 — Диаграмма алгоритма триггера на обновление товара на складе при покупке

2.5 Описание проектируемой ролевой модели на уровне базы данных

Три имеющиеся типа пользователей обладают следующим набором прав. Кладовщики – загрузка информации о партии и просмотр каталога доступных товаров; продавцы – внесение информации о заказах и продажах, просмотр каталога доступных товаров; администраторы – всё выше перечисленное.

Вывод

В конструкторском разделе была разработана структура базы данных, включающая 12 нормализованных сущностей, связанных между собой через внешние ключи, что обеспечивает целостность данных. Диаграмма базы данных наглядно демонстрирует взаимосвязи между таблицами: от справочников (номенклатура, страны, места хранения) до операционных сущностей (партии, заказы, продажи, списания). Особое внимание уделено механизмам поддержания актуальности данных через систему триггеров, автоматически обновляющих остатки на складе при поступлении новых партий и продажах. Ролевая модель, реализованная на уровне СУБД, разделяет права доступа между тремя типами пользователей (администраторами, продавцами и кладовщиками), обеспечивая безопасность и соответствие бизнес-процессам компании.

3 Технологический раздел

3.1 Обоснование выбора средств реализации базы данных и приложения

Выбор PostgreSQL обусловлен его надежностью, производительностью и соответствием требованиям предметной области. Система предоставляет: полноценную поддержку ACID-транзакций для гарантии целостности данных, гибкую систему ограничений и триггеров для автоматизации бизнес-процессов, встроенную ролевую модель с детализированным управлением доступом, поддержку сложных типов данных (JSONB, ENUM), что упрощает хранение структурированной информации.

Для клиентской части выбор пал на *C#* в силу: строгой типизации и высокой производительности исполняемого кода, наличия проверенных инструментов интеграции с PostgreSQL (Npgsql), поддержки современных парадигм программирования и паттернов проектирования.

Данный технологический стек оптимально соответствует требованиям к безопасности, надежности и масштабируемости проектируемой системы учета.

3.2 Реализация

3.2.1 Создание таблиц

В листинге 3.1 представлен код создания таблиц для базы данных Цветочный магазин.

```
1  create table country(  
2  id int primary key,  
3  name text  
4  );  
5  
6  create table nomenclature(  
7  id int primary key,  
8  name text,  
9  country_id int references country(id)  
10 );  
11  
12 create type counterpart_role as enum ('поставщик', 'покупатель');  
13 create type legal_status_type as enum (  
14 'Физическое лицо',
```

```

15     'Юридическое лицо',
16     'Индивидуальный предприниматель'
17 );
18
19 create table counterpart(
20     id int primary key,
21     name text,
22     type counterpart_role,
23     legal_status legal_status_type,
24     legal_address JSONB,
25     contact_person text,
26     phone varchar(20)
27 );
28
29 create type user_role as enum ('администратор', 'продавец', 'кладовщик');
30 create table "user"(
31     id int primary key,
32     name text,
33     type user_role
34 );
35
36 alter table "user" add column password varchar(255);
37
38 create table batch_of_products(
39     id_product_batch int,
40     id_nomenclature int references nomenclature(id),
41     production_date date,
42     expiration_date date,
43     cost_price decimal(12,2), -- Себестоимость
44     amount int,
45     responsible int references "user"(id),
46     suppliers int references counterpart(id),
47
48     primary key (id_product_batch, id_nomenclature)
49 );
50
51 create table write_off(
52     id int primary key,
53     id_product_batch int,
54     id_nomenclature int,
55     amount int,
56     responsible int references "user"(id),
57     foreign key (id_product_batch, id_nomenclature)
58     references batch_of_products(id_product_batch, id_nomenclature)
59 );
60

```

```

61 create table price(
62 id int primary key,
63 id_nomenclature int REFERENCES nomenclature(id),
64 selling_price decimal(12,2), -- Цена продажи
65 id_product_batch int,
66 foreign key (id_product_batch, id_nomenclature)
67 references batch_of_products(id_product_batch, id_nomenclature)
68 );
69
70 create table storage_place(
71 id int primary key,
72 name text,
73 address JSONB
74 );
75
76
77 CREATE TABLE product_in_stock(
78 id int PRIMARY KEY,
79 id_nomenclature int,
80 id_product_batch int,
81 amount int,
82 storage_place int references storage_place(id),
83 foreign key (id_product_batch, id_nomenclature)
84 references batch_of_products(id_product_batch, id_nomenclature)
85 );
86
87 create table "order"(
88 id int primary key,
89 reg_date date,
90 counterpart int references counterpart(id),
91 responsible int references "user"(id)
92 );
93
94 create table "order_product_in_stock"(
95 id_order int references "order"(id),
96 id_product int references product_in_stock(id),
97 amount int,
98 price int references price(id)
99 );
100
101 create type order_status_type as enum ('Новый', 'Подтверждён', 'Собран', '
    Получен');
102
103 create table sales(
104 receipt_number int primary key,
105 counterpart int references counterpart(id),

```

```

106 order_id int references "order"(id),
107 order_status order_status_type,
108 final_price decimal(12,2)
109 );

```

Листинг 3.1 — Создание таблиц базы данных

В листинге 3.2 представлен код создания ограничений для таблиц базы данных Цветочный магазин.

```

1  -- Для таблицы country
2  alter table country add constraint check_country_name_length check (length
   (name) > 0);
3
4  -- Для таблицы nomenclature
5  alter table nomenclature add constraint check_nomenclature_name_length
   check (length(name) > 0);
6  alter table nomenclature add constraint check_country_reference check (
   country_id is not null);
7
8  -- Для таблицы counterpart
9  alter table counterpart add constraint check_counterpart_name_length check
   (length(name) > 0);
10 alter table counterpart add constraint check_contact_person_length check (
   length(contact_person) > 0);
11 alter table counterpart add constraint check_phone_format check (phone ~ '
   ^\+?[0-9]{10,15}$');
12
13 -- Для таблицы user
14 alter table "user" add constraint check_user_name_length check (length(
   name) > 1);
15
16 -- Для таблицы batch_of_products
17 alter table batch_of_products add constraint check_production_date check (
   production_date <= current_date);
18 alter table batch_of_products add constraint check_expiration_date check (
   expiration_date > production_date);
19 alter table batch_of_products add constraint check_positive_cost_price
   check (cost_price > 0);
20 alter table batch_of_products add constraint check_positive_amount check (
   amount > 0);
21
22 -- Для таблицы write_off
23 alter table write_off add constraint check_positive_write_off_amount check
   (amount >= 0);
24
25 -- Для таблицы price

```

```

26 alter table price add constraint check_positive_selling_price check (
    selling_price > 0);
27
28 -- Для таблицы storage_place
29 alter table storage_place add constraint check_storage_name_length check (
    length(name) > 0);
30
31 -- Для таблицы product_in_stock
32 alter table product_in_stock add constraint check_positive_stock_amount
    check (amount >= 0);
33
34 -- Для таблицы order
35 alter table "order_product_in_stock"
36 add constraint fk_product_id foreign key (id_product) references
    product_in_stock;
37
38 alter table "order_product_in_stock"
39 add constraint fk_order_id foreign key (id_order) references "order";
40 -- Для таблицы sales
41 alter table sales ADD constraint check_positive_final_price check (
    final_price > 0);

```

Листинг 3.2 — Создание ограничений для таблиц базы данных

3.2.2 Создание триггеров

В листингах 3.3 и 3.4 представлены коды создания триггеров на обновление товара на складе при поступлении новой партии товара и на обновление товара на складе при покупке, соответственно.

```

1 create or replace function public.update_product_in_stock()
2 returns trigger
3 language plpgsql
4 as $function$
5 begin
6     -- Проверяем, существует ли уже запись о данном товаре и партии на складе
7     if exists (
8         select 1
9             from product_in_stock
10            where id_nomenclature = new.id_nomenclature
11              and id_product_batch = new.id_product_batch
12        ) then
13         -- Если запись существует, обновляем количество
14         update product_in_stock
15         set amount = amount + new.amount
16         where id_nomenclature = new.id_nomenclature

```



```

17     and id_product_batch = new.id_product_batch;
18 else
19     -- Если записи нет, создаем новую
20     insert into product_in_stock (id, id_nomenclature, id_product_batch,
21     amount, storage_place)
22     values (
23         coalesce((select max(id) from product_in_stock), 0) + 1,
24         new.id_nomenclature,
25         new.id_product_batch,
26         new.amount,
27         (select id from storage_place order by random() limit 1) -- случайное
28         место хранения
29     );
30
31     with avg_cost_by_year as (
32         select
33             id_nomenclature,
34             EXTRACT(YEAR FROM production_date) AS production_year,
35             AVG(cost_price) * 1.55 AS calculated_price
36         from
37             batch_of_products
38         group by
39             id_nomenclature,
40             extract(year from production_date)
41     )
42     insert into price (id, id_nomenclature, selling_price, id_product_batch)
43     select
44         row_number() over () + coalesce((select max(id) from price), 0),
45         bop.id_nomenclature,
46         ac.calculated_price,
47         bop.id_product_batch
48     from
49         batch_of_products bop
50     join
51         avg_cost_by_year ac on bop.id_nomenclature = ac.id_nomenclature
52         and extract(year from bop.production_date) = ac.production_year
53     where bop.id_nomenclature = new.id_nomenclature and bop.id_product_batch
54         = new.id_product_batch;
55 end if;
56
57 return new;
58 end;
59 $function$
60 ;

```

Листинг 3.3 — Создание триггера на обновление товара на складе при поступлении новой партии товара

```
1 create or replace function public.update_stock_on_order_product_insert()
2 returns trigger
3 language plpgsql
4 as $function$
5 declare
6     batch_rec record;
7     remaining_amount numeric := new.amount;
8     nomencl_id int;
9     price_rec record;
10 begin
11     -- Получаем id_nomenclature из связанной записи product_in_stock
12     select id_nomenclature into nomencl_id
13     from product_in_stock
14     where id = new.id_product;
15
16     -- Проверяем доступное количество с учетом срока годности
17     while remaining_amount > 0 loop
18         -- Ищем подходящую партию с неистекшим сроком годности
19         select pis.id, pis.amount, p.id as price_id, bop.expiration_date
20         into batch_rec
21         from product_in_stock pis
22         join batch_of_products bop on pis.id_product_batch = bop.
23         id_product_batch
24         and pis.id_nomenclature = bop.id_nomenclature
25         join price p on p.id_nomenclature = bop.id_nomenclature
26         and p.id_product_batch = bop.id_product_batch
27         where pis.id_nomenclature = nomencl_id
28         and pis.amount > 0
29         and bop.expiration_date > current_date
30         order by bop.expiration_date asc, pis.amount desc
31         limit 1
32         for update of pis;
33
34         if not found then
35             raise exception 'Недостаточно товара на складе с неистекшим сроком год
36             ности. ID товара: %, запрошено: %, осталось: %',
37             new.id_product,
38             new.amount,
39             new.amount - remaining_amount;
40         end if;
41
42         -- Определяем сколько можем списать с этой партии
```

```

41 declare
42     to_reduce numeric := least(batch_rec.amount, remaining_amount);
43 begin
44     -- Списание
45     update product_in_stock
46     set amount = amount - to_reduce
47     where id = batch_rec.id;
48
49     -- Сохраняем цену в заказе
50     if remaining_amount = new.amount then
51         new.price := batch_rec.price_id;
52     end if;
53
54     remaining_amount := remaining_amount - to_reduce;
55 end;
56 end loop;
57
58 return new;
59 end;
60 $function$
61 ;

```

Листинг 3.4 — Создание триггера на обновление товара на складе при покупке

3.2.3 Создание ролей

В листинге 3.5 представлен код создания ролей для базы данных Цветочный магазин.

```

1 -- Создаем роли
2 create role flower_admin with login password 'admin_password';
3 create role flower_seller with login password 'seller_password';
4 create role flower_storekeeper with login password 'storekeeper_password';
5
6 -- Комментарии к ролям
7 comment on role flower_admin is 'Администратор цветочного магазина (полные п
   права)';
8 comment on role flower_seller is 'Продавец (просмотр товаров, оформление зак
   азов)';
9 comment on role flower_storekeeper is 'Кладовщик (просмотр товаров, загрузка
   партий)';
10
11 -- Права для кладовщика
12 grant select on nomenclature, price, product_in_stock, batch_of_products,
   country to flower_storekeeper;
13 grant insert on price to flower_storekeeper;

```

```

14 grant insert on batch_of_products to flower_storekeeper;
15 grant select, insert on product_in_stock to flower_storekeeper;
16 grant select on storage_place to flower_storekeeper;
17 -- Права на последовательности (если используются автоинкрементные ID)
18 grant usage, select on ALL SEQUENCES IN SCHEMA public to flower_storekeeper;
19 grant usage on schema public to flower_storekeeper;
20 grant execute on function update_product_in_stock() to flower_storekeeper;
21
22 -- Права для продавца
23 grant flower_storekeeper to flower_seller;
24 grant insert, select on "order", order_product_in_stock, sales to
    flower_seller;
25 grant update (amount) on product_in_stock to flower_seller;
26 grant select on "user" to flower_seller;
27
28 grant select on product_in_stock to flower_seller;
29 grant select on batch_of_products to flower_seller;
30 grant select on price to flower_seller;
31 grant execute on function update_stock_on_order_product_insert() to
    flower_seller;
32
33 -- Права для администратора
34 grant flower_storekeeper to flower_admin;
35 grant flower_seller to flower_admin;

```

Листинг 3.5 — Создание ролей для базы данных

3.3 Тестирование

В таблице 3.1 приведены тесты для проверки корректности работы триггеров. Для этого были написаны интеграционные тесты.

Также были написаны модульные тесты для программы управления базой данных. Результат тестирования представлен на изображении 3.1

Обозреватель тестов	
<div> <div>▶ ▶ ▶ ↺ ✕</div> <div>🔍</div> <div> <div>🧪 18</div> <div>✅ 18</div> <div>❌ 0</div> </div> <div> <div>📄</div> <div>⌵</div> <div>⌶</div> <div>⌷</div> <div>⚙</div> </div> </div>	
Запуск тестов завершен: тестов запущено в 958 мс: 18 (пройдено: 18, не пройдено: 0, пропущено: 0).	
Тестирование ▲	Длительн...
▲ ✅ Domain.Tests (5)	103 мс
▲ ✅ Domain.Tests (5)	103 мс
▲ ✅ ProductServiceTests (5)	103 мс
✅ CheckNewAmount_WithEmptyProductId_ShouldThrowArgumentException	72 мс
✅ CheckNewAmount_WithInvalidProduct_ShouldReturnFalse	21 мс
✅ CheckNewAmount_WithNegativeAmount_ShouldThrowArgumentException	< 1 мс
✅ CheckNewAmount_WithValidProductAndAmount_ShouldReturnTrue	2 мс
✅ GetAllAvailableProducts_ShouldReturnInventory	8 мс
▲ ✅ Integration.Tests (4)	1,1 с
▲ ✅ Integration.Tests (4)	1,1 с
▲ ✅ InventoryRepoTest (2)	363 мс
✅ CheckNewAmount_ShouldReturnCorrectResults	348 мс
✅ GetAvailableProduct_ShouldReturnCorrectInventory	15 мс
▲ ✅ ProductBatchLoaderTest (1)	336 мс
✅ Load_ShouldInsertBatchAndUpdateStock	336 мс
▲ ✅ ReceiptRepoTest (1)	407 мс
✅ ProcessOrder_ShouldCreateOrderAndUpdateStock	407 мс
▲ ✅ ProductBatchReader.Tests (3)	8 мс
▲ ✅ ProductBatchReader.Tests (3)	8 мс
▲ ✅ ProductBatchReaderTests (3)	8 мс
✅ Create_WithEmptyFile_ThrowsException	< 1 мс
✅ Create_WithInvalidData_ThrowsException	< 1 мс
✅ Create_WithValidData_ReturnsProductBatch	8 мс
▲ ✅ ReceiptOfSale.Tests (6)	27 мс
▲ ✅ ReceiptOfSale.Tests (6)	27 мс
▲ ✅ ReceiptRepoTests (6)	27 мс
✅ Create_WithEmptyItems_ThrowsArgumentException	13 мс
✅ Create_WithMultipleItems_CorrectlyCalculatesTotals	< 1 мс
✅ Create_WithNullItems_ThrowsArgumentNullException	< 1 мс
✅ Create_WithSingleItem_CorrectlyCalculatesTotal	1 мс
✅ Create_WithValidItemsAndCustomer_ReturnsReceipt	13 мс
✅ Create_WithZeroPriceItem_IncludesInTotal	< 1 мс

Рисунок 3.1 — Результат модульного и интеграционного тестирования

Таблица 3.1 — Тестирование корректности работы триггеров и ограничений

Название	Входные данные	Ожидаемый результат	Фактический результат
Триггер на обновление товара на складе при поступлении новой партии товара.	Корректные данные.	Добавление соответствующих строк в таблицы <i>batch of products</i> , <i>product in stock</i> и <i>price</i> .	Нужные строки были добавлены.
Триггер на обновление товара на складе при поступлении новой партии товара.	Данные с повторяющимся номером партии	Вызов исключения.	Ошибка при загрузке партии: 23505: повторяющееся значение ключа нарушает ограничение уникальности "batch of products pkey"
Триггер на обновление товара на складе при покупке.	Корректные данные.	Добавление соответствующих строк в таблицы <i>order</i> , <i>sales</i> и изменение количества соответствующего товара в таблице <i>product in stock</i> .	Нужные строки были добавлены и количество в наличие было изменено.
Триггер на обновление товара на складе при покупке.	Попытка купить больше чем есть в наличие.	Вызов исключения.	Исключение 'Недостаточно товара на складе с истекшим сроком годности.'

3.4 Пример работы программы

В листинге 3.6 представлен пример работы программы для сценария совершения покупки. Многоточием заменены постоянно повторяющийся вывод меню.

```
1 Введите логин (Ваш ID): 1
2 Введите пароль: pass1
3 Вы вошли как администратор.
4 === ГЛАВНОЕ МЕНЮ ===
5 1. Сделать заказ
6 2. Загрузка информации о новой партии
7 0. Выход
8 Выберите пункт меню: 1
9 === Сделать заказ ===
10 1. Показать доступные товары
11 2. Добавить товар в корзину
12 3. Изменить количество товара в корзине
13 4. Удалить товар из корзины
14 5. Показать содержание корзины
15 6. Заказать
16 0. Назад (если вернуться, содержание корзины обнулится)
17 Выберите пункт меню: 1
18 Товар: 1 Телопея, United Kingdom, количество: 32, цена: 3725,24
19 Товар: 4 Сетария, Cape Verde, количество: 2, цена: 5260,35
20 Товар: 5 Амбrellа, Bolivia, количество: 55, цена: 3703,16
21 Товар: 9 Антуриум, Botswana, количество: 27, цена: 4181,22
22 Товар: 11 Ваза керамическая белая 25см, Zimbabwe, количество: 23, цена:
    6872,71
23 Товар: 13 Книфофия, Philippines, количество: 45, цена: 3888,72
24 Товар: 15 Сетка декоративная медовая 45смx100м, Namibia, количество: 60, ц
    ена: 4255,04
25 Товар: 19 Лента репсовая Пудра 100м, Uganda, количество: 11, цена: 4667,08
26 Товар: 21 Эрика, Tanzania, количество: 2, цена: 5698,18
27 Товар: 23 Пион, Turkmenistan, количество: 53, цена: 4522,5
28 Товар: 36 Пихта, Andorra, количество: 13, цена: 3515,79
29 Товар: 44 Вибурнум, Monaco, количество: 4, цена: 3889,92
30 Товар: 45 Пион, Iraq, количество: 9, цена: 5187,55
31 Товар: 51 Кипарисовик, United States of America, количество: 47, цена:
    4526,53
32 Товар: 56 Тилландсия, Grenada, количество: 7, цена: 3207,71
33 Товар: 61 Лента атласная красная 100м, Mexico, количество: 3, цена:
    3909,84
34 Товар: 64 Лента рафия окрашенная (10 цветов) 100м, Dominican Republic, кол
    ичество: 1, цена: 5139,66
35 Товар: 66 Подставка для вазы 30см, Haiti, количество: 84, цена: 2917,62
```

```

36 Товар: 67 Лента атласная золотая 100м, The Gambia, количество: 12, цена:
    3458,98
37 Товар: 70 Астильба, Democratic Republic of Congo, количество: 6, цена:
    3005,13
38 -----
39 Продолжить вывод доступных товаров? (0 - остановиться, 1 - продолжить): 0
40 === Сделать заказ ===
41 ...
42 Выберите пункт меню: 2
43 Введите id товара для добавления в корзину: 5
44 Введите количество товара для добавления в корзину: 10
45 Товар добавлен в корзину.
46 === Сделать заказ ===
47 ...
48 Выберите пункт меню: 2
49 Введите id товара для добавления в корзину: 66
50 Введите количество товара для добавления в корзину: 14
51 Товар добавлен в корзину.
52 === Сделать заказ ===
53 ...
54 Выберите пункт меню: 5
55 Корзина:
56 5 Амбrellа, Bolivia, количество: 10, цена за шт.: 3703,16
57 66 Подставка для вазы 30см, Haiti, количество: 14, цена за шт.: 2917,62
58 === Сделать заказ ===
59 ...
60 Выберите пункт меню: 6
61 Заказ оформлен. Номер чека 5024

```

Листинг 3.6 — Совершение покупки

В листинге 3.7 представлен пример работы программы для сценария загрузки информации о новой партии.

```

1 Введите логин (Ваш ID): 1
2 Введите пароль: pass1
3 Вы вошли как администратор.
4 === ГЛАВНОЕ МЕНЮ ===
5 1. Сделать заказ
6 2. Загрузка информации о новой партии
7 0. Выход
8 Выберите пункт меню: 2
9 === Загрузка информации о новой партии ===
10 Введите путь к файлу с данными о партии:
11 "D:\bmstu\Курсовая по БД\software_design\ProjectForCourseWorkForDB\
    FlowerShop\batch4.txt"
12 Needs to be loaded

```


Номенклатура	Даты (произв./годен)	Кол-во	Цена	Срок годности
854	30.05.2025 / 30.05.2026	51	1404,50 руб.	363 дней
626	30.05.2025 / 28.08.2025	39	4590,14 руб.	88 дней
646	30.05.2025 / 28.08.2025	69	3887,61 руб.	88 дней
383	30.05.2025 / 13.06.2025	60	3958,11 руб.	12 дней
705	30.05.2025 / 13.06.2025	8	1054,00 руб.	12 дней
205	30.05.2025 / 29.06.2025	66	2568,34 руб.	28 дней
55	30.05.2025 / 26.11.2025	71	1962,61 руб.	178 дней
340	30.05.2025 / 29.07.2025	16	1384,28 руб.	58 дней
109	30.05.2025 / 30.05.2026	71	1115,37 руб.	363 дней
901	30.05.2025 / 06.06.2025	63	4529,26 руб.	5 дней
98	30.05.2025 / 29.07.2025	8	3565,52 руб.	58 дней
311	30.05.2025 / 29.07.2025	12	1847,95 руб.	58 дней
425	30.05.2025 / 30.05.2026	54	4314,36 руб.	363 дней
33	30.05.2025 / 26.11.2025	48	2973,71 руб.	178 дней
Загрузка прошла успешно!				
=== ГЛАВНОЕ МЕНЮ ===				
1. Сделать заказ				
2. Загрузка информации о новой партии				
0. Выход				
Выберите пункт меню: 1				
=== Сделать заказ ===				
1. Показать доступные товары				
2. Добавить товар в корзину				
3. Изменить количество товара в корзине				
4. Удалить товар из корзины				
5. Показать содержание корзины				
6. Заказать				
0. Назад (если вернуться, содержание корзины обнулится)				
Выберите пункт меню: 1				
Товар: 1 Телопея, United Kingdom, количество: 32, цена: 3725,24				
Товар: 4 Сетария, Cape Verde, количество: 2, цена: 5260,35				
Товар: 5 Амбрелла, Bolivia, количество: 45, цена: 3703,16				
Товар: 9 Антуриум, Botswana, количество: 27, цена: 4181,22				
Товар: 11 Ваза керамическая белая 25см, Zimbabwe, количество: 23, цена: 6872,71				
Товар: 13 Книфофия, Philippines, количество: 45, цена: 3888,72				
Товар: 15 Сетка декоративная медовая 45смx100м, Namibia, количество: 60, цена: 4255,04				
Товар: 19 Лента репсовая Пудра 100м, Uganda, количество: 11, цена: 4667,08				
Товар: 21 Эрика, Tanzania, количество: 2, цена: 5698,18				
Товар: 23 Пион, Turkmenistan, количество: 53, цена: 4522,5				
Товар: 33 Молюцелла, Uganda, количество: 48, цена: 3963,05				
Товар: 36 Пихта, Andorra, количество: 13, цена: 3515,79				

56	Товар: 44 Вибурнум, Монако, количество: 4, цена: 3889,92
57	Товар: 45 Пион, Iraq, количество: 9, цена: 5187,55
58	Товар: 51 Кипарисовик, United States of America, количество: 47, цена: 4526,53
59	Товар: 55 Мускари, Lebanon, количество: 71, цена: 3631,19
60	Товар: 56 Тилландсия, Grenada, количество: 7, цена: 3207,71
61	Товар: 61 Лента атласная красная 100м, Mexico, количество: 3, цена: 3909,84
62	Товар: 64 Лента рафия окрашенная (10 цветов) 100м, Dominican Republic, количество: 1, цена: 5139,66
63	Товар: 66 Подставка для вазы 30см, Haiti, количество: 70, цена: 2917,62

Листинг 3.7 — Загрузка информации о новой партии товара

Вывод

В рамках технологического раздела была успешно реализована база данных цветочного магазина, включающая все необходимые таблицы, ограничения целостности и триггеры для автоматизации бизнес-процессов. Разработанное приложение на C# обеспечило удобный интерфейс для взаимодействия с системой. Проведенные интеграционные и модульные тесты подтвердили корректность работы всех компонентов.

4 Исследовательский раздел

Было проведено исследование влияния индексов на скорость выполнения запросов при различных объемах данных. На тестовой таблице *batch_of_products_copy* (имеет такую же структуру как и *batch_of_products*) с количеством записей от 10 до 1 миллиона измерялось время выполнения типового запроса с фильтрацией и сортировкой. Замеры проводились в двух режимах: без использования индексов и с оптимизированными индексами (составным для условий фильтрации и отдельным для сортировки).

В листинге 4.1 представлен запрос, для которого проводилось исследование и использованные индексы.

```
1  select id_product_batch, id_nomenclature, production_date, expiration_date
2  from batch_of_products_copy
3  where
4  production_date between '2023-01-01' and '2023-12-31'
5  and cost_price between 100.00 and 500.00
6  and amount > 50
7  order by expiration_date desc;
8
9  create index idx_composite_filter on batch_of_products_copy(
10     production_date, cost_price, amount);
11
12 create index idx_sorting on batch_of_products_copy(expiration_date desc);
```

Листинг 4.1 — Запрос и индексы

Таблица 4.1 — Сравнение времени выполнения запросов с индексами и без (в секундах)

Количество записей	Без индекса	С индексом
10	0.000040	0.000008
100	0.000051	0.000014
1,000	0.000146	0.000055
10,000	0.002802	0.000699
100,000	0.030805	0.017392
1,000,000	0.129758	0.128987

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. "Полный список документации по программным продуктам 1С [Электронный ресурс]". "Режим доступа: https://its.1c.ru/docs/program_documentation (дата обращения: 21.05.2025)".
2. А.О. Фирсов. Исследование способов применения NoSQL и реляционных баз данных. E-Scio, 2019.
3. ТКАЧЕНКО А.В. ВАСИЛЬЧИКОВА А.В. ГРИШУНОВ С.С. ОБЗОР КЛАССОВ НЕРЕЛЯЦИОННЫХ БАЗ ДАННЫХ. ЭЛЕКТРОННЫЙ ЖУРНАЛ: НАУКА, ТЕХНИКА И ОБРАЗОВАНИЕ Издательство: Общество с ограниченной ответственностью "Манускрипт 2016.