

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Казанский национальный исследовательский технический  
университет им. А.Н. Туполева-КАИ»  
(КНИТУ-КАИ)

Отделение СПО ИКТЗИ, КИТ

(наименование института (факультета))

**КУРСОВАЯ РАБОТА**

по дисциплине «Основы проектирования баз данных»

тема «Сеть магазинов одежды»

Выполнила: обучающаяся группы 4235

Никулина Е.Д.

(ФИО)

(подпись)

Проверил: преподаватель Шумилкин А.О.

(должность)

(Ф.И.О.)

(подпись)

Курсовая работа защищена с оценкой: \_\_\_\_\_

Дата защиты «\_\_\_» 20\_\_ г.

Казань 2025г.

## **АННОТАЦИЯ**

Курсовая работа 60 страниц, 96 рисунков, 4 таблицы, 10 источников литературы.

В данном курсовом проекте будет спроектирована база данных для сети магазинов одежды.

Для создания базы данных была использована стандартная среда администрирования баз данных выбранной СУБД MS SQL Server – Среда администрирования MS SQL Server Management.

База данных «Сеть магазинов одежды» включает таблицы, представления, триггеры и хранимые процедуры.

## **СОДЕРЖАНИЕ**

ВВЕДЕНИЕ.....	4
ГЛАВА 1 ОСНОВНАЯ ЧАСТЬ .....	6
1.1 Концептуальное проектирование .....	6
1.2 Обоснование выбора СУБД.....	10
1.3 Даталогическое проектирование .....	12
1.3.1 Преобразование концептуальной модели в реляционную модель	
12	
1.3.2 Нормализация базы данных.....	14
1.3.3 Определение характеристик атрибутов.....	15
1.4 Создание БД в выбранной СУБД .....	17
1.5 Поддержка целостности данных .....	19
1.5.1 Декларативная поддержка ограничений целостности .....	19
1.5.2 Процедурная поддержка ограничений целостности .....	27
ГЛАВА 2 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ .....	51
ЗАКЛЮЧЕНИЕ .....	59
СПИСОК ЛИТЕРАТУРЫ.....	60

## **ВВЕДЕНИЕ**

Система управления сетью магазинов одежды представляет собой инструмент для автоматизации процессов, связанных с продажей, поставками, управлением ассортиментом, персоналом и клиентами. В современных условиях конкурентного розничного рынка наличие эффективной информационной системы позволяет магазинам обеспечивать высокий уровень обслуживания клиентов, оперативное обновление ассортимента и эффективное управление поставками и возвратами.

В рамках данного проекта будет спроектирована база данных для пользователей системы, включая персонал магазинов, поставщиков и покупателей. Будет реализована логика работы ключевых бизнес-процессов: продажи товаров, учет остатков, возврат продукции, работа с клиентами и управление персоналом.

**Объектом исследования** является процесс проектирования базы данных для сети магазинов одежды, а предметом — разработка и реализация реляционной базы данных «Сеть магазинов одежды».

**Цель курсовой работы** — разработка базы данных «Сеть магазинов одежды».

Для достижения поставленной цели необходимо выполнить следующие задачи:

1. Ознакомиться с литературой по данной теме.
2. Проанализировать предметную область.
3. Построить ER-модель.
4. Построить логическую модель БД.
5. Построить физическую модель БД.
6. Создать объекты БД (таблицы, запросы).
7. Определить условия целостности БД, разработать триггеры и хранимые процедуры.
8. Разработать запросы к базе данных.

БД «Сеть магазинов одежды» обеспечит централизованное управление

ассортиментом, упростит процессы продаж и поставок, повысит контроль над операциями и улучшит взаимодействие с клиентами и сотрудниками.

# **ГЛАВА 1 ОСНОВНАЯ ЧАСТЬ**

## **1.1 Концептуальное проектирование**

Деятельность по управлению сетью магазинов одежды связана с выполнением множества операций, где большое количество товаров, поставок и клиентов усложняет процессы обработки данных и замедляет работу персонала. Трудности, связанные с учетом ассортимента, отслеживанием остатков на складах магазинов, регистрацией клиентов и продажами, требуют высокой точности и автоматизации.

В системе ведется база данных сотрудников. О каждом сохраняются следующие сведения: должность, ФИО, дата рождения и устройства на работу, контактная информация. Каждому сотруднику присваивается уникальный идентификационный номер.

При добавлении новых товаров в систему, каждому товару присваивается уникальный идентификатор, указывается его наименование, описание, цена, категория, поставщик, изображение и другие атрибуты. При продаже учитываются данные покупателя и сотрудника, количество, цена и название магазина, в котором производится покупка.

Предусматривается автоматизация учета продаж и возвратов, поставок товаров, добавления новых позиций в каталог, а также управление клиентской сотрудниками.

Каждая покупка в системе содержит информацию о магазине, сотруднике, оформившем продажу, клиенте, дате покупки и перечне купленных товаров.

Поставки содержат информацию о товарах, поступивших от поставщиков, дате поставки и распределении по магазинам. Магазины связаны с адресами и контактной информацией.

Автоматизация деятельности сети магазинов одежды обеспечивает сотрудникам более быстрое и качественное обслуживание клиентов, прозрачный учет товаров и эффективную аналитику продаж.

Работа системы включает в себя ряд стандартных действий:

1. Хранение информации о магазинах.
2. Покупка/возврат товаров.
3. Хранение информации о сотрудниках.
4. Управление поставками.
5. Хранение информации о поставщиках.
6. Хранение информации о категориях и товарах в них.
7. Управление количеством товара и его наличия в магазинах.

База данных хранит информацию о товарах, магазинах, поставках, покупателях, покупках и возвратах. Для каждого товара указываются его характеристики, цена, количество в каждом магазине, а также поставщик. Ведется учет клиентских заказов, возвратов, а также действий сотрудников.

В систему можно вносить следующие изменения:

- управление базой товаров (добавление, редактирование, удаление);
- управление базой сотрудников;
- контроль поставок и остатков;
- оформление покупок;
- обработка возврата покупок.

Программа должна формировать следующие отчеты:

- отчет об остатках товаров на складах;
- анализ спроса по категориям и конкретным товарным позициям;
- эффективность поставщиков.

На основе проведенного анализа предметной области были выделены следующие сущности:

- продукты – хранение сведений о товарах;
- категории – хранение сведений о имеющихся категориях;
- продукт-категория – промежуточная таблица между продуктами и категориями;
- поставщик – хранение сведений о имеющихся поставщиках;
- поставка – хранение сведений о прошедших и будущих поставках;

- магазин – хранение сведений о магазине в сети;
- магазин-товар-инвентарь – промежуточная таблица между магазинами и товарами;
- поставка-магазин – промежуточная таблица между магазином и поставкой;
- сотрудник – хранение сведений о сотрудниках;
- покупатель – хранение сведений о покупателях;
- покупка – хранение сведений о сделанных ранее покупках;
- возврат – хранение сведений о возвращенных покупках;
- покупка-единица – промежуточная таблица между покупкой и товаром.

В таблице 1 представлен список атрибутов каждой выделенной сущности.

Таблица 1 – Перечень атрибутов выделенных сущностей

Сущность	Атрибут	Описание
1	2	3
Продукты	Код продукта	Уникальный код товара
	Название	Название товара
	Описание	Описание товара
	Цена	Стоимость товара
	Код поставки	Уникальный код поставки
	Изображение	Изображение товара
Категории	Код категории	Уникальный код категории
	Название	Название категории
Продукт-категория	Код категории	Уникальный код категории
	Код продукта	Уникальный код продукта
Поставщик	Код поставщика	Уникальный код поставщика
	Название	Название поставщика
	Номер телефона	Номер телефона поставщика
	Электронная почта	Электронная почта для связи с поставщиком
	Код поставки	Уникальный код поставки
Поставка	Код поставки	Уникальный код поставки
	Название	Название поставки
Магазин	Код магазина	Уникальный код магазина
	Название	Название магазина
	Номер телефона	Номер телефона магазина
Магазин-товар-инвентарь	Адрес	Адрес магазина
	Изображение	Изображение магазина
Магазин-товар-инвентарь	Код магазина	Уникальный код магазина
	Код продукта	Уникальный код продукта

## Продолжение таблицы 1

Сущность	Атрибут	Описание
1	2	3
Магазин-товар-инвентарь	Количество	Количество товара в магазине
Поставка-магазин	Код поставки	Уникальный код поставки
	Код магазина	Уникальный код магазина
Сотрудник	Код сотрудника	Уникальный код сотрудника
	Электронная почта	Электронная почта сотрудника
	ФИО	Фамилия имя отчество сотрудника
	Код магазина	Уникальный код магазина
	Дата устройства	Дата устройства на работу сотрудника
	Должность	Должность сотрудника
	Дата рождения	Дата рождения сотрудника
Покупатель	Код покупателя	Уникальный код покупателя
	ФИО	Фамилия имя отчество покупателя
	Электронная почта	Электронная почта покупателя
	Номер телефона	Номер телефона покупателя
Покупка	Код покупки	Уникальный код покупки
	Дата покупки	Дата совершения покупки
	Код магазина	Уникальный код магазина
	Код сотрудника	Уникальный код сотрудника
	Код покупателя	Уникальный код покупателя
Возврат	Код возврата	Уникальный код возврата
	Код продукта	Уникальный код продукта
	Дата возврата	Дата возврата покупки
Покупка-единица	Код покупки-единицы	Уникальный код покупки-единицы
	Код покупки	Уникальный код покупки
	Код продукта	Уникальный код продукта
	Количество	Количество товара в покупке
	Цена	Цена товара в покупке

На рисунке 1 представлена ER-диаграмма «сущность-связь».

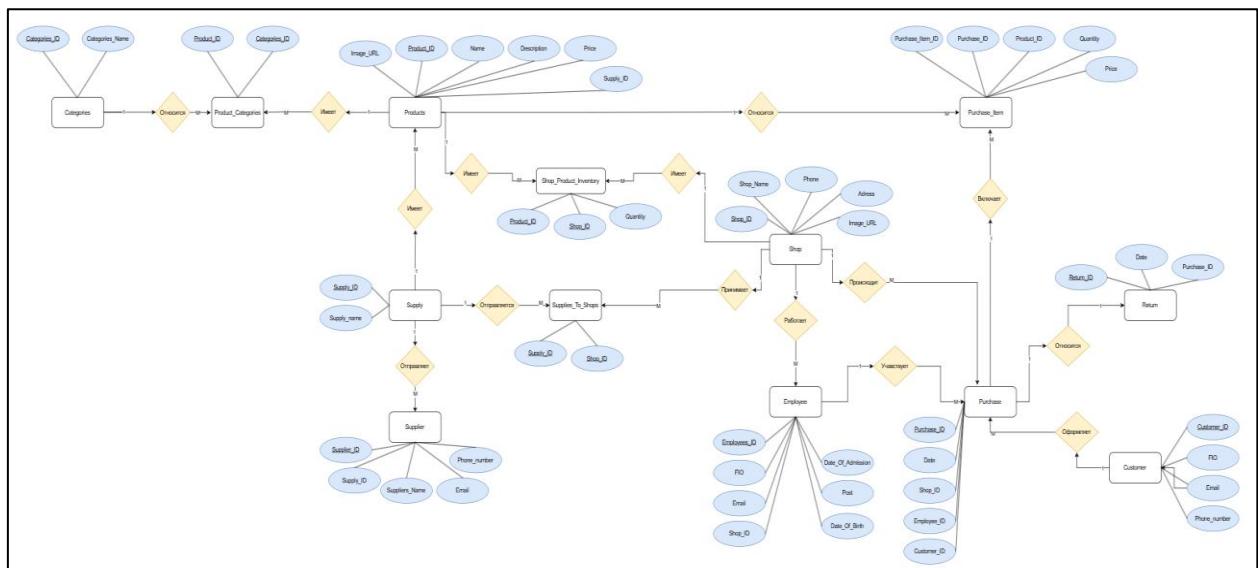


Рисунок 1 – ER-диаграмма «сущность-связь»

Между выделенными сущностями созданы следующие связи:

- М:М – у множества продуктов может быть множество категорий;
- 1:М – каждая поставка может содержать множество товаров;
- 1:М – каждый поставщик может иметь множество поставок;
- М:М – у множества магазинов может быть множество товаров;
- М:М – у множества магазинов может быть множество поставок;
- М:М – множество покупок может содержать множество товаров;
- 1:М – каждый магазин может содержать множество сотрудников;
- 1:М – каждый магазин может иметь множество покупок;
- 1:М – каждый покупатель может иметь множество покупок;
- 1:1 – каждая покупка может иметь один возврат.

## **1.2 Обоснование выбора СУБД**

СУБД (система управления базами данных) – это информационная среда для создания баз данных и управления хранимыми в них сведениями об объектах реального мира.

Выбор СУБД определяется многими факторами, но главный из них – возможность работы с построенной моделью данных. Поэтому одной из важнейших характеристик является тип модели (иерархический, сетевой, реляционный), который поддерживается СУБД. База данных для решения поставленной задачи является реляционной. Помимо модели данных важным показателем является стоимость лицензии для разработки базы данных и стоимость поддержки данной СУБД. Также необходимо обращать внимание на минимальные технические требования для оперативной и эффективной работы СУБД.

В настоящее время на рынке представлен огромный выбор СУБД и каждая СУБД «заточена» под определенный круг задач, которые она может выполнять. Проведем сравнительный анализ СУБД для разработки информационной системы сети магазинов одежды.

Для сравнения выберем такие СУБД, как:

- MS Access;

- MySQL;
- MS SQL Server.

Сравнительный анализ СУБД представлен в таблице 2.

Таблица 2 – Сравнение СУБД

Критерии оценки	MS Access	MySQL	MS SQL Server
1	2	3	4
<b>Размер базы данных</b>			
несколько Мб	+	+	
до 100Мб	+	+	
несколько ГБ		+	+
100 Гб и более			+
<b>Количество одновременных пользователей, работающих с базой данных напрямую</b>			
1 пользователь	+	+	
до 10	+	+	
более 10		+	+
более 100		+	+
более 1000			+
<b>Стоимость лицензии</b>			
бесплатно (только лицензия на пакет)	+	+	
дешево			
дорого			+
<b>Платформа, на которой способна работать СУБД</b>			
Windows	+		+
Unix/Linux			
Windows+Linux		+	
Кластеры			+
<b>Встроенный язык программирования</b>			
Microsoft	+		+
Borland	+		+
Языки для Linux			+
<b>Тип программы</b>			
Локальное приложение	+		
WEB		+	
Сложная система			+
<b>Защита данных</b>			
низкая			
слабая	+		
сильная		+	+
<b>Мощность языка SQL и возможности СУБД</b>			
очень слабые			
слабые	+		
развитые		+	
мощные			+
<b>Требования к техническим характеристикам ПК</b>			
неприхотливые	+	+	
чувствительные			

## Продолжение таблицы 2

Критерии оценки	MS Access	MySQL	MS SQL Server
1	2	3	4
Требования к техническим характеристикам ПК			
необходимы мощные сервера			+
Возможные способы доступа к данным			
ODBC			
OLEDB/ADO	+	+	
DAO	+		
BDE			
Другое	+	+	+

Как видно из представленного анализа рассмотренные СУБД могут быть использованы для разработки базы данных автоматизации работы системы бронирования билетов. Окончательный выбор был сделан в пользу MS SQL Server, так как данная СУБД обладает высокими показателями производительности, способна хранить большие объемы информации без потери быстродействия при реализации доступа к данным из внешних источников.

### 1.3 Даталогическое проектирование

#### 1.3.1 Преобразование концептуальной модели в реляционную модель

При переходе от концептуальной модели предметной области к схеме БД (например, реляционной) используются три различных подхода, которые подробно описаны в списке использованных источников.

При первом подходе преобразование осуществляется вручную.

Второй подход основан на автоматизированной компиляции концептуальной модели предметной области в схему БД, в результате которой создается реляционная база данных в третьей нормальной форме.

В рамках курсового проекта будем использовать первый подход. При этом сущности модели «сущность-связь» соответствует отношение (таблица) реляционной БД. Связи между сущностями типа «один – к – одному (1:1)» и «один – ко – многим (1:M)» соответствуют связям между отношениями. Атрибуты, которые идентифицируют, определяют или моделируют сущности

в модели «сущность-связь» соответствуют атрибутам отношений.

На рисунке 2 представлена схема функциональных связей.

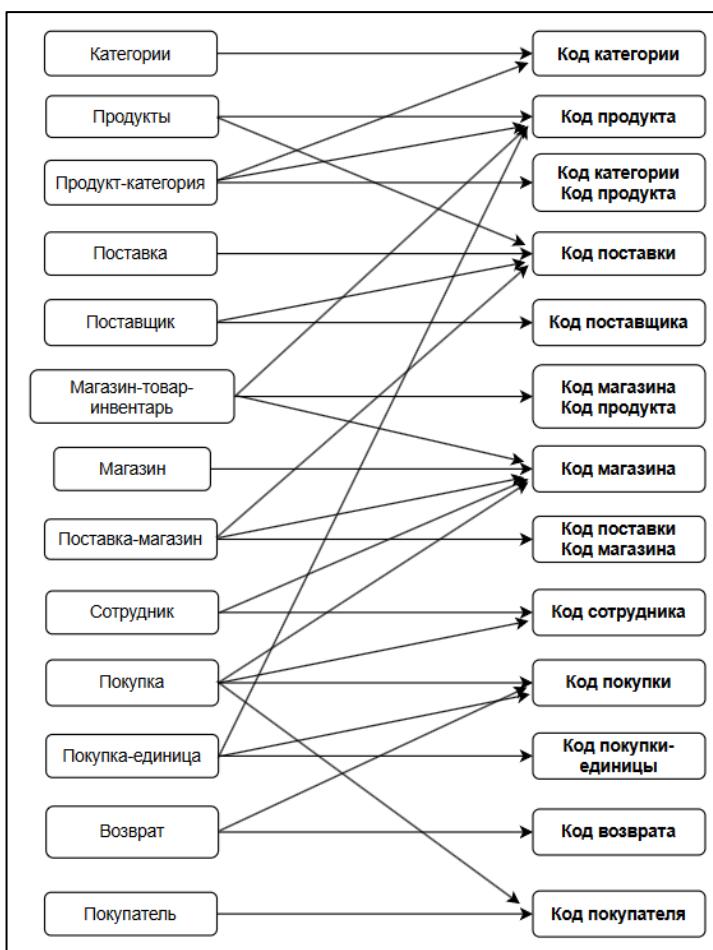


Рисунок 2 – Схема функциональных связей

При разработке реляционной базы данных проанализированы функциональные зависимости между атрибутами всех отношений и выбраны идентифицирующие атрибуты, которые в реляционной модели данных используются в качестве первичных ключей:

- Код категории;
- Код продукта;
- Код категории-Код продукта;
- Код поставки;
- Код поставщика;
- Код магазина-Код продукта;
- Код магазина;
- Код поставки-Код магазина;

- Код сотрудника;
- Код покупки;
- Код покупки-единицы;
- Код возврата;
- Код покупателя.

### 1.3.2 Нормализация базы данных

Изначально разрабатываемая БД представляла собой логически организованную структуру данных, представленную на таблице 3.

Отношения находятся в 3НФ, если оно находится в 2НФ и в нем нет зависимости атрибутов от возможных ключей (удаление данных, не зависящих от первичного ключа). Отношения находятся во 2НФ, если отношения находятся в 1НФ, и каждый не первичный атрибут формально полно зависит от каждого возможного ключа.

Таблица 3 – 3НФ базы данных

Продукты	Категории	Продукт-категория	Поставка	Поставщик
Код продукта	Код категории	Код категории	Код поставки	Код поставщика
Название	Название	Код продукта	Название	Название
Описание				Номер телефона
Цена				Почта
Код поставки				Код поставки
Изображение				
Магазин-товар-инвентарь	Магазин	Поставка-магазин	Сотрудник	Покупка
Код магазина	Код магазина	Код поставки	Код сотрудника	Код покупки
Код продукта	Название	Код магазина	Почта	Дата покупки
Количество	Телефон		ФИО	Код магазина
	Адрес		Код магазина	Код сотрудника
	Изображение		Дата	Код покупателя
			Должность	
			Дата рождения	
Покупка-единица	Возврат	Покупатель		
Номер	Код возврата	Номер покуп.		
Код покупки	Код продукта	ФИО		
Код продукта	Дата возврата	Почта		
Количество		Телефон		
Цена				

Таким образом, база данных «Сеть магазинов одежды» была приведена к ЗНФ.

На рисунке 3 представлена логическая модель базы данных, отражающая связи таблиц БД, разработанная с помощью «drawio».

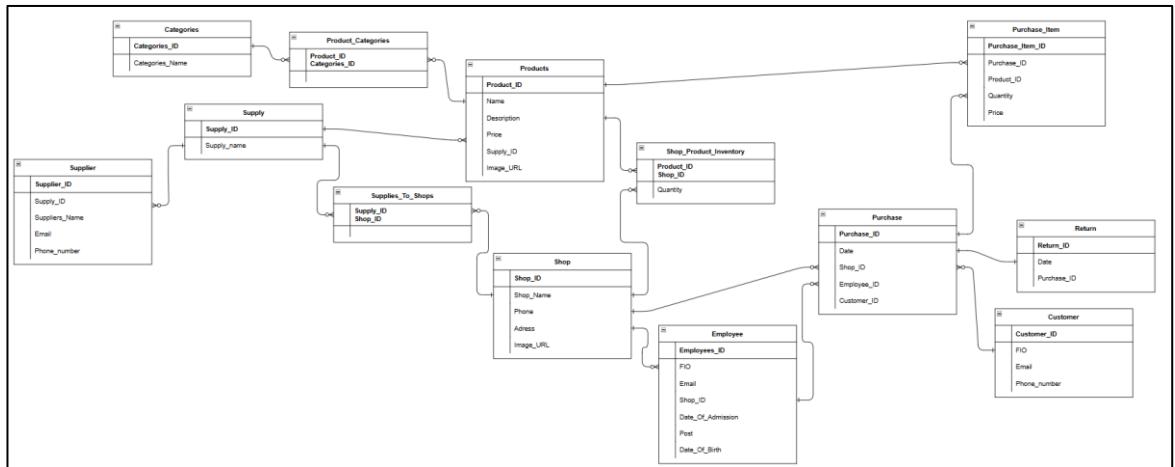


Рисунок 3 – Логическая модель базы данных

### 1.3.3 Определение характеристик атрибутов

Описание даталогической модели БД представлено в таблице 4.

Таблица 4 – Описание даталогической модели БД

Таблица	Поле	Тип поля	Размер поля	Допустимость неопределенных значений
Продукты	Id	Счётчик	-	NOT NULL
	Название	Короткий текст	30	NOT NULL
	Описание	Короткий текст	50	NOT NULL
	Цена	Decimal (10,2)	-	NOT NULL
	Код поставки	Счётчик	-	NOT NULL
	Изображение	Короткий текст	255	NOT NULL
Категории	Id	Счётчик	-	NOT NULL
	Название	Короткий текст	20	NOT NULL
Продукт-категория	Код категории	Счётчик	-	NOT NULL
	Код продукта	Счётчик	-	NOT NULL
Поставщик	Id поставщика	Счётчик	-	NOT NULL
	Название	Короткий текст	30	NOT NULL
	Номер телефона	Короткий текст	20	NOT NULL
	Почта	Короткий текст	30	NOT NULL
	Код поставки	Счётчик	-	NOT NULL
Поставка	Id	Счётчик	-	NOT NULL
	Название	Короткий текст	30	NOT NULL
Магазин	Id	Счётчик	-	NOT NULL
	Название	Короткий текст	50	NOT NULL
	Номер телефона	Короткий текст	20	NOT NULL
	Адрес	Короткий текст	50	NOT NULL

## Продолжение таблицы 4

Таблица	Поле	Тип поля	Размер поля	Допустимость неопределенных значений
Магазин	Изображение	Короткий текст	255	NOT NULL
Магазин-товар-инвентарь	Код магазина	Счётчик	-	NOT NULL
	Код продукта	Счётчик	-	NOT NULL
	Количество	Счётчик	-	NOT NULL
Поставка-магазин	Код поставки	Счётчик	-	NOT NULL
	Код магазина	Счётчик	-	NOT NULL
Сотрудник	Id	Счётчик	-	NOT NULL
	Почта	Короткий текст	30	NOT NULL
	ФИО	Короткий текст	50	NOT NULL
	Код магазина	Счётчик	-	NOT NULL
	Дата устройства	Дата	Дата	NOT NULL
	Дата рождения	Дата	Дата	NOT NULL
	Должность	Короткий текст	30	NOT NULL
	Id	Счётчик	-	NOT NULL
	ФИО	Короткий текст	50	NOT NULL
Покупатель	Почта	Короткий текст	30	NOT NULL
	Номер телефона	Короткий текст	30	NOT NULL
Покупка	Id	Счётчик	-	NOT NULL
	Дата покупки	Дата	Дата	NOT NULL
	Код магазина	Счётчик	-	NOT NULL
	Код сотрудника	Счётчик	-	NOT NULL
	Код покупателя	Счётчик	-	NOT NULL
Возврат	Id	Счётчик	-	NOT NULL
	Код продукта	Счётчик	-	NOT NULL
	Дата возврата	Дата	Дата	NOT NULL
Покупка-единица	Id	Счётчик	-	NOT NULL
	Код покупки	Счётчик	-	NOT NULL
	Код продукта	Счётчик	-	NOT NULL
	Количество	Счётчик	-	NOT NULL
	Цена	Decimal (10,2)	-	NOT NULL

На рисунке 4 представлена даталогическая модель БД.

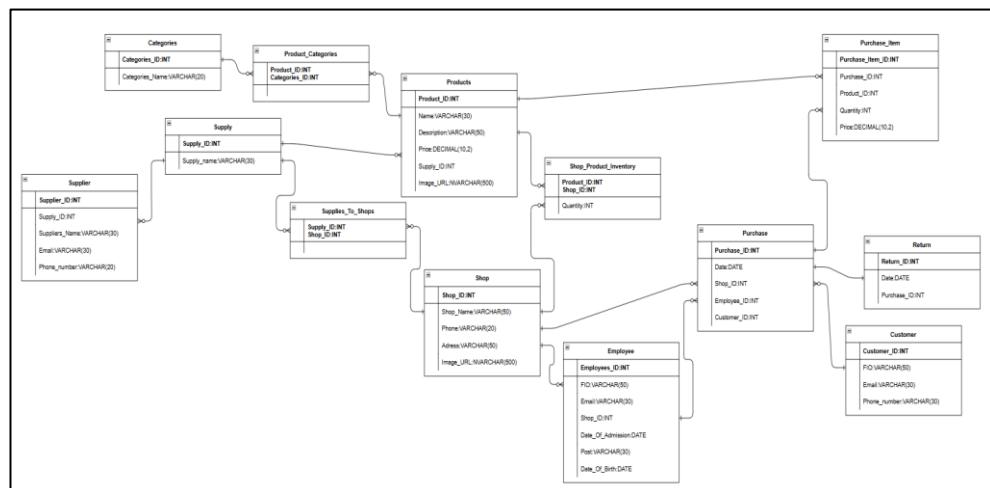


Рисунок 4 – Даталогическая модель БД

## 1.4 Создание БД в выбранной СУБД

Для создания базы данных была использована стандартная среда администрирования баз данных выбранной СУБД MS SQL Server – Среда администрирования MS SQL Server Management.

На рисунках 5 – 17 представлена структура созданных таблиц БД.

Column Name	Data Type	Allow Nulls
Categories_ID	int	<input type="checkbox"/>
Categories_Name	varchar(20)	<input type="checkbox"/>

Рисунок 5 – Проект таблицы «Категории»

Column Name	Data Type	Allow Nulls
Customer_ID	int	<input type="checkbox"/>
FIO	varchar(50)	<input type="checkbox"/>
Email	varchar(30)	<input type="checkbox"/>
Phone_number	varchar(30)	<input checked="" type="checkbox"/>

Рисунок 6 – Проект таблицы «Покупатель»

Column Name	Data Type	Allow Nulls
Employees_ID	int	<input type="checkbox"/>
FIO	varchar(50)	<input type="checkbox"/>
Email	varchar(30)	<input type="checkbox"/>
Shop_ID	int	<input type="checkbox"/>
Date_of_Admission	date	<input type="checkbox"/>
Post	varchar(30)	<input type="checkbox"/>
Date_of_Birth	date	<input type="checkbox"/>

Рисунок 7 – Проект таблицы «Сотрудник»

Column Name	Data Type	Allow Nulls
Product_ID	int	<input type="checkbox"/>
Categories_ID	int	<input type="checkbox"/>

Рисунок 8 – Проект таблицы «Продукт-категория»

Column Name	Data Type	Allow Nulls
Product_ID	int	<input type="checkbox"/>
Product_Name	varchar(30)	<input type="checkbox"/>
Description_product	varchar(50)	<input checked="" type="checkbox"/>
Price	decimal(10, 2)	<input type="checkbox"/>
Supply_ID	int	<input type="checkbox"/>
Image_URL	nvarchar(255)	<input checked="" type="checkbox"/>

Рисунок 9 – Проект таблицы «Продукты»

Column Name	Data Type	Allow Nulls
Purchase_ID	int	<input type="checkbox"/>
Date_of_purchase	date	<input type="checkbox"/>
Shop_ID	int	<input type="checkbox"/>
Employees_ID	int	<input type="checkbox"/>
Customer_ID	int	<input type="checkbox"/>

Рисунок 10 – Проект таблицы «Покупка»

Column Name	Data Type	Allow Nulls
Purchase_Item_ID	int	<input type="checkbox"/>
Purchase_ID	int	<input type="checkbox"/>
Product_ID	int	<input type="checkbox"/>
Quantity	int	<input type="checkbox"/>
Price	decimal(10, 2)	<input type="checkbox"/>

Рисунок 11 – Проект таблицы «Покупка-единица»

Column Name	Data Type	Allow Nulls
Return_ID	int	<input type="checkbox"/>
Date_of_return	date	<input type="checkbox"/>
Purchase_ID	int	<input type="checkbox"/>

Рисунок 12 – Проект таблицы «Возврат»

Column Name	Data Type	Allow Nulls
Shop_ID	int	<input type="checkbox"/>
Product_ID	int	<input type="checkbox"/>
Quantity	int	<input type="checkbox"/>

Рисунок 13 – Проект таблицы «Магазин-товар-инвентарь»

Column Name	Data Type	Allow Nulls
Shop_ID	int	<input type="checkbox"/>
Shop_Name	varchar(50)	<input type="checkbox"/>
Phone	varchar(20)	<input type="checkbox"/>
Adress	varchar(50)	<input type="checkbox"/>
Image_URL	nvarchar(255)	<input checked="" type="checkbox"/>

Рисунок 14 – Проект таблицы «Магазин»

Column Name	Data Type	Allow Nulls
Supplier_ID	int	<input type="checkbox"/>
Suppliers_Name	varchar(30)	<input type="checkbox"/>
Phone_number	varchar(20)	<input type="checkbox"/>
Email	varchar(30)	<input type="checkbox"/>
Supply_ID	int	<input type="checkbox"/>

Рисунок 15 – Проект таблицы «Поставщик»

Column Name	Data Type	Allow Nulls
Supply_ID	int	<input type="checkbox"/>
Shop_ID	int	<input type="checkbox"/>

Рисунок 16 – Проект таблицы «Поставка-магазин»

Column Name	Data Type	Allow Nulls
Supply_ID	int	<input type="checkbox"/>
Supply_Name	varchar(30)	<input type="checkbox"/>

Рисунок 17 – Проект таблицы «Поставка»

На рисунке 18 представлена модель базы данных, отражающая индексы и связи таблиц БД.

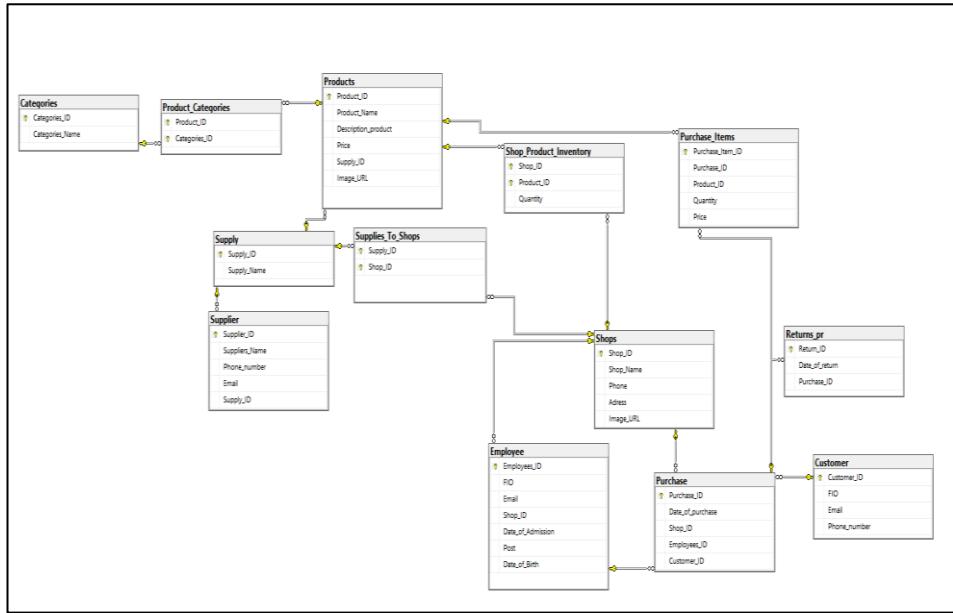


Рисунок 18 – Модель связей таблиц базы данных

## 1.5 Поддержка целостности данных

### 1.5.1 Декларативная поддержка ограничений целостности

Обеспечение целостности базы данных означает выполнение ряда ограничений, соблюдение которых необходимо для поддержания непротиворечивости хранимых данных. Среди ограничений целостности можно выделить ограничения диапазонов значений атрибутов отношений и структурные ограничения на кортежи отношений.

Первый тип ограничений предполагает контроль значений атрибутов отношений.

Структурные отношения фиксируют два требования целостности, которые должны поддерживаться реляционными СУБД: требование целостности сущностей и целостности ссылок. Каждому экземпляру сущности, представленному в отношении, соответствует только один его кортеж. Первое требование состоит в том, что любой кортеж отношения должен быть отличим от любого другого кортежа этого отношения, иными словами, любое отношение должно обладать первичным ключом.

Формулировка второго требования тесно связана с понятием внешнего ключа. Требование целостности по ссылкам состоит в том, что для каждого значения внешнего ключа родительской таблицы должна найтись строка в

дочерней таблице с таким же значением первичного ключа.

В MS SQL Server поддержка ссылочной целостности выполняется с помощью настройки связей таблиц, например, в диаграмме связей.

В данном курсовом проекте ссылочная целостность формировалась с помощью настройки связей таблиц так, как показано на рисунке 19.

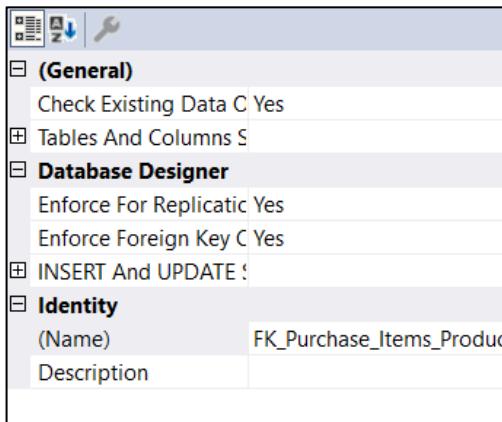


Рисунок 19 – Настройка связей таблиц БД для обеспечения целостности

Реализация операций над данными: в информационной системе бронирования ЖД билетов были созданы 20 запросов.

- **INSERT**

```
INSERT INTO Supply (Supply_ID, Supply_Name) VALUES  
(1, 'Коллекция Amor'),  
(2, 'Летние новинки'),  
(3, 'Осенняя коллекция InOctober'),  
(4, 'Зимняя коллекция Snow'),  
(5, 'Молодежный стиль');
```

Supply_ID	Supply_Name
1	Коллекция Amor
2	Летние новинки
3	Осенняя коллекция InOctober
4	Зимняя коллекция Snow
5	Молодежный стиль

Рисунок 20 – 1 запрос

```
INSERT INTO Shops (Shop_ID, Shop_Name, Phone, Adress, Image_URL)  
VALUES  
(1, 'Aure - Puskin', '+79161234567', 'ул. Пушкина,  
10', '/images/shops/e2f10506ccd106454ec0abd1e0b80f40.jpg'),  
(2, 'Aure - Len', '+79167654321', 'пр. Ленина,  
25', '/images/shops/484fff96a5e0cbefaf141161371050f.jpg'),  
(3, 'Aure - Atrasev', '+79169998877', 'ул. Атрасева,  
5', '/images/shops/577e33a48f5fa6dc79c5ffbd6efae032.jpg'),  
(4, 'Aure - Revolution', '+79165554433', 'пл. Революции,  
3', '/images/shops/a2d0fff37bd4ed20305f37828b0c2959.jpg')
```

	Shop_ID	Shop_Name	Phone	Adress	Image_URL
1	1	Aure - Puskin	+79161234567	ул. Пушкина, 10	/images/shops/e2f10506cc0106454ec0abd1e0b8040.jpg
2	2	Aure - Len	+79167654321	пр. Ленина, 25	/images/shops/484ff96a5e0cbefbf141161371050.jpg
3	3	Aure - Altsev	+79169998877	ул. Атласова, 5	/images/shops/577e33a495fa6dc795fb6d6fae032.jpg
4	4	Aure - Revolution	+79165554433	пл. Революции, 3	/images/shops/a200ff37bd4ed20305f3782b0c2959.jpg
5	5	Aure - Peace	+79162223344	ул. Мира, 15	/images/shops/e2f10506cc0106454ec0abd1e0b8040.jpg

Рисунок 21 – 2 запрос

```
INSERT INTO Employee (Employees_ID, FIO, Email, Shop_ID,
Date_of_Admission, Post, Date_of_Birth)
VALUES
(1, 'Иванова Анна Сергеевна', 'ivanova@mail.com', 1, '2020-05-15', 'Продавец', '1990-08-20'),
(2, 'Петров Дмитрий Игоревич', 'petrov@mail.com', 2, 'Администратор', '1985-11-15'),
(3, 'Сидорова Елена Викторовна', 'sidorova@mail.com', 3, '2019-07-22', 'Менеджер', '1988-04-30'),
(4, 'Козлов Алексей Петрович', 'kozlov@mail.com', 1, '2022-01-05', 'Кассир', '1995-09-12'),
(5, 'Михайлова Ольга Дмитриевна', 'mihailova@mail.com', 4, '2020-11-18', 'Продавец', '1992-07-25'),
(6, 'Николаев Иван Сергеевич', 'nikolaev@mail.com', 5, '2021-09-30', 'Консультант', '1987-03-08'),
(7, 'Нигаматзянова Камиля Завдатовна', 'niga@mail.com', 5, '2024-10-22', 'Владелец', '1999-10-22'),
(8, 'Кошкин Дмитрий Андреевич', 'koshkin@mail.com', 5, '2021-09-30', 'Консультант', '1952-03-08'),
(9, 'Никулин Никита Нурисламович', 'nurisSvai@mail.com', 5, '2021-09-30', 'Кассир', '1987-05-20');
```

	Employees_ID	FIO	Email	Shop_ID	Date_of_Admission	Post	Date_of_Birth
1	1	Иванова Анна Сергеевна	ivanova@mail.com	1	2020-05-15	Продавец	1990-08-20
2	2	Петров Дмитрий Игоревич	petrov@mail.com	2	2021-02-10	Администратор	1985-11-15
3	3	Сидорова Елена Викторовна	sidorova@mail.com	3	2019-07-22	Менеджер	1988-04-30
4	4	Козлов Алексей Петрович	kozlov@mail.com	1	2022-01-05	Кассир	1995-09-12
5	5	Михайлова Ольга Дмитриевна	mihailova@mail.com	4	2020-11-18	Продавец	1992-07-25
6	6	Николаев Иван Сергеевич	nikolaev@mail.com	5	2021-09-30	Консультант	1987-03-08
7	7	Нигаматзянова Камиля Завдатовна	niga@mail.com	5	2024-10-22	Владелец	1999-10-22
8	8	Кошкин Дмитрий Андреевич	koshkin@mail.com	5	2021-09-30	Консультант	1952-03-08
9	9	Никулин Никита Нурисламович	nurisSvai@mail.com	5	2021-09-30	Кассир	1987-05-20

Рисунок 22 – 3 запрос

```
INSERT INTO Customer (Customer_ID, FIO, Email, Phone_number)
VALUES
(1, 'Смирнова Мария Алексеевна', 'smirnova@mail.com', '+79161112233'),
(2, 'Волков Андрей Игоревич', 'volkov@mail.com', '+79162223344'),
(3, 'Белова Екатерина Сергеевна', 'belova@mail.com', '+79163334455'),
(4, 'Григорьев Павел Дмитриевич', 'grigorev@mail.com', '+79164445566'),
(5, 'Кудрявцева Анна Викторовна', 'kudryavceva@mail.com', '+79165556677'),
(6, 'Орлов Денис Александрович', 'orlov@mail.com', '+79166667788');
```

Customer_ID	FIO	Email	Phone_number
1	Смирнова Мария Алексеевна	smirnova@mail.com	+7911112233
2	Волков Андрей Игоревич	volkov@mail.com	+79162223344
3	Белова Екатерина Сергеевна	belova@mail.com	+79163334455
4	Григорьев Павел Дмитриевич	griгорев@mail.com	+79164445566
5	Кудрявцева Анна Викторовна	kudryavtseva@mail.com	+79165556677
6	Орлов Денис Александрович	orlov@mail.com	+79166667788

Рисунок 23 – 4 запрос

```
INSERT INTO Purchase (Purchase_ID, Date_of_purchase, Shop_ID, Employees_ID, Customer_ID) VALUES
(1, '2023-01-15', 1, 1, 1),
(2, '2023-01-16', 1, 1, 2),
(3, '2023-01-17', 2, 2, 3),
(4, '2023-01-18', 2, 2, 4),
(5, '2023-01-19', 3, 3, 5),
(6, '2023-01-20', 3, 3, 6),
(7, '2023-01-21', 4, 4, 1),
(8, '2023-01-22', 4, 4, 2),
(9, '2023-01-23', 5, 5, 3),
(10, '2023-01-24', 5, 6, 4);
```

Purchase_ID	Date_of_purchase	Shop_ID	Employees_ID	Customer_ID
1	2023-01-15	1	1	1
2	2023-01-16	1	1	2
3	2023-01-17	2	2	3
4	2023-01-18	2	2	4
5	2023-01-19	3	3	5
6	2023-01-20	3	3	6
7	2023-01-21	4	4	1
8	2023-01-22	4	4	2
9	2023-01-23	5	5	3
10	2023-01-24	5	6	4

Рисунок 24 – 5 запрос

## • UPDATE

```
UPDATE Products SET Price = 4500.00 WHERE Product_ID = 1;
```

Product_ID	Product_Name	Description_product	Price	Supply_ID	Image_URL
1	Джинсы женские	Джинсы клеш темно-синего цвета	4000.00	3	NULL
2	Футболка мужская	Хлопковая футболка белого цвета	2000.00	1	NULL
3	Кроссовки	Спортивные кроссовки унисекс	6000.00	2	NULL
4	Сумка	Сумка женская кросс-боди	7000.00	4	NULL
Product_ID	Product_Name	Description_product	Price	Supply_ID	Image_URL
1	Джинсы женские	Джинсы клеш темно-синего цвета	4500.00	3	NULL
2	Футболка мужская	Хлопковая футболка белого цвета	2000.00	1	NULL
3	Кроссовки	Спортивные кроссовки унисекс	6000.00	2	NULL
4	Сумка	Сумка черная кросс-боди	7000.00	4	NULL
5	Юбка миди	Юбка плиссе черного цвета	3500.00	5	NULL
6	Блузка	Блузка белая с кружевными вставками	2800.00	1	NULL

Рисунок 25 – 6 запрос

```
UPDATE Employee SET Post = 'Генеральный директор' WHERE Employees_ID = 7;
```

Employees_ID	FIO	Email	Shop_ID	Date_of_Admission	Post	Date_of_Birth
1	Иванова Анна Сергеевна	ivanova@mail.com	1	2020-05-15	Продавец	1990-08-20
2	Петров Дмитрий Игоревич	petrov@mail.com	2	2021-02-10	Администратор	1985-11-15
3	Сидорова Елена Викторовна	sidorova@mail.com	3	2019-07-22	Менеджер	1988-04-30
4	Козлов Алексей Тетярович	kozlov@mail.com	1	2022-01-05	Кассир	1995-09-12
5	Михайлова Ольга Дмитриевна	mikhailova@mail.com	4	2020-11-18	Продавец	1992-07-25
6	Николаев Иван Сергеевич	nikolaev@mail.com	5	2021-09-30	Консультант	1987-03-08
7	Нигаматзянова Камиля Рифатовна	niga@mail.com	5	2024-10-22	Генеральный директор	1999-10-22
8	Кошкин Дмитрий Андреевич	koshkin@mail.com	5	2021-09-30	Консультант	1952-03-08

Рисунок 26 – 7 запрос

```
UPDATE Shop_Product_Inventory SET Quantity = 15 WHERE Shop_ID = 1 AND Product_ID = 3;
```

	Shop_ID	Product_ID	Quantity
1	1	1	15
2	1	2	20
3	1	3	10
4	1	4	8
5	1	5	12

	Shop_ID	Product_ID	Quantity
1	1	1	15
2	1	2	20
3	1	3	15
4	1	4	8

Рисунок 27 – 8 запрос

```
UPDATE Customer SET Email = 'volkov.andrey@newmail.com' WHERE Customer_ID = 2;
```

Customer_ID	FIO	Email	Phone_number
1	Смирнова Мария Алексеевна	smirnova@mail.com	+79161112233
2	Волков Андрей Игоревич	volkov@mail.com	+79162223344
3	Белова Екатерина Сергеевна	belova@mail.com	+79163334455
4	Григорьев Павел Дмитриевич	grigorev@mail.com	+79164445566
5	Кудрявцева Анна Викторовна	kudryavtseva@mail.com	+79165556677
6	Орлов Денис Александрович	orlov@mail.com	+79166667788

Customer_ID	FIO	Email	Phone_number
1	Смирнова Мария Алексеевна	smirnova@mail.com	+79161112233
2	Волков Андрей Игоревич	volkov.andrey@newmail.com	+79162223344
3	Белова Екатерина Сергеевна	belova@mail.com	+79163334455
4	Григорьев Павел Дмитриевич...	grigorev@mail.com	+79164445566
5	Кудрявцева Анна Викторовна	kudryavtseva@mail.com	+79165556677
6	Орлов Денис Александрович	orlov@mail.com	+79166667788

Рисунок 28 – 9 запрос

```
UPDATE Products SET Description_product = 'Юбка плиссе черного цвета, длина до колена' WHERE Product_ID = 5;
```

Product_ID	Product_Name	Description_product	Price	Supply_ID	Image_URL
1	Джинсы женские	Джинсы клеш темно-синего цвета	4500.00	3	NULL
2	Футболка мужская	Хлопковая футболка белого цвета	2000.00	1	NULL
3	Кроссовки	Спортивные кроссовки унисекс	6000.00	2	NULL
4	Сумка	Сумка черная кросс-боди	7000.00	4	NULL
5	Юбка миди	Юбка плиссе черного цвета	3500.00	5	NULL
6	Блузка	Блузка белая с кружевными вставками	2800.00	1	NULL

Product_ID	Product_Name	Description_product	Price	Supply_ID	Image_URL
1	Джинсы женские	Джинсы клеш темно-синего цвета	4500.00	3	NULL
2	Футболка мужская	Хлопковая футболка белого цвета	2000.00	1	NULL
3	Кроссовки	Спортивные кроссовки унисекс	6000.00	2	NULL
4	Сумка	Сумка черная кросс-боди	7000.00	4	NULL
5	Юбка миди	Юбка плиссе черного цвета, длина до колена	3500.00	5	NULL
6	Блузка	Блузка белая с кружевными вставками	2800.00	1	NULL
7	Платье летнее	Платье хлопковое с цветочным принтом	4500.00	2	NULL
8	Кардиган	Кардиган удлиненный бежевый	5000.00	3	NULL
9	Брюки офисные	Брюки классические серые	3800.00	4	NULL

Рисунок 29 – 10 запрос

- SELECT

```
SELECT p.Product_Name, c.Categories_Name FROM Products p JOIN Product_Categories pc
ON p.Product_ID = pc.Product_ID JOIN Categories c ON
pc.Categories_ID = c.Categories_ID;
```

	Product_Name	Categories_Name
1	Джинсы женские	Женская одежда
2	Футболка мужская	Мужская одежда
3	Кроссовки	Обувь
4	Сумка	Аксессуары
5	Юбка миди	Женская одежда
6	Блузка	Женская одежда
7	Платье летнее	Женская одежда
8	Кардиган	Женская одежда
9	Брюки офисные	Женская одежда
10	Топ	Женская одежда
11	Куртка джинсовая	Женская одежда
12	Свитшот	Женская одежда
13	Леггинсы	Женская одежда
14	Пальто	Женская одежда
15	Рубашка	Мужская одежда
16	Джинсы мужские	Мужская одежда
17	Свитер	Мужская одежда
18	Брюки чинос	Мужская одежда
19	Куртка ветровка	Мужская одежда
20	Футболка поло	Мужская одежда
21	Толстовка	Мужская одежда
22	Шорты	Мужская одежда
23	Пиджак	Мужская одежда
24	Термобелье	Мужская одежда

Рисунок 30 – 11 запрос

```
SELECT e.FIO, e.Post, s.Shop_Name FROM Employee e JOIN Shops s
ON e.Shop_ID = s.Shop_ID;
```

	FIO	Post	Shop_Name
1	Иванова Анна Сергеевна	Продавец	Aure - Puskin
2	Петров Дмитрий Игоревич	Администратор	Aure - Len
3	Сидорова Елена Викторовна	Менеджер	Aure - Atrasev
4	Козлов Алексей Петрович	Кассир	Aure - Puskin
5	Михайлова Ольга Дмитриевна	Продавец	Aure - Revolution
6	Николаев Иван Сергеевич	Консультант	Aure - Peace
7	Нигаматзянова Камиля Завдатовна	Генеральный директор	Aure - Peace
8	Кошкин Дмитрий Андреевич	Консультант	Aure - Peace
9	Никулин Никита Нурисламович	Кассир	Aure - Peace

Рисунок 31 – 12 запрос

```
SELECT sp.Supply_Name, s.Suppliers_Name FROM Supply sp JOIN
Supplier s ON sp.Supply_ID = s.Supply_ID;
```

	Supply_Name	Suppliers_Name
1	Коллекция Amor	Fashion Forward Inc.
2	Летние новинки	Trendy Styles Ltd.
3	Осенняя коллекция InOctober	Urban Wear Co.
4	Зимняя коллекция Snow	Classic Apparel LLC
5	Молодежный стиль	Modern Outfitters

Рисунок 32 – 13 запрос

```
SELECT pu.Purchase_ID, c.FIO AS Customer, s.Shop_Name,
pu.Date_of_purchase
FROM Purchase pu JOIN Customer c ON pu.Customer_ID =
c.Customer_ID JOIN Shops s ON pu.Shop_ID = s.Shop_ID;
```

Purchase_ID	Customer	Shop_Name	Date_of_purchase
1	Смирнова Мария Алексеевна	Aure - Pushkin	2023-01-15
2	Волков Андрей Игоревич	Aure - Pushkin	2023-01-16
3	Белова Екатерина Сергеевна	Aure - Len	2023-01-17
4	Григорьев Павел Дмитриевич	Aure - Len	2023-01-18
5	Кудрявцева Анна Викторовна	Aure - Atrasev	2023-01-19
6	Орлов Денис Александрович	Aure - Atrasev	2023-01-20
7	Смирнова Мария Алексеевна	Aure - Revolution	2023-01-21
8	Волков Андрей Игоревич	Aure - Revolution	2023-01-22
9	Белова Екатерина Сергеевна	Aure - Peace	2023-01-23
10	Григорьев Павел Дмитриевич	Aure - Peace	2023-01-24

Рисунок 33 – 14 запрос

```
SELECT      s.Shop_Name,      p.Product_Name,      spi.Quantity      FROM
Shop_Product_Inventory spi JOIN Shops s
ON spi.Shop_ID = s.Shop_ID JOIN Products p ON spi.Product_ID =
p.Product_ID ;
```

Shop_Name	Product_Name	Quantity
1 Aure - Atrasev	Куртка джинсовая	7
2 Aure - Atrasev	Свитшот	13
3 Aure - Atrasev	Леггинсы	20
4 Aure - Atrasev	Пальто	5
5 Aure - Atrasev	Рубашка	15
6 Aure - Len	Блузка	18
7 Aure - Len	Платье летнее	14
8 Aure - Len	Кардиган	9
9 Aure - Len	Брюки офисные	11
10 Aure - Len	Топ	16
11 Aure - Peace	Толстовка	9
12 Aure - Peace	Шорты	17
13 Aure - Peace	Пиджак	4
14 Aure - Peace	Термобелье	11
15 Aure - Peace	Туфли	8
16 Aure - Pushkin	Джинсы женские	15
17 Aure - Pushkin	Футболка мужская	20
18 Aure - Pushkin	Кроссовки	15
19 Aure - Pushkin	Сумка	8
20 Aure - Pushkin	Юбка мини	12
21 Aure - Revolution	Джинсы мужские	12
22 Aure - Revolution	Свитер	8
23 Aure - Revolution	Брюки чинос	14

Рисунок 34 – 15 запрос

- DELETE

```
DELETE FROM Products WHERE Product_ID = 10;
```

Product_ID	Product_Name	Description_product	Price	Supply_ID	Image_URL
1 10	Топ	Топ облегающий черный	1500.00	5	NULL
Product_ID	Product_Name	Description_product	Price	Supply_ID	Image_URL

Рисунок 35– 16 запрос

```
DELETE FROM Returns_pr WHERE Date_of_return BETWEEN '2023-01-01'
AND '2023-01-31';
```

Results			Messages		
	Return_ID	Date_of_return	Purchase_ID		
1	1	2023-01-20	1		
2	2	2023-01-25	3		
3	3	2023-01-28	5		
4	4	2023-02-01	7		
5	5	2023-02-05	9		

	Return_ID	Date_of_return	Purchase_ID		
1	4	2023-02-01	7		
2	5	2023-02-05	9		

Рисунок 36 – 17 запрос

```
DELETE FROM Employee WHERE FIO LIKE '%Кошкин%';
```

Results			Messages				
	Employees_ID	FIO	Email	Shop_ID	Date_of_Admission	Post	Date_of_Birth
1	1	Иванова Анна Сергеевна	ivanova@mail.com	1	2020-05-15	Продавец	1990-08-20
2	2	Петров Дмитрий Игоревич	petrov@mail.com	2	2021-02-10	Администратор	1985-11-15
3	3	Сидорова Елена Викторовна	sidorova@mail.com	3	2019-07-22	Менеджер	1988-04-30
4	4	Козлов Алексей Петрович	kozlov@mail.com	1	2022-01-05	Кассир	1995-09-12
5	5	Михайлова Ольга Дмитриевна	mihailova@mail.com	4	2020-11-18	Продавец	1992-07-25
6	6	Николаев Иван Сергеевич	nikolaev@mail.com	5	2021-09-30	Консультант	1987-03-08
7	7	Нигаматзянова Камиля Завдатовна	niga@mail.com	5	2024-10-22	Генеральный директор	1999-10-22
8	8	Кошкин Дмитрий Андреевич	koshkin@mail.com	5	2021-09-30	Консультант	1952-03-08
9	9	Никулин Никита Нурисламович	nurisSvai@mail.com	5	2021-09-30	Кассир	1987-05-20

	Employees_ID	FIO	Email	Shop_ID	Date_of_Admission	Post	Date_of_Birth
1	1	Иванова Анна Сергеевна	ivanova@mail.com	1	2020-05-15	Продавец	1990-08-20
2	2	Петров Дмитрий Игоревич	petrov@mail.com	2	2021-02-10	Администратор	1985-11-15
3	3	Сидорова Елена Викторовна	sidorova@mail.com	3	2019-07-22	Менеджер	1988-04-30
4	4	Козлов Алексей Петрович	kozlov@mail.com	1	2022-01-05	Кассир	1995-09-12
5	5	Михайлова Ольга Дмитриевна	mihailova@mail.com	4	2020-11-18	Продавец	1992-07-25
6	6	Николаев Иван Сергеевич	nikolaev@mail.com	5	2021-09-30	Консультант	1987-03-08
7	7	Нигаматзянова Камиля Завдатовна	niga@mail.com	5	2024-10-22	Генеральный директор	1999-10-22
8	9	Никулин Никита Нурисламович	nurisSvai@mail.com	5	2021-09-30	Кассир	1987-05-20

Рисунок 37 – 18 запрос

```
DELETE FROM Products WHERE Price < 2000;
```

Results			Messages				
	Product_ID	Product_Name	Description_product	Price	Supply_ID	Image_URL	
1	36	Шарф	Шарф шерстяной серый	1500.00	4	NULL	
2	42	Галстук	Галстук шелковый	1800.00	2	NULL	

Product_ID	Product_Name	Description_product	Price	Supply_ID	Image_URL

Рисунок 38 – 19 запрос

```
DELETE FROM Customer WHERE Email = 'belova@mail.com';
```

Results		Messages		
	Customer_ID	FIO	Email	Phone_number
1	1	Смирнова Мария Алексеевна	smirnova@mail.com	+79161112233
2	2	Волков Андрей Игоревич	volkov.andrey@newmail.com	+79162223344
3	3	Белова Екатерина Сергеевна	belova@mail.com	+79163334455
4	4	Григорьев Павел Дмитриевич	grigorev@mail.com	+79164445566
5	5	Кудрявцева Анна Викторовна	kudryavceva@mail.com	+79165556677
6	6	Орлов Денис Александрович	orlov@mail.com	+79166667788

	Customer_ID	FIO	Email	Phone_number
1	1	Смирнова Мария Алексеевна	smirnova@mail.com	+79161112233
2	2	Волков Андрей Игоревич	volkov.andrey@newmail.com	+79162223344
3	4	Григорьев Павел Дмитриевич	grigorev@mail.com	+79164445566
4	5	Кудрявцева Анна Викторовна	kudryavceva@mail.com	+79165556677
5	6	Орлов Денис Александрович	orlov@mail.com	+79166667788

Рисунок 39 – 20 запрос

### 1.5.2 Процедурная поддержка ограничений целостности

Для организации процедурной поддержки ограничения целостности данных разработаны хранимые процедуры и триггеры.

**1 хранимая процедура.** Она называется AddCustomer\_ и предназначена для добавления нового клиента в таблицу Customer.

Процедура принимает четыре параметра:

- @Customer\_ID — идентификатор клиента;
- @FIO — полное имя клиента;
- @Email — адрес электронной почты;
- @Phone\_number — номер телефона.

При вызове с указанными значениями параметров процедура выполняет операцию вставки (INSERT) новой записи в таблицу Customer, добавляя данные о клиенте.

Таким образом, процедура позволяет регистрировать новых клиентов в системе.

Листинг

```
GO
CREATE PROCEDURE AddCustomer_
    @Customer_ID INT,
    @FIO VARCHAR(50),
```

## Продолжение листинга

```
@Email VARCHAR(30),  
@Phone_number VARCHAR(30)  
AS  
BEGIN  
    INSERT INTO Customer (Customer_ID, FIO, Email, Phone_number)  
    VALUES (@Customer_ID, @FIO, @Email, @Phone_number);  
END;
```

```
CREATE PROCEDURE _AddCustomer  
    @Customer_ID INT,  
    @FIO VARCHAR(50),  
    @Email VARCHAR(30),  
    @Phone_number VARCHAR(30)  
AS  
BEGIN  
    INSERT INTO Customer (Customer_ID, FIO, Email, Phone_number)  
    VALUES (@Customer_ID, @FIO, @Email, @Phone_number);  
END;  
EXEC _AddCustomer 21, 'Микаэлян Артур', 'mika@test.com', '+79192564230';
```

Messages  
Commands completed successfully.  
Completion time: 2025-05-14T18:31:31.3287628+03:00

Рисунок 40 – Создание 1 хранимой процедуры

```
SELECT * FROM Customer
```

	Customer_ID	FIO	Email	Phone_number
1	1	Смирнова Мария Алексеевна	smirnova@mail.com	+79161112233
2	2	Волков Андрей Игоревич	volkov.andrey@newmail.com	+79162223344
3	4	Григорьев Павел Дмитриевич	grigorev@mail.com	+79164445566
4	5	Кудрявцева Анна Викторовна	kudryavtseva@mail.com	+79165556677
5	6	Орлов Денис Александрович	orlov@mail.com	+79166667788
6	20	Иванов Сергей	ivanov@test.com	+79160001122
7	21	Микаэлян Артур	mika@test.com	+79192564230

Рисунок 41 – Выполнение 1 хранимой процедуры

**2 хранимая процедура.** Она называется `_UpdateShopPhone` и предназначена для обновления номера телефона магазина в таблице `Shops`.

Процедура принимает два параметра:

- `@ShopName` — название магазина;
- `@NewPhone` — новый номер телефона.

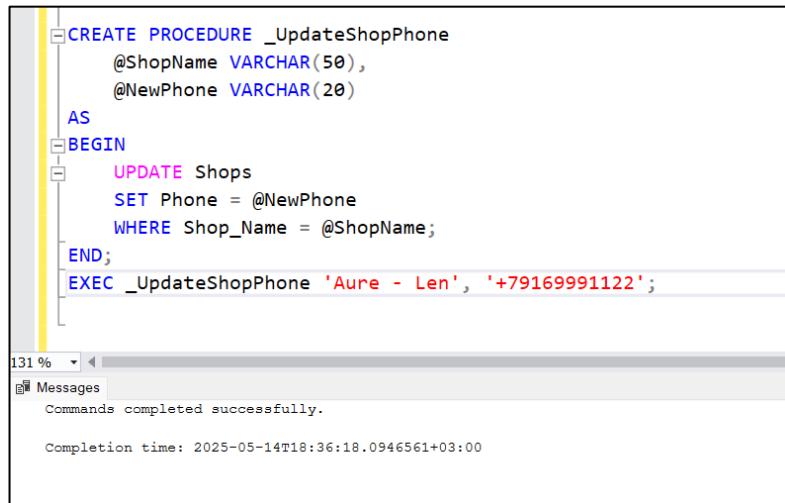
При вызове процедура выполняет обновление (UPDATE) записи в таблице `Shops`, устанавливая новый номер телефона для магазина с указанным

названием.

Таким образом, процедура позволяет актуализировать контактную информацию магазинов.

### Листинг

```
GO
CREATE PROCEDURE _UpdateShopPhone
    @ShopName VARCHAR(50),
    @NewPhone VARCHAR(20)
AS
BEGIN
    UPDATE Shops
    SET Phone = @NewPhone
    WHERE Shop_Name = @ShopName;
END;
```

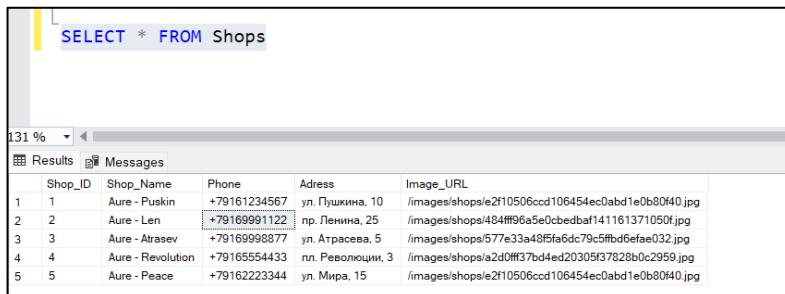


```
CREATE PROCEDURE _UpdateShopPhone
    @ShopName VARCHAR(50),
    @NewPhone VARCHAR(20)
AS
BEGIN
    UPDATE Shops
    SET Phone = @NewPhone
    WHERE Shop_Name = @ShopName;
END;
EXEC _UpdateShopPhone 'Aure - Len', '+79169991122';

131 % ▶
Messages
Commands completed successfully.

Completion time: 2025-05-14T18:36:18.0946561+03:00
```

Рисунок 42 – Создание 2 хранимой процедуры



	Shop_ID	Shop_Name	Phone	Adress	Image_URL
1	1	Aure - Puskin	+79161234567	ул. Пушкина, 10	/images/shops/e2f10506cd106454ec0abd1e0b80f40.jpg
2	2	Aure - Len	+79169991122	пр. Ленина, 25	/images/shops/484ff96a5e0cbdbaf141161371050f.jpg
3	3	Aure - Atrasev	+79169998877	ул. Атрасеева, 5	/images/shops/57e33a48f5fa6dc79c5fbde6fae032.jpg
4	4	Aure - Revolution	+79165554433	пл. Революции, 3	/images/shops/a2d0ff37bd4ed20305f37828b0c2959.jpg
5	5	Aure - Peace	+79162223344	ул. Мира, 15	/images/shops/e2f10506cd106454ec0abd1e0b80f40.jpg

Рисунок 43 – Выполнение 2 хранимой процедуры

**3 хранимая процедура.** Она называется DeleteProductByName и предназначена для удаления товара из таблицы Products по его названию.

Процедура принимает один параметр:

- @ProductName — название товара.

При вызове с заданным параметром процедура выполняет удаление (DELETE) записи в таблице Products, где имя товара соответствует переданному значению.

Таким образом, процедура позволяет удалять устаревшие или ненужные товары из базы данных.

### Листинг

```
GO
CREATE PROCEDURE DeleteProductByName
    @ProductName VARCHAR(30)
AS
BEGIN
    DELETE FROM Products
    WHERE Product_Name = @ProductName;
END;
```

The screenshot shows the SQL Server Management Studio interface. In the center pane, a code editor displays the T-SQL script for creating a stored procedure:

```
CREATE PROCEDURE DeleteProductByName
    @ProductName VARCHAR(30)
AS
BEGIN
    DELETE FROM Products
    WHERE Product_Name = @ProductName;
END;
```

Below the code editor, the 'Messages' pane shows the execution results:

```
Commands completed successfully.

Completion time: 2025-05-14T18:39:32.1159691+03:00
```

Рисунок 44 – Создание 3 хранимой процедуры

The screenshot shows the SQL Server Management Studio interface. In the center pane, a code editor displays the T-SQL script for executing the stored procedure:

```
SELECT * FROM Products
EXEC DeleteProductByName 'Зонт';
SELECT * FROM Products
```

Below the code editor, the 'Results' pane shows the product data before and after the deletion:

Product_ID	Product_Name	Description_product	Price	Supply_ID	Image_URL
34	35	Ремень	2000.00	3	NULL
35	37	Перчатки	2500.00	5	NULL
36	38	Шляпа	3000.00	5	NULL
37	39	Кошелек	3500.00	1	NULL
38	40	Очки	4000.00	1	NULL
39	41	Зонт	2000.00	2	NULL
40	43	Часы	10000.00	2	NULL
41	44	Браслет	2500.00	3	NULL

Product_ID	Product_Name	Description_product	Price	Supply_ID	Image_URL
33	34	Слипсоны	4000.00	2	NULL
34	35	Ремень	2000.00	3	NULL
35	37	Перчатки	2500.00	5	NULL
36	38	Шляпа	3000.00	5	NULL
37	39	Кошелек	3500.00	1	NULL
38	40	Очки	4000.00	1	NULL
39	43	Часы	10000.00	2	NULL

Рисунок 45 – Выполнение 3 хранимой процедуры

**4 хранимая процедура.** Она называется \_GetProductsByCategory и предназначена для получения списка товаров, относящихся к определённой категории.

Процедура принимает один параметр:

- @CategoryName — название категории товаров.

При вызове процедура выполняет выборку (SELECT) первых 10 товаров из таблицы Products, которые связаны с заданной категорией через таблицу связей Product\_Categories. Результат включает идентификатор товара, название и цену.

Таким образом, процедура позволяет просматривать товары нужной категории с ограничением по количеству (максимум 10 записей).

Листинг

```
GO
CREATE PROCEDURE _GetProductsByCategory
    @CategoryName VARCHAR(20)
AS
BEGIN
    SELECT TOP 10 p.Product_ID, p.Product_Name, p.Price
    FROM Products p
    JOIN Product_Categories pc ON p.Product_ID = pc.Product_ID
    JOIN Categories c ON pc.Categories_ID = c.Categories_ID
    WHERE c.Categories_Name = @CategoryName;
END;
```

The screenshot shows the SQL Server Management Studio interface. A new stored procedure named '\_GetProductsByCategory' is being created. The code is as follows:

```
CREATE PROCEDURE _GetProductsByCategory
    @CategoryName VARCHAR(20)
AS
BEGIN
    SELECT TOP 10 p.Product_ID, p.Product_Name, p.Price
    FROM Products p
    JOIN Product_Categories pc ON p.Product_ID = pc.Product_ID
    JOIN Categories c ON pc.Categories_ID = c.Categories_ID
    WHERE c.Categories_Name = @CategoryName;
END;
```

The code is highlighted with syntax coloring. Below the code, a message window displays:

31 %

Messages

Commands completed successfully.

Completion time: 2025-05-14T18:51:29.4286767+03:00

Рисунок 46 – Создание 4 хранимой процедуры

The screenshot shows a SQL query window with the following content:

```
EXEC _GetProductsByCategory 'Аксессуары';
```

Below the query, a progress bar indicates "31 %". The results tab is selected, displaying the following data:

	Product_ID	Product_Name	Price
1	4	Сумка	7000.00
2	35	Ремень	2000.00
3	37	Перчатки	2500.00
4	38	Шляпа	3000.00
5	39	Кошелек	3500.00
6	40	Очки	4000.00
7	43	Часы	10000.00
8	44	Браслет	2500.00

Рисунок 47 – Выполнение 4 хранимой процедуры

**5 хранимая процедура.** Она называется CountEmployeesPerShop и предназначена для подсчёта количества сотрудников в каждом магазине. Процедура не принимает параметров.

При вызове она выполняет агрегирующий запрос (SELECT с COUNT), который считает количество сотрудников (из таблицы Employee), работающих в каждом магазине (из таблицы Shops), включая те магазины, где сотрудников может не быть (через LEFT JOIN).

Таким образом, процедура позволяет получить общее количество сотрудников по каждому магазину, включая те, где пока никого не наняли.

#### Листинг

```
GO
CREATE PROCEDURE CountEmployeesPerShop
AS
BEGIN
    SELECT s.Shop_Name, COUNT(e.Employees_ID) AS Employee_Count
    FROM Shops s
    LEFT JOIN Employee e ON s.Shop_ID = e.Shop_ID
    GROUP BY s.Shop_Name;
END;
```

```

CREATE PROCEDURE CountEmployeesPerShop
AS
BEGIN
    SELECT s.Shop_Name, COUNT(e.Employees_ID) AS Employee_Count
    FROM Shops s
    LEFT JOIN Employee e ON s.Shop_ID = e.Shop_ID
    GROUP BY s.Shop_Name;
END;

```

131 % < Messages  
Commands completed successfully.  
Completion time: 2025-05-14T18:52:30.0811710+03:00

Рисунок 48 – Создание 5 хранимой процедуры

```

EXEC CountEmployeesPerShop;

```

131 % < Results  
Shop\_Name Employee\_Count  
1 Aure - Puskin 2  
2 Aure - Len 1  
3 Aure - Atrasev 1  
4 Aure - Revolution 1  
5 Aure - Peace 3

Рисунок 49 – Выполнение 5 хранимой процедуры

**1 триггер.** Этот триггер называется `trg_CheckEmployeeAge` и привязан к таблице `Employee`. Он срабатывает вместо вставки (INSTEAD OF INSERT) строк в таблицу `Employee`.

После срабатывания триггера он выполняет следующие действия:

1. Проверяет, есть ли среди вставляемых сотрудников те, кому менее 18 лет на текущую дату;
2. Если такие сотрудники есть, вызывается ошибка (RAISERROR), и выполняется откат транзакции (ROLLBACK TRANSACTION);
3. Если все сотрудники удовлетворяют возрастному критерию,

производится вставка данных в таблицу Employee.

Таким образом, триггер обеспечивает, чтобы в таблицу Employee не могли быть добавлены сотрудники моложе 18 лет, поддерживая корректность данных в HR-системе.

### Листинг

```
CREATE TRIGGER trg_CheckEmployeeAge
ON Employee
INSTEAD OF INSERT
AS
BEGIN
    IF EXISTS (
        SELECT 1
        FROM inserted
        WHERE DATEDIFF(YEAR, Date_of_Birth, GETDATE()) < 18
    )
    BEGIN
        RAISERROR ('Сотруднику должно быть не менее 18 лет.', 16, 1);
        ROLLBACK TRANSACTION;
    END
    ELSE
    BEGIN
        INSERT INTO Employee (Employees_ID, FIO, Email, Shop_ID,
Date_of_Admission, Post, Date_of_Birth)
        SELECT Employees_ID, FIO, Email, Shop_ID,
Date_of_Admission, Post, Date_of_Birth
        FROM inserted;
    END
END;
```

The screenshot shows the SQL Server Management Studio interface with the following details:

- Object Explorer:** Shows the database structure with the trigger definition.
- Script pane:** Displays the T-SQL code for creating the trigger:CREATE TRIGGER trg\_CheckEmployeeAge
ON Employee
INSTEAD OF INSERT
AS
BEGIN
 IF EXISTS (
 SELECT 1
 FROM inserted
 WHERE DATEDIFF(YEAR, Date\_of\_Birth, GETDATE()) < 18
 )
 BEGIN
 RAISERROR ('Сотруднику должно быть не менее 18 лет.', 16, 1);
 ROLLBACK TRANSACTION;
 END
 ELSE
 BEGIN
 INSERT INTO Employee (Employees\_ID, FIO, Email, Shop\_ID,
Date\_of\_Admission, Post, Date\_of\_Birth)
 SELECT Employees\_ID, FIO, Email, Shop\_ID,
Date\_of\_Admission, Post, Date\_of\_Birth
 FROM inserted;
 END
END;
- Status bar:** Shows "89 %".
- Messages pane:** Shows the completion message:

```
Commands completed successfully.
Completion time: 2025-05-14T20:10:05.5213294+03:00
```

Рисунок 50 – Создание 1 триггера



Рисунок 51 – Проверка 1 триггера

**2 триггер.** Этот триггер называется `trg_CheckReturnDate` и привязан к таблице `Returns_pr`. Он срабатывает вместо вставки (INSTEAD OF INSERT) строк в таблицу `Returns_pr`.

1. Проверяет, есть ли среди новых возвратов такие, у которых дата возврата раньше даты покупки, связанной с этим возвратом;
2. Если такие случаи найдены, вызывается ошибка (RAISERROR) и транзакция отменяется;
3. В противном случае данные корректно вставляются в таблицу `Returns_pr`.

Таким образом, триггер предотвращает появление логически неверных данных, где возврат товара происходит до его покупки.

#### Листинг

```
CREATE TRIGGER trg_CheckReturnDate
ON Returns_pr
INSTEAD OF INSERT
AS
BEGIN
    IF EXISTS (
        SELECT 1
        FROM inserted i
        JOIN Purchase p ON i.Purchase_ID = p.Purchase_ID
        WHERE i.Date_of_return < p.Date_of_purchase
    )
    BEGIN
        RAISERROR ('Дата возврата не может быть раньше даты покупки.', 16, 1);
        ROLLBACK TRANSACTION;
    END
    ELSE
    BEGIN
        INSERT INTO Returns_pr (Return_ID, Date_of_return,
Purchase_ID)
        SELECT Return_ID, Date_of_return, Purchase_ID
        FROM inserted;
    END
END
```

## Продолжение листинга

```
END;
```

The screenshot shows the SQL Editor window of SSMS. The code listed is the creation of a trigger named `trg_CheckReturnDate` on the `Returns_pr` table instead of insert. The trigger checks if the return date is before the purchase date. If it is, it raises an error and rolls back the transaction. Otherwise, it inserts the data into the `Returns_pr` table. The status bar at the bottom indicates the command was completed successfully.

```
CREATE TRIGGER trg_CheckReturnDate
ON Returns_pr
INSTEAD OF INSERT
AS
BEGIN
    IF EXISTS (
        SELECT 1
        FROM inserted i
        JOIN Purchase p ON i.Purchase_ID = p.Purchase_ID
        WHERE i.Date_of_return < p.Date_of_purchase
    )
    BEGIN
        RAISERROR ('Дата возврата не может быть раньше даты покупки.', 16, 1);
        ROLLBACK TRANSACTION;
    END
    ELSE
    BEGIN
        INSERT INTO Returns_pr (Return_ID, Date_of_return, Purchase_ID)
        SELECT Return_ID, Date_of_return, Purchase_ID
        FROM inserted;
    END
END;
```

Рисунок 52 – Создание 2 триггера

The screenshot shows the SQL Editor window of SSMS. A query is run to insert a new row into the `Returns_pr` table with a return date of '2022-01-01'. The status bar at the bottom indicates the command was completed successfully. The messages pane shows an error message from the trigger: 'Дата возврата не может быть раньше даты покупки.' (The return date cannot be earlier than the purchase date). It also shows the transaction was rolled back.

```
INSERT INTO Returns_pr (Return_ID, Date_of_return, Purchase_ID)
VALUES (10, '2022-01-01', 1);
```

Рисунок 53 – Выполнение 2 триггера

**3 триггер.** Этот триггер называется `trg_CheckProductPrice` и привязан к таблице `Products`. Он срабатывает вместо вставки (INSTEAD OF INSERT) строк в таблицу `Products`.

После срабатывания триггера, он выполняет следующие действия:

1. Проверяет, есть ли среди вставляемых товаров те, у которых цена отрицательная;
2. Если такие записи есть, вызывается ошибка (RAISERROR) и выполняется откат транзакции;
3. Если все цены допустимы (0 и выше), данные вставляются в таблицу `Products`.

Таким образом, триггер предотвращает добавление товаров с отрицательной стоимостью, обеспечивая достоверность ценовой информации.

#### Листинг

```
CREATE TRIGGER trg_CheckProductPrice
ON Products
INSTEAD OF INSERT
AS
BEGIN
    IF EXISTS (
        SELECT 1 FROM inserted WHERE Price < 0
    )
    BEGIN
        RAISERROR ('Цена товара не может быть отрицательной.', 16, 1);
        ROLLBACK TRANSACTION;
    END
    ELSE
    BEGIN
        INSERT INTO Products (Product_ID, Product_Name,
Description_product, Price, Supply_ID, Image_URL)
        SELECT Product_ID, Product_Name, Description_product,
Price, Supply_ID, Image_URL
        FROM inserted;
    END
END;
```

The screenshot shows the SQL Server Management Studio interface with the following details:

- Object Explorer:** A tree view on the left showing the database structure.
- Script pane:** The main area contains the T-SQL code for creating the trigger:

```
CREATE TRIGGER trg_CheckProductPrice
ON Products
INSTEAD OF INSERT
AS
BEGIN
    IF EXISTS (
        SELECT 1 FROM inserted WHERE Price < 0
    )
    BEGIN
        RAISERROR ('Цена товара не может быть отрицательной.', 16, 1);
        ROLLBACK TRANSACTION;
    END
    ELSE
    BEGIN
        INSERT INTO Products (Product_ID, Product_Name, Description_product, Price, Supply_ID, Image_URL)
        SELECT Product_ID, Product_Name, Description_product, Price, Supply_ID, Image_URL
        FROM inserted;
    END
END;
```

- Status bar:** Shows "89 %".
- Messages pane:** Displays the completion message:

```
Commands completed successfully.
Completion time: 2025-05-14T20:15:18.1594660+03:00
```

Рисунок 54 – Создание 3 триггера

Рисунок 55 – Проверка 3 триггера

**4 триггер.** Этот триггер называется `trg_CheckPurchaseItemQuantity` и привязан к таблице `Purchase_Items`. Он срабатывает вместо вставки (INSTEAD OF INSERT) строк в таблицу `Purchase_Items`.

После срабатывания триггера, он выполняет следующие действия:

1. Проверяет, есть ли среди вставляемых позиций покупки такие, где количество товара ноль или меньше;
2. При наличии таких записей вызывается ошибка (RAISERROR) и выполняется откат транзакции;
3. Если количество корректное (больше нуля), данные вставляются в таблицу `Purchase_Items`.

Таким образом, триггер обеспечивает, что каждая покупка содержит только реальные, ненулевые количества товаров, поддерживая корректную статистику продаж.

#### Листинг

```
CREATE TRIGGER trg_CheckPurchaseItemQuantity
ON Purchase_Items
INSTEAD OF INSERT
AS
BEGIN
    IF EXISTS (
        SELECT 1 FROM inserted WHERE Quantity <= 0
    )
    BEGIN
        RAISERROR ('Количество товара в покупке должно быть положительным числом.', 16, 1);
        ROLLBACK TRANSACTION;
    END
    ELSE
    BEGIN
        INSERT INTO Purchase_Items (Purchase_ID, Product_ID,
Quantity, Price)
        SELECT Purchase_ID, Product_ID, Quantity, Price
        FROM inserted;
```

```

CREATE TRIGGER trg_CheckPurchaseItemQuantity
ON Purchase_Items
INSTEAD OF INSERT
AS
BEGIN
    IF EXISTS (
        SELECT 1 FROM inserted WHERE Quantity <= 0
    )
    BEGIN
        RAISERROR ('Количество товара в покупке должно быть положительным числом.', 16, 1);
        ROLLBACK TRANSACTION;
    END
    ELSE
    BEGIN
        INSERT INTO Purchase_Items (Purchase_ID, Product_ID, Quantity, Price)
        SELECT Purchase_ID, Product_ID, Quantity, Price
        FROM inserted;
    END
END;

```

89 %

Messages

Commands completed successfully.

Completion time: 2025-05-14T20:18:01.8915622+03:00

Рисунок 56 – Создание 4 триггера

```

INSERT INTO Purchase_Items (Purchase_ID, Product_ID, Quantity, Price)
VALUES (1, 1, 0, 4000.00);

```

89 %

Messages

Msg 50000, Level 16, State 1, Procedure trg\_CheckPurchaseItemQuantity, Line 10 [Batch Start Line 638]
Количество товара в покупке должно быть положительными числами.

Msg 3609, Level 16, State 1, Line 639
Транзакция завершилась в триггере. Выполнение пакета прервано.

Completion time: 2025-05-14T20:19:17.9983051+03:00

Рисунок 57 – Проверка 4 триггера

**5 триггер.** Этот триггер называется `trg_ReturnProductToInvent` и привязан к таблице `Returns_pr`. Он срабатывает после вставки (AFTER INSERT) строк в таблицу `Returns_pr`.

После срабатывания триггера, он выполняет следующие действия:

1. Находит соответствующие записи в таблице `Shop_Product_Inventory`, основываясь на возвращённой покупке;
2. Увеличивает количество товара на складе магазина на значение, равное количеству возвращённых товаров.

Итак, данный триггер автоматически возвращает товар на склад после оформления возврата, обеспечивая актуальность данных об остатках продукции в магазине.

#### Листинг

```

CREATE TRIGGER trg_ReturnProductToInvent
ON Returns_pr
AFTER INSERT

```

## Продолжение листинга

```
AS
BEGIN
    SET NOCOUNT ON;

    UPDATE spi
    SET spi.Quantity = spi.Quantity + pi.Quantity
    FROM Shop_Product_Inventory spi
    JOIN Purchase_Items pi ON spi.Product_ID = pi.Product_ID
    JOIN Purchase p ON pi.Purchase_ID = p.Purchase_ID
    JOIN inserted i ON p.Purchase_ID = i.Purchase_ID
    WHERE spi.Shop_ID = p.Shop_ID;
END;
```

The screenshot shows the SQL Server Management Studio interface. A new trigger is being created named 'trg\_ReturnProductToInvent'. The script pane contains the T-SQL code for the trigger:

```
CREATE TRIGGER trg_ReturnProductToInvent
ON Returns_pr
AFTER INSERT
AS
BEGIN
    SET NOCOUNT ON;

    UPDATE spi
    SET spi.Quantity = spi.Quantity + pi.Quantity
    FROM Shop_Product_Inventory spi
    JOIN Purchase_Items pi ON spi.Product_ID = pi.Product_ID
    JOIN Purchase p ON pi.Purchase_ID = p.Purchase_ID
    JOIN inserted i ON p.Purchase_ID = i.Purchase_ID
    WHERE spi.Shop_ID = p.Shop_ID;
END;
```

The status bar at the bottom indicates the command completed successfully and provides the completion time.

Рисунок 58 – Создание 5 триггера

The screenshot shows the SQL Server Management Studio interface with two result sets. The top result set shows the initial state of the 'Shop\_Product\_Inventory' table:

```
SELECT * FROM Shop_Product_Inventory WHERE Shop_ID = 1 AND Product_ID = 2;
```

The bottom result set shows the state after a return was recorded:

```
INSERT INTO Returns_pr (Return_ID, Date_of_return, Purchase_ID)
VALUES (6, '2023-02-10', 2);

SELECT * FROM Shop_Product_Inventory WHERE Shop_ID = 1 AND Product_ID = 2;
```

Shop_ID	Product_ID	Quantity
1	2	20

The quantity has been updated to 22.

Рисунок 59 – Проверка 5 триггера

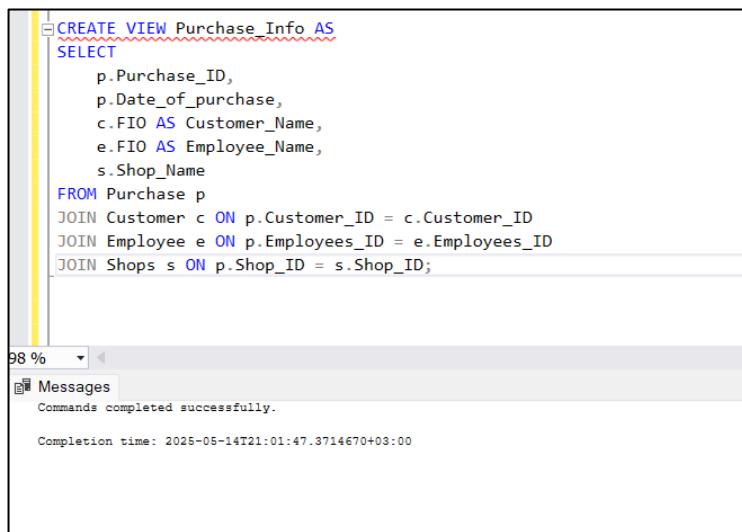
Также были разработаны представления и транзакции.

**1 представление.** Этот код создает представление под названием Purchase\_Info. Представление Purchase\_Info будет содержать информацию о покупках, включая идентификатор покупки (Purchase\_ID), дату покупки (Date\_of\_purchase), имя клиента (Customer\_Name), имя сотрудника,

оформившего покупку (Employee\_Name), а также название магазина (Shop\_Name), в котором была совершена покупка.

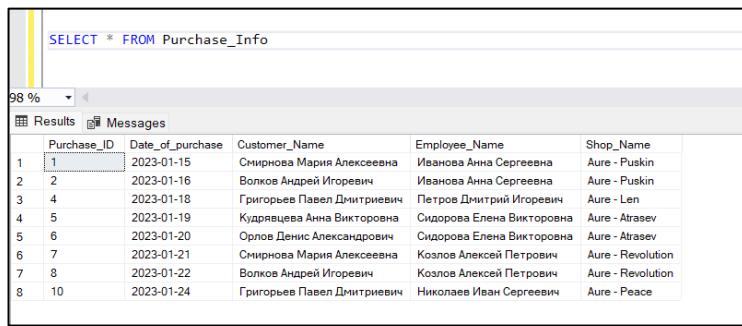
#### Листинг

```
CREATE VIEW Purchase_Info AS
SELECT
    p.Purchase_ID,
    p.Date_of_purchase,
    c.FIO AS Customer_Name,
    e.FIO AS Employee_Name,
    s.Shop_Name
FROM Purchase p
JOIN Customer c ON p.Customer_ID = c.Customer_ID
JOIN Employee e ON p.Employees_ID = e.Employees_ID
JOIN Shops s ON p.Shop_ID = s.Shop_ID;
```



The screenshot shows the SQL Server Management Studio interface. In the top pane, there is a code editor window containing the SQL script for creating the 'Purchase\_Info' view. The script includes SELECT statements and JOIN clauses from four tables: Purchase, Customer, Employee, and Shops. Below the code editor is a message window titled 'Messages' which displays the success message 'Commands completed successfully.' and the completion time 'Completion time: 2025-05-14T21:01:47.3714670+03:00'.

Рисунок 60 – Создание 1 представления



The screenshot shows the SQL Server Management Studio interface with a results grid. The query 'SELECT \* FROM Purchase\_Info' has been executed. The results grid contains 8 rows of data, each representing a purchase record with columns: Purchase\_ID, Date\_of\_purchase, Customer\_Name, Employee\_Name, and Shop\_Name. The data includes various names like Смирнова Мария Алексеевна, Иванова Анна Сергеевна, etc., and shop names like Aure - Puskin, Aure - Revolution, etc.

	Purchase_ID	Date_of_purchase	Customer_Name	Employee_Name	Shop_Name
1	1	2023-01-15	Смирнова Мария Алексеевна	Иванова Анна Сергеевна	Aure - Puskin
2	2	2023-01-16	Волков Андрей Игоревич	Иванова Анна Сергеевна	Aure - Puskin
3	4	2023-01-18	Григорьев Павел Дмитриевич	Петров Дмитрий Игоревич	Aure - Len
4	5	2023-01-19	Кудрявцева Анна Викторовна	Сидорова Елена Викторовна	Aure - Atrasev
5	6	2023-01-20	Орлов Денис Александрович	Сидорова Елена Викторовна	Aure - Atrasev
6	7	2023-01-21	Смирнова Мария Алексеевна	Козлов Алексей Петрович	Aure - Revolution
7	8	2023-01-22	Волков Андрей Игоревич	Козлов Алексей Петрович	Aure - Revolution
8	10	2023-01-24	Григорьев Павел Дмитриевич	Николаев Иван Сергеевич	Aure - Peace

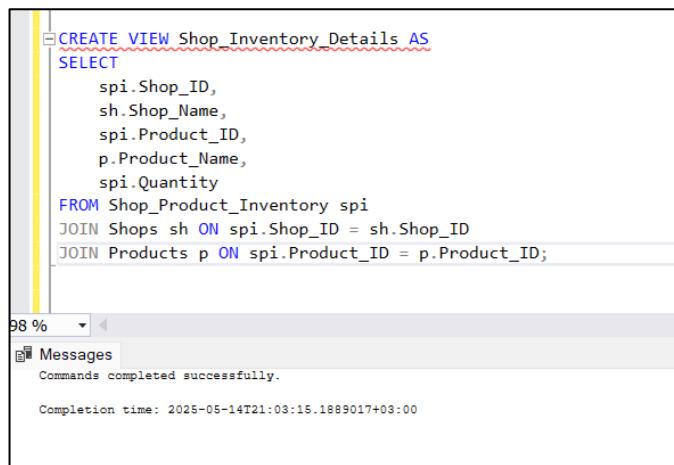
Рисунок 61 – Вызов 1 представления

**2 представление.** Этот код создает представление под названием Shop\_Inventory\_Details. Представление Shop\_Inventory\_Details содержит информацию о запасах товаров в магазинах, включая идентификатор магазина (Shop\_ID), название магазина (Shop\_Name), идентификатор товара

(Product\_ID), название товара (Product\_Name) и количество единиц на складе (Quantity).

#### Листинг

```
CREATE VIEW Shop_Inventory_Details AS
SELECT
    spi.Shop_ID,
    sh.Shop_Name,
    spi.Product_ID,
    p.Product_Name,
    spi.Quantity
FROM Shop_Product_Inventory spi
JOIN Shops sh ON spi.Shop_ID = sh.Shop_ID
JOIN Products p ON spi.Product_ID = p.Product_ID;
```

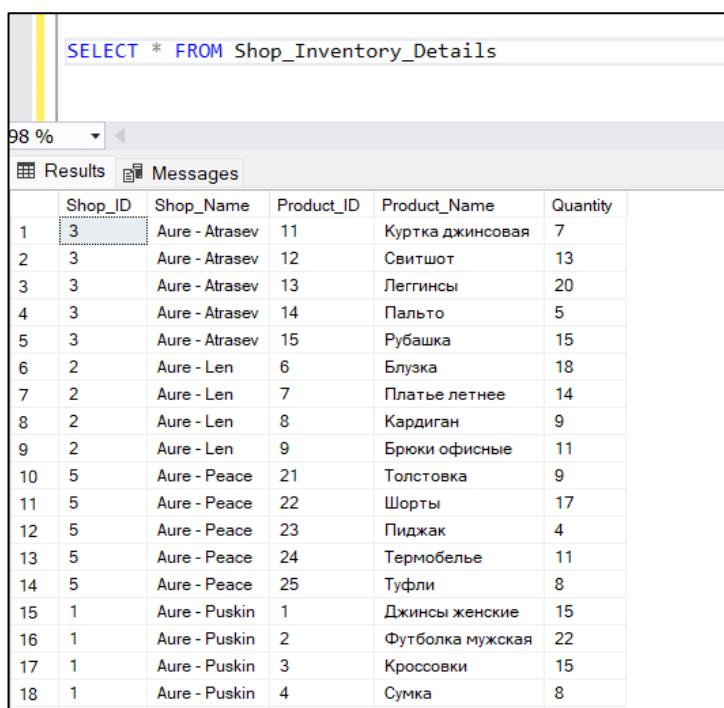


The screenshot shows the SQL query for creating the view:

```
CREATE VIEW Shop_Inventory_Details AS
SELECT
    spi.Shop_ID,
    sh.Shop_Name,
    spi.Product_ID,
    p.Product_Name,
    spi.Quantity
FROM Shop_Product_Inventory spi
JOIN Shops sh ON spi.Shop_ID = sh.Shop_ID
JOIN Products p ON spi.Product_ID = p.Product_ID;
```

Below the query, the status bar indicates "98 %". In the "Messages" pane, it says "Commands completed successfully." and "Completion time: 2025-05-14T21:03:15.1889017+03:00".

Рисунок 62 – Создание 2 представления



The screenshot shows the results of a SELECT query:

```
SELECT * FROM Shop_Inventory_Details
```

The results table has columns: Shop\_ID, Shop\_Name, Product\_ID, Product\_Name, and Quantity. The data is as follows:

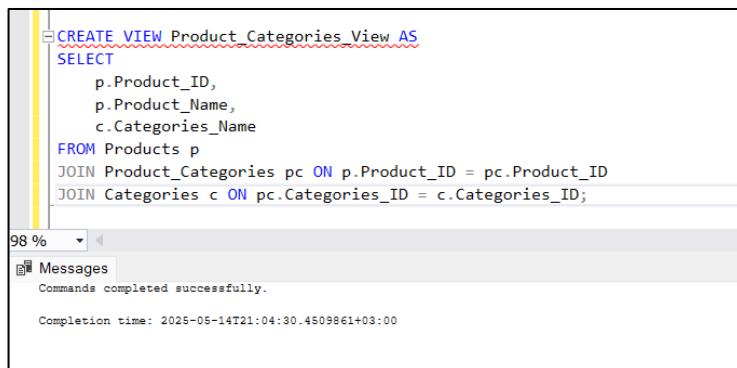
	Shop_ID	Shop_Name	Product_ID	Product_Name	Quantity
1	3	Aure - Atrasev	11	Куртка джинсовая	7
2	3	Aure - Atrasev	12	Свитшот	13
3	3	Aure - Atrasev	13	Леггинсы	20
4	3	Aure - Atrasev	14	Пальто	5
5	3	Aure - Atrasev	15	Рубашка	15
6	2	Aure - Len	6	Блузка	18
7	2	Aure - Len	7	Платье летнее	14
8	2	Aure - Len	8	Кардиган	9
9	2	Aure - Len	9	Брюки офисные	11
10	5	Aure - Peace	21	Толстовка	9
11	5	Aure - Peace	22	Шорты	17
12	5	Aure - Peace	23	Пиджак	4
13	5	Aure - Peace	24	Термобелье	11
14	5	Aure - Peace	25	Туфли	8
15	1	Aure - Puskin	1	Джинсы женские	15
16	1	Aure - Puskin	2	Футболка мужская	22
17	1	Aure - Puskin	3	Кроссовки	15
18	1	Aure - Puskin	4	Сумка	8

Рисунок 63 – Вызов 2 представления

**3 представление.** Этот код создает представление под названием Product\_Categories\_View. Представление Product\_Categories\_View содержит информацию о принадлежности товаров к категориям, включая идентификатор товара (Product\_ID), название товара (Product\_Name) и название категории (Categories\_Name), к которой этот товар относится.

#### Листинг

```
CREATE VIEW Product_Categories_View AS
SELECT
    p.Product_ID,
    p.Product_Name,
    c.Categories_Name
FROM Products p
JOIN Product_Categories pc ON p.Product_ID = pc.Product_ID
JOIN Categories c ON pc.Categories_ID = c.Categories_ID;
```

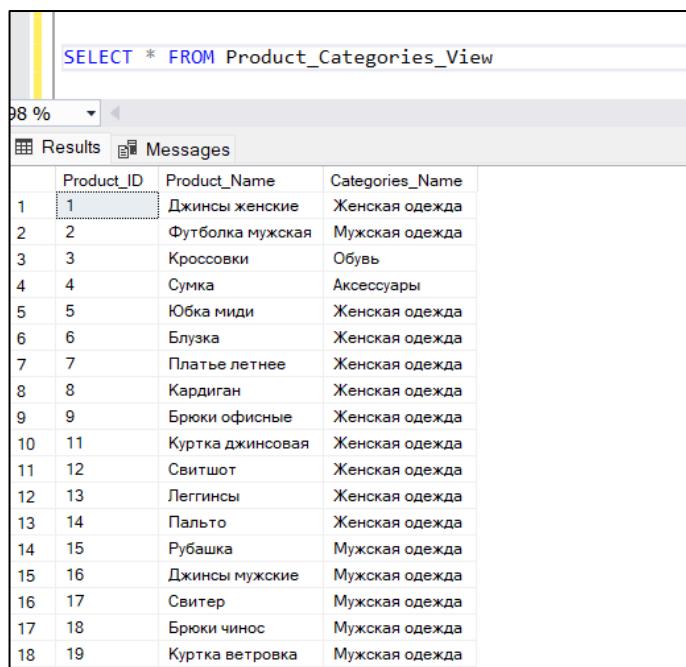


```
CREATE VIEW Product_Categories_View AS
SELECT
    p.Product_ID,
    p.Product_Name,
    c.Categories_Name
FROM Products p
JOIN Product_Categories pc ON p.Product_ID = pc.Product_ID
JOIN Categories c ON pc.Categories_ID = c.Categories_ID;
```

98 %

Messages  
Commands completed successfully.  
Completion time: 2025-05-14T21:04:30.4509861+03:00

Рисунок 64 – Создание 3 представления



```
SELECT * FROM Product_Categories_View
```

98 %

	Product_ID	Product_Name	Categories_Name
1	1	Джинсы женские	Женская одежда
2	2	Футболка мужская	Мужская одежда
3	3	Кроссовки	Обувь
4	4	Сумка	Аксессуары
5	5	Юбка миди	Женская одежда
6	6	Блузка	Женская одежда
7	7	Платье летнее	Женская одежда
8	8	Кардиган	Женская одежда
9	9	Брюки офисные	Женская одежда
10	11	Куртка джинсовая	Женская одежда
11	12	Свитшот	Женская одежда
12	13	Леггинсы	Женская одежда
13	14	Пальто	Женская одежда
14	15	Рубашка	Мужская одежда
15	16	Джинсы мужские	Мужская одежда
16	17	Свитер	Мужская одежда
17	18	Брюки чинос	Мужская одежда
18	19	Куртка ветровка	Мужская одежда

Рисунок 65 – Вызов 3 представления

**4 представление.** Этот код создает представление под названием Supply\_Details. Представление Supply\_Details содержит информацию о поставках товаров в магазины, включая название поставщика (Suppliers\_Name), название поставки (Supply\_Name) и название магазина (Shop\_Name), в который осуществляется поставка..

#### Листинг

```
CREATE VIEW Supply_Details AS
SELECT
    su.Suppliers_Name,
    s.Supply_Name,
    sh.Shop_Name
FROM Supplier su
JOIN Supply s ON su.Supply_ID = s.Supply_ID
JOIN Supplies_To_Shops sts ON s.Supply_ID = sts.Supply_ID
JOIN Shops sh ON sts.Shop_ID = sh.Shop_ID;
```

```
CREATE VIEW Supply_Details AS
SELECT
    su.Suppliers_Name,
    s.Supply_Name,
    sh.Shop_Name
FROM Supplier su
JOIN Supply s ON su.Supply_ID = s.Supply_ID
JOIN Supplies_To_Shops sts ON s.Supply_ID = sts.Supply_ID
JOIN Shops sh ON sts.Shop_ID = sh.Shop_ID;
```

98 %

Messages  
Commands completed successfully.  
Completion time: 2025-05-14T21:05:55.1841990+03:00

Рисунок 66 – Создание 4 представления

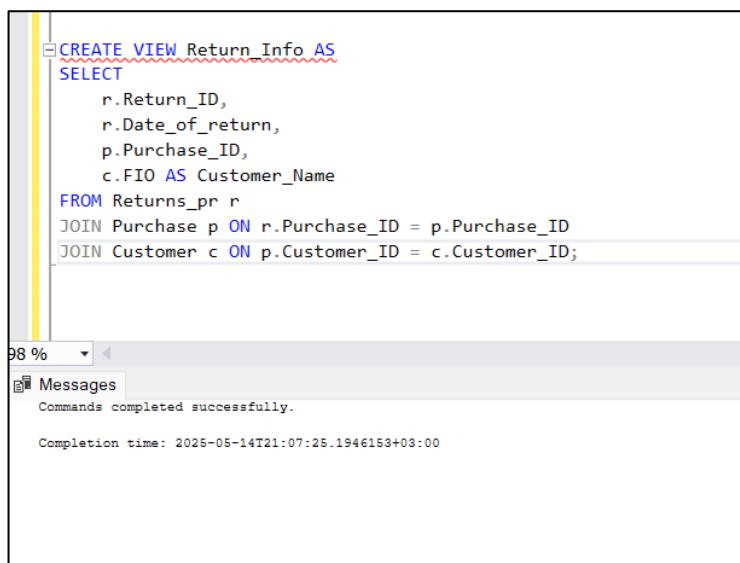
	Suppliers_Name	Supply_Name	Shop_Name
1	Fashion Forward Inc.	Коллекция Amor	Aure - Puskin
2	Fashion Forward Inc.	Коллекция Amor	Aure - Len
3	Fashion Forward Inc.	Коллекция Amor	Aure - Atrasev
4	Trendy Styles Ltd.	Летние новинки	Aure - Len
5	Trendy Styles Ltd.	Летние новинки	Aure - Atrasev
6	Trendy Styles Ltd.	Летние новинки	Aure - Revolution
7	Urban Wear Co.	Осенняя коллекция InOctober	Aure - Atrasev
8	Urban Wear Co.	Осенняя коллекция InOctober	Aure - Revolution
9	Urban Wear Co.	Осенняя коллекция InOctober	Aure - Peace
10	Classic Apparel LLC	Зимняя коллекция Snow	Aure - Puskin
11	Classic Apparel LLC	Зимняя коллекция Snow	Aure - Revolution
12	Classic Apparel LLC	Зимняя коллекция Snow	Aure - Peace
13	Modern Outfitters	Молодежный стиль	Aure - Puskin
14	Modern Outfitters	Молодежный стиль	Aure - Len
15	Modern Outfitters	Молодежный стиль	Aure - Peace

Рисунок 67 – Вызов 4 представления

**5 представление.** Этот код создает представление под названием Return\_Info. Представление Return\_Info содержит информацию о возвратах покупок, включая идентификатор возврата (Return\_ID), дату возврата (Date\_of\_return), идентификатор связанной покупки (Purchase\_ID), а также имя клиента (Customer\_Name), который оформил возврат.

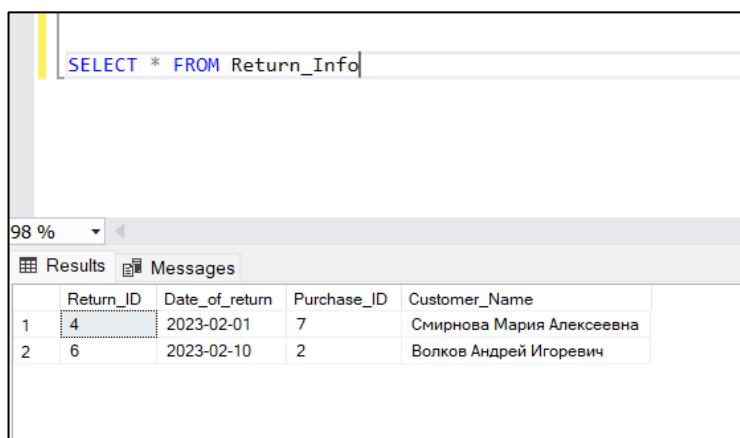
#### Листинг

```
CREATE VIEW Return_Info AS
SELECT
    r.Return_ID,
    r.Date_of_return,
    p.Purchase_ID,
    c.FIO AS Customer_Name
FROM Returns_pr r
JOIN Purchase p ON r.Purchase_ID = p.Purchase_ID
JOIN Customer c ON p.Customer_ID = c.Customer_ID;
```



The screenshot shows the SQL Server Management Studio interface. In the main pane, the code for creating the 'Return\_Info' view is displayed. The code includes a SELECT statement that joins three tables: 'Returns\_pr', 'Purchase', and 'Customer'. The 'Returns\_pr' table is aliased as 'r', 'Purchase' as 'p', and 'Customer' as 'c'. The 'Customer' table is used to get the customer's name based on their purchase ID. Below the code, the status bar shows '98 %' completion. In the bottom pane, under the 'Messages' tab, it says 'Commands completed successfully.' and provides the completion time: 'Completion time: 2025-05-14T21:07:25.1946153+03:00'.

Рисунок 68 – Создание 5 представления



The screenshot shows the SQL Server Management Studio interface. In the main pane, the query 'SELECT \* FROM Return\_Info' is entered. In the bottom pane, under the 'Results' tab, the output is shown as a table with two rows. The columns are 'Return\_ID', 'Date\_of\_return', 'Purchase\_ID', and 'Customer\_Name'. The first row has values 1, 2023-02-01, 7, and Смирнова Мария Алексеевна. The second row has values 2, 2023-02-10, 2, and Волков Андрей Игоревич.

Return_ID	Date_of_return	Purchase_ID	Customer_Name
1	2023-02-01	7	Смирнова Мария Алексеевна
2	2023-02-10	2	Волков Андрей Игоревич

Рисунок 69 – Вызов 5 представления

**1 транзакция.** Этот код используется для оформления покупки. Он создаёт новую запись в таблице покупок (Purchase) и добавляет позиции покупки в таблицу Purchase\_Items. При успешном выполнении все изменения фиксируются, после чего выводится информация о добавленной покупке и товарах.

#### Листинг

```
BEGIN TRANSACTION;
INSERT INTO Purchase (Purchase_ID, Date_of_purchase, Shop_ID,
Employees_ID, Customer_ID)
VALUES (11, '2025-05-14', 1, 1, 2);
INSERT INTO Purchase_Items (Purchase_ID, Product_ID, Quantity,
Price)
VALUES
(11, 1, 1, 4000.00),
(11, 2, 2, 2000.00);
COMMIT;
```

The screenshot shows the SQL Server Management Studio interface. In the main pane, a transaction script is being run:

```
BEGIN TRANSACTION;
INSERT INTO Purchase (Purchase_ID, Date_of_purchase, Shop_ID, Employees_ID, Customer_ID)
VALUES (11, '2025-05-14', 1, 1, 2);
INSERT INTO Purchase_Items (Purchase_ID, Product_ID, Quantity, Price)
VALUES
(11, 1, 1, 4000.00),
(11, 2, 2, 2000.00);
COMMIT;
```

Below the script, the 'Messages' tab displays the results of the execution:

- (1 row affected)
- (2 rows affected)
- (2 rows affected)

The completion time is shown as 2025-05-14T21:33:39.1251284+03:00.

Рисунок 70 – Создание 1 транзакции

The screenshot shows the SQL Server Management Studio interface with two queries in the 'Messages' tab:

```
SELECT * FROM Purchase WHERE Purchase_ID = 11;
SELECT * FROM Purchase_Items WHERE Purchase_ID = 11;
```

Below the queries, the 'Results' tab displays the data from the tables:

	Purchase_ID	Date_of_purchase	Shop_ID	Employees_ID	Customer_ID
1	11	2025-05-14	1	1	2

	Purchase_Item_ID	Purchase_ID	Product_ID	Quantity	Price
1	8	11	1	1	4000.00
2	9	11	2	2	2000.00

Рисунок 71 – Проверка 1 транзакции

**2 транзакция.** Этот код используется для оформления возврата покупки. Он создаёт запись в таблице возвратов (Returns\_pr) с привязкой к конкретной покупке. После выполнения транзакции можно просмотреть добавленный возврат по его идентификатору.

Листинг

```
BEGIN TRANSACTION;
INSERT INTO Returns_pr (Return_ID, Date_of_return, Purchase_ID)
VALUES (20, '2025-05-14', 11);
COMMIT;
```

The screenshot shows the SQL Server Management Studio interface. In the main pane, there is a code editor window containing the following SQL script:

```
BEGIN TRANSACTION;
INSERT INTO Returns_pr (Return_ID, Date_of_return, Purchase_ID)
VALUES (20, '2025-05-14', 11);
COMMIT;
```

Below the code editor is a progress bar indicating "98 %". Underneath the progress bar, there are two tabs: "Messages" and "Results". The "Messages" tab is selected and displays the following output:

```
(2 rows affected)
(1 row affected)
(1 row affected)
Completion time: 2025-05-14T22:04:04.1094108+03:00
```

Рисунок 72 – Создание 2 транзакции

The screenshot shows the SQL Server Management Studio interface. In the main pane, there is a results grid window containing the following SQL query:

```
SELECT * FROM Returns_pr WHERE Return_ID = 20;
```

Below the query is a progress bar indicating "98 %". Underneath the progress bar, there are two tabs: "Results" and "Messages". The "Results" tab is selected and displays a table with one row of data:

	Return_ID	Date_of_return	Purchase_ID
1	20	2025-05-14	11

Рисунок 73 – Проверка 2 транзакции

**3 транзакция.** Этот код используется для увеличения остатка товара на складе магазина. Он увеличивает количество единиц указанного товара в таблице Shop\_Product\_Inventory. После фиксации изменений можно проверить новое количество.

Листинг

```
BEGIN TRANSACTION;
UPDATE Shop_Product_Inventory
```

## Продолжение листинга

```
SET Quantity = Quantity + 5  
WHERE Shop_ID = 1 AND Product_ID = 5;  
COMMIT;
```

The screenshot shows a SQL query window with the following content:

```
BEGIN TRANSACTION;  
UPDATE Shop_Product_Inventory  
SET Quantity = Quantity + 5  
WHERE Shop_ID = 1 AND Product_ID = 5;  
COMMIT;
```

Below the query window, there is a progress bar at 98% completion and a message window titled "Messages" containing the text "(1 row affected)". At the bottom, the completion time is shown as "Completion time: 2025-05-14T22:05:23.3254998+03:00".

Рисунок 74 – Создание 3 транзакции

The screenshot shows a SQL query window with the following content:

```
SELECT * FROM Shop_Product_Inventory WHERE Shop_ID = 1 AND Product_ID = 5;
```

Below the query window, there is a results grid titled "Results" showing one row of data:

Shop_ID	Product_ID	Quantity
1	5	17

Рисунок 75 – Проверка 3 транзакции

**4 транзакция.** Этот код используется для удаления покупателя. Сначала проверяется, совершил ли покупатель покупки. Если нет — выполняется удаление и фиксируется транзакция. Если есть покупки — удаление отменяется, и транзакция откатывается. Выводятся сообщения в зависимости от результата.

## Листинг

```
BEGIN TRANSACTION;  
DECLARE @DelCustomerID INT = 6;  
BEGIN TRY  
    IF NOT EXISTS (  
        SELECT 1 FROM Purchase WHERE Customer_ID = @DelCustomerID  
    )  
    BEGIN  
        DELETE FROM Customer WHERE Customer_ID = @DelCustomerID;  
        COMMIT;  
        PRINT 'Транзакция : Покупатель удалён.';  
    END
```

## Продолжение листинга

```
ELSE
BEGIN
    ROLLBACK;
    PRINT 'Транзакция : У клиента есть покупки — удаление
невозможно.';
END
END TRY
BEGIN CATCH
    ROLLBACK;
    PRINT 'Ошибка в транзакции. Откат.';
END CATCH;
```

```
BEGIN TRANSACTION;
DECLARE @DelCustomerID INT = 6;
BEGIN TRY
    IF NOT EXISTS (
        SELECT 1 FROM Purchase WHERE Customer_ID = @DelCustomerID )
    BEGIN
        DELETE FROM Customer WHERE Customer_ID = @DelCustomerID;
        COMMIT;
        PRINT 'Транзакция : Покупатель удалён.';
    END
    ELSE
    BEGIN
        ROLLBACK;
        PRINT 'Транзакция : У клиента есть покупки — удаление невозможно.';
    END
END TRY
BEGIN CATCH
    ROLLBACK;
    PRINT 'Ошибка в транзакции. Откат.';
END CATCH;
```

Рисунок 76 – Создание 4 транзакции

**5 транзакция.** Этот код добавляет новую категорию в таблицу Categories. При успешном добавлении транзакция фиксируется, и выводится сообщение об успехе. В случае ошибки — происходит откат и отображается сообщение об ошибке.

## Листинг

```
BEGIN TRANSACTION;
DECLARE @CategoryID INT = 5;
DECLARE @CategoryName NVARCHAR(50) = 'Игрушки';
BEGIN TRY
    INSERT INTO Categories(Categories_ID, Categories_Name)
    VALUES (@CategoryID, @CategoryName);
    COMMIT;
    PRINT 'Транзакция : Новая категория добавлена.';
END TRY
BEGIN CATCH
```

## Продолжение листинга

```
ROLLBACK;
PRINT 'Ошибка в транзакции. Откат.';
END CATCH;
```

The screenshot shows a SQL script window with the following code:

```
BEGIN TRANSACTION;
DECLARE @CategoryID INT = 5;
DECLARE @CategoryName NVARCHAR(50) = 'Игрушки';
BEGIN TRY
    INSERT INTO Categories(Categories_ID, Categories_Name)
    VALUES (@CategoryID, @CategoryName);
    COMMIT;
    PRINT 'Транзакция : Новая категория добавлена.';
END TRY
BEGIN CATCH
    ROLLBACK;
    PRINT 'Ошибка в транзакции. Откат.';
END CATCH;
```

Below the script, the status bar shows "98 %". Underneath the script, there is a "Messages" tab with the following output:

```
(1 row affected)
Транзакция : Новая категория добавлена.

Completion time: 2025-05-14T22:15:58.1638332+03:00
```

Рисунок 77 – Создание 5 транзакции

The screenshot shows a query window with the following SQL command:

```
SELECT * FROM Categories
```

Below the query, the status bar shows "98 %". Underneath the query, there are two tabs: "Results" and "Messages". The "Results" tab displays the following data:

	Categories_ID	Categories_Name
1	1	Женская одежда
2	2	Мужская одежда
3	3	Обувь
4	4	Аксессуары
5	5	Игрушки

Рисунок 78 – Проверка 5 транзакции

## ГЛАВА 2 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

При запуске программы вас встречает главный экран (Рисунок 79), на котором представлены все магазины, имеющиеся в сети.

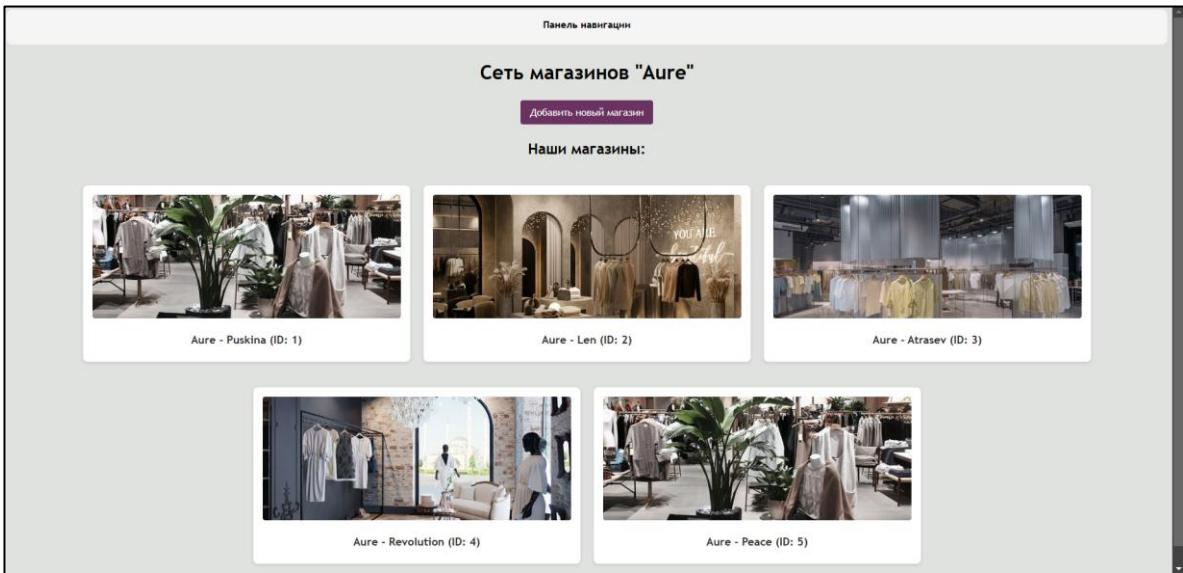


Рисунок 79 – Главный экран

На верху каждой из страниц представлена панель навигации, которая открывается при наведении на неё. Благодаря панели мы можем без затруднений перемещаться на вкладку с нужной нам таблицей.

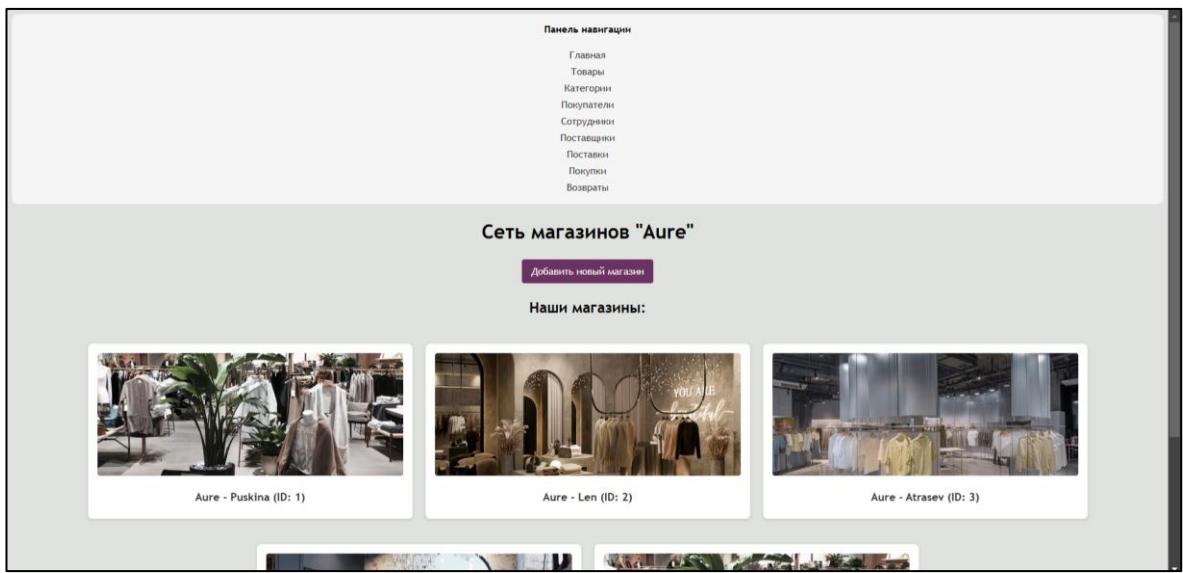


Рисунок 80 – Панель навигации

После выбора таблицы (Рисунок 81), мы переходим на страницу, которая отображает ее данные, и позволяет редактировать, удалить старые или добавить новые данные.

Покупатели		<a href="#">+ Добавить покупателя</a>
Смирнова Мария Алексеевна	ID: 1	<a href="#">Изменить</a> <a href="#">Удалить</a>
✉ smirnova@mail.com	📞 +79161112233	
Волков Андрей Игоревич	ID: 2	<a href="#">Изменить</a> <a href="#">Удалить</a>
✉ volkov@mail.com	📞 +79162223344	
Белова Екатерина Сергеевна	ID: 3	<a href="#">Изменить</a> <a href="#">Удалить</a>
✉ belova@mail.com	📞 +79163334455	
Григорьев Павел Дмитриевич	ID: 4	<a href="#">Изменить</a> <a href="#">Удалить</a>
✉ grigorev@mail.com	📞 +79164445566	
Кудрявцева Анна Викторовна	ID: 5	<a href="#">Изменить</a> <a href="#">Удалить</a>
✉ kudryavtseva@mail.com	📞 +79165556677	
Орлов Денис Александрович	ID: 6	<a href="#">Изменить</a> <a href="#">Удалить</a>
✉ orlov@mail.com	📞 +79166667788	

Рисунок 81 – Таблица «Покупатели»

При нажатии на кнопку «Добавить покупателя» нам открывается новая страница (Рисунок 82), на которой мы можем добавить новую запись в выбранную нами таблицу, заполнив все данные.

### Create New Customer

Customer ID:	20
ФИО:	Кошечкин Александр Илья
Email:	ilyusha22@mail.ru
Номер телефона:	+79162345678
<input type="button" value="Сохранить"/>	

Рисунок 82 – Добавление новой записи в таблицу

После нажатия на «Сохранить» мы вновь оказываемся на странице таблицы (Рисунок 83) и можем убедиться в добавлении новой записи – она появилась в списке.

Волков Андрей Игоревич ID: 2	<a href="#">Изменить</a>	<a href="#">Удалить</a>
Белова Екатерина Сергеевна ID: 3	<a href="#">Изменить</a>	<a href="#">Удалить</a>
Григорьев Павел Дмитриевич ID: 4	<a href="#">Изменить</a>	<a href="#">Удалить</a>
Кудрявцева Анна Викторовна ID: 5	<a href="#">Изменить</a>	<a href="#">Удалить</a>
Орлов Денис Александрович ID: 6	<a href="#">Изменить</a>	<a href="#">Удалить</a>
Кошечкин Александр Ильич ID: 20	<a href="#">Изменить</a>	<a href="#">Удалить</a>

Рисунок 83 – Сохранение новой записи в таблице

Если же есть необходимость изменить ранее добавленные данные, то существует кнопка «Изменить», нажав на которую мы окажемся на новой вкладке редактирования данных (Рисунок 84), где имеется возможность изменения выбранной записи.

Панель навигации

## Edit Customer

ID покупателя: 20

ФИО: Кошечкин Александр Илья

Email: ilyushaAvtomat52@mail.ru

Номер телефона: +79162345678

Рисунок 84 – Редактирование записи в таблице

После нажатия на «Сохранить» мы вновь оказываемся на странице таблицы (Рисунок 85) и можем убедиться в том, что данные обновлены.

Волков Андрей Игоревич ID: 2 ✉ volkov@mail.com ☎ +79162223344	Изменить  Удалить
Белова Екатерина Сергеевна ID: 3 ✉ belova@mail.com ☎ +79163334455	Изменить  Удалить
Григорьев Павел Дмитриевич ID: 4 ✉ grigorev@mail.com ☎ +79164445566	Изменить  Удалить
Кудрявцева Анна Викторовна ID: 5 ✉ kudryavceva@mail.com ☎ +79165556677	Изменить  Удалить
Орлов Денис Александрович ID: 6 ✉ orlov@mail.com ☎ +79166667788	Изменить  Удалить
Кошечкин Александр Ильич ID: 20 ✉ ilyushaAvtomat52@mail.test ☎ +79162345678	Изменить  Удалить

Рисунок 85 – Обновленные данные в таблице

При нажатии на кнопку «Удалить» (Рисунок 86) - выбранная запись в таблице будет удалена.

Смирнова Мария Алексеевна ID: 1 ✉ smirnova@mail.com ☎ +79161112233	Изменить  Удалить
Волков Андрей Игоревич ID: 2 ✉ volkov@mail.com ☎ +79162223344	Изменить  Удалить
Белова Екатерина Сергеевна ID: 3 ✉ belova@mail.com ☎ +79163334455	Изменить  Удалить
Григорьев Павел Дмитриевич ID: 4 ✉ grigorev@mail.com ☎ +79164445566	Изменить  Удалить
Кудрявцева Анна Викторовна ID: 5 ✉ kudryavceva@mail.com ☎ +79165556677	Изменить  Удалить
Орлов Денис Александрович ID: 6 ✉ orlov@mail.com ☎ +79166667788	Изменить  Удалить

Рисунок 86 – Удаление записи

На каждой странице присутствуют проверки на наличие ошибок. Например, при попытке добавить в покупку определенное количество товара, которого не имеется в наличии магазина – возникнет ошибка.

Server Error: Not enough stock for product 9 in shop 5. Available: 0, requested: 10.

Рисунок 87 – Обработка ошибок

По такому принципу были созданы все страницы (Рисунки 88-96) с таблицами и обработки ошибок для них.

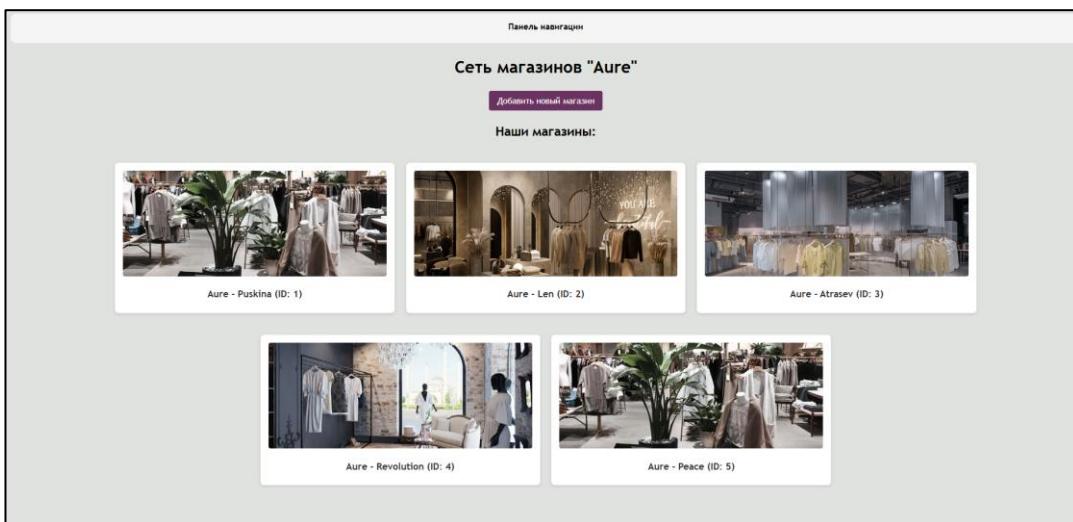


Рисунок 88 – Таблица «Магазины»

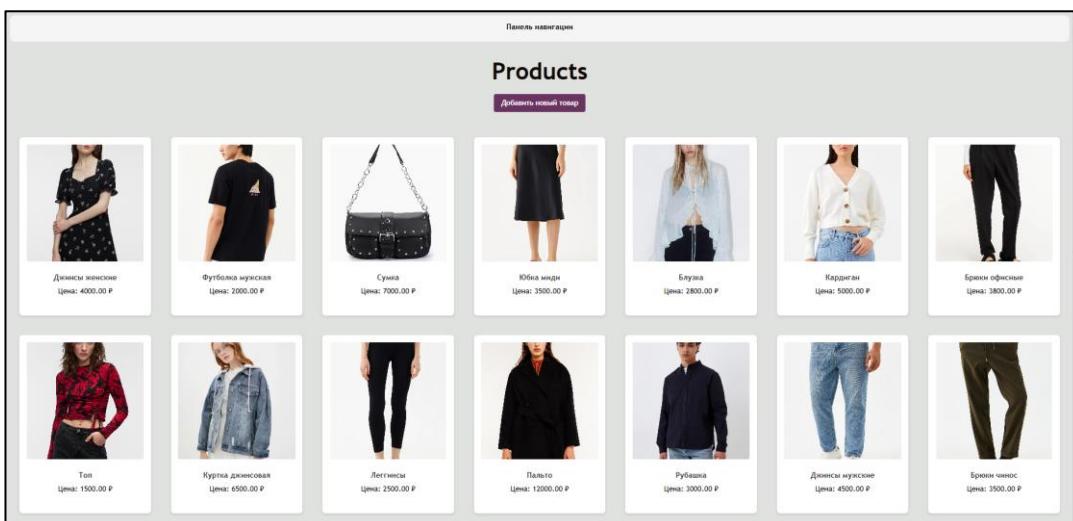


Рисунок 89 – Таблица «Продукты»

Категории товаров																							
Панель навигации																							
<a href="#">Создать новую категорию</a>																							
<table border="1"> <tr> <td>ID: 0 Без категории</td> <td><a href="#">Посмотреть товары</a></td> <td><a href="#">Редактировать</a></td> <td><a href="#">Удалить</a></td> </tr> <tr> <td>ID: 1 Женская одежда</td> <td><a href="#">Посмотреть товары</a></td> <td><a href="#">Редактировать</a></td> <td><a href="#">Удалить</a></td> </tr> <tr> <td>ID: 2 Мужская одежда</td> <td><a href="#">Посмотреть товары</a></td> <td><a href="#">Редактировать</a></td> <td><a href="#">Удалить</a></td> </tr> <tr> <td>ID: 3 Обувь</td> <td><a href="#">Посмотреть товары</a></td> <td><a href="#">Редактировать</a></td> <td><a href="#">Удалить</a></td> </tr> <tr> <td>ID: 4 Аксессуары</td> <td><a href="#">Посмотреть товары</a></td> <td><a href="#">Редактировать</a></td> <td><a href="#">Удалить</a></td> </tr> </table>				ID: 0 Без категории	<a href="#">Посмотреть товары</a>	<a href="#">Редактировать</a>	<a href="#">Удалить</a>	ID: 1 Женская одежда	<a href="#">Посмотреть товары</a>	<a href="#">Редактировать</a>	<a href="#">Удалить</a>	ID: 2 Мужская одежда	<a href="#">Посмотреть товары</a>	<a href="#">Редактировать</a>	<a href="#">Удалить</a>	ID: 3 Обувь	<a href="#">Посмотреть товары</a>	<a href="#">Редактировать</a>	<a href="#">Удалить</a>	ID: 4 Аксессуары	<a href="#">Посмотреть товары</a>	<a href="#">Редактировать</a>	<a href="#">Удалить</a>
ID: 0 Без категории	<a href="#">Посмотреть товары</a>	<a href="#">Редактировать</a>	<a href="#">Удалить</a>																				
ID: 1 Женская одежда	<a href="#">Посмотреть товары</a>	<a href="#">Редактировать</a>	<a href="#">Удалить</a>																				
ID: 2 Мужская одежда	<a href="#">Посмотреть товары</a>	<a href="#">Редактировать</a>	<a href="#">Удалить</a>																				
ID: 3 Обувь	<a href="#">Посмотреть товары</a>	<a href="#">Редактировать</a>	<a href="#">Удалить</a>																				
ID: 4 Аксессуары	<a href="#">Посмотреть товары</a>	<a href="#">Редактировать</a>	<a href="#">Удалить</a>																				

Рисунок 90 – Таблица «Категории»

Покупатели																																							
Панель навигации																																							
<a href="#">+ Добавить покупателя</a> <table border="1"> <tr> <td>Смирнова Мария Алексеевна ID: 1</td> <td><a href="#">Изменить</a></td> <td><a href="#">Удалить</a></td> </tr> <tr> <td>✉ smirnova@mail.com ✉ +79161112233</td> <td></td> <td></td> </tr> <tr> <td>Волков Андрей Игоревич ID: 2</td> <td><a href="#">Изменить</a></td> <td><a href="#">Удалить</a></td> </tr> <tr> <td>✉ volkov@mail.com ✉ +79162233444</td> <td></td> <td></td> </tr> <tr> <td>Белова Екатерина Сергеевна ID: 3</td> <td><a href="#">Изменить</a></td> <td><a href="#">Удалить</a></td> </tr> <tr> <td>✉ belova@mail.com ✉ +79183334455</td> <td></td> <td></td> </tr> <tr> <td>Григорьев Павел Дмитриевич ID: 4</td> <td><a href="#">Изменить</a></td> <td><a href="#">Удалить</a></td> </tr> <tr> <td>✉ grigorev@mail.com ✉ +79164445566</td> <td></td> <td></td> </tr> <tr> <td>Кудрявцева Анна Викторовна ID: 5</td> <td><a href="#">Изменить</a></td> <td><a href="#">Удалить</a></td> </tr> <tr> <td>✉ kudryavtseva@mail.com ✉ +79163556677</td> <td></td> <td></td> </tr> <tr> <td>Орлов Денис Александрович ID: 6</td> <td><a href="#">Изменить</a></td> <td><a href="#">Удалить</a></td> </tr> <tr> <td>✉ orlov@mail.com ✉ +79166667788</td> <td></td> <td></td> </tr> </table>				Смирнова Мария Алексеевна ID: 1	<a href="#">Изменить</a>	<a href="#">Удалить</a>	✉ smirnova@mail.com ✉ +79161112233			Волков Андрей Игоревич ID: 2	<a href="#">Изменить</a>	<a href="#">Удалить</a>	✉ volkov@mail.com ✉ +79162233444			Белова Екатерина Сергеевна ID: 3	<a href="#">Изменить</a>	<a href="#">Удалить</a>	✉ belova@mail.com ✉ +79183334455			Григорьев Павел Дмитриевич ID: 4	<a href="#">Изменить</a>	<a href="#">Удалить</a>	✉ grigorev@mail.com ✉ +79164445566			Кудрявцева Анна Викторовна ID: 5	<a href="#">Изменить</a>	<a href="#">Удалить</a>	✉ kudryavtseva@mail.com ✉ +79163556677			Орлов Денис Александрович ID: 6	<a href="#">Изменить</a>	<a href="#">Удалить</a>	✉ orlov@mail.com ✉ +79166667788		
Смирнова Мария Алексеевна ID: 1	<a href="#">Изменить</a>	<a href="#">Удалить</a>																																					
✉ smirnova@mail.com ✉ +79161112233																																							
Волков Андрей Игоревич ID: 2	<a href="#">Изменить</a>	<a href="#">Удалить</a>																																					
✉ volkov@mail.com ✉ +79162233444																																							
Белова Екатерина Сергеевна ID: 3	<a href="#">Изменить</a>	<a href="#">Удалить</a>																																					
✉ belova@mail.com ✉ +79183334455																																							
Григорьев Павел Дмитриевич ID: 4	<a href="#">Изменить</a>	<a href="#">Удалить</a>																																					
✉ grigorev@mail.com ✉ +79164445566																																							
Кудрявцева Анна Викторовна ID: 5	<a href="#">Изменить</a>	<a href="#">Удалить</a>																																					
✉ kudryavtseva@mail.com ✉ +79163556677																																							
Орлов Денис Александрович ID: 6	<a href="#">Изменить</a>	<a href="#">Удалить</a>																																					
✉ orlov@mail.com ✉ +79166667788																																							

Рисунок 91 – Таблица «Покупатели»

Сотрудники																																																			
Панель навигации																																																			
<a href="#">Добавить нового сотрудника</a> <table border="1"> <tr> <td>Иванова Анна Сергеевна ID: 1 <span style="background-color: #e0e0e0; border-radius: 50%; padding: 2px 5px;">Продавец</span></td> <td><a href="#">Редактировать информацию</a></td> <td><a href="#">Удалить</a></td> </tr> <tr> <td>✉ ivanova@mail.com ✉ Auro - Pusinka</td> <td></td> <td></td> </tr> <tr> <td>✉ Устроена с: 15.05.2020 ✉ Дата рождения: 20.08.1990</td> <td></td> <td></td> </tr> <tr> <td>Петров Дмитрий Игоревич ID: 2 <span style="background-color: #e0e0e0; border-radius: 50%; padding: 2px 5px;">Администратор</span></td> <td><a href="#">Редактировать информацию</a></td> <td><a href="#">Удалить</a></td> </tr> <tr> <td>✉ petrov@mail.com ✉ Auro - Len</td> <td></td> <td></td> </tr> <tr> <td>✉ Устроена с: 10.02.2021 ✉ Дата рождения: 15.11.1985</td> <td></td> <td></td> </tr> <tr> <td>Сидорова Елена Викторовна ID: 3 <span style="background-color: #e0e0e0; border-radius: 50%; padding: 2px 5px;">Менеджер</span></td> <td><a href="#">Редактировать информацию</a></td> <td><a href="#">Удалить</a></td> </tr> <tr> <td>✉ sidorova@mail.com ✉ Auro - Atrasav</td> <td></td> <td></td> </tr> <tr> <td>✉ Устроена с: 22.07.2019 ✉ Дата рождения: 30.04.1988</td> <td></td> <td></td> </tr> <tr> <td>Козлов Алексей Петрович ID: 4 <span style="background-color: #e0e0e0; border-radius: 50%; padding: 2px 5px;">Кассир</span></td> <td><a href="#">Редактировать информацию</a></td> <td><a href="#">Удалить</a></td> </tr> <tr> <td>✉ kozlov@mail.com ✉ Auro - Pusinka</td> <td></td> <td></td> </tr> <tr> <td>✉ Устроена с: 05.01.2022 ✉ Дата рождения: 12.09.1995</td> <td></td> <td></td> </tr> <tr> <td>Михайлова Ольга Дмитриевна ID: 5 <span style="background-color: #e0e0e0; border-radius: 50%; padding: 2px 5px;">Продавец</span></td> <td><a href="#">Редактировать информацию</a></td> <td><a href="#">Удалить</a></td> </tr> <tr> <td>✉ mihailova@mail.com ✉ Auro - Revolution</td> <td></td> <td></td> </tr> <tr> <td>✉ Устроена с: 18.11.2020 ✉ Дата рождения: 25.07.1992</td> <td></td> <td></td> </tr> <tr> <td>Николаев Иван Сергеевич ID: 6 <span style="background-color: #e0e0e0; border-radius: 50%; padding: 2px 5px;">Консультант</span></td> <td><a href="#">Редактировать информацию</a></td> <td><a href="#">Удалить</a></td> </tr> </table>				Иванова Анна Сергеевна ID: 1 <span style="background-color: #e0e0e0; border-radius: 50%; padding: 2px 5px;">Продавец</span>	<a href="#">Редактировать информацию</a>	<a href="#">Удалить</a>	✉ ivanova@mail.com ✉ Auro - Pusinka			✉ Устроена с: 15.05.2020 ✉ Дата рождения: 20.08.1990			Петров Дмитрий Игоревич ID: 2 <span style="background-color: #e0e0e0; border-radius: 50%; padding: 2px 5px;">Администратор</span>	<a href="#">Редактировать информацию</a>	<a href="#">Удалить</a>	✉ petrov@mail.com ✉ Auro - Len			✉ Устроена с: 10.02.2021 ✉ Дата рождения: 15.11.1985			Сидорова Елена Викторовна ID: 3 <span style="background-color: #e0e0e0; border-radius: 50%; padding: 2px 5px;">Менеджер</span>	<a href="#">Редактировать информацию</a>	<a href="#">Удалить</a>	✉ sidorova@mail.com ✉ Auro - Atrasav			✉ Устроена с: 22.07.2019 ✉ Дата рождения: 30.04.1988			Козлов Алексей Петрович ID: 4 <span style="background-color: #e0e0e0; border-radius: 50%; padding: 2px 5px;">Кассир</span>	<a href="#">Редактировать информацию</a>	<a href="#">Удалить</a>	✉ kozlov@mail.com ✉ Auro - Pusinka			✉ Устроена с: 05.01.2022 ✉ Дата рождения: 12.09.1995			Михайлова Ольга Дмитриевна ID: 5 <span style="background-color: #e0e0e0; border-radius: 50%; padding: 2px 5px;">Продавец</span>	<a href="#">Редактировать информацию</a>	<a href="#">Удалить</a>	✉ mihailova@mail.com ✉ Auro - Revolution			✉ Устроена с: 18.11.2020 ✉ Дата рождения: 25.07.1992			Николаев Иван Сергеевич ID: 6 <span style="background-color: #e0e0e0; border-radius: 50%; padding: 2px 5px;">Консультант</span>	<a href="#">Редактировать информацию</a>	<a href="#">Удалить</a>
Иванова Анна Сергеевна ID: 1 <span style="background-color: #e0e0e0; border-radius: 50%; padding: 2px 5px;">Продавец</span>	<a href="#">Редактировать информацию</a>	<a href="#">Удалить</a>																																																	
✉ ivanova@mail.com ✉ Auro - Pusinka																																																			
✉ Устроена с: 15.05.2020 ✉ Дата рождения: 20.08.1990																																																			
Петров Дмитрий Игоревич ID: 2 <span style="background-color: #e0e0e0; border-radius: 50%; padding: 2px 5px;">Администратор</span>	<a href="#">Редактировать информацию</a>	<a href="#">Удалить</a>																																																	
✉ petrov@mail.com ✉ Auro - Len																																																			
✉ Устроена с: 10.02.2021 ✉ Дата рождения: 15.11.1985																																																			
Сидорова Елена Викторовна ID: 3 <span style="background-color: #e0e0e0; border-radius: 50%; padding: 2px 5px;">Менеджер</span>	<a href="#">Редактировать информацию</a>	<a href="#">Удалить</a>																																																	
✉ sidorova@mail.com ✉ Auro - Atrasav																																																			
✉ Устроена с: 22.07.2019 ✉ Дата рождения: 30.04.1988																																																			
Козлов Алексей Петрович ID: 4 <span style="background-color: #e0e0e0; border-radius: 50%; padding: 2px 5px;">Кассир</span>	<a href="#">Редактировать информацию</a>	<a href="#">Удалить</a>																																																	
✉ kozlov@mail.com ✉ Auro - Pusinka																																																			
✉ Устроена с: 05.01.2022 ✉ Дата рождения: 12.09.1995																																																			
Михайлова Ольга Дмитриевна ID: 5 <span style="background-color: #e0e0e0; border-radius: 50%; padding: 2px 5px;">Продавец</span>	<a href="#">Редактировать информацию</a>	<a href="#">Удалить</a>																																																	
✉ mihailova@mail.com ✉ Auro - Revolution																																																			
✉ Устроена с: 18.11.2020 ✉ Дата рождения: 25.07.1992																																																			
Николаев Иван Сергеевич ID: 6 <span style="background-color: #e0e0e0; border-radius: 50%; padding: 2px 5px;">Консультант</span>	<a href="#">Редактировать информацию</a>	<a href="#">Удалить</a>																																																	

Рисунок 92 – Таблица «Сотрудники»

Поставщики							
Поставщик	Контактная информация			Действия			
Fashion Forward Inc. ID: 1	Телефон: +9161234565 Email: fashion.forward@example.com			<a href="#">Подробнее</a>	<a href="#">Изменить</a>	<a href="#">Добавить поставку</a>	<a href="#">Удалить</a>
Trendy Styles Ltd ID: 2	Телефон: +9162345678 Email: trendy.styles@example.com			<a href="#">Подробнее</a>	<a href="#">Изменить</a>	<a href="#">Добавить поставку</a>	<a href="#">Удалить</a>
Urban Wear Co. ID: 3	Телефон: +9163456789 Email: urban.wear@example.com			<a href="#">Подробнее</a>	<a href="#">Изменить</a>	<a href="#">Добавить поставку</a>	<a href="#">Удалить</a>
Classic Apparel LLC ID: 4	Телефон: +9164567890 Email: classic.apparel@example.com			<a href="#">Подробнее</a>	<a href="#">Изменить</a>	<a href="#">Добавить поставку</a>	<a href="#">Удалить</a>
Modern Outfitters ID: 5	Телефон: +9165578901 Email: modern.out@example.com			<a href="#">Подробнее</a>	<a href="#">Изменить</a>	<a href="#">Добавить поставку</a>	<a href="#">Удалить</a>

Рисунок 93 – Таблица «Поставщики»

Поставки			
Название поставки	ID	Поставщик	Действия
Коллекция Amog	#1	Fashion Forward Inc.	<a href="#">Изменить</a> <a href="#">Удалить</a>
Летние новинки	#2	Fashion Forward Inc.	<a href="#">Изменить</a> <a href="#">Удалить</a>
Осенняя коллекция InOctober	#3	Fashion Forward Inc.	<a href="#">Изменить</a> <a href="#">Удалить</a>
Зимняя коллекция Snow	#4	Urban Wear Co.	<a href="#">Изменить</a> <a href="#">Удалить</a>
Молодежный стиль	#5	Trendy Styles Ltd	<a href="#">Изменить</a> <a href="#">Удалить</a>

Рисунок 94 – Таблица «Поставки»

Покупки							
# ID	Дата	Магазин	Сотрудник	Покупатель	Товары	Действия	
300	05.10.2024	Aure - Peace	Петров Дмитрий Игоревич	Волков Андрей Игоревич	• Карадиган - 3	<a href="#">Изменить</a>	<a href="#">Удалить</a>
10	24.01.2023	Aure - Atrasev	Николаев Иван Сергеевич	Волков Андрей Игоревич	No items in this purchase.	<a href="#">Изменить</a>	<a href="#">Удалить</a>
9	23.01.2023	Aure - Atrasev	Михайлова Ольга Дмитриевна	Белова Екатерина Сергеевна	• Куртка джинсовая - 2 • Леггинсы - 10	<a href="#">Изменить</a>	<a href="#">Удалить</a>
8	22.01.2023	Aure - Revolution	Козлов Алексей Петрович	Волков Андрей Игоревич	No items in this purchase.	<a href="#">Изменить</a>	<a href="#">Удалить</a>
7	21.01.2023	Aure - Revolution	Козлов Алексей Петрович	Смирнова Мария Алексеевна	No items in this purchase.	<a href="#">Изменить</a>	<a href="#">Удалить</a>
6	20.01.2023	Aure - Atrasev	Сидорова Елена Викторовна	Соловьев Денис Александрович	• Сумка - 2	<a href="#">Изменить</a>	<a href="#">Удалить</a>
5	19.01.2023	Aure - Atrasev	Сидорова Елена Викторовна	Кудрявцева Анна Викторовна	No items in this purchase.	<a href="#">Изменить</a>	<a href="#">Удалить</a>
4	18.01.2023	Aure - Len	Петров Дмитрий Игоревич	Григорьев Павел Дмитриевич	No items in this purchase.	<a href="#">Изменить</a>	<a href="#">Удалить</a>
3	17.01.2023	Aure - Len	Петров Дмитрий Игоревич	Белова Екатерина Сергеевна	No items in this purchase.	<a href="#">Изменить</a>	<a href="#">Удалить</a>
2	16.01.2023	Aure - Puskin	Иванова Анна Сергеевна	Волков Андрей Игоревич	• Джинсы женские - 1	<a href="#">Изменить</a>	<a href="#">Удалить</a>
20	22.02.2020	Aure - Puskin	Иванова Анна Сергеевна	Григорьев Павел Дмитриевич	• Сумка - 4 • Юбка миди - 2 • Джинсы женские - 1	<a href="#">Изменить</a>	<a href="#">Удалить</a>

Рисунок 95 – Таблица «Покупки»

Панель навигации				
Сформировать возврат				
Возвраты				
#	ID возврата	Дата оформления	Относится к покупке с ID	Действие
5		20.02.2025	№ 4	<input checked="" type="checkbox"/> Рассмотреть <input type="button" value="Удалить"/>
5		05.02.2023	№ 5	<input checked="" type="checkbox"/> Рассмотреть <input type="button" value="Удалить"/>
4		01.02.2023	№ 3	<input checked="" type="checkbox"/> Рассмотреть <input type="button" value="Удалить"/>
3		28.01.2023	№ 5	<input checked="" type="checkbox"/> Рассмотреть <input type="button" value="Удалить"/>
2		25.01.2023	№ 3	<input checked="" type="checkbox"/> Рассмотреть <input type="button" value="Удалить"/>

Рисунок 96 – Таблица «Возвраты»

## **ЗАКЛЮЧЕНИЕ**

В ходе выполнения курсовой работы была создана база данных «Сеть магазинов одежды», направленная на облегчение обработки данных.

Исследована предметная область, рассмотрены особенности работы системы бронирования.

Подробно описана постановка задачи курсового проекта. Определены основные моменты, которые должны быть учтены при проектировании базы данных.

Рассмотрен список выделенных сущностей и их атрибутов. Создана модель «Сущность-связь».

В состав базы данных входят таблицы, между которыми определена связь. Рассмотрены функциональные связи между атрибутами выделенных сущностями, проведена нормализация базы данных к ЗНФ.

Создана инфологическая и даталогическая модели БД.

Для проектирования базы данных была выбрана СУБД MS SQL Server.

Созданы основные запросы к базе данных. Рассмотрены механизмы защиты данных и требования к техническому обеспечению.

Рассмотрена инструкция по использованию базы данных.

В ходе выполнения курсового проекта были приобретены устойчивые навыки проектирования баз данных и разработки запросов. Полученные знания могут быть использованы при разработке более серьезных коммерческих проектов.

## **СПИСОК ЛИТЕРАТУРЫ**

1. Гарсиа-Молина Г., Ульман Д., Уидом Д. Системы баз данных. Полный курс. – М.: Вильямс, 2018. – 1088 с.
2. Коннолли Т., Бегг К. Базы данных: проектирование, реализация и сопровождение. Теория и практика. – М.: Вильямс, 2020. – 1440 с.
3. Карпова Т.С. Базы данных: модели, разработка, реализация. – СПб.: Питер, 2019. – 496 с.
4. Дейт К.Дж. Введение в системы баз данных. – М.: Вильямс, 2018. – 1328 с.
5. Мухин К.А. Проектирование баз данных, М.: Горячая линия – Телеком, 2017. – 240 с.
6. Таненбаум Э., Вудхалл А. Операционные системы. Разработка и реализация. – СПб.: Питер, 2018. – 1120 с.
7. Роб П., Коронел К. Системы баз данных: проектирование, реализация и управление. – М.: БИНОМ, 2019. – 1040 с.
8. Кузнецов С.Д. Основы баз данных. – М.: Интернет-Университет Информационных Технологий, 2020. – 484 с.
9. Грофф Дж.Р., Вайнберг П.Н. SQL: полное руководство. – М.: Вильямс, 2019. – 960 с.
10. Хансен Г., Хансен Д. Базы данных: разработка и управление. – М.: ЭКОМ, 2018. – 512 с.