областное государственное бюджетное профессиональное образовательное учреждение

**«КОСТРОМСКОЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ»**

Специальность «Информационные системы и программирование»

**КУРСОВОЙ ПРОЕКТ**

МДК 11.01 «Технология разработки и

защиты баз данных»

Курсовой проект по теме “Проектирование и разработка

базы данных интернет-магазина”

Проверил: Зайцев Д. В.

Работа защищена: Оценка \_\_\_\_\_\_\_\_\_

«\_\_»\_\_\_\_\_\_ 2023 г.

Выполнила: студентка группы

4-2П9

Чистова Софья

Кострома – 2023

**Образовательное государственное бюджетное профессиональное учреждение**

**«КОСТРОМСКОЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ»**

Специальность: 09.02.07 «Информационные системы и программирование»

|  |
| --- |
| УТВЕРЖДАЮ  Зам. директора по УР.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е.О. Калинина  «18» декабря 2023 г. |

**ЗАДАНИЕ**

на курсовой проект

студентке

Чистовой Софье

группа **4-2П9**

**Тема курсового проекта:**

**ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА БАЗЫ ДАННЫХ ИНТЕРНЕТ-МАГАЗИНА**

**Исходные данные:**

Разработать план создания программного обеспечения (далее – ПО) для автоматизации процессов оформления заказов и управления интернет-магазином.

База данных должна содержать сведения о пользователях, включая персональные данные, информацию о товарах и заказах. Обеспечить возможность оформления заказа, добавления пользователей и оставления отзывов.

**СОСТАВ КУРСОВОГО ПРОЕКТА**

**Введение**

**Раздел 1.** Проектирование базы данных

**Раздел 2.** Создание базы данных

**Заключение**

**Список литературы**

**Приложения**

Срок проектирования:

Начало: «28» ноября 2023 г.; окончание: «18» декабря 2023 г.

Руководитель курсового проекта \_\_\_\_\_\_\_\_\_\_\_\_\_ Зайцев Д.В.

**ОГЛАВЛЕНИЕ**

[ВВЕДЕНИЕ 4](#_i5ti8ignbuwn)

[1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 5](#_w3hgc6a1tr7)

[2. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ 8](#_9cvl9r19hnp2)

[2.1 Реляционная модель 8](#_joofw4ra3v44)

[2.2 Создание ERD 13](#_vp1qu5czgq5g)

[3. СОЗДАНИЕ БАЗЫ ДАННЫХ 14](#_ittc64a1egn9)

[3.1 Выбор СУБД 14](#_8fkhc5ht978d)

[3.2 Создание таблиц с полями 15](#_d2hkjllakmne)

[3.3 Настройка связей 21](#_wcxw3b9me5j6)

[3.4 Настройка ограничений 23](#_421d6i76vado)

[3.5 Заполнение базы данных 25](#_4ynmw1us9qgr)

[3.6 Экспорт данных 28](#_8cn4enlvm1x)

[3.7 Создание представлений 29](#_8k4j3kban58i)

[3.8 Создание индексов 30](#_e98x25t0wxjg)

[3.9 Написание хранимых процедур 31](#_nxor4ft9x7fh)

[3.10 Написание триггеров 32](#_nz0k46d6zw46)

[3.11 Настройка резервного копирования 33](#_e30n93apxrh5)

# **ВВЕДЕНИЕ**

Интернет-магазины стали незаменимым элементом современной торговли. Их рост и влияние на розничную торговлю стали фундаментальной частью потребительского опыта. Они обеспечивают покупателям широкий доступ к товарам и услугам, облегчая процесс поиска, сравнения и приобретения продукции. Благодаря интернет-магазинам у потребителей есть возможность делать покупки в любое время суток и исчезла необходимость физически присутствовать при покупке нужного товара. А также, компании и предприниматели находят новые возможности для доступа к мировым рынкам через интернет-продажи. Онлайн-платформы помогают компаниям расширить свою аудиторию за счет доступа к глобальному рынку. Это открывает двери для предпринимателей и малых бизнесов, позволяя им продвигать свои товары и услуги за пределы местного рынка.

Актуальность разработки базы данных обусловлена необходимостью использования информационных систем и технологий как средства повышения производительности и эффективности работы.

Целью данной работы является проектирование и разработка базы данных интернет-магазина для обеспечения эффективного управления товарами, заказами, клиентами и другими важными данными, необходимыми для успешного функционирования онлайн-торговли.

# **АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ**

Интернет-магазин — сайт, торгующий товарами посредством сети Интернет. Позволяет пользователям онлайн, в своём браузере или через мобильное приложение, сформировать заказ на покупку, выбрать способ оплаты и доставки заказа, оплатить заказ.

Важно учесть, что интернет-магазин будет расти со временем, поэтому база данных должна быть гибкой и масштабируемой. База данных должна быть спроектирована так, чтобы легко масштабироваться, добавляя новые таблицы или расширяя существующие. Также важно обеспечить производительность системы при увеличении объема данных и нагрузки.

В условиях постоянного расширения объема информации в базе данных, необходимо учесть все проблемы, чтобы обеспечить стабильную работу интернет-магазина даже при постоянном росте данных. База данных должна иметь:

* Эффективную структуру данных: Важно иметь хорошо спроектированную структуру базы данных. Для этого нужно провести анализ типов данных, связей между таблицами, а также типичных запросов, которые будут подаваться к базе данных.
* Нормализацию данных: Проектирование базы данных с учетом нормализации поможет избежать дублирования данных и поддерживать целостность. Это позволит избежать проблем точности и актуальности данных.
* Масштабируемость: База данных должна быть спроектирована с учётом возможности масштабирования. Таблицы, которые могут быстро расти, должны быть спроектированы с учетом горизонтального масштабирования.

Проанализировав данную предметную область, база данных будет иметь сущности:

Заказ:

* Сумма заказа,
* Статус,
* Номер для отслеживания,
* Адрес доставки,
* Дата доставки.

Товар:

* Цена,
* Размер,
* Цвет,
* Материал,
* Описание,
* Фото,
* Дата добавления.

Пользователь:

* Имя,
* Фамилия,
* Отчество,
* Роль,
* Возраст,
* Адрес,
* Скидка,
* Логин,
* Пароль.

Оплата:

* Метод оплаты.

Отзыв:

* Оценка,
* Пользователь,
* Описание,
* Дата написания.

Категория:

* Тип товара.

Фирма:

* Название,
* Описание,
* Дата регистрации.

Роль:

* Роль.

.Товар-Заказ:

* Количество,
* Цена,
* Сумма.

Пользователь-Фирма:

* Роль

# **ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ**

## **2.1 Реляционная модель**

1. Таблица “Заказ (Order\_of\_customer)”  
   Атрибуты:

* Идентификатор (Первичный ключ) id\_order;
* ID пользователя(Внешний ключ, ссылается на id\_user);
* ID оплаты (Внешний ключ, ссылается на id\_payment);
* Сумма заказа (total\_value);
* Статус (status);
* Номер для отслеживания (tracking\_number);
* Адрес доставки (delivery\_address);
* Дата доставки (delivery\_date).

Связи:

* Многие-к-одному с таблицей “Пользователь” (по внешнему ключу id\_user в таблице “Заказ”);
* Многие-к-одному с таблицей “Оплата” (по внешнему ключу id\_payment в таблице “Заказ”);
* Один-ко-многим с таблицей “Товар-Заказ” (по первичному ключу id\_order в таблице “Товар-Заказ”);

1. Таблица “Товар (Product)”  
   Атрибуты:

* Идентификатор (Первичный ключ) id\_product;
* ID фирмы (Внешний ключ, ссылается на id\_firm);
* ID категории (Внешний ключ, ссылается на id\_category);
* Цена (price);
* Размер (size);
* Цвет (color);
* Материал (fabric);
* Описание (description);
* Фото (photo);
* Дата добавления (added\_date).

Связи:

* Один-ко-многим с таблицей “Товар-Заказ” (по первичному ключу id\_product в таблице “Товар-Заказ”);
* Один-ко-многим с таблицей “Отзыв” (по внешнему ключу id\_product в таблице “Отзыв”);
* Многие-к-одному с таблицей “Фирма” (по внешнему ключу id\_firm в таблице “Товар”);
* Многие-к-одному с таблицей “Категория” (по внешнему ключу id\_category в таблице “Товар”).

1. Таблица “Товар-Заказ (ProductOrder)”

Атрибуты:

* Идентификатор (Первичный ключ) id\_product;
* Идентификатор (Первичный ключ) id\_order;
* Количество (quantity);
* Цена (price);
* Сумма заказа (total\_value).

Связи:

* Многие к одному с таблицей “Товар” (по первичному ключу id\_product в таблице “Товар”);
* Многие к одному с таблицей “Заказ” (по первичному ключу id\_order в таблице “Заказ”);

1. Таблица “Пользователь(UserShop)”  
   Атрибуты:

* Идентификатор (Первичный ключ) id\_user;
* ID роли (Внешний ключ, ссылается на id\_role);
* Имя (name);
* Фамилия (surname);
* Отчество (patronymic);
* Возраст (age);
* Адрес (address);
* Скидка (discount);
* Логин (login);
* Пароль (password).

Связи:

* Один-ко-многим с таблицей “Заказ” (по внешнему ключу id\_user в таблице “Заказ”).
* Один-ко-многим с таблицей “Пользователь-Фирма” (по первичному ключу id\_user в таблице “Пользователь-Фирма”).
* Один-ко-многим с таблицей “Отзыв” (по внешнему ключу id\_user в таблице “Отзыв”).
* Многие-к-одному с таблицей “Роль” (по внешнему ключу id\_role в таблице “Пользователь”).

1. Таблица “Роль (RoleUser)”  
   Атрибуты:

* Идентификатор (Первичный ключ) id\_role;
* Роль (role).

Связи:

* Один-ко-многим с таблицей “Пользователь-Фирма” (по внешнему ключу id\_role в таблице “Пользователь-Фирма”).
* Один-ко-многим с таблицей “Пользователь” (по внешнему ключу id\_role в таблице “Пользователь”).

1. Таблица “Пользователь-Фирма (UserFirm)”  
   Атрибуты:

* Идентификатор (Первичный ключ) id\_firm;
* Идентификатор (Первичный ключ) id\_user;
* ID роли (Внешний ключ, ссылается на id\_role).

Связи:

* Один-ко-многим с таблицей “Пользователь” (по первичному ключу id\_user в таблице “Пользователь”).
* Один-ко-многим с таблицей “Фирма” (по первичному ключу id\_firm в таблице “Фирма”).

1. Таблица “Фирма (Firm)”  
   Атрибуты:

* Идентификатор (Первичный ключ) id\_firm;
* Имя (name);
* Описание (description);
* Дата регистрации (registered\_date).

Связи:

* Один-ко-многим с таблицей “Пользователь-Фирма” (по первичному ключу id\_firm в таблице “Пользователь-Фирма”).
* Один-ко-многим с таблицей “Товар” (по внешнему ключу id\_firm в таблице “Товар”).

1. Таблица “Оплата (Payment)”  
   Атрибуты:

* Идентификатор (Первичный ключ) id\_payment;
* Способ оплаты (payment\_method).

Связи:

* Один-ко-многим с таблицей “Заказ” (по внешнему ключу id\_payment в таблице “Заказ”).

1. Таблица “Отзыв (Review)”  
   Атрибуты:

* Идентификатор (Первичный ключ) id\_review;
* ID товара (Внешний ключ, ссылается на id\_product);
* ID пользователя (Внешний ключ, ссылается на id\_user).
* Оценка (evaluation);
* Описание (description).

Связи:

* Многие-к-одному с таблицей “Товар” (по внешнему ключу id\_product в таблице “Отзыв”).

1. Таблица “Категория (Category)”  
   Атрибуты:

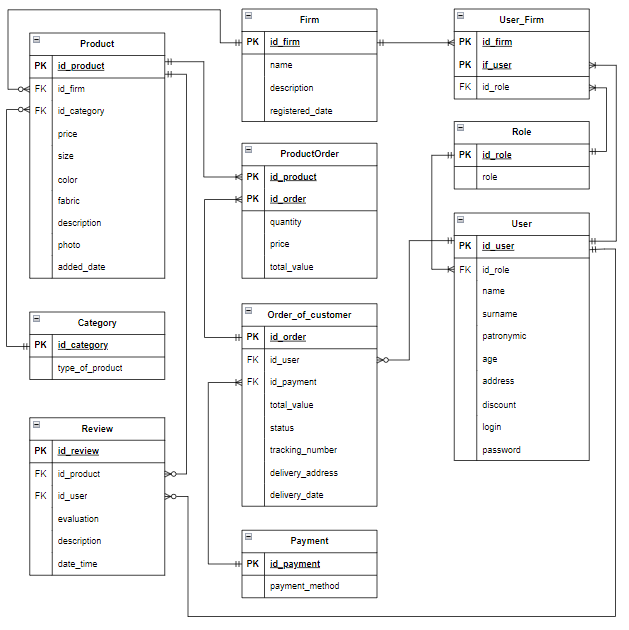
* Идентификатор (Первичный ключ) id\_category;
* Тип продукта (type\_of\_product).

Связи:

* Один-ко-многим с таблицей “Товар” (по внешнему ключу id\_category в таблице “Товар”).

## **2.2 Создание ERD**

Диаграмма отношений сущностей (ERD) — это визуальное представление базы данных, которое показывает, как связаны элементы внутри. ERD-атрибуты характеризуют сущности, позволяя пользователям лучше разобраться в устройстве базы данных. Атрибуты содержат информацию о сущностях, выделенных в концептуальной ER-диаграмме. Характеризует сущность, а также отношения между двумя или более элементами.



*Рисунок 2.2.1 Схема ERD*

# **СОЗДАНИЕ БАЗЫ ДАННЫХ**

## **3.1 Выбор СУБД**

При существовании большого количества различных СУБД со своими особенностями, для проектирования базы данных интернет-магазина будет использоваться СУБД MSSQL, так как имеет такие качества, как:

1. Отличная производительность: SQL Server обеспечивает отличную производительность как для OLTP (Online Transaction Processing), так и для OLAP (Online Analytical Processing) работ. Он обладает мощным оптимизатором запросов, индексацией, а также поддержкой параллельной обработки запросов.
2. Широкий спектр функций: SQL Server предлагает широкий спектр встроенных функций и хранимых процедур, что делает его мощным инструментом для обработки данных, вычислений и управления базой данных.
3. Безопасность: SQL Server обеспечивает многоуровневую систему безопасности, позволяющую управлять доступом к данным и обеспечивать их конфиденциальность. Это включает в себя авторизацию, аутентификацию, шифрование и аудит базы данных.
4. Простота интеграции: Благодаря широкой популярности и хорошей документации, SQL Server легко интегрируется с другими инструментами и платформами, что обеспечивает удобство разработки и сопровождения приложений.
5. Поддержка различных типов данных: SQL Server предоставляет поддержку для различных типов данных, включая географические данные, XML, JSON и другие, что делает его универсальным инструментом для хранения и обработки разнообразной информации.

## 

## **3.2 Создание таблиц с полями**

Код создания таблиц:

CREATE TABLE RoleUser(

id\_role INT IDENTITY PRIMARY KEY NOT NULL,

role NVARCHAR(20) NOT NULL

);

CREATE TABLE UserShop(

id\_user INT IDENTITY PRIMARY KEY NOT NULL,

id\_role INT NOT NULL REFERENCES RoleUser(id\_role) ON DELETE CASCADE,

name NVARCHAR(50) NOT NULL,

surname NVARCHAR(50) NOT NULL,

patronymic NVARCHAR(50),

age INT CHECK(age>=18) NOT NULL,

address NVARCHAR(50) NOT NULL,

discount INT,

login NVARCHAR(30) NOT NULL,

password NVARCHAR(20) NOT NULL

);

CREATE TABLE Firm(

id\_firm INT IDENTITY PRIMARY KEY NOT NULL,

name NVARCHAR(20) NOT NULL,

description NVARCHAR(100),

registered\_date DATE NOT NULL

);

CREATE TABLE UserFirm(

id\_firm INT NOT NULL REFERENCES Firm(id\_firm) ON DELETE CASCADE,

id\_user INT NOT NULL REFERENCES UserShop(id\_user) ON DELETE CASCADE,

CONSTRAINT id\_firm\_user PRIMARY KEY (id\_firm, id\_user),

id\_role INT NOT NULL REFERENCES RoleUser(id\_role)

);

CREATE TABLE Category(

id\_category INT IDENTITY PRIMARY KEY,

type\_of\_category NVARCHAR(20) NOT NULL

);

CREATE TABLE Payment(

id\_payment INT IDENTITY PRIMARY KEY NOT NULL,

payment\_method NVARCHAR(50) NOT NULL

);

CREATE TABLE Order\_of\_customer(

id\_order INT IDENTITY PRIMARY KEY NOT NULL,

id\_user INT NOT NULL REFERENCES UserShop(id\_user) ON DELETE CASCADE,

id\_payment INT NOT NULL REFERENCES Payment(id\_payment) ON DELETE CASCADE,

total\_value MONEY NOT NULL,

status NVARCHAR(20) NOT NULL,

tracking\_number INT NOT NULL,

delivery\_address NVARCHAR(50) NOT NULL,

delivery\_date DATE NOT NULL

);

CREATE TABLE Product(

id\_product INT IDENTITY PRIMARY KEY NOT NULL,

id\_firm INT NOT NULL REFERENCES Firm(id\_firm) ON DELETE CASCADE,

id\_category INT NOT NULL REFERENCES Category(id\_category) ON DELETE CASCADE,

price MONEY NOT NULL,

size NVARCHAR(10),

color NVARCHAR(20),

fabric NVARCHAR(50),

description NVARCHAR(100),

photo IMAGE,

added\_date DATE NOT NULL

);

CREATE TABLE ProductOrder(

id\_product INT NOT NULL REFERENCES Product(id\_product) ON DELETE CASCADE,

id\_order INT NOT NULL REFERENCES Order\_of\_customer(id\_order) ON DELETE CASCADE,

CONSTRAINT id\_product\_order PRIMARY KEY (id\_product, id\_order),

quantity INT NOT NULL,

price MONEY NOT NULL,

total\_value MONEY NOT NULL

);

CREATE TABLE Review(

id\_review INT IDENTITY PRIMARY KEY NOT NULL,

id\_product INT NOT NULL REFERENCES Product(id\_product) ON DELETE CASCADE,

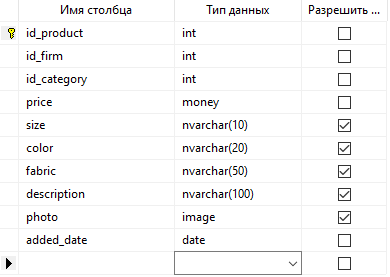
id\_user INT NOT NULL REFERENCES UserShop(id\_user) ON DELETE CASCADE,

evaluation INT NOT NULL,

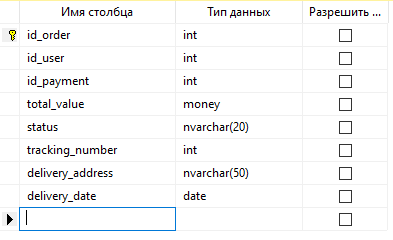
description NVARCHAR(100),

date\_time SMALLDATETIME NOT NULL

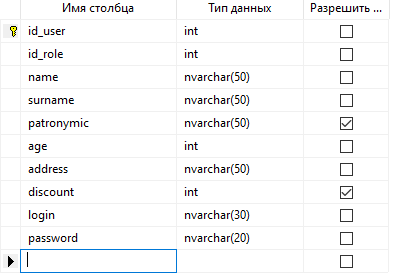
);



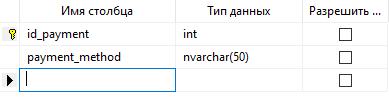
*Рисунок 3.2.1 Таблица “Товар”*

**

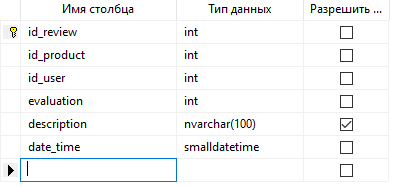
*Рисунок 3.2.2 Таблица “Заказ”*

**

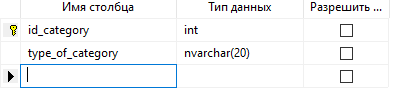
*Рисунок 3.2.3 Таблица “Пользователь”*

**

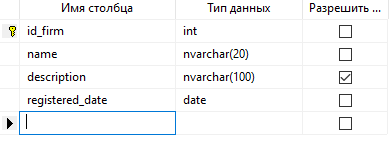
*Рисунок 3.2.4 Таблица “Оплата”*

**

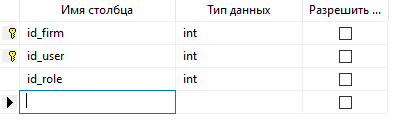
*Рисунок 3.2.5 Таблица “Отзыв”*

**

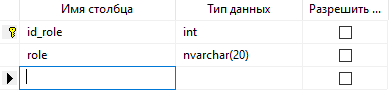
*Рисунок 3.2.6 Таблица “Категория”*

**

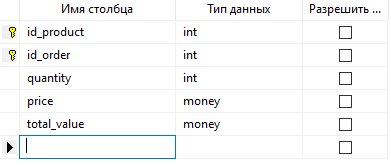
*Рисунок 3.2.7 Таблица “Фирма”*

****

*Рисунок 3.2.8 Таблица “Пользователь-Фирма”*

**

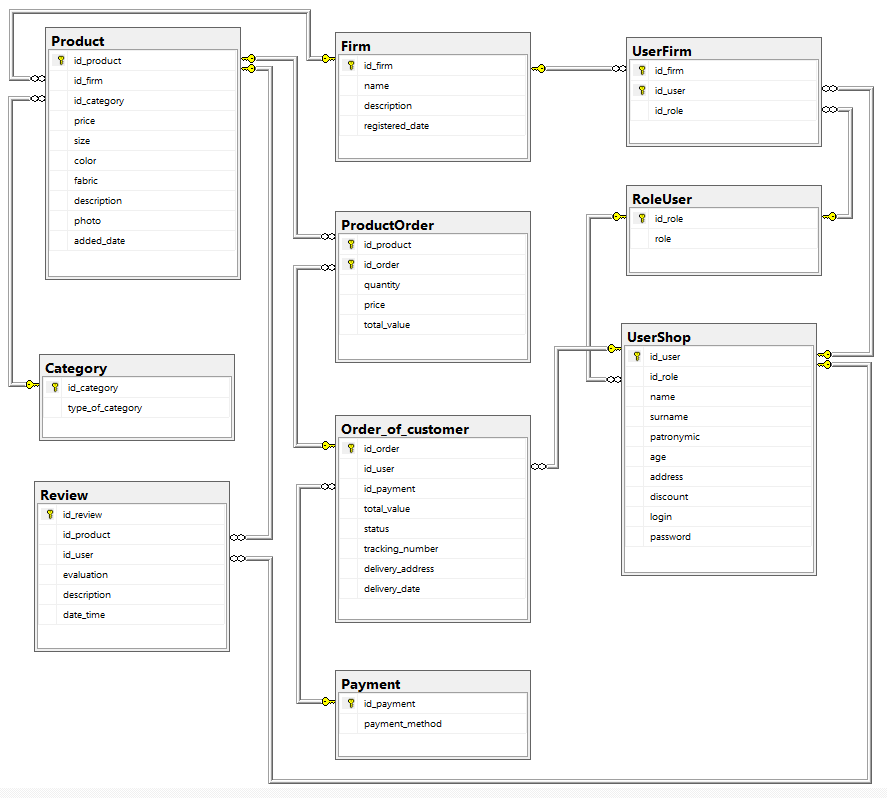
*Рисунок 3.2.9 Таблица “Роль”*

**

*Рисунок 3.2.10 Таблица “Товар-Заказ”*

## 

## **3.3 Настройка связей**



*Рисунок 3.3.1 Диаграмма*

Связи:

1. Таблица Product по первичному ключу id\_product связана с таблицей ProductOrder(id\_product);
2. Таблица Product по первичному ключу id\_product связана с таблицей Review(id\_product);
3. Таблица Category по первичному ключу id\_category связана с таблицей Product(id\_category) ;
4. Таблица Firm по первичному ключу id\_firm связана с таблицей Product(id\_firm) ;
5. Таблица Firm по первичному ключу id\_firm связана с таблицей UserFirm(id\_firm);
6. Таблица Order\_of\_customer по первичному ключу id\_order связана с таблицей ProductOrder(id\_order);
7. Таблица Payment по первичному ключу id\_payment связана с таблицей Order\_of\_customer(id\_payment);
8. Таблица RoleUser по первичному ключу id\_role связана с таблицей UserFirm(id\_role);
9. Таблица RoleUser по первичному ключу id\_role связана с таблицей UserShop(id\_role);
10. Таблица UserShop по первичному ключу id\_user связана с таблицей Order\_of\_customer(id\_user);
11. Таблица UserShop по первичному ключу id\_user связана с таблицей Userfirm(id\_user);
12. Таблица UserShop по первичному ключу id\_user связана с таблицей Review(id\_user).

## 

## **3.4 Настройка ограничений**

Первичные ключи были применены к следующим таблицам:

* Таблица UserShop: Первичный ключ id\_user
* Таблица Payment: Первичный ключ id\_payment
* Таблица Review: Первичный ключ id\_review
* Таблица Category: Первичный ключ id\_category
* Таблица Firm: Первичный ключ id\_firm
* Таблица Product: Первичный ключ id\_product
* Таблица Order\_of\_customer: Первичный ключ id\_order
* Таблица RoleUser: Первичный ключ id\_role
* Таблица UserFirm: Первичные ключи id\_firm, id\_user
* Таблица ProductOrder: Первичные ключи id\_order, id\_product

Внешние ключи были применены для обеспечения связей между таблицами:

* Таблица Product:
  + Внешний ключ id\_firm связан с таблицей Firm;
  + Внешний ключ id\_category связан с таблицей Category;
* Таблица Order\_of\_customer:
  + Внешний ключ id\_user связан с таблицей UserShop;
  + Внешний ключ id\_payment связан с таблицей Payment;
* Таблица Review:
  + Внешний ключ id\_product связан с таблицей Product;
  + Внешний ключ id\_user связан с таблицей UserShop.
* Таблица ProductOrder:
  + Первичный ключ id\_product связан с таблицей Product;
  + Первичный ключ id\_order связан с таблицей Order\_of\_customer.
* Таблица UserFirm:
  + Первичный ключ id\_firm связан с таблицей Firm;
  + Первичный ключ id\_user связан с таблицей UserShop;
  + Внешний ключ id\_role связан с таблицей RoleUser.
* Таблица UserShop:
  + Внешний ключ id\_role связан с таблицей RoleUser.

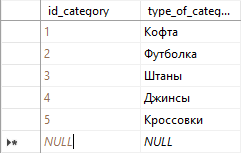
Условия проверки:

* Таблица UserShop: Установлено условие проверки для возраста (age), требуя, чтобы он был не менее 18 лет.

Ограничение NOT NULL было установлено для следующих столбцов таблиц:

* Product: id\_product, id\_firm, id\_category, price, added\_date;
* Order\_of\_customer: id\_order, id\_user, id\_payment, total\_value, status, tracking\_number, delivery\_address, delivery\_date;
* UserShop: id\_user, id\_rple, name, surname, age, address, login, password;
* Payment: id\_payment, payment\_method;
* Review: id\_review, id\_product, id\_user, evaluation, date\_time;
* Category: id\_category, type\_of\_category;
* Firm: id\_firm, name, registered\_date;
* UserFirm: id\_firm, id\_user, id\_role;
* RoleUser: id\_role, role;
* ProductOrder: id\_product, id\_order, quantity, price, total\_value.

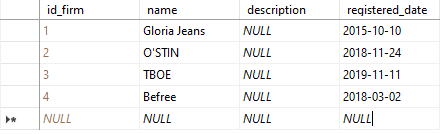
## **3.5 Заполнение базы данных**



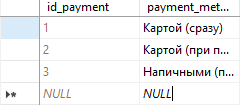
*Рисунок 3.5.1 Данные в таблице “Категория”*



*Рисунок 3.5.2 Данные в таблице “Пользователь”*

**

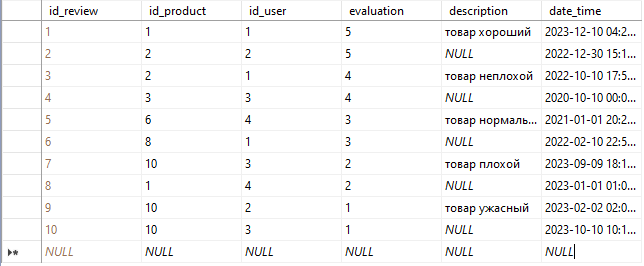
*Рисунок 3.5.3 Данные в таблице “Фирма”*



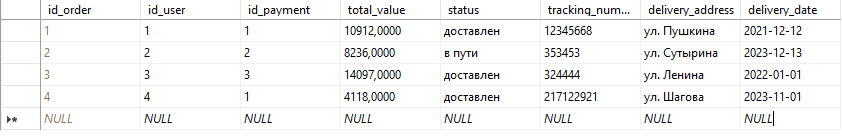
*Рисунок 3.5.4 Данные в таблице “Оплата”*



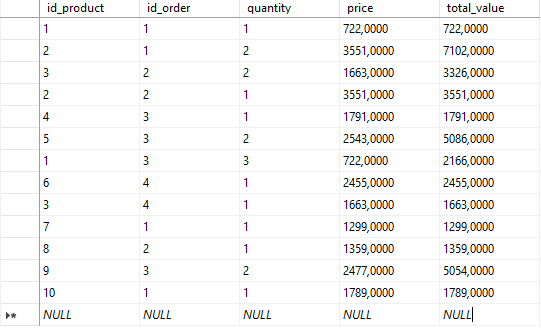
*Рисунок 3.5.5 Данные в таблице “Товар”*

**

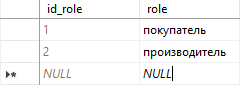
*Рисунок 3.5.6 Данные в таблице “Отзыв”*

**

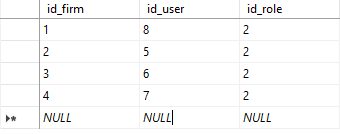
*Рисунок 3.5.7 Данные в таблице “Заказ”*

**

*Рисунок 3.5.8 Данные в таблице “Товар-Заказ”*

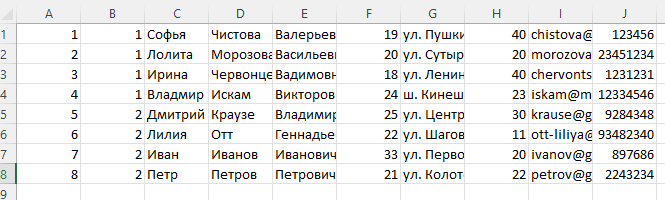
**

*Рисунок 3.5.9 Данные в таблице “Роль”*

**

*Рисунок 3.5.10 Данные в таблице “Пользователь-Фирма”*

## **3.6 Экспорт данных**



*Рисунок 3.6.1 Файл экспорта таблицы UserShop в csv*

## **3.7 Создание представлений**

Отзывы о продуктах с дополнительной информацией о пользователе и продукте:

CREATE VIEW ReviewsUserProduct AS

SELECT C.type\_of\_category AS [Категория],

P.color AS [Цвет],

P.price AS [Цена],

U.name AS [Имя],

R.date\_time AS [Дата/Время],

R.evaluation AS [Оценка],

R.description AS [Описание]

FROM Review R

JOIN UserShop U ON R.id\_user = U.id\_user

JOIN Product P ON R.id\_product = P.id\_product

JOIN Category C ON P.id\_category = C.id\_category;



*Рисунок 3.7.1 Представление ReviewsUserProduct*

Заказы клиентов с дополнительной информацией о пользователе и оплате:

CREATE VIEW CustomerOrdersInfo AS

SELECT O.id\_order AS [Код заказа],

O.total\_value AS [Сумма],

O.status AS [Статус],

O.tracking\_number AS [Номер отслеживания],

O.delivery\_address AS [Адрес доставки],

O.delivery\_date AS [Дата доставки],

U.name AS [Имя],

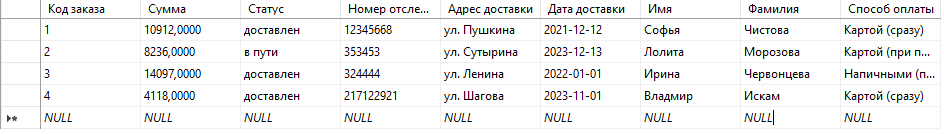
U.surname AS [Фамилия],

P.payment\_method AS [Способ оплаты]

FROM Order\_of\_customer O

JOIN UserShop U ON O.id\_user = U.id\_user

JOIN Payment P ON O.id\_payment = P.id\_payment;



*Рисунок 3.7.2 Представление CustomerOrdersInfo*

## **3.8 Создание индексов**

В данной работе не используются индексы, помимо созданных по умолчанию (по умолчанию для каждого PRIMARY KEY создается индекс).

## **3.9 Написание хранимых процедур**

Создание хранимой процедуры, добавляющей нового пользователя:

USE Kurs\_DB;

GO

CREATE PROCEDURE AddUser

@roleID INT,

@name NVARCHAR(50),

@surname NVARCHAR(50),

@patronymic NVARCHAR(50),

@age INT,

@address NVARCHAR(50),

@discount INT,

@login NVARCHAR(30),

@password NVARCHAR(20) AS

INSERT INTO UserShop(id\_role, name, surname, patronymic, age, address, discount, login, password)

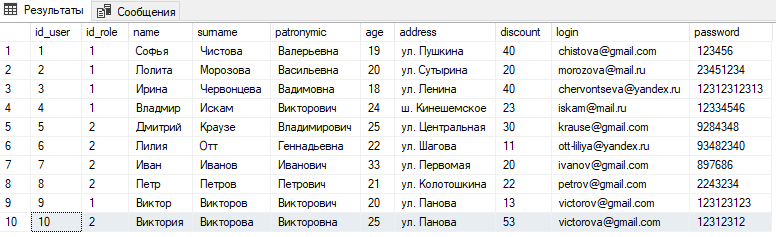
VALUES(@roleID, @name, @surname, @patronymic, @age, @address, @discount, @login, @password)

Выполнение процедуры:

USE Kurs\_DB;

EXEC AddUser 2, 'Виктория', 'Викторова','Викторовна',25,'ул. Панова',53,'victorovа@gmail.com','12312312'

SELECT \* FROM UserShop



*Рисунок 3.9.1 Результат работы хранимой процедуры AddUser*

## **3.10 Написание триггеров**

Триггер для обновления общей стоимости заказа при изменении количества товаров в таблице ProductOrder:

CREATE TRIGGER UpdateTotalValue

ON ProductOrder

AFTER INSERT, UPDATE, DELETE

AS

BEGIN

IF EXISTS (SELECT \* FROM inserted)

BEGIN

UPDATE Order\_of\_customer

SET total\_value =

(SELECT SUM(PO.quantity \* P.price)

FROM inserted I

JOIN ProductOrder PO ON PO.id\_order = I.id\_order

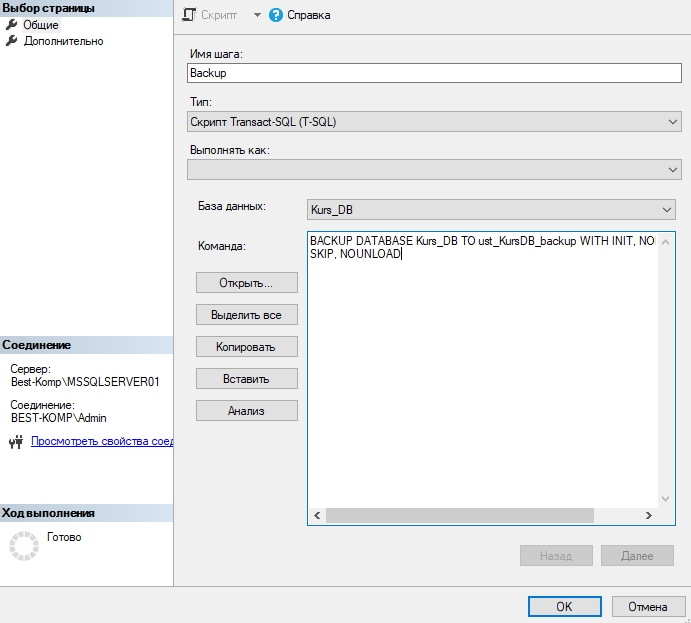
JOIN Product P ON PO.id\_product = P.id\_product

WHERE PO.id\_order = I.id\_order);

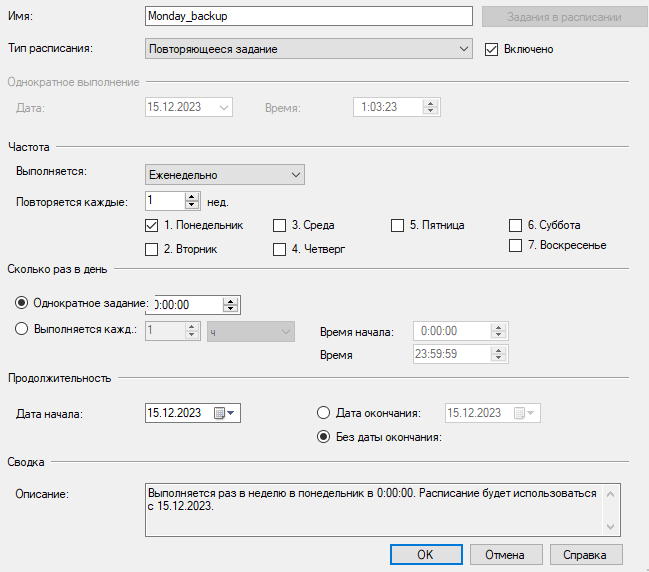
END

END;

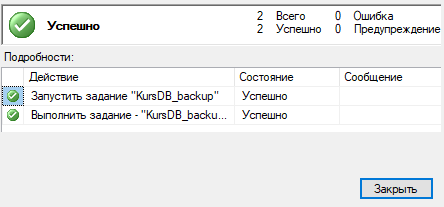
## **3.11 Настройка резервного копирования**



*Рисунок 3.11.1 Создание задания Backup*



*Рисунок 3.11.2 Настройка еженедельного резервного копирования*



*Рисунок 3.11.3 Результат выполнения задания*

# **ЗАКЛЮЧЕНИЕ**

Разработка базы данных для интернет-магазина представляет собой не только техническую задачу, но и стратегическую возможность обеспечить надежное и гибкое хранение информации об ассортименте, заказах, клиентах и других важных аспектах бизнеса. Проектирование и разработка базы данных позволит обеспечить эффективное управление данными, улучшить процессы принятия решений и повысить производительность онлайн-торговли.

Проектирование и разработка базы данных позволит обеспечить эффективное управление данными, улучшить процессы принятия решений и повысить производительность онлайн-торговли. С учетом быстрого роста онлайн-торговли и постоянно меняющихся потребностей рынка, разработка базы данных становится ключевым аспектом успешного функционирования и развития бизнеса.

В этой работе были решены следующие задачи:

* Написание реляционной модели базы данных;
* Создание ER диаграммы;
* Создание базы данных, включая таблицы, поля и связи;
* Создание хранимых процедур, триггеров, представлений;
* Настройка резервного копирования.

# **СПИСОК ЛИТЕРАТУРЫ**

1. Синтаксис SQL [Сетевой ресурс] <https://sql-academy.org/ru/guide>
2. Различные статьи [Сетевой ресурс] <https://ru.wikipedia.org>
3. Документация по Microsoft SQL [Сетевой ресурс] <https://learn.microsoft.com/ru-ru/sql>
4. Резервное копирование базы данных [Сетевой ресурс] <https://winitpro.ru/index.php/2020/02/26/backup-i-vosstanovlenie-baz-dannyx-mssql/>
5. Установка MS SQL Server [Сетевой ресурс] <https://metanit.com/sql/sqlserver/1.2.php>