

## Проектирование базы данных интернет-магазина в dbForge Studio for MySQL

*Матвеев Сергей Сергеевич*

*Приамурский государственный университет имени Шолом-Алейхема*  
*Студент*

### Аннотация

В статье исследуется методология управления данными при проектировании реляционной базы данных для интернет-магазина в контексте цифровизации экономики и развития электронной коммерции. Актуальность работы обусловлена повышенными требованиями к целостности, производительности и масштабируемости информационных систем, обрабатывающих большие объемы разнородных данных. В качестве инструментальной среды разработки использована dbForge Studio for MySQL. Целью исследования является разработка и практическая апробация методологии, включающей анализ требований, проектирование структуры базы данных с применением нормализации, обеспечение ссылочной целостности и реализацию механизмов автоматизации бизнес-логики. В результате создана функциональная база данных, включающая ключевые сущности: Customers, Categories, Products, Orders, OrderDetails, Payments. Для поддержания актуальности данных реализован триггер ReduceStock, автоматически корректирующий остатки товаров. Проведенное исследование подтверждает, что применение системного подхода к проектированию с использованием специализированных инструментов позволяет создать надежную, производительную и легко администрируемую базу данных, удовлетворяющую потребностям современных систем электронной коммерции.

**Ключевые слова:** Управление данными, база данных, проектирование БД, MySQL, dbForge Studio, реляционная модель, целостность данных, нормализация, триггер, представление.

### Designing an internet store database in dbForge Studio for MySQL

*Matveev Sergey Sergeevich*

*Sholom-Aleichem Priamursky State University*  
*Student*

### Abstract

The article explores the methodology of data management in designing a relational database for an e-commerce store within the context of digitalization of the economy and the development of e-commerce. The relevance of the work is driven by increased demands for the integrity, performance, and scalability of information

systems that process large volumes of heterogeneous data. The dbForge Studio for MySQL was used as the development tool. The aim of the research is the development and practical testing of a methodology that includes requirements analysis, database structure design using normalization, ensuring referential integrity, and implementing business logic automation mechanisms. As a result, a functional database was created, encompassing key entities: Customers, Categories, Products, Orders, OrderDetails, Payments. To maintain data relevance, the ReduceStock trigger was implemented, automatically adjusting product stock levels. The conducted research confirms that applying a systematic design approach using specialized tools enables the creation of a reliable, high-performance, and easily manageable database that meets the needs of modern e-commerce systems.

**Keywords:** Data management, database, database design, MySQL, dbForge Studio, relational model, data integrity, normalization, trigger, view.

## 1. Введения

### 1.1 Актуальность

В условиях цифровизации экономики и стремительного развития электронной коммерции интернет-магазины становятся одним из ключевых каналов взаимодействия между бизнесом и потребителями. Эффективное функционирование интернет-магазина напрямую зависит от качества организации и управления данными, поскольку именно данные о товарах, клиентах, заказах и платежах являются основой бизнес-процессов и управленческих решений.

Рост объёмов данных, их разнородность и высокая интенсивность обновления предъявляют повышенные требования к проектированию баз данных и методам управления ими. Ошибки на этапе разработки структуры базы данных могут приводить к снижению производительности системы, потере целостности данных, усложнению масштабирования и повышению затрат на сопровождение информационной системы. В связи с этим особую значимость приобретает разработка методологии управления данными, ориентированной на обеспечение целостности, согласованности и доступности информации.

Использование специализированных инструментальных средств, таких как dbForge Studio for MySQL, позволяет автоматизировать процессы проектирования, администрирования и оптимизации баз данных. Однако эффективность применения данных инструментов во многом зависит от наличия системного подхода к управлению данными и соблюдения методологических принципов их организации.

Таким образом, исследование методологии управления данными при разработке базы данных интернет-магазина с использованием dbForge Studio for MySQL является актуальным, поскольку способствует повышению надёжности, производительности и масштабируемости информационных систем электронной коммерции, а также формированию практических рекомендаций по эффективному управлению данными.

## **1.2 Обзор исследований**

Э.С. Анисимова характеризует DbForge Studio как универсальный и удобный инструмент для MySQL, который предоставляет разработчикам и администраторам комплексные возможности по созданию запросов, отладке процедур и автоматизированному управлению объектами БД [1]. С. Крутилин в своем исследовании предлагает решение проблемы удаленного мониторинга журналов и делегирования задач, описывая средства, которые устраняют зависимость от прямого доступа к серверу и специальных инструментов для удаленного подключения [2]. Исследование И.В. Тудвасева представляет собой комплексный процесс проектирования собственной системы резервного копирования MySQL, включающий анализ предметной области и существующих решений, выработку требований, выбор инструментов проектирования (UML) и разработки, а также детальное проектирование системы с использованием диаграмм прецедентов, последовательности, развертывания и пользовательского интерфейса [3]. В исследовании, основанном на работе О.И. Жалолова и Х.У. Хаятова, рассматриваются особенности реляционной модели баз данных и использование SQL для эффективной и быстрой обработки запросов, обеспечивающей оперативный доступ к данным [4]. Исследование Л.М. Останковой посвящено анализу систем управления базами данных, в рамках которого рассматриваются их основные понятия, содержание, а также актуальность и эффективность применения в мировом масштабе с оценкой достоинств и недостатков реляционной модели [5].

## **1.3 Цель исследования**

Цель исследования – провести сравнительный анализ функциональных возможностей и эффективности применения специализированной среды разработки dbForge Studio for MySQL по сравнению со стандартными инструментами в процессе проектирования, реализации и администрирования реляционной базы данных для интернет-магазина. Исследование направлено на оценку влияния выбранного инструментария на скорость разработки, обеспечение целостности данных, реализацию бизнес-логики и общую надежность итогового решения.

## **2. Материалы и методы**

В качестве материала для исследования использовалась база данных интернет-магазина, разработанная с использованием среды dbForge Studio for MySQL. Основной акцент делался на структурирование данных и обеспечение их целостности и согласованности. Для проектирования базы данных применялись методы анализа требований к данным, моделирования бизнес-процессов и построения реляционной схемы с использованием нормализации, что позволяло минимизировать избыточность и дублирование данных. В процессе исследования использовались функциональные возможности dbForge Studio for MySQL, включая создание таблиц и связей между ними,

настройку ограничений целостности, разработку индексов для оптимизации запросов, формирование представлений, триггеров и хранимых процедур.

Методы тестирования и валидации данных включали проверку корректности и полноты записей, согласованности связей между таблицами, а также моделирование типовых операций интернет-магазина. Кроме того, проводился сравнительный анализ различных подходов к проектированию реляционных баз данных для выявления оптимальных решений с точки зрения управления данными, производительности системы и удобства администрирования. Применяемая методология позволяла не только создать функционально полноценную базу данных, но и продемонстрировать практические принципы управления данными на всех этапах жизненного цикла информационной системы интернет-магазина

### 3. Результаты

В результате исследования разработана реляционная база данных интернет-магазина с использованием среды dbForge Studio for MySQL. База данных включает основные сущности, необходимые для функционирования интернет-магазина:

Таблица «Customers» - хранение информации о клиентах, включая ФИО, контактные данные и адрес. Добавлено 20 клиентов для тестирования (рис. 1).

The screenshot displays the 'Data' tab in dbForge Studio for MySQL. The top section shows a table with 11 records. The bottom section shows the SQL script for creating and altering the 'Customers' table.

| CustomerID | FirstName | LastName  | Email              | Phone        | Address                              | CreatedAt          |
|------------|-----------|-----------|--------------------|--------------|--------------------------------------|--------------------|
| 1          | Иван      | Иванов    | ivanov@mail.com    | +79161234501 | Москва, ул. Ленина, 1                | 18.01.2026 7:30:19 |
| 2          | Мария     | Петрова   | petrova@mail.com   | +79161234502 | Санкт-Петербург, пр. Невский, 2      | 18.01.2026 7:30:19 |
| 3          | Алексей   | Сидоров   | sidorov@mail.com   | +79161234503 | Казань, ул. Баумана, 3               | 18.01.2026 7:30:19 |
| 4          | Елена     | Кузнецова | kuznecova@mail.com | +79161234504 | Новосибирск, ул. Красный проспект, 4 | 18.01.2026 7:30:19 |
| 5          | Дмитрий   | Морозов   | morozov@mail.com   | +79161234505 | Екатеринбург, ул. Ленина, 5          | 18.01.2026 7:30:19 |
| 6          | Ольга     | Васильева | vaslieva@mail.com  | +79161234506 | Самара, ул. Гагарина, 6              | 18.01.2026 7:30:19 |
| 7          | Сергей    | Николаев  | nikolaev@mail.com  | +79161234507 | Воронеж, ул. Пушкина, 7              | 18.01.2026 7:30:19 |
| 8          | Анастасия | Соколова  | sokolova@mail.com  | +79161234508 | Ростов-на-Дону, ул. Чехова, 8        | 18.01.2026 7:30:19 |
| 9          | Михаил    | Попов     | popov@mail.com     | +79161234509 | Краснодар, ул. Карла Маркса, 9       | 18.01.2026 7:30:19 |
| 10         | Татьяна   | Лебедева  | lebedeva@mail.com  | +79161234510 | Тюмень, ул. Республики, 10           | 18.01.2026 7:30:19 |
| 11         | Никита    | Григорьев | griqorev@mail.com  | +79161234511 | Ижевск, ул. Пушкина, 11              | 18.01.2026 7:30:19 |

```

CREATE TABLE InternetShop.Customers (
  CustomerID INT NOT NULL AUTO_INCREMENT,
  FirstName VARCHAR(50) NOT NULL,
  LastName VARCHAR(50) NOT NULL,
  Email VARCHAR(100) NOT NULL,
  Phone VARCHAR(20) DEFAULT NULL,
  Address VARCHAR(255) DEFAULT NULL,
  CreatedAt DATETIME DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (CustomerID)
)
ENGINE = INNODB,
AUTO_INCREMENT = 21,
AVG_ROW_LENGTH = 1365,
CHARACTER SET utf8mb4,
COLLATE utf8mb4_0900_ai_ci;

ALTER TABLE InternetShop.Customers

```

Рисунок 1 – Таблица Customers

Таблица Categories - категории товаров, что позволяет структурировать ассортимент. Добавлено 5 категорий (рис. 2).

The screenshot shows a database management interface with the 'Data' tab selected. The 'Categories' table is displayed with 5 records. Below the table, the SQL code for creating the table is shown.

| CategoryID | CategoryName | Description                         |
|------------|--------------|-------------------------------------|
| 1          | Электроника  | Телефоны, ноутбуки, гаджеты         |
| 2          | Одежда       | Футболки, брюки, платья             |
| 3          | Книги        | Художественная литература, учебники |
| 4          | Дом и кухня  | Посуда, мебель, бытовая техника     |
| 5          | Спорт        | Спортивный инвентарь и одежда       |

```

CREATE TABLE InternetShop.Categories (
  CategoryID INT NOT NULL AUTO INCREMENT,
  CategoryName VARCHAR(100) NOT NULL,
  Description TEXT DEFAULT NULL,
  PRIMARY KEY (CategoryID)
)
ENGINE = INNODB,
AUTO INCREMENT = 6,
AVG ROW LENGTH = 3276,
CHARACTER SET utf8mb4,
COLLATE utf8mb4_0900_ai_ci;

```

Рисунок 2 – Таблица Categories

Таблица Products - хранение информации о товарах, включая наименование, описание, цену, количество на складе и принадлежность к категории. Добавлено 20 товаров. (рис. 3).

The screenshot shows a database management interface with the 'Data' tab selected. The 'Products' table is displayed with 11 records. Below the table, the SQL code for creating the table is shown.

| ProductID | ProductName                | Description             | Price    | StockQuantity | CategoryID | CreatedAt          |
|-----------|----------------------------|-------------------------|----------|---------------|------------|--------------------|
| 1         | Смартфон XYZ               | 128GB, 6GB RAM          | 29999,99 | 50            | 1          | 18.01.2026 7:30:19 |
| 2         | Ноутбук ABC                | i7, 16GB RAM            | 75999,99 | 20            | 1          | 18.01.2026 7:30:19 |
| 3         | Футболка "Cool"            | Хлопок, размер M        | 999,99   | 100           | 2          | 18.01.2026 7:30:19 |
| 4         | Джинсы "Classic"           | Синие, размер 32        | 1999,99  | 50            | 2          | 18.01.2026 7:30:19 |
| 5         | Книга "Мастер и Маргарита" | Булгаков                | 499,99   | 200           | 3          | 18.01.2026 7:30:19 |
| 6         | Книга "Война и мир"        | Толстой                 | 599,99   | 150           | 3          | 18.01.2026 7:30:19 |
| 7         | Миксер бытовой             | 500W, 2 скорости        | 3499,99  | 30            | 4          | 18.01.2026 7:30:19 |
| 8         | Стул офисный               | Эргономичный, черный    | 4599,99  | 40            | 4          | 18.01.2026 7:30:19 |
| 9         | Беговая дорожка            | 150 кг макс. вес        | 45999,99 | 10            | 5          | 18.01.2026 7:30:19 |
| 10        | Гантели 10 кг              | Пара                    | 2499,99  | 25            | 5          | 18.01.2026 7:30:19 |
| 11        | Смарт-часы A1              | Функции фитнес-трекинга | 7999,99  | 60            | 1          | 18.01.2026 7:30:19 |

```

CREATE TABLE InternetShop.Products (
  ProductID INT NOT NULL AUTO INCREMENT,
  ProductName VARCHAR(100) NOT NULL,
  Description TEXT DEFAULT NULL,
  Price DECIMAL(10, 2) NOT NULL,
  StockQuantity INT DEFAULT 0,
  CategoryID INT DEFAULT NULL,
  CreatedAt DATETIME DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (ProductID)
)
ENGINE = INNODB,
AUTO INCREMENT = 21,
AVG ROW LENGTH = 819,
CHARACTER SET utf8mb4,
COLLATE utf8mb4_0900_ai_ci;

```

Рисунок 3 – Таблица Products

Таблица Orders - хранение информации о заказах клиентов, с указанием статуса и суммы заказа. Создано 20 заказов. (рис. 4).

| OrderID | CustomerID | OrderDate          | Status  | TotalAmount |
|---------|------------|--------------------|---------|-------------|
| 1       | 1          | 18.01.2026 7:30:19 | Pending | 30999,98    |
| 2       | 2          | 18.01.2026 7:30:19 | Pending | 999,99      |
| 3       | 3          | 18.01.2026 7:30:19 | Pending | 75999,99    |
| 4       | 4          | 18.01.2026 7:30:19 | Pending | 1999,99     |
| 5       | 5          | 18.01.2026 7:30:19 | Pending | 499,99      |
| 6       | 6          | 18.01.2026 7:30:19 | Pending | 599,99      |
| 7       | 7          | 18.01.2026 7:30:19 | Pending | 3499,99     |
| 8       | 8          | 18.01.2026 7:30:19 | Pending | 4599,99     |
| 9       | 9          | 18.01.2026 7:30:19 | Pending | 45999,99    |
| 10      | 10         | 18.01.2026 7:30:19 | Pending | 2499,99     |
| 11      | 11         | 18.01.2026 7:30:19 | Pending | 7999,99     |

```
CREATE TABLE InternetShop Orders (  
  OrderID INT NOT NULL AUTO_INCREMENT,  
  CustomerID INT NOT NULL,  
  OrderDate DATETIME DEFAULT CURRENT_TIMESTAMP,  
  Status VARCHAR(50) DEFAULT 'Pending',  
  TotalAmount DECIMAL(10, 2) NOT NULL,  
  PRIMARY KEY (OrderID)  
)  
ENGINE = INNODB,  
AUTO_INCREMENT = 21,  
AVG_ROW_LENGTH = 819,  
CHARACTER SET utf8mb4,  
COLLATE utf8mb4_0900_ai_ci;  
  
ALTER TABLE InternetShop Orders  
  ADD INDEX idx_orders_customer(CustomerID);
```

Рисунок 4 – Таблица Orders

Таблица OrderDetails - состав каждого заказа, включая товары и их количество. Добавлено 20 записей. (рис. 5).

| OrderDetailID | OrderID | ProductID | Quantity | Price    |
|---------------|---------|-----------|----------|----------|
| 1             | 1       | 1         | 1        | 29999,99 |
| 2             | 1       | 3         | 1        | 999,99   |
| 3             | 2       | 3         | 1        | 999,99   |
| 4             | 3       | 2         | 1        | 75999,99 |
| 5             | 4       | 4         | 1        | 1999,99  |
| 6             | 5       | 5         | 1        | 499,99   |
| 7             | 6       | 6         | 1        | 599,99   |
| 8             | 7       | 7         | 1        | 3499,99  |
| 9             | 8       | 8         | 1        | 4599,99  |
| 10            | 9       | 9         | 1        | 45999,99 |
| 11            | 10      | 10        | 1        | 2499,99  |

```
CREATE TABLE InternetShop OrderDetails (  
  OrderDetailID INT NOT NULL AUTO_INCREMENT,  
  OrderID INT NOT NULL,  
  ProductID INT NOT NULL,  
  Quantity INT NOT NULL,  
  Price DECIMAL(10, 2) NOT NULL,  
  PRIMARY KEY (OrderDetailID)  
)  
ENGINE = INNODB,  
AUTO_INCREMENT = 21,  
CHARACTER SET utf8mb4,  
COLLATE utf8mb4_0900_ai_ci;  
  
ALTER TABLE InternetShop OrderDetails  
  ADD INDEX idx_orderdetails_order(OrderID);  
  
ALTER TABLE InternetShop OrderDetails  
  ADD CONSTRAINT fk_orderdetails_order FOREIGN KEY (OrderID) REFERENCES InternetShop Orders (OrderID);
```

Рисунок 5 – Таблица OrderDetails

Таблица Payments - информация о платежах за заказы, с указанием способа оплаты и статуса платежа. Добавлено 20 записей. (рис. 6).

| PaymentID<br>INT | OrderID<br>INT | PaymentDate<br>DATETIME | Amount<br>DECIMAL(10, 2) | PaymentMethod<br>VARCHAR(50) | Status<br>VARCHAR(50) |
|------------------|----------------|-------------------------|--------------------------|------------------------------|-----------------------|
| 1                | 1              | 18.01.2026 7:30:19      | 30999,98                 | Банковская карта             | Completed             |
| 2                | 2              | 18.01.2026 7:30:19      | 999,99                   | Онлайн кошелек               | Completed             |
| 3                | 3              | 18.01.2026 7:30:19      | 75999,99                 | Банковская карта             | Completed             |
| 4                | 4              | 18.01.2026 7:30:19      | 1999,99                  | Наличные                     | Completed             |
| 5                | 5              | 18.01.2026 7:30:19      | 499,99                   | Онлайн кошелек               | Completed             |
| 6                | 6              | 18.01.2026 7:30:19      | 599,99                   | Банковская карта             | Completed             |
| 7                | 7              | 18.01.2026 7:30:19      | 3499,99                  | Наличные                     | Completed             |
| 8                | 8              | 18.01.2026 7:30:19      | 4599,99                  | Банковская карта             | Completed             |
| 9                | 9              | 18.01.2026 7:30:19      | 45999,99                 | Онлайн кошелек               | Completed             |
| 10               | 10             | 18.01.2026 7:30:19      | 2499,99                  | Наличные                     | Completed             |
| 11               | 11             | 18.01.2026 7:30:19      | 7999,99                  | Банковская карта             | Completed             |

```

CREATE TABLE InternetShop Payments (
  PaymentID INT NOT NULL AUTO_INCREMENT,
  OrderID INT NOT NULL,
  PaymentDate DATETIME DEFAULT CURRENT_TIMESTAMP,
  Amount DECIMAL(10, 2) NOT NULL,
  PaymentMethod VARCHAR(50) DEFAULT NULL,
  Status VARCHAR(50) DEFAULT 'Pending',
  PRIMARY KEY (PaymentID)
)
ENGINE = INNODB,
AUTO_INCREMENT = 21,
CHARACTER SET utf8mb4,
COLLATE utf8mb4_0900_ai_ci;

ALTER TABLE InternetShop Payments
  ADD INDEX idx_payments_order(OrderID);

```

Рисунок 6 – Таблица Payments

Для управления данными реализованы следующие механизмы:

Триггер ReduceStock - автоматическое уменьшение количества товара на складе при добавлении нового заказа, что обеспечивает актуальность данных о запасах (рис. 7).

Name: ReduceStock Database: InternetShop

Table: OrderDetails Definer: root@

Timing: ☒ After ☐ Before

Event: ☒ Insert ☐ Delete ☐ Update

Order: ☐ Follows trigger ☐ Precedes trigger

```

CREATE
  DEFINER = 'root'@'%'
  TRIGGER InternetShop ReduceStock
  AFTER INSERT
  ON InternetShop OrderDetails
  FOR EACH ROW
  BEGIN
    UPDATE Products
    SET StockQuantity = StockQuantity - NEW Quantity
    WHERE ProductID = NEW ProductID;
  END

```

Рисунок 7 - Триггер ReduceStock

Представление OrderSummary - позволяет просматривать заказы вместе с информацией о клиентах и общей суммой, упрощая аналитику и контроль (рис. 8).

| OrderID | FirstName | LastName  | OrderDate          | Status  | TotalAmount |
|---------|-----------|-----------|--------------------|---------|-------------|
| 1       | Иван      | Иванов    | 18.01.2026 7:30:19 | Pending | 30999,98    |
| 2       | Мария     | Петрова   | 18.01.2026 7:30:19 | Pending | 999,99      |
| 3       | Алексей   | Сидоров   | 18.01.2026 7:30:19 | Pending | 75999,99    |
| 4       | Елена     | Кузнецова | 18.01.2026 7:30:19 | Pending | 1999,99     |
| 5       | Дмитрий   | Морозов   | 18.01.2026 7:30:19 | Pending | 499,99      |
| 6       | Ольга     | Васильева | 18.01.2026 7:30:19 | Pending | 599,99      |
| 7       | Сергей    | Николаев  | 18.01.2026 7:30:19 | Pending | 3499,99     |
| 8       | Анастасия | Соколова  | 18.01.2026 7:30:19 | Pending | 4599,99     |
| 9       | Михаил    | Попов     | 18.01.2026 7:30:19 | Pending | 45999,99    |
| 10      | Татьяна   | Лебедева  | 18.01.2026 7:30:19 | Pending | 2499,99     |
| 11      | Никита    | Григорьев | 18.01.2026 7:30:19 | Pending | 7999,99     |
| 12      | Виктория  | Федорова  | 18.01.2026 7:30:19 | Pending | 12999,99    |
| 13      | Роман     | Егоров    | 18.01.2026 7:30:19 | Pending | 8999,99     |
| 14      | Ксения    | Михайлова | 18.01.2026 7:30:19 | Pending | 4999,99     |
| 15      | Игорь     | Никитин   | 18.01.2026 7:30:19 | Pending | 399,99      |
| 16      | Юлия      | Козлова   | 18.01.2026 7:30:19 | Pending | 1999,99     |
| 17      | Владимир  | Волков    | 18.01.2026 7:30:19 | Pending | 6999,99     |
| 18      | Алина     | Семенова  | 18.01.2026 7:30:19 | Pending | 8999,99     |
| 19      | Павел     | Тимофеев  | 18.01.2026 7:30:19 | Pending | 10999,99    |
| 20      | Светлана  | Морозова  | 18.01.2026 7:30:19 | Pending | 3999,99     |

Рисунок 8 - Представление OrderSummary

В рамках разработки базы данных настроены все необходимые связи между таблицами (рис. 9) для обеспечения целостности и корректного взаимодействия данных. Использованы внешние ключи (FOREIGN KEY).

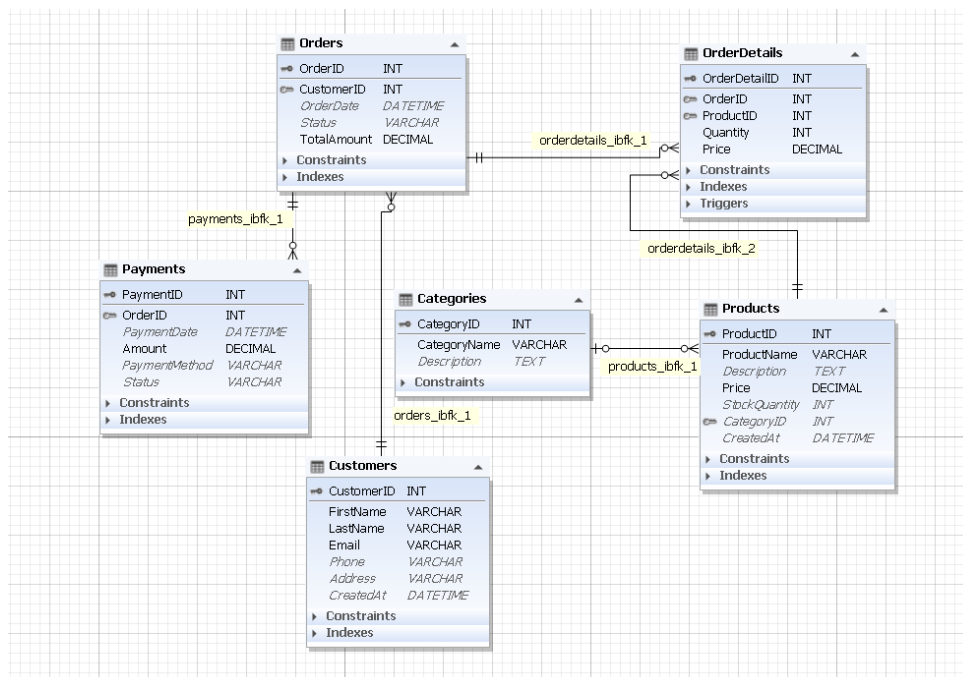


Рисунок 9 – Схема связи между таблицами

#### 4. Вывод

В ходе исследования разработана реляционная база данных интернет-магазина с использованием среды dbForge Studio for MySQL, включающая таблицы для хранения информации о клиентах, товарах, категориях, заказах, составе заказов и платежах. Все таблицы заполнены тестовыми данными, что

позволило проверить корректность работы базы и эффективность предложенной методологии управления данными.

Особое внимание уделено построению связей между таблицами через внешние ключи, что обеспечило целостность данных, исключило появление несогласованных записей и позволило корректно выполнять бизнес-операции, такие как оформление заказов и проведение платежей. Настройка этих связей также обеспечила возможность создания удобных аналитических представлений и отчетов для контроля деятельности интернет-магазина.

Реализована автоматизация некоторых процессов управления данными с помощью триггера ReduceStock, который автоматически уменьшает количество товара на складе при оформлении заказа. Дополнительно созданы представления, обеспечивающие удобство анализа данных и повышение производительности запросов.

Таким образом, разработанная база данных демонстрирует практическую применимость и эффективность методологии управления данными, позволяя вести учет клиентов, товаров, заказов и платежей в условиях интернет-магазина. Работа подтверждает, что правильно спроектированная структура базы данных с корректными связями и средствами автоматизации обеспечивает надежность, согласованность и управляемость информации, что является ключевым фактором успешного функционирования электронной коммерции.

### Библиографический список

1. Анисимова Э. С. Интеллектуальный инструмент dbforge Studio for MySQL в работе с базами данных // Экономика и социум. 2015. № 6-1(19). С. 770-773 URL: <https://elibrary.ru/item.asp?id=25623294>
2. Крутилин С. Настраиваем хранение логов в базе данных MySQL // Системный администратор. 2009. № 5(78). С. 26-28 URL: <https://elibrary.ru/item.asp?id=20406939>
3. Тудвасев И. В. Проектирование и документирование системы для создания, хранения и восстановления резервных копий баз данных MySQL // Актуальные проблемы математики, механики и информатики: сборник статей по материалам студенческой конференции, Пермь, 25 мая – 10 2021 года. Пермь: 2021. С. 172-176 URL: <http://elibrary.ru/item.asp?id=46446513>
4. Жалолов О. И., Хаятов Х. У. Понятие SQL и реляционной базы данных // Universum: технические науки. 2020. С. 26-29. URL: <https://cyberleninka.ru/article/n/ponyatie-sql-i-relyatsionnoy-bazy-dannyh>
5. Останкова Л. М. База данных и системы управления базами данных // Экономика и социум. 2017. № 5-2(36). С. 600-603 URL: <https://elibrary.ru/item.asp?id=29907876>