|  |  |
| --- | --- |
| Федеральное государственное образовательное бюджетное учреждение высшего образования  «ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ  РОССИЙСКОЙ ФЕДЕРАЦИИ»  (Финансовый университет)  **Колледж информатики и программирования** | |
| ПМ.11 Разработка, администрирование и защита баз данных  Группа: 3ИСИП—522 | УТВЕРЖДАЮ  Председатель предметно—цикловой комиссии информационных систем и программирования  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Т.Г. Аксёнова/  « » \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 г. |
| **КУРСОВОЙ ПРОЕКТ**  **На тему: Проектирование и разработка базы данных для ритейла** | |
|  | Руководитель курсового проекта  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Н.А. Хасанова  Исполнитель курсового проекта  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Т.Д. Молчанов  Оценка за проект: \_\_\_\_\_\_\_\_\_\_\_\_ «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 г. |
| Москва 2025 | |

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc197606651)

[Глава 1. Анализ требований для базы данных для ритейла 5](#_Toc197606652)

[1.1. Исследование предметной области 5](#_Toc197606653)

[1.2. Постановка задачи 7](#_Toc197606654)

[1.3. Характеристика инструментальных средств разработки 9](#_Toc197606655)

[Глава 2. Построение и анализ баз данных для ритейла 11](#_Toc197606656)

[2.1. Проектирование базы данных 11](#_Toc197606657)

[2.2. Разработка базы данных и интерфейса 13](#_Toc197606658)

[2.3. Отладка и тестирование 17](#_Toc197606659)

[2.4. Руководство администратора базы данных 20](#_Toc197606660)

[ЗАКЛЮЧЕНИЕ 23](#_Toc197606661)

[СПИСОК ЛИТЕРАТУРЫ (ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ) И ИНТЕРНЕТ-РЕСУРСОВ 26](#_Toc197606662)

[ПРИЛОЖЕНИЕ 27](#_Toc197606663)

# ВВЕДЕНИЕ

В современном мире ритейл—индустрия представляет собой одну из наиболее динамично развивающихся отраслей, что обусловлено растущим интересом потребителей к удобству покупок и широкому ассортименту товаров. Магазины, как физические, так и онлайн, играют ключевую роль в обеспечении людей необходимыми товарами, а также в организации удобного покупательского процесса. Однако эффективное функционирование таких организаций требует грамотного управления их внутренними процессами, включая работу с клиентами, товарами, заказами, скидками и финансовыми потоками.

Целью данной курсовой работы является проектирование и разработка базы данных для ритейл, которая обеспечит эффективное управление ключевыми процессами, включая учет товаров, клиентов, заказов, скидок и автоматизацию внутреннего документооборота.

Для достижения поставленной цели решаются следующие задачи:  
— анализ существующих подходов к хранению и обработке данных в ритейл—системах;  
— определение функциональных требований к проектируемой базе данных;  
— разработка структуры базы данных с учетом требований к надежности, целостности и удобству работы с данными;  
— реализация базы данных и разработка пользовательского интерфейса для работы с ней с использованием технологий WPF и языка программирования C  
— тестирование системы на предмет корректности функционирования и соответствия требованиям.

Объектом исследования является процесс управления и автоматизации бизнес—процессов в ритейл, включая учет товаров, клиентов, заказов, скидок и транзакций.

Актуальность темы обусловлена необходимостью цифровизации процессов в ритейл—индустрии, что позволяет повысить уровень обслуживания клиентов и обеспечить эффективное управление товарными запасами и продажами. Разработка базы данных для ритейл является важным инструментом автоматизации, обеспечивающим высокую степень оперативности и точности работы, а также способствующим росту бизнеса и укреплению его позиций на рынке.

Основное функциональное назначение разрабатываемого программного средства заключается в централизованном хранении данных о товарах, клиентах, заказах и транзакциях, обеспечении их доступности для редактирования и просмотра, а также предоставлении удобного интерфейса для поиска и анализа информации.

Для реализации проекта были выбраны следующие инструментальные средства:  
— язык структурированных запросов SQL для проектирования и работы с базой данных;  
— фреймворк WPF для разработки пользовательского интерфейса;  
— язык программирования C для реализации логики взаимодействия с базой данных и управления пользовательским интерфейсом.

В процессе работы будет рассмотрен теоретический аспект проектирования баз данных, включая методы анализа требований и моделирования данных. На основании полученных данных будет создана структура базы данных, реализованная с использованием современных технологий управления данными. Результаты работы могут быть полезны для реализации подобных проектов в реальных условиях, что подтверждает практическую значимость исследования.

# Глава 1. Анализ требований для базы данных для ритейла

# 1.1. Исследование предметной области

Разрабатываемая база данных предназначена для автоматизации процессов учета и управления информацией в ритейле.

Основные объекты предметной области:

— Клиенты: информация о каждом клиенте включает в себя идентификатор клиента (CustomerID), имя (FirstName), фамилию (LastName), электронную почту (Email), номер телефона (Phone) и дату регистрации (CreatedAt). Клиенты совершают заказы, участвуют в программах скидок и могут получать уведомления.

— Категории товаров: содержит идентификатор категории (CategoryID), название категории (CategoryName) и описание (Description). Категории помогают структурировать ассортимент товаров.

— Товары: каждый товар включает идентификатор товара (ProductID), название (ProductName), цену (Price), количество на складе (StockQuantity) и идентификатор категории (CategoryID), к которой он принадлежит.

— Заказы: каждый заказ имеет идентификатор заказа (OrderID), идентификатор клиента (CustomerID), дату заказа (OrderDate) и общую сумму заказа (TotalAmount). В заказе могут присутствовать несколько товаров.

— Детали заказа: содержит информацию о каждом товаре в заказе, включая идентификатор детали заказа (OrderDetailID), идентификатор заказа (OrderID), идентификатор товара (ProductID), количество (Quantity), цену за единицу (Price) и итоговую сумму по позиции (LineTotal).

— Сотрудники: информация о персонале магазина, включая идентификатор сотрудника (EmployeeID), имя (FirstName), фамилию (LastName), должность (Position), дату найма (HireDate) и зарплату (Salary). Сотрудники обрабатывают заказы, управляют поставками и обслуживают клиентов.

— Поставщики: включает идентификатор поставщика (SupplierID), название поставщика (SupplierName), контактное лицо (ContactName), номер телефона (Phone) и адрес (Address). Поставщики поставляют товары в магазин.

— Поставки: каждая поставка содержит идентификатор поставки (ShipmentID), идентификатор поставщика (SupplierID), дату поставки (ShipmentDate) и общую стоимость поставки (TotalCost).

— Детали поставки: информация о каждом товаре в поставке, включая идентификатор детали поставки (ShipmentDetailID), идентификатор поставки (ShipmentID), идентификатор товара (ProductID), количество (Quantity) и стоимость за единицу (CostPerUnit).

— Скидки: включает идентификатор скидки (DiscountID), идентификатор товара (ProductID), процент скидки (DiscountPercentage), дату начала (StartDate) и дату окончания скидки (EndDate). Позволяет управлять акциями и предложениями для клиентов.

Эта база данных предназначена для эффективного управления всеми процессами в ритейле, включая продажу товаров, учет клиентов, управление запасами и обработку заказов, обеспечивая надежное хранение данных и их безопасность.

Основные события и процессы:

— Регистрация клиента: добавление нового клиента в базу данных и сохранение его контактных данных.

— Создание заказа: клиент оформляет заказ, который фиксируется в системе, включая список товаров и их количество.

— Добавление товаров в заказ: связь между заказом и товарами через таблицу деталей заказа.

— Обновление остатков: при оформлении заказа или поставке обновляется количество товаров на складе.

— Прием поставок: регистрация поступления товаров от поставщиков, обновление складских остатков.

— Назначение скидок: управление акциями и снижением цен на определенные товары в заданный период.

— Управление персоналом: учет сотрудников, их ролей и заработной платы.

— Анализ продаж: сбор и анализ данных о заказах, продажах и эффективности скидок.

Дополнительные аспекты:

— Разграничение прав доступа: администраторы и сотрудники должны иметь разные уровни доступа к данным.

— Резервное копирование данных: регулярное создание резервных копий для предотвращения потери данных.

— Обеспечение безопасности данных: защита персональной информации клиентов и финансовых данных с использованием современных методов шифрования и аутентификации.

Эта база данных обеспечивает эффективное управление всеми процессами ритейл, помогая оптимизировать продажи, учет товаров и взаимодействие с клиентами.

# 1.2. Постановка задачи

Категории пользователей

— **Администратор**: обладает полным доступом ко всем функциям системы.

— **Сотрудник**: имеет ограниченный доступ к данным, управляет заказами и поставками.

Администратор

**Управление пользователями:**

— Добавление, изменение, удаление информации о сотрудниках.

— Управление правами доступа сотрудников.

**Управление контентом:**

— Добавление, редактирование и удаление товаров.

— Управление категориями товаров.

**Управление заказами:**

— Мониторинг заказов и их статусов.

— Формирование отчетов по продажам.

**Управление поставками:**

— Создание и редактирование информации о поставщиках.

— Отслеживание поставок и управление складскими запасами.

**Формирование отчетов:**

— Анализ продаж и товарооборота.

— Отчеты по остаткам на складе.

**Обеспечение безопасности:**

— Настройка системы уведомлений.

— Регулярное резервное копирование и восстановление данных.

Сотрудник

**Управление заказами:**

— Создание и обработка заказов.

— Контроль выполнения заказов и статусов доставки.

**Работа с клиентами:**

— Просмотр информации о клиентах.

— Оформление возвратов и управление скидками.

**Работа с поставками:**

— Прием и учет новых поставок.

— Обновление остатков товаров.

Информация о необходимых отчетах указаны в таблице 1 и таблице 2:

Таблица 1 — Отчет о продажах

|  |  |
| --- | --- |
| Колонка | Содержание |
| OrderID | Уникальный идентификатор заказа |
| CustomerID | Идентификатор клиента |
| OrderDate | Дата оформления заказа |
| TotalAmount | Общая сумма заказа |

Таблица 2 — Отчет о складских остатках

|  |  |
| --- | --- |
| Колонка | Содержание |
| ProductID | Уникальный идентификатор товара |
| ProductName | Название товара |
| StockQuantity | Количество на складе |
| CategoryID | Категория товара |

Нефункциональные требования

**— Требования к реализации:**

1) База данных должна работать на операционной системе Windows.

2) Приложение должно быть совместимо с компьютерами под управлением Windows, имеющими доступ к серверу баз данных.

**— Требования к надежности:**

1. Защита данных с помощью паролей и разграничения прав доступа.
2. Регулярное резервное копирование данных для предотвращения их потери.
3. Использование защищенных соединений для передачи данных между приложением и сервером базы данных.

# 1.3. Характеристика инструментальных средств разработки

Для разработки базы данных для ритейл использовались Microsoft SQL Server, так как это мощная и надежная система управления базами данных. MS SQL Server обладает широкими функциональными возможностями, включая поддержку транзакций, масштабируемость, а также возможности по защите данных, такие как шифрование и управление доступом.

Для создания графического пользовательского интерфейса (GUI) использовалась технология WPF (Windows Presentation Foundation), которая обеспечивает гибкость и привлекательный дизайн интерфейса.

Язык программирования C является основным языком разработки приложений для платформы Microsoft, включая работу с базами данных. C обладает широкими возможностями по созданию функциональных и эффективных приложений.

Для разработки проекта использовалась среда программирования Microsoft Visual Studio, которая предоставляет удобные инструменты для работы с C, WPF интеграцию с базой данных MS SQL Server.

Для создания документации и презентаций курсового проекта использовались программные продукты, такие как Microsoft Word и Microsoft PowerPoint, которые обеспечат удобное оформление и презентацию результатов работы.

# Глава 2. Построение и анализ баз данных для ритейла

# 2.1. Проектирование базы данных

1. Инфологическая модель базы данных ритейл должна отражать структуру информационных потоков и связей между различными элементами данных. На рис. 1 изображена инфологическая модель ритейл, на которой определили основные сущности (entities) и их атрибуты (attributes), и связи между этими сущностями.

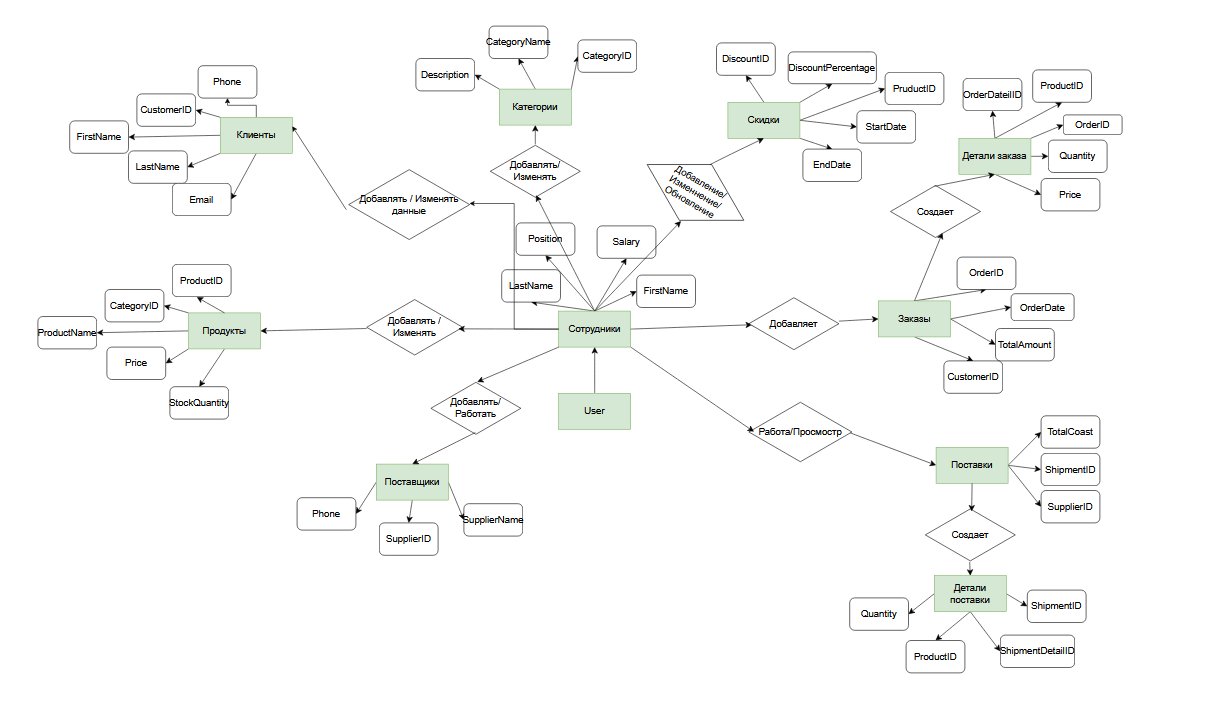


Рисунок 1 — Инфологическая модель

При построении даталогической модели для базы данных ритейла в контексте реляционной модели данных необходимо определить схемы отношений, первичные ключи и связи между отношениями в виде внешних ключей. На рис. 2 изображена даталогическая модель для базы данных ритейла

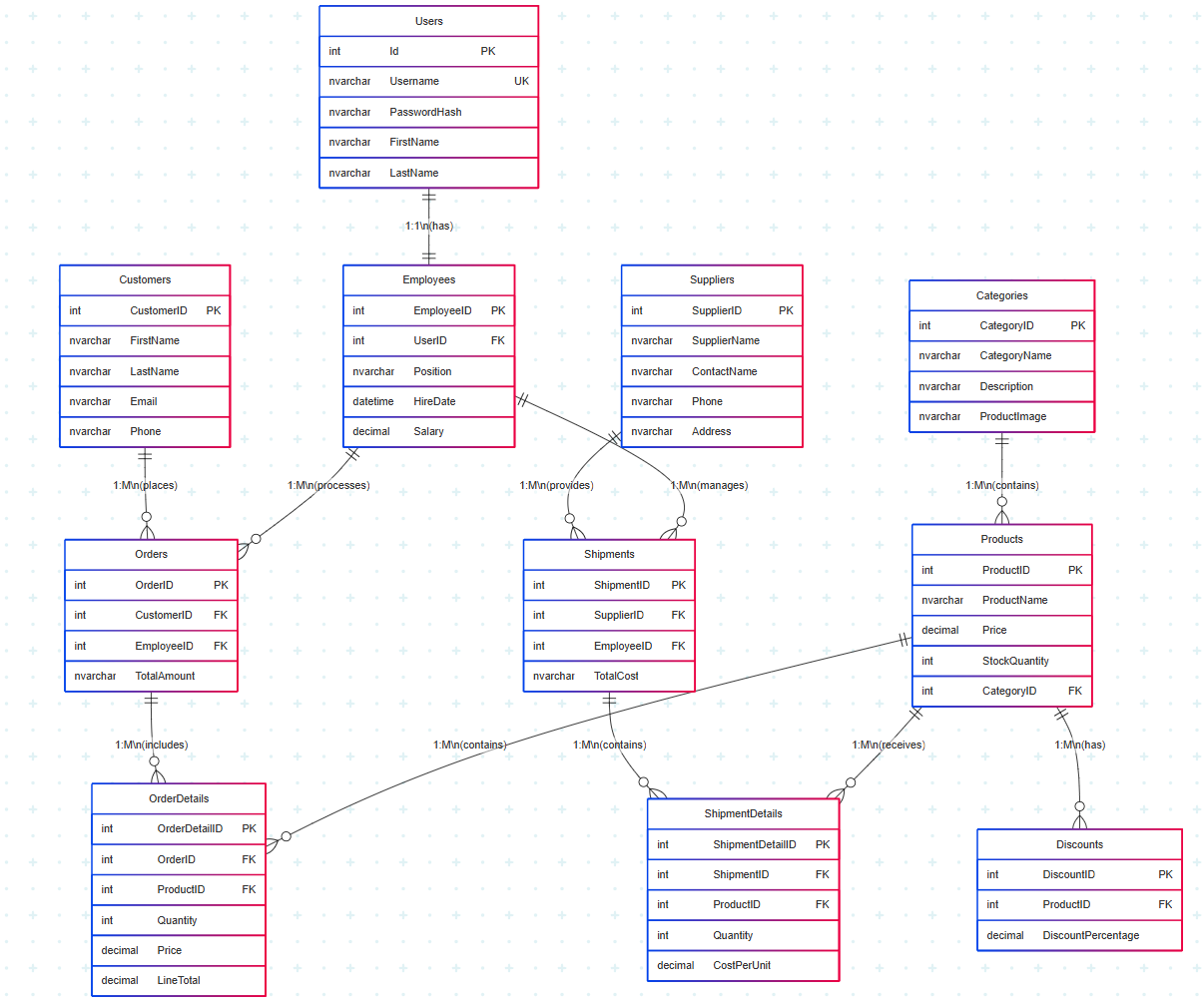


Рисунок 2 — Даталогическая модель

Таким образом, даталогическая модель базы данных ритейла включает в себя определение сущностей, их атрибутов, первичных ключей для каждой сущности, а также связи между этими сущностями через использование внешних ключей. Это обеспечивает структурированное хранение данных и позволяет эффективно управлять информацией.

2. Моделирование бизнес—процессов, связанных с функционированием ритейла

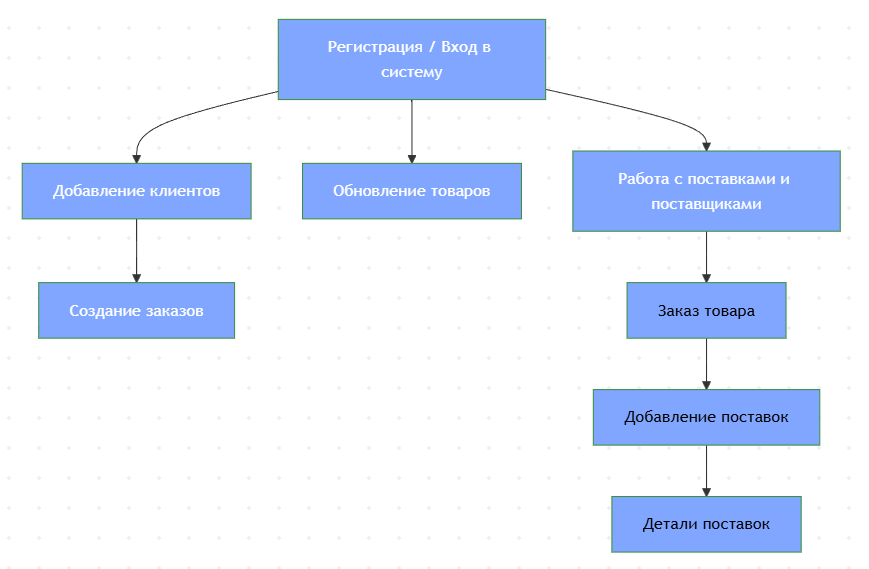


Рисунок 3 — Моделирование бизнес—процессов

Моделирование этих и других бизнес—процессов ритейла позволит визуализировать и проанализировать их эффективность, выявить потенциальные проблемы и области для оптимизации.

# 2.2. Разработка базы данных и интерфейса

В данном разделе разрабатывается база данных в MS SQL SERVER и

приложение WPF для взаимодействия с базой данных.

На рис. 4 пример создание таблицы в MS SQL SERVER и выведение

внешнего ключа на другую таблицу.

Листинг 1 — Создание таблиц

|  |
| --- |
| CREATE TABLE OrderDetails (  OrderDetailID INT PRIMARY KEY IDENTITY(1,1),  OrderID INT NOT NULL,  ProductID INT NOT NULL,  Quantity INT NOT NULL,  Price DECIMAL(10, 2) NOT NULL,  LineTotal AS (Quantity \* Price) PERSISTED,  FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),  FOREIGN KEY (ProductID) REFERENCES Products(ProductID)  ); |

Подобным образом мы создаем остальные таблицы (1—11) и выводим внешние ключи. Также в таблицах указаны типы данных и ограничения целостности, которые важны для целостности данных.  
Каждое ограничение, включая PRIMARY KEY, FOREIGN KEY, UNIQUE, CHECK и NOT NULL, играет ключевую роль в обеспечении надежности и непротиворечивости хранимой информации.  
Кроме того, связи между таблицами позволяют реализовать логическую структуру базы данных и избежать избыточности данных.

Типы данных, которые использовались в данных таблицах:

— INT: целые числа;

— VARCHAR(n): строки переменной длины до n символов;

— DATE: дата;

— BOOL: хранит значение в формате — истина/ложь

Ограничения, которые использовались в данных таблицах:

— PRIMARY KEY: первичный ключ, гарантирует уникальность каждой

строки;

— NOT NULL: столбец не может быть пустым

— NVARCHAR(200): Уникальное значение

— DECIMAL: Уникальное значение

— DATETIME: Дата и время

— DATE: Дата

— TIME: Время

В таблице 3 представлены поля, ключи и примечания таблиц в базе данных.

Таблица 3 – Таблицы БД

|  |  |  |  |
| --- | --- | --- | --- |
| **Типы товаров, категории** | | | |
| **Ключ** | **Поле** | **Обязательное** | **Примечание** |
| Первичный | ID | Да | Порядковый номер типа товара |
|  | Тип | Да | Название типа товара (например, одежда, электроника) |
| **Продукты** | | | |
| **Ключ** | **Поле** | **Обязательное** | **Примечание** |
| Первичный | ID | Да | Порядковый номер товара |
|  | Название | Да | Название или описание товара |
| Внешний | ID типа товара | Да | |  | | --- | |  |  |  | | --- | | Номер типа товара | |
|  | Описание | Да | Описание товара |
|  | Цена | Да | Цена товара |
|  | Количество в наличии | Да | Количество на складе |
|  | Дата поступления | Да | Поступление на склад |
| **Заказы** | | | |
| **Ключ** | **Поле** | **Обязательное** | **Примечание** |
| Первичный | ID | Да | Порядковый номер заказа |
| Внешний | ID клиента | Да | Номер клиента |
|  | Дата заказа | Да | Дата оформления |
|  | Сумма | Да | Сумма |
|  | Статус | Да | Статус заказа |
| **Клиенты** | | | |
| Первичный | ID | Да | Порядковый номер пользователя |
|  | Фамилия | Да | Фамилия пользователя |
|  | Имя | Да | Имя пользователя |
|  |  |  |  |
|  | Адрес | Да | Адрес |
|  | Дата регистрации | Да | Дата   |  | | --- | |  |  |  | | --- | |  | |
|  | E-mail | Да | Электронная почта пользователя |

Продолжение таблицы 3

|  |  |  |  |
| --- | --- | --- | --- |
| **Детали заказов** | | | |
| Первичный | ID | Да | Порядковый номер пользователя |
| Внешний | ID заказа | Да | Фамилия пользователя |
| Внешний | ID товара | Да | Номера товара |
|  | Количество | Да | Количество товара |
|  | Цена | Да | Цена за единицу   |  | | --- | |  |  |  | | --- | |  | |
|  | Сумма | Да | Сумма заказа |
| **Поставщики** | | | |
| Первичный | ID | Да | Порядковый номер пользователя |
|  | Название | Да | Название компании |
|  | Контактное лицо | Да | Контактное лицо поставщика |
|  | Телефон |  | Телефон поставщика |
|  | Адрес | Да | Адрес поставщика |
| **Поставки** | | | |
| Первичный | ID | Да | Порядковый номер пользователя |
| Внешний | ID поставщика | Да | Номер поставщика |
|  | Дата | Да | Дата |
|  | Стоимость |  | Общая стоимость |
| **Детали поставок** | | | |
| Первичный | ID | Да | Номер детали поставки |
| Внешний | ID поставки | Да | Номер поставки |
| Внешний | ID товара | Да | Номер товара |
|  | Количество |  | Количество товара в поставке |
| **Скидки** | | | |
| Первичный | ID | Да | Номер скидки |
| Внешний | ID товара | Да | Номер товара |
|  | Процент | Да | Процент от товара |
|  | Дата начала | Да | Дата начала скидки |
| **Платежи** | | | |
| Первичный | ID | Да | Номер платежа |
| Внешний | ID заказа | Да | Номер заказа |
|  | Сумма | Да | Сумма |
|  | Дата платежа | Да | Дата и время платежа |
| **Сотрудники** | | | |
| Первичный | ID | Да | ID сотрудника |
|  | Имя | Да | Имя |
|  | Фамилия | Да | Фамилия |
|  | Позиция | Да | Профессия |

Это все таблицы для рабочей БД ритейла

На рис. 4 представлен фрагмент кода, на котором показана строка

подключения базы данных к WPF.

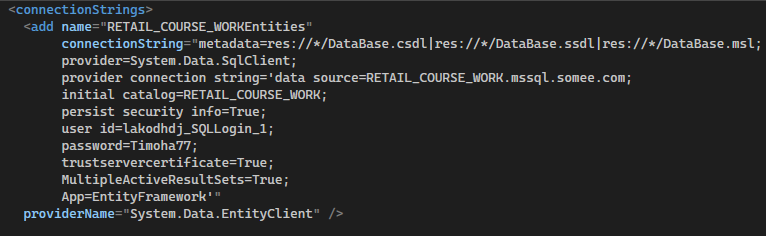


Рисунок 4 — Подключение

На листинге 2 представлено добавление таблиц в меню навигации

Листинг 2 – Добавление таблиц в меню навигации

|  |
| --- |
| using (SqlConnection conn = new SqlConnection(connectionString))  {  conn.Open();  DataTable schema = conn.GetSchema("Tables");  foreach (DataRow row in schema.Rows)  {  string tableName = row["TABLE\_NAME"].ToString();  if (allowedTables.Contains(tableName))  {  tableNames.Add(tableName);  }  }  } |

# 2.3. Отладка и тестирование

На рис. 5 мы вносим данные администратора (почта и пароль) и при

правильном вводе, система допускает нас до окна взаимодействия с базой

данных.

На рис. 6 мы не вносим данные администратора. Программа выдаёт ошибку.

При внесении неверных данных программа так же выдаёт ошибку. Это

показано на рис. 7.

В таблице 4 представлены результаты отладки и тестирования программы.

Таблица 4 — Результаты отладки и тестирования программы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № теста | Входные данные | Вводимое значение | Ожидаемая Реакция программы | Фактическая реакция программы | Ошибка выявлена |
| 1 | Логин/Пароль | Верные значения | Допуск к управлению базой данных | На рисунке 7 | нет |
| 2 | Логин/Пароль | — | Открытие окна «Ошибка» | На рисунке 8 | нет |
| 3 | Логин/Пароль | Выдуманные данные | Открытие окна «Ошибка» | На рисунке 9 | нет |

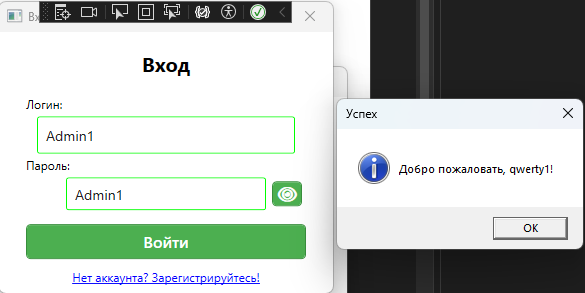


Рисунок 5 — Успешный вход в систему

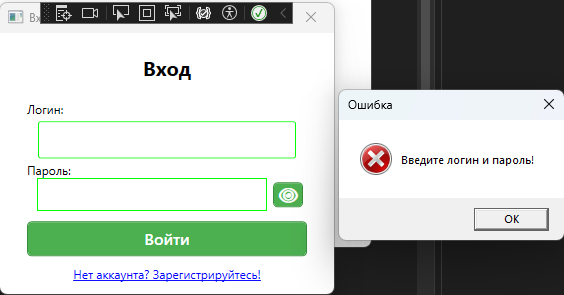


Рисунок 6 — Попытка входа без данных

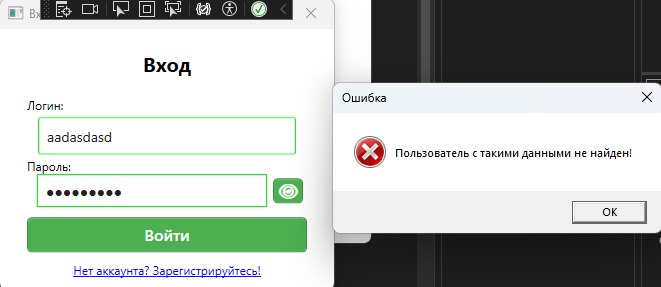


Рисунок 7 — Внесение неверных данных

Методы отладки, используемые для исправления ошибок:

— проверка ограничений:

1) ограничение NOT NULL: Убедитесь, что в обязательных полях вводятся данные;

2) ограничение FOREIGN KEY: Проверьте, что вводимое значение для

внешнего ключа существует в связанной таблице.

— использование функций валидации данных:

1) создаётся функция, которые будут проверять корректность вводимых

данных перед их сохранением в базу;

2) вызываются эти функции в триггерах или хранимых процедурах.

— использование инструментов отладки SSMS:

1) установите точки останова в коде для пошагового выполнения и аналиа

значений переменных;

2) используйте окно "Locals" для просмотра значений переменных;

3) используйте окно "Output" для просмотра сообщений об ошибках и

другой отладочной информации.

— анализ логов ошибок MS SQL Server:

1) просматриваются логи ошибок MS SQL Server для получения

информации об ошибках, возникающих при вводе данных.

— профилирование запросов:

1. используются SQL Server Profiler для анализа производительности запросов на вставку данных и выявления потенциальных проблем.

# 2.4. Руководство администратора базы данных

Установка MS SQL Server:

загрузите установочный файл MS SQL Server с официального сайта

Microsoft;

— запустите установочный файл и следуйте инструкциям мастера установки; Выберите тип установки:

— Basic: Установка основных компонентов SQL Server; Custom: Выбор компонентов для установки.

Выберите расположение для установки SQL Server.

Настройте параметры экземпляра SQL Server:

— Instance Name: Имя экземпляра SQL Server;

— Authentication Mode: Режим аутентификации (Windows Authentication или SQL Server Authentication).

Настройте параметры базы данных:

— Data Directories: Расположение файлов базы данных.

— TempDB Configuration: Параметры временной базы данных.

Завершите установку.

Инструкция по восстановлению БД:

— откройте SQL Server Management Studio (SSMS);

— подключитесь к экземпляру SQL Server;

— щелкните правой кнопкой мыши на Databases и выберите Restore Database; Выберите Source для восстановления:

— Device: Выберите файл резервной копии с устройства хранения;

Database: Выберите базу данных для восстановления из существующей резервной копии.

Укажите Destination для восстановленной базы данных; Выберите параметры восстановления:

— Overwrite the existing database (WITH REPLACE): Перезаписать существующую базу данных;

— Restore with recovery: Восстановить базу данных в режиме восстановления;

— Restore with norecovery: Восстановить базу данных без восстановления.

Нажмите OK для начала процесса восстановления.

Установка соответствующих ролей, привилегий и запретов;

— Откройте SSMS и подключитесь к экземпляру SQL Server;

— Разверните Security> Logins;

— Щелкните правой кнопкой мыши на учетной записи пользователя и выберите Properties;

— Перейдите на вкладку Server Roles;

Выберите роли сервера, которые нужно назначить пользователю:

— sysadmin: Полный доступ к серверу;

— securityadmin: Управление учетными записями пользователей и ролями; dbcreator: Создание баз данных.

Перейдите на вкладку User Mapping.

Выберите базы данных, к которым нужно предоставить доступ пользователю.

Выберите роли базы данных, которые нужно назначить пользователю:

— db\_owner: Полный доступ к базе данных;

— db\_datareader: Доступ к чтению данных из базы данных; db\_datawriter: Доступ к записи данных в базу данных.

Нажмите OK для сохранения изменений.

Важные рекомендации:

Регулярно создавать резервные копии базы данных;

— Хранить резервные копии в безопасном месте;

— Тщательно управлять ролями, привилегиями и запретами для обеспечения безопасности базы данных.

# ЗАКЛЮЧЕНИЕ

В ходе выполнения курсового проекта на тему "Проектирование и разработка базы данных для ритейла" были успешно достигнуты все поставленные цели. Разработана и реализована база данных на платформе MS SQL Server, обеспечивающая хранение и обработку информации, необходимой для управления розничной торговлей. Созданное WPF—приложение предоставляет удобный интерфейс для взаимодействия пользователей с базой данных, позволяя эффективно выполнять основные операции с данными.

Выполненные задачи:

1. Проектирование структуры БД: разработана схема данных, включающая таблицы для хранения информации о товарах, поставщиках, заказах, клиентах, скидках и отгрузках. Учтены ключевые аспекты ритейл—логистики и управления продажами.

2. Реализация БД в MS SQL Server: с использованием языка SQL созданы таблицы, установлены связи между ними, добавлены индексы для оптимизации запросов, а также реализованы хранимые процедуры и триггеры для автоматизации бизнес—процессов.

3. Разработка WPF—приложения: создан интуитивно понятный графический интерфейс, позволяющий пользователям:

— Добавлять, просматривать, изменять и удалять данные.

— Формировать отчеты по продажам, остаткам товаров и другим ключевым метрикам.

— Управлять заказами и поставками.

Достоинства разработанного решения:

— Функциональность: приложение охватывает все основные процессы ритейла, включая управление товарными запасами, работу с поставщиками и клиентами.

— Удобство использования: графический интерфейс WPF обеспечивает простоту взаимодействия, что делает систему доступной для пользователей без глубоких технических знаний.

— Надежность: использование MS SQL Server гарантирует безопасное хранение данных и высокую производительность при обработке запросов.

Недостатки и ограничения:

1. Локальность решения: приложение работает только на компьютерах с установленной ОС Windows, что ограничивает возможности удаленного доступа

2. Масштабируемость: при значительном увеличении объема данных или числа пользователей может потребоваться переход на более мощные серверные решения или облачные технологии.

Перспективы развития:

— Разработка веб—версии: создание веб—интерфейса позволит предоставить доступ к системе с любого устройства через браузер.

— Интеграция с облачными сервисами: использование облачных платформ (например, Azure SQL Database) улучшит масштабируемость и отказоустойчивость.

— Дополнительные модули: внедрение аналитики продаж, CRM—функционала и интеграция с системами электронных платежей расширит возможности системы.

Курсовой проект позволил закрепить теоретические знания по проектированию баз данных и приобрести практический опыт разработки WPF—приложений. Созданная система может быть использована в реальных условиях малого и среднего ритейла, а также послужить основой для дальнейшего расширения функционала. Работа над проектом также способствовала освоению новых технологий, включая MS SQL Server, WPF, C, и методам оптимизации запросов.

Разработанное решение демонстрирует эффективный подход к автоматизации процессов розничной торговли и может быть доработано для применения в более сложных бизнес-сценариях.

# СПИСОК ЛИТЕРАТУРЫ (ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ) И ИНТЕРНЕТ-РЕСУРСОВ

1. ГОСТ Р 7.0.12-2022 Библиографическая запись. Сокращение слов и словосочетаний на русском языке. Общие требования и правила составления. – М.: ИПК Издательство стандартов, 2022. – 172 с.
2. ГОСТ 7.1-2023 Библиографическая запись. Библиографическое описание. Общие требования и правила составления. – М.: Стандартинформ, 2023. – 95 с.
3. ГОСТ 7.32-2021 Отчет о научно-исследовательской работе. Структура и правила оформления. – М.: Стандартинформ, 2021. – 48 с.
4. ГОСТ 7.82-2022 Библиографическая запись. Библиографическое описание электронных ресурсов. Общие требования и правила составления. – М.: Стандартинформ, 2022. – 56 с.
5. ГОСТ Р 7.0.100-2023 Библиографическая запись. Библиографическое описание. Общие требования и правила составления. – М.: Стандартинформ, 2023. – 114 с.
6. ГОСТ Р 7.0.5-2022 Библиографическая ссылка. Общие требования и правила составления. – М.: Стандартинформ, 2022. – 80 с.
7. Единая система программной документации. – М.: Стандартинформ, 2021. – 130 с.
8. Агальцов, В.П. Базы данных. В 2-х т. Т. 1. Локальные базы данных: Учебник / В.П. Агальцов. – М.: ИД ФОРУМ, НИЦ ИНФРА-М, 2022. – 432 с.
9. Албахари, Дж., Албахари, Б. C# 10.0. Справочник. Полное описание языка / Дж. Албахари, Б. Албахари. – М.: Вильямс, 2023. – 1160 с.
10. Голицына, О.Л. и др. Базы данных: учебник / О.Л. Голицына и др. – М.: Форум; Инфра-М, 2022. – 312 с.
11. Дейт, К.Дж. Введение в системы баз данных / К.Дж. Дейт. – М.: Диалектика, 2021. – Изд. 8-е. – 1088 с.

# ПРИЛОЖЕНИЕ

**ПРИЛОЖЕНИЕ А**

**Листинг страницы с Навигацией**

|  |
| --- |
| using System.Collections.Generic;  using System.Windows;  using System.Windows.Controls;  using System.Windows.Navigation;  namespace Coursework.Pages  {  public partial class NavigatePage : Page  {  public NavigatePage()  {  InitializeComponent();  LoadStaticTableButtons();  }  private void LoadStaticTableButtons()  {  var tableNames = new List<string>  {  "Shipments",  "Suppliers",  "Categories",  "Orders",  "Customers",  "Chart"  };  MenuPanel.ItemsSource = tableNames;  }  private void TableButton\_Click(object sender, RoutedEventArgs e)  {  if (sender is Button button && button.Content is string tableName)  {  NavigateToPage(tableName);  }  }  private void NavigateToPage(string tableName)  {  switch (tableName)  {  case "Shipments":  NavigationService.Navigate(new ShipmentsPage());  break;  case "Suppliers":  NavigationService.Navigate(new SuppliersPage());  break;  case "Categories":  NavigationService.Navigate(new CategoriesPage());  break;  case "Orders":  NavigationService.Navigate(new OrdersPage());  break;  case "Customers":  NavigationService.Navigate(new CustomersPage());  break;  case "Chart":  NavigationService.Navigate(new Page1());  break;  default:  MessageBox.Show("Страница не найдена.");  break;  }  }  private void ShowLegend\_Click(object sender, RoutedEventArgs e)  {  MessageBox.Show("Легенда:\n\nАдминистратор имеет доступ ко всем таблицам базы данных.\nСотрудник — только к таблицам заказов и клиентов.",  "Легенда",  MessageBoxButton.OK,  MessageBoxImage.Information);  }  }  } |