



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

ФАКУЛЬТЕТ
КАФЕДРА

ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ (ИУ)
ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЭВМ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ (ИУ7)

КУРСОВАЯ РАБОТА

НА ТЕМУ:

*Разработка базы данных для хранения и обработки
данных компании-реселлера электроники*

Студент

ИУ7-62Б
(группа)

(подпись, дата)

Кладницкий А.Б.
(И.О. Фамилия)

Руководитель курсового проекта

(подпись, дата)

Тассов К.Л.
(И.О. Фамилия)

2024 г.

РЕФЕРАТ

Расчетно-пояснительная записка 34 с., 9 рис., 2 табл., 1 прил., 13 источн.

РЕСЕЛЛИНГ, ФИРМА, БАЗА ДАННЫХ, СУБД, WEB

Объектом разработки являются БД и ПО, предоставляющее web-интерфейс к ней.

Цель работы — разработка БД для хранения и обработки данных компании-реселлера электроники.

В результате проделанной работы была разработана БД для хранения и обработки данных компании-реселлера электроники и ПО с web-интерфейсом для работы с этой БД.

Область применения — организация бизнеса по реселлингу электротоваров.

СОДЕРЖАНИЕ

СПИСОК СОКРАЩЕНИЙ	6
ВВЕДЕНИЕ	7
1 Аналитический раздел	8
1.1 Анализ существующих решений	8
1.2 Классификация СУБД	8
1.2.1 Модель хранения данных	9
1.2.2 Модель обработки данных	9
1.2.3 Архитектура	10
1.2.4 Способ доступа к БД	10
1.3 Формулирование требований к БД и ПО	11
1.4 Ролевая модель	12
1.5 ER-модель	13
1.6 Вывод из аналитического раздела	14
2 Конструкторский раздел	15
2.1 Описание сущностей и ограничений целостности	15
2.2 Проектирование функции на стороне БД	18
2.3 Архитектура программы	18
2.4 Вывод из конструкторского раздела	20
3 Технологический раздел	21
3.1 Выбор средств реализации	21
3.2 Реализация функции на стороне БД	21
3.3 Реализация сущностей и ограничений целостности	22
3.4 Интерфейс доступа к БД	25
3.5 Тестирование	27
3.6 Вывод из технологического раздела	28
4 Исследовательский раздел	29
4.1 Описание исследования	29
4.2 Вывод из исследовательского раздела	30

ЗАКЛЮЧЕНИЕ	31
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	32
ПРИЛОЖЕНИЕ А	34

СПИСОК СОКРАЩЕНИЙ

ПО — программное обеспечение

БД — база данных

СУБД — система управления базами данных

ER-МОДЕЛЬ — Entity-Relationship модель, модель сущность-связь

ВВЕДЕНИЕ

Реселлинг (от англ. resale, также реселл/ресейл) — это процесс покупки товаров с целью их последующей перепродажи, как правило, с прибылью [1].

Актуальность реселлинга обусловлена несколькими факторами [1].

1. Возможность продать товар компании, которая будет его реализовывать сама (удобство).
2. Возможность дать вторую жизнь в других руках неиспользуемым электронным товарам, а не просто выкинуть их (экологичность).
3. Наличие множества складов и возможность использования любого из них для реализации товара.

Цель курсовой работы — разработка БД для хранения и обработки данных компании-реселлера электроники.

Для достижения поставленной цели требуется решить следующие задачи:

- проанализировать существующие решения;
- сформулировать ограничения целостности и выделить роли;
- описать сущности и ограничения целостности БД;
- разработать БД и ПО для работы с ней;
- реализовать ПО;
- провести исследование характеристик разработанного ПО.

1 Аналитический раздел

В данном разделе анализируются существующие решения, рассматривается классификация СУБД, а также приводятся ER-диаграмма сущностей, формализация пользователей и диаграмма вариантов использования.

1.1 Анализ существующих решений

Среди компаний, занимающихся реселлингом товаров на территории РФ, можно выделить SYRO [2], OZON [3], Яндекс.Маркет [4], Second Friend Store [5]. В таблице 1.1 приведен сравнительный анализ этих компаний по следующим критериям:

- наличие собственных складов;
- наличие ремонтного сервиса;
- ассортимент товаров;
- наличие комиссии.

Таблица 1.1 – Сравнение компаний-реселлеров

Название	Наличие складов	Наличие ремонтного сервиса	Ассортимент товаров	Наличие комиссии
SYRO	+	+	Только электротовары	+
OZON	+	-	Все товары	+
Яндекс.Маркет	+	-	Все товары	+
Second Friend Store	-	-	Только одежда	-

1.2 Классификация СУБД

СУБД классифицируются по различным критериям, в зависимости от их архитектуры, способа доступа к БД, модели хранения данных и модели обработки данных [7].

1.2.1 Модель хранения данных

По модели хранения данных выделяют несколько типов СУБД [7]:

- иерархические;
- сетевые;
- реляционные;
- объектно-ориентированные.

Иерархические СУБД организуют данные в виде иерархии, где каждый узел имеет родителя и может иметь несколько дочерних узлов.

Сетевые СУБД хранят данные в виде графа, где сущность может быть связана с несколькими другими сущностями.

Реляционные СУБД хранят данные в виде таблиц, которые состоят из строк и столбцов. Реляционные базы используют структурированный язык запросов (SQL) для доступа к данным.

Объектно-ориентированные СУБД организуют хранение данных в виде объектов и классов, что позволяет эффективно работать с комплексными данными и их отношениями.

1.2.2 Модель обработки данных

По модели обработки данных, СУБД разделяются на OLTP и OLAP [6].

OLTP (Online Transaction Processing) базы данных предназначены для обработки оперативных транзакций (оформление заказов, бронирование и т.д.) в реальном времени. OLTP-модель как правило используется для быстрой записи, обновления и удаления данных. Главная цель OLTP — обеспечить быструю и надежную обработку операций.

OLAP (Online Analytical Processing) базы данных предназначены для анализа и отчетности. Эффективно работают с крупными объемами данных. OLAP-модель обычно используется для агрегации данных, создания сводных таблиц, проведения аналитических запросов и прогнозирования.

1.2.3 Архитектура

По архитектуре организации хранения данных, СУБД разделяются на локальные и распределенные [7].

Локальная СУБД устанавливается и работает на одном компьютере (сервере). Она хранит и обрабатывает данные только на нем. Примерами локальных СУБД являются SQLite, Microsoft Access и PostgreSQL (если установлена на одном сервере).

Распределенные СУБД распределяют хранение и обработку данных между несколькими устройствами (серверами). Распределенные СУБД обычно имеют более высокую производительность и масштабируемость, поскольку распределение данных по разным узлам сети уменьшает нагрузку на отдельные устройства. Примерами распределенных СУБД являются MongoDB и Cassandra.

1.2.4 Способ доступа к БД

По способу доступа к БД выделяют следующие виды СУБД [7]:

- файл-серверные (Microsoft Access);
- клиент-серверные (PostgreSQL, Oracle, MS SQL Server);
- встраиваемые (Redis, SQLite);
- сервисно-ориентированные;
- другие.

Файл-серверные СУБД используются для хранения данных в файлах на сервере. В этом случае клиенты обращаются к серверу для доступа ко всем данным сразу. При такой организации доступа, каждый пользователь хранит на своем компьютере локальную копию БД, а запросы выполняются локально.

Клиент-серверные СУБД представляют собой систему, в которой клиентские компьютеры соединены с сервером, который управляет хранением и обработкой данных. Клиенты обращаются к серверу для доступа только к нужным данным. Запросы обрабатываются на сервере.

Встраиваемые СУБД представляют собой хранилище, интегрированное в ПО или устройство. Они работают непосредственно в рамках программы и не требуют отдельной установки или запуска сервера.

Сервисно-ориентированные СУБД используются для хранения данных, предоставляемых через веб-сервисы. В этом случае доступ к данным и обработка запросов для клиентов осуществляется через API.

1.3 Формулирование требований к БД и ПО

ПО должно представлять собой приложение с Web-интерфейсом.

Разрабатываемое ПО должно предоставлять пользователям доступ к данным о товарах, их персональных данных и данных их фирм. Должна быть реализована возможность выбора всех товаров и товаров по категориям. Должна быть реализована система регистрации и авторизации. Представители фирм должны иметь возможность заключать и подписывать контракты.

Администратор должен иметь возможность управлять сущностями системы (фирмами, пользователями, складами, товарами, производителями, категориями).

Аудитор должен иметь возможность просматривать все контракты и сопутствующие им документы.

1.4 Ролевая модель

Выделены следующие роли:

1. Менеджер — менеджер фирмы; может управлять товарами (добавлять, изменять и удалять), а также составлять контракты от лица своей фирмы.
2. Руководитель — руководитель фирмы; может управлять товарами (добавлять, изменять и удалять), а также подписывать контракты от лица своей фирмы.
3. Администратор — суперпользователь; может управлять (добавлять, изменять и удалять) всеми компонентами системы (фирмами, складами, товарами, производителями, категориями и пользователями).
4. Аудитор — может просматривать список всех контрактов и приложенные к ним документы.

Каждая фирма при заключении контракта может выступать в роли как покупателем, так и продавца.

В соответствии с ролевой моделью, на рисунке 1.1 приведена диаграмма прецедентов.

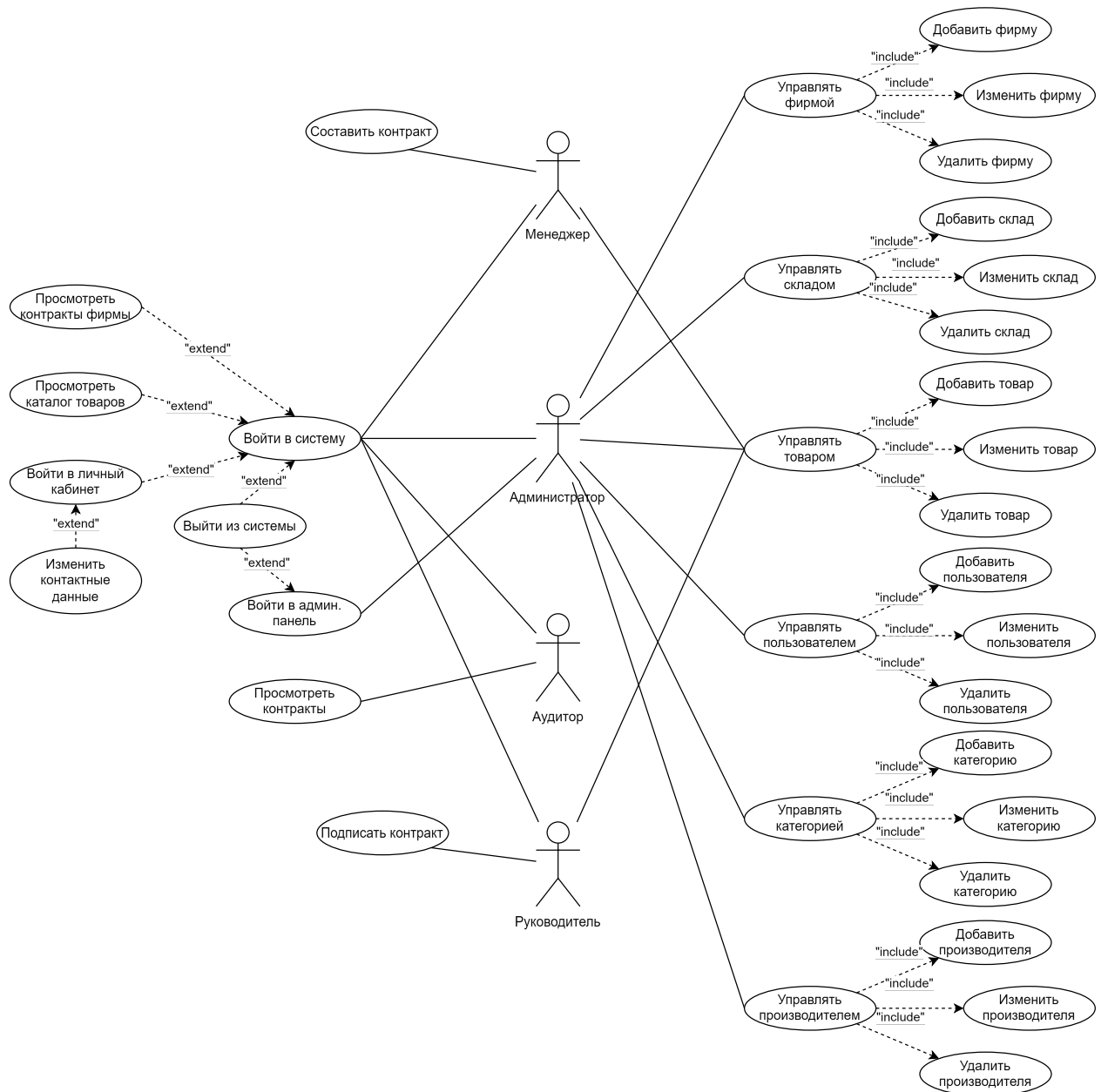


Рисунок 1.1 – Диаграмма прецедентов

1.5 ER-модель

На рисунке 1.2 представлена ER-модель в нотации Чена.

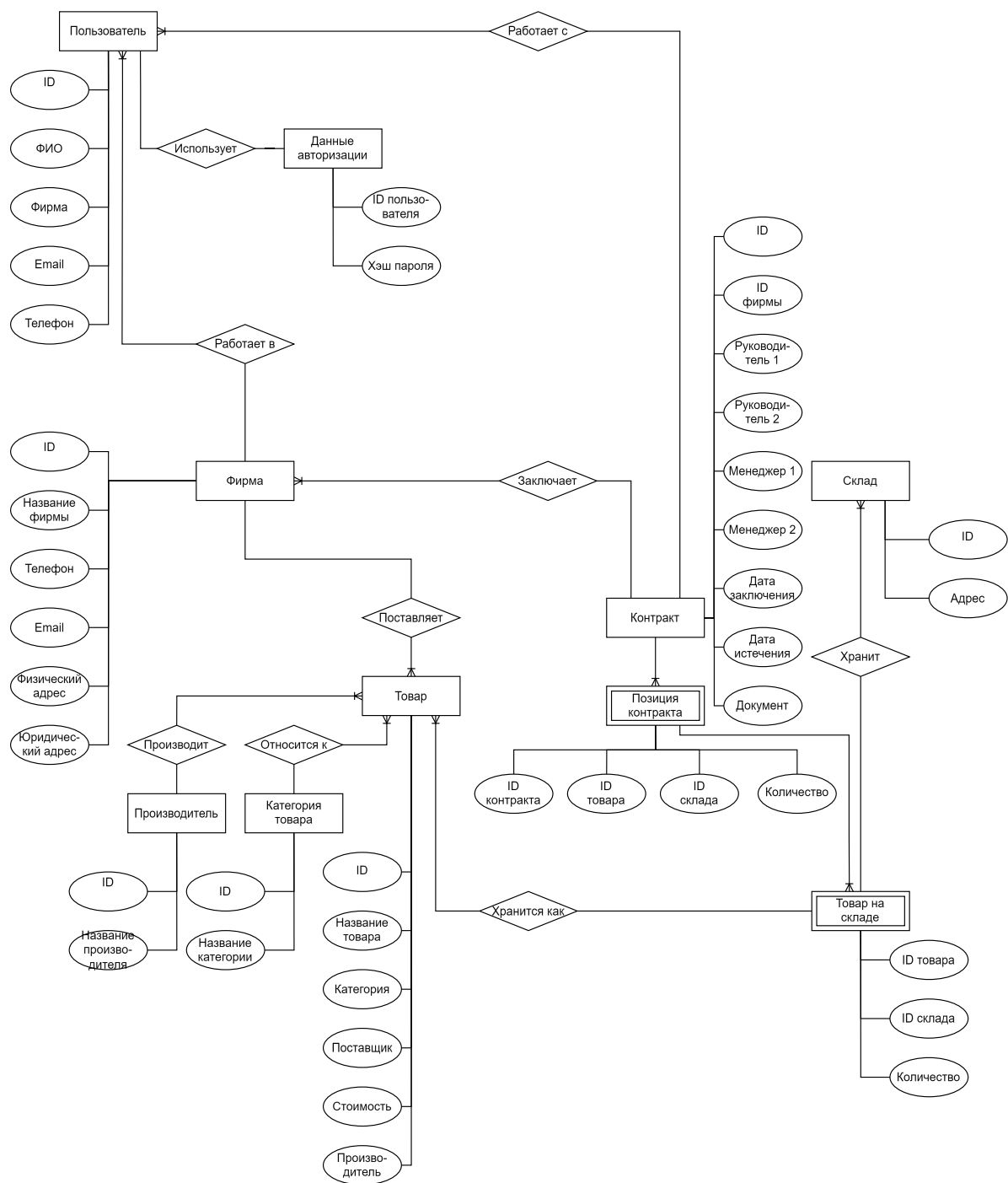


Рисунок 1.2 – ER-модель в нотации Чена

1.6 Вывод из аналитического раздела

В данном разделе были проанализированы существующие решения, рассмотрена классификация СУБД, а также были приведены ER-диаграмма сущностей, формализация пользователей и диаграмма вариантов использования.

2 Конструкторский раздел

В данном разделе рассмотрено проектирование БД и ПО, а также приведено описание сущностей и ограничений целостности.

2.1 Описание сущностей и ограничений целостности

1. Фирма имеет следующие атрибуты:

- ID, PK;
- название;
- телефон;
- email;
- физический адрес;
- юридический адрес.

2. Пользователь имеет следующие атрибуты:

- ID, PK;
- ФИО;
- фирма, FK;
- телефон;
- email UNIQUE.

3. Товар имеет следующие атрибуты:

- ID, PK;
- название;
- категория, FK;
- поставщик, FK;
- стоимость;
- производитель, FK.

4. Категория товара имеет следующие атрибуты:

- ID, PK;
- название.

5. Производитель имеет следующие атрибуты:

- ID, PK;
- название.

6. Склад имеет следующие атрибуты:

- ID, PK;
- адрес.

7. Контракт имеет следующие атрибуты:

- ID, PK;
- ID фирмы, FK;
- руководитель 1, может быть Null, FK;
- руководитель 2, может быть Null, FK;
- менеджер 1, может быть Null, FK;
- менеджер 2, может быть Null, FK;
- дата заключения;
- дата истечения;
- документ.

8. Товар на складе имеет следующие атрибуты:

- ID товара, FK;
- ID склада, FK;
- количество.

9. Позиция контракта имеет следующие атрибуты:

- ID контракта, FK;
- ID товара, FK;
- ID склада, FK;
- количество.

10. Данные аутентификации имеют следующие атрибуты:

- ID пользователя, FK;
- хэш пароля.

Все атрибуты описываемых сущностей, кроме помеченных атрибутов контракта, не могут принимать значение Null.

Атрибуты, помеченные как РК, являются ключевыми. Атрибуты, помеченные как FK, являются внешними.

Атрибут пользователя email также является уникальным.

Диаграмма проектируемой БД представлена на рисунке 2.1.

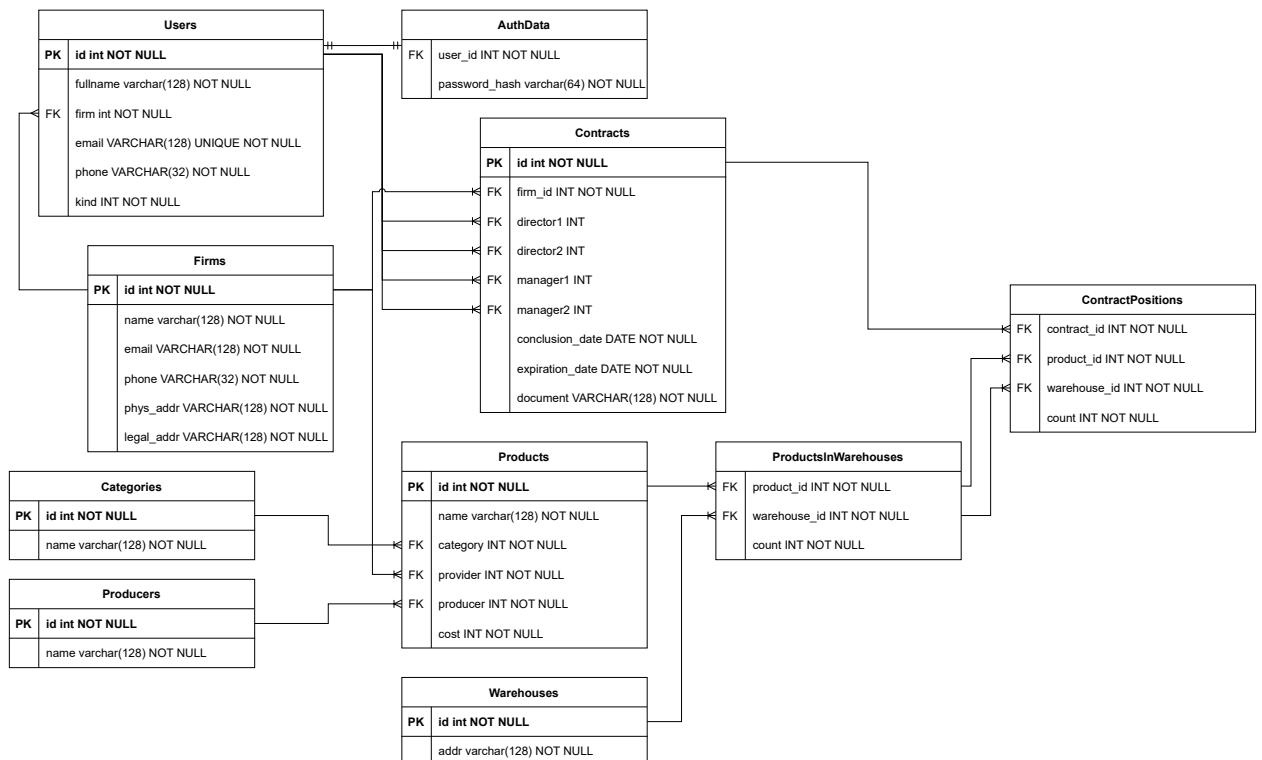


Рисунок 2.1 – Диаграмма БД

2.2 Проектирование функции на стороне БД

Проектируемая функция будет возвращать значение атрибута Id для добавляемого в таблицу объекта. Схема проектируемой функции приведена на рисунке 2.2.

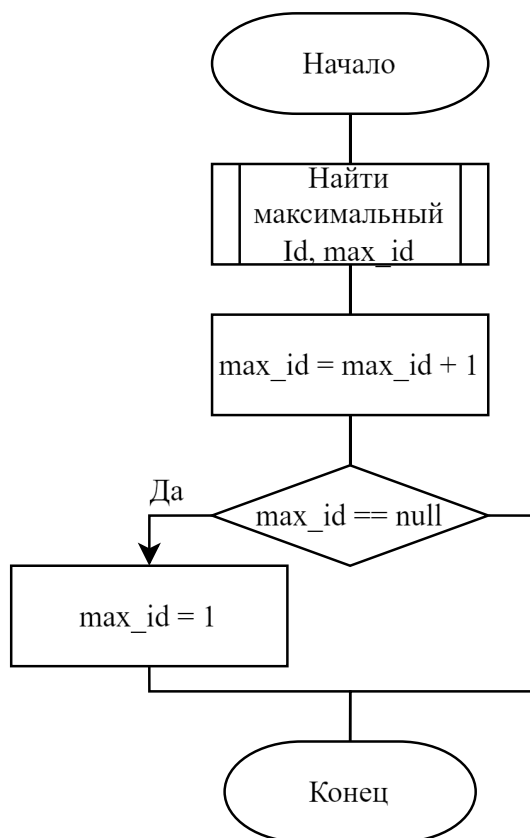


Рисунок 2.2 – Проектируемая функция

2.3 Архитектура программы

На рисунке 2.3 приведена верхнеуровневая UML-диаграмма модулей. ПО состоит из трех компонентов: компонент бизнес-логики, компонент доступа к данным и компонент пользовательского интерфейса.

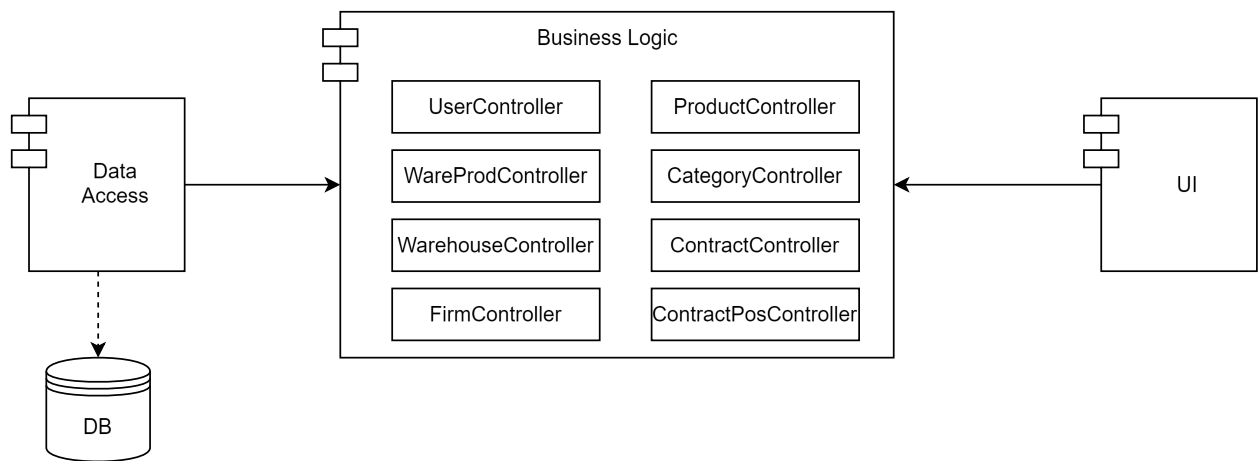


Рисунок 2.3 – Верхнеуровневая UML-диаграмма модулей

На рисунке 2.4 приведена UML-диаграмма классов разрабатываемого ПО.

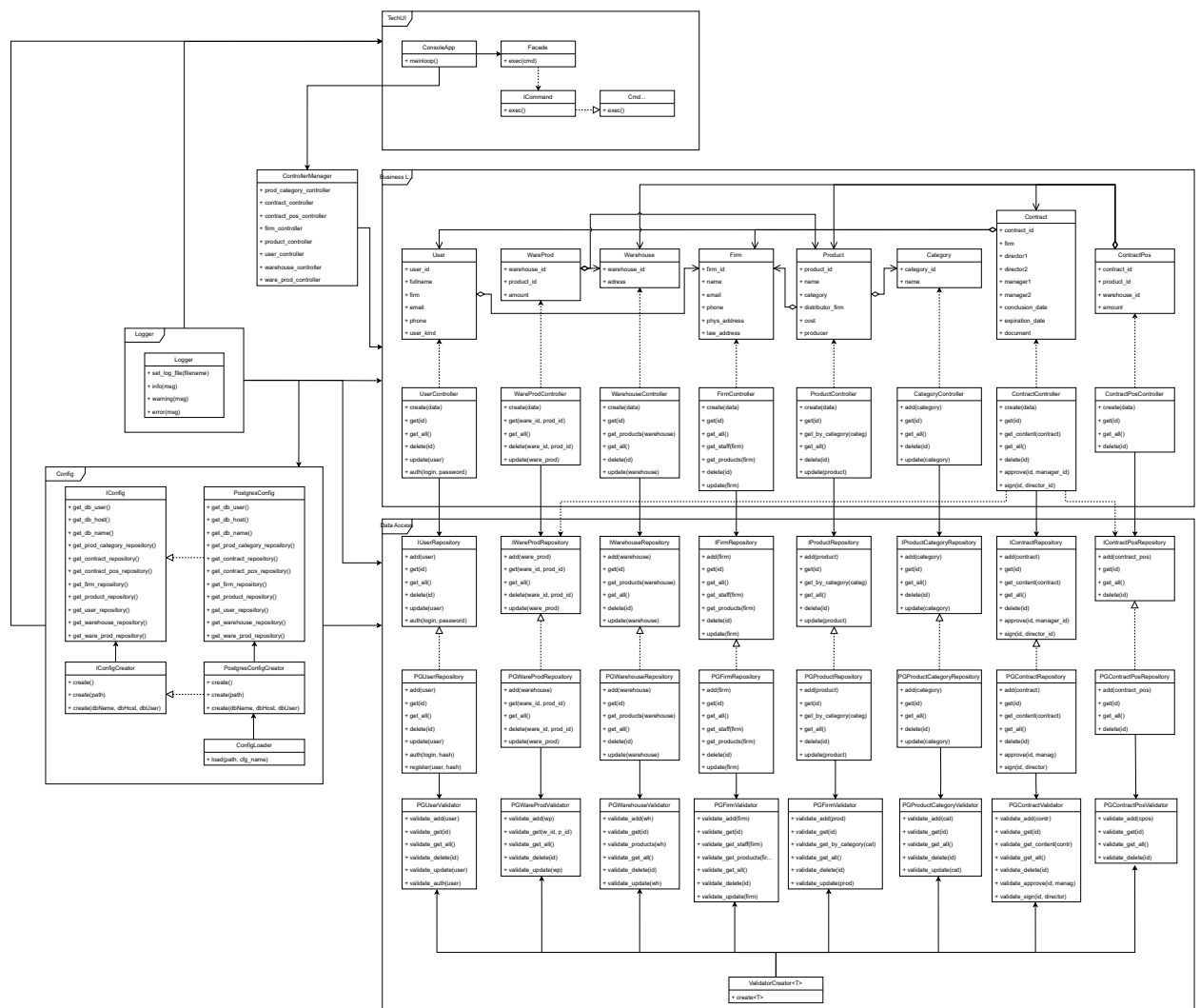


Рисунок 2.4 – UML-диаграмма классов

2.4 Вывод из конструкторского раздела

В данном разделе было рассмотрено проектирование БД и ПО, описаны сущности и ограничения целостности.

3 Технологический раздел

В данном разделе приведены выбор средств реализации, реализация функции на стороне БД, реализация сущностей и ограничений целостности, описание интерфейса доступа к БД и описание тестов.

3.1 Выбор средств реализации

В качестве языка программирования для разработки ПО был выбран язык C#, потому что средствами этого языка можно реализовать все требуемые возможности [8].

В качестве СУБД выбран PostgreSQL, поскольку в нем поддерживаются принципы ACID и стандарты SQL, а также он обладает высокой производительностью [9].

Для работы с СУБД выбрана библиотека Npgsql, поскольку она полностью поддерживает работу с PostgreSQL из кода C# [10].

Для реализации пользовательского интерфейса ПО был выбран фреймворк ASP.NET. Данный фреймворк предоставляет необходимые инструменты для создания пользовательского WEB-интерфейса [11].

В качестве среды разработки выбрана Visual Studio, поскольку эта среда хорошо подходит для разработки приложений на платформе .NET (C#) [12].

3.2 Реализация функции на стороне БД

На листинге 3.1 приведена реализация функции на стороне БД, возвращающая значение атрибута Id для добавляемого в таблицу объекта.

Листинг 3.1 – Реализация функции на стороне БД

```
1 CREATE OR REPLACE FUNCTION next_id(tname text)
2 RETURNS INT
3 AS $$
4 DECLARE ret INT;
5 BEGIN
6     EXECUTE 'SELECT max(id) + 1 FROM ' || tname INTO ret;
7     IF ret IS NULL THEN
8         ret = 1;
9     END IF;
10    RETURN ret;
11 END
12 $$
13 LANGUAGE plpgsql;
```

3.3 Реализация сущностей и ограничений целостности

На листингах 3.2–3.4 приведена реализация сущностей в БД.

Листинг 3.2 – Реализация сущностей в БД

```
1 CREATE TABLE IF NOT EXISTS Firms(
2     id INT PRIMARY KEY,
3     name VARCHAR(128) NOT NULL,
4     phone VARCHAR(32) NOT NULL,
5     email VARCHAR(128) NOT NULL,
6     physical_addr VARCHAR(128) NOT NULL,
7     legal_addr VARCHAR(128) NOT NULL
8 );
9 CREATE TABLE IF NOT EXISTS ProductCategories(
10    id INT PRIMARY KEY,
11    name VARCHAR(128) NOT NULL
12 );
13 CREATE TABLE IF NOT EXISTS Producers(
14    id INT PRIMARY KEY,
15    name VARCHAR(128) NOT NULL
16 );
```

Листинг 3.3 – Реализация сущностей в БД (продолжение)

```
1 CREATE TABLE IF NOT EXISTS Products(  
2     id INT PRIMARY KEY,  
3     name VARCHAR(128) NOT NULL,  
4     category INT NOT NULL,  
5     provider INT NOT NULL,  
6     cost INT NOT NULL,  
7     producer INT NOT NULL  
8 );  
9 CREATE TABLE IF NOT EXISTS Users(  
10    id INT PRIMARY KEY,  
11    fullname VARCHAR(128) NOT NULL,  
12    firm INT NOT NULL,  
13    email VARCHAR(128) UNIQUE NOT NULL,  
14    phone VARCHAR(32) NOT NULL,  
15    kind INT NOT NULL  
16 );  
17 CREATE TABLE IF NOT EXISTS Warehouses(  
18    id INT PRIMARY KEY,  
19    addr VARCHAR(128) NOT NULL  
20 );  
21 CREATE TABLE IF NOT EXISTS Contracts(  
22    id INT PRIMARY KEY,  
23    firm_id INT NOT NULL,  
24    director1_id INT,  
25    director2_id INT,  
26    manager1_id INT,  
27    manager2_id INT,  
28    conclusion_date DATE NOT NULL,  
29    expiration_date DATE NOT NULL,  
30    document VARCHAR(128) NOT NULL  
31 );  
32 CREATE TABLE IF NOT EXISTS ContractPositions(  
33    contract_id INT NOT NULL,  
34    product_id INT NOT NULL,  
35    warehouse_id INT NOT NULL,  
36    count INT NOT NULL  
37 );
```

Листинг 3.4 – Реализация сущностей в БД (окончание)

```
1 CREATE TABLE IF NOT EXISTS ProductsInWarehouses (  
2     product_id INT NOT NULL,  
3     warehouse_id INT NOT NULL,  
4     count INT NOT NULL  
5 );  
6 CREATE TABLE IF NOT EXISTS AuthData(  
7     user_id INT NOT NULL,  
8     password_hash VARCHAR(64) NOT NULL  
9 );
```

На листингах 3.5–3.6 приведена реализация ограничений целостности.

Листинг 3.5 – Реализация ограничений целостности

```
1 ALTER TABLE IF EXISTS Products  
2     ADD CONSTRAINT fk_products_cat FOREIGN KEY (category)  
3     REFERENCES ProductCategories(id) ON DELETE CASCADE,  
4     ADD CONSTRAINT fk_products_prod FOREIGN KEY (producer)  
5     REFERENCES Producers(id) ON DELETE CASCADE,  
6     ADD CONSTRAINT fk_products_prov FOREIGN KEY (provider)  
7     REFERENCES Firms(id) ON DELETE CASCADE;  
8 ALTER TABLE IF EXISTS Users  
9     ADD CONSTRAINT fk_users_firm FOREIGN KEY (firm) REFERENCES  
10    Firms(id) ON DELETE CASCADE;  
11 ALTER TABLE IF EXISTS Contracts  
12     ADD CONSTRAINT fk_contracts_u1 FOREIGN KEY (director1_id)  
13     REFERENCES Users(id) ON DELETE CASCADE,  
14     ADD CONSTRAINT fk_contracts_u2 FOREIGN KEY (director2_id)  
15     REFERENCES Users(id) ON DELETE CASCADE,  
16     ADD CONSTRAINT fk_contracts_u3 FOREIGN KEY (manager1_id)  
17     REFERENCES Users(id) ON DELETE CASCADE,  
18     ADD CONSTRAINT fk_contracts_u4 FOREIGN KEY (manager2_id)  
19     REFERENCES Users(id) ON DELETE CASCADE,  
20     ADD CONSTRAINT fk_contracts_firm FOREIGN KEY (firm_id)  
21     REFERENCES Firms(id) ON DELETE CASCADE;
```

Листинг 3.6 – Реализация ограничений целостности (окончание)

```
1 ALTER TABLE IF EXISTS ContractPositions
2     ADD CONSTRAINT fk_contr_pos_contract FOREIGN KEY
3         (contract_id) REFERENCES Contracts(id) ON DELETE CASCADE,
4     ADD CONSTRAINT fk_contr_pos_product FOREIGN KEY (product_id)
5         REFERENCES Products(id) ON DELETE CASCADE,
6     ADD CONSTRAINT fk_contr_pos_warehouse FOREIGN KEY
7         (warehouse_id) REFERENCES Warehouses(id) ON DELETE CASCADE;
8 ALTER TABLE IF EXISTS ProductsInWarehouses
9     ADD CONSTRAINT fk_ware_prod_product FOREIGN KEY (product_id)
10        REFERENCES Products(id) ON DELETE CASCADE,
11     ADD CONSTRAINT fk_ware_prod_warehouse FOREIGN KEY
12        (warehouse_id) REFERENCES Warehouses(id) ON DELETE CASCADE;
13 ALTER TABLE IF EXISTS AuthData
14     ADD CONSTRAINT fk_auth_user FOREIGN KEY (user_id) REFERENCES
15        Users(id) ON DELETE CASCADE;
```

3.4 Интерфейс доступа к БД

Реализован Web-интерфейс доступа к данным.

Наверху каждой Web-страницы расположена шапка. У неавторизованного пользователя в ней расположены ссылки на страницы регистрации и авторизации. У авторизованного — ссылки на главную страницу, личный кабинет, кабинет фирмы, каталог и выход, а также строка с его ФИО. У администратора также есть ссылка на страницу админ-панели, а у аудитора — на страницу аудита.

В личном кабинете выведена информация о пользователе (рис. 3.1). По нажатию кнопки «изменить» открывается форма для изменения этой информации.

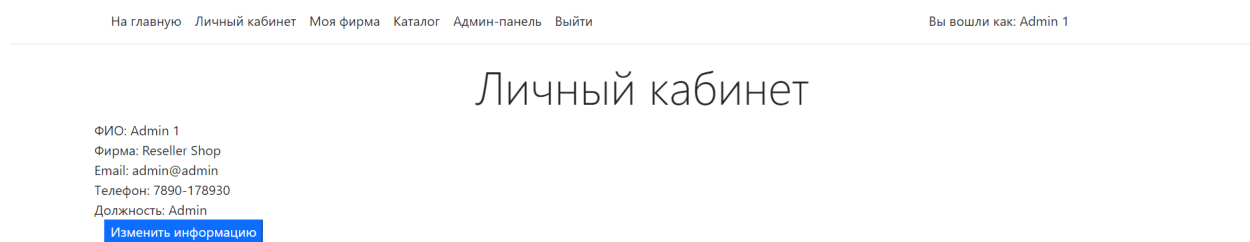


Рисунок 3.1 – Личный кабинет

В кабинете фирмы выведена информация о фирме пользователя (рис. 3.2). По нажатию кнопки «контракты» открывается страница, на которой перечислены все контракты фирмы. Также там расположена кнопка, открывающая форму добавления нового контракта (для менеджера). У руководителя есть кнопка удаления контракта. Можно перейти на страницу любого контракта: там выведена информация о нем и его содержанием, а также есть ссылка на его документ.

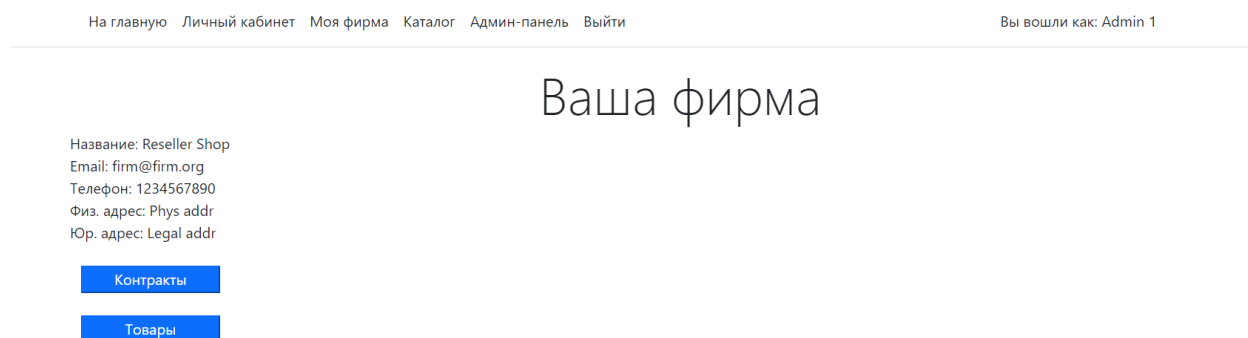


Рисунок 3.2 – Кабинет фирмы

По нажатию кнопки «товары» открывается страница, на которой перечислены все товары фирмы. Там же расположена кнопка, открывающая форму для добавления нового товара. На страницу любого товара можно также перейти: там выведена информация о нем и складах, на которых он есть в наличии. Существует возможность изменить или удалить товар, а также добавить его на склад.

В каталоге перечислены все товары, которые фирма пользователя может приобрести. Есть возможность выбора категории товаров.

На странице аудита выводится список всех контрактов. На страницу каждого контракта также можно войти: там перечислена вся информация о нем, а также есть ссылка на его документ.

В админ-панели можно получить доступ к управлению сущностями: категориями, производителями, пользователями, фирмами, товарами и складами (рис. 3.3). Есть страницы, на которых перечислены все сущности указанного типа, а также страницы каждой такой сущности. Можно добавлять новые объекты, а также изменять или удалять существующие.

Админ панель

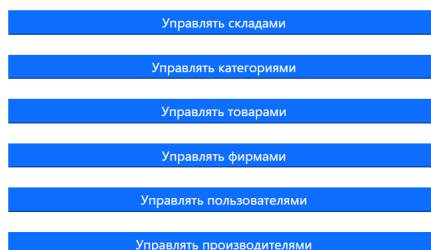


Рисунок 3.3 – Админ-панель

3.5 Тестирование

Проведено модульное для компонента бизнес-логики и интеграционное тестирование для компонента доступа к данным.

В ходе модульного тестирования для каждого класса объекта модели тестируется:

- добавление нового объекта;
- удаление объекта;
- получение объекта;
- изменение объекта;
- получение всех объектов объекта.

Для контракта тестируется также составление, подпись и получение всего содержимого контракта.

Для фирмы тестируется также получение всего штата сотрудников и получение всего списка товаров.

Для товаров тестируется получение всех товаров определенной категории.

В ходе модульного тестирования проведены следующие тесты:

- добавление нового склада, его обновление и последующее удаление;
- добавление нового товара, категории и производителя, их обновление и последующее удаление;
- создание нового контракта, утверждение и подпись его с обеих сторон и последующее удаление;

Все тесты были пройдены успешно.

3.6 Вывод из технологического раздела

В данном разделе были приведены выбор средств реализации, реализация функции на стороне БД, реализация сущностей и ограничений целостности, описание интерфейса доступа к БД и описание тестов.

4 Исследовательский раздел

В данном разделе проведено исследование зависимости времени выполнения запроса от размера таблицы в БД в двух случаях: без использования дополнительного индекса и с его использованием.

4.1 Описание исследования

Исследование проводилось с использованием PgAdmin — графической оболочки для СУБД PostgreSQL. В этом приложении есть встроенная возможность для измерения времени выполнения запросов [13].

Исследование производилось с использованием таблицы Users. Поиск в таблице производился в трех вариантах: по полю Id, по полю email и по обоим полям сразу.

Предварительно таблица была заполнена случайными данными. Поиск выполнялся для последнего объекта.

В таблице 4.1 представлены результаты исследования, время в мс.

Таблица 4.1 – Результаты исследования

К-во записей	Без дополнительного индекса			С дополнительным индексом		
	Id	Email	Оба	Id	Email	Оба
1000	174	166	172	162	156	153
5000	171	185	177	161	163	167
10000	178	194	195	170	177	177
50000	184	199	205	172	188	190
100000	187	212	223	176	190	193

4.2 Вывод из исследовательского раздела

В данном разделе было проведено исследование зависимости времени выполнения запроса от размера таблицы в БД в двух случаях: без использования дополнительного индекса и с его использованием. По результатам исследования выявлено следующее:

- с увеличением количества записей в таблице увеличивается время поиска;
- при отсутствии дополнительного индекса затрачиваемое время возрастает быстрее, чем при его наличии;
- при наличии дополнительного индекса время запроса по полю email все равно больше, чем по полю id; возможно, это объясняется тем, что сравнение числовых значений работает быстрее, чем текстовых;
- при количестве записей в пределах 100000 индекс ускоряет запрос примерно на 5–10%.

ЗАКЛЮЧЕНИЕ

В ходе выполнения курсовой работы была проанализирована предметная область и существующие решения, формализованы пользователи и сущности.

На основе этой формализации были спроектированы БД и ПО, описаны сущности и ограничения целостности.

Разработанные БД и ПО были реализованы. Был реализован также Web-интерфейс для ПО.

Было проведено исследование зависимости времени выполнения запроса от размера таблицы в БД в двух случаях: без использования дополнительного индекса и с его использованием. По результатам исследования выявлено следующее:

- с увеличением количества записей в таблице увеличивается время поиска;
- при отсутствии дополнительного индекса затрачиваемое время возрастает быстрее, чем при его наличии;
- при наличии дополнительного индекса время запроса по полю email все равно больше, чем по полю id; возможно, это объясняется тем, что сравнение числовых значений работает быстрее, чем текстовых;
- при количестве записей в пределах 100000 индекс ускоряет запрос примерно на 5–10%.

В дальнейшем, работа может быть расширена: например, могут быть добавлены новые сценарии использования (поиск товара по названию, сортировка по цене и т.д.) и поддержка других СУБД.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Что такое реселлинг [Электронный ресурс]. Режим доступа: <https://weblising.com/blog/internet-marketing/что-такое-reselling> (дата обращения 04.04.2024)
2. SYRO. Ресейл оборудования [Электронный ресурс]. Режим доступа: <https://syro.in/> (дата обращения 06.04.2024)
3. OZON [Электронный ресурс]. Режим доступа: <https://www.ozon.ru/> (дата обращения 06.04.2024)
4. Яндекс.Маркет [Электронный ресурс]. Режим доступа: <https://market.yandex.ru/> (дата обращения 06.04.2024)
5. Second Friend Store [Электронный ресурс]. Режим доступа: <https://secondfriendstore.ru/> (дата обращения 07.04.2024)
6. Conn, Samuel — OLTP and OLAP data integration: A review of feasible implementation methods and architectures for real time data analysis. — Regis University School for Professional Studies, 2005. — С. 515–520.
7. Волк В. К. — Базы данных. Проектирование, программирование, управление и администрирование. — Санкт-Петербург : Лань, 2020. — С. 12–34.
8. Документация по языку C# [Электронный ресурс]. Режим доступа: <https://learn.microsoft.com/ru-ru/dotnet/csharp/> (дата обращения 10.04.2024)
9. Документация PostgreSQL [Электронный ресурс]. Режим доступа: <https://www.postgresql.org/docs/> (дата обращения 16.04.2024)
10. Npgsql - .NET Access to PostgreSQL [Электронный ресурс]. Режим доступа: <https://www.npgsql.org/> (дата обращения 18.04.2024)
11. ASP.NET Core [Электронный ресурс]. Режим доступа: <https://dotnet.microsoft.com/en-us/apps/aspnet> (дата обращения 02.05.2024)
12. Visual Studio Code [Электронный ресурс]. Режим доступа: <https://code.visualstudio.com/> (дата обращения: 26.03.2024)

13. pgAdmin PostgreSQL Tools [Электронный ресурс]. Режим доступа:
<https://www.pgadmin.org/> (дата обращения 16.04.2024)

ПРИЛОЖЕНИЕ А

Презентация к курсовой работе