Государственное образовательное учреждение высшего профессионального образования

|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | ***«Московский государственный технический университет  имени Н.Э. Баумана»***  ***(МГТУ им. Н.Э. Баумана)*** |

ФАКУЛЬТЕТ \_\_\_Информатика и системы управления \_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_Програмное обеспечение ЭВМ и инфомационные технологии\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**РАСЧЁТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**к курсовому проекту на тему:**

«Веб приложение проведения аукционных торгов для закупки лекарственных препаратов»

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_**Мороков М. Л.\_**\_**

(Подпись, дата) (И.О.Фамилия)

Руководитель курсового проекта \_\_\_\_\_\_\_ **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_**Рогозин О. В.\_**\_\_**

(Подпись, дата) (И.О.Фамилия)

Москва, 2019

Содержание

[**Введение** 3](#_Toc20943191)

[**1. Аналитический раздел** 4](#_Toc20943192)

[**1.1 Теоретические основы разработки баз данных** 4](#_Toc20943193)

[**1.2 Анализ предметной области** 7](#_Toc20943194)

[**2. Конструкторский раздел** 7](#_Toc20943195)

[**2.1 Сущности модели базы данных** 8](#_Toc20943196)

[2.1.1 Информация по аукциону (AuctionInfo) 8](#_Toc20943197)

[2.1.2 Аукционы (Auction) 8](#_Toc20943198)

[2.1.3 Дистрибьюторы (Distributor) 8](#_Toc20943199)

[2.1.4 Информация по отгрузкам (ShipmentInfo) 8](#_Toc20943200)

[2.1.5 Отгрузка (Shipment) 8](#_Toc20943201)

[2.1.6 Информация о заказе (OrderInfoId) 9](#_Toc20943202)

[2.1.7 Препарат (Preparation) 9](#_Toc20943203)

[2.1.8 Заказ (Order) 9](#_Toc20943204)

[2.1.9 Препараты на складе (PreparationsBase) 9](#_Toc20943205)

[**2.2 Реляционная модель базы данных** 10](#_Toc20943206)

[**2.3 Структура базы данных** 12](#_Toc20943207)

[**2.4 Требования к API** 15](#_Toc20943208)

[**3. Технологический раздел** 16](#_Toc20943209)

[**3.1 Выбор СУБД** 16](#_Toc20943210)

[3.1.1 MySQL 16](#_Toc20943211)

[3.1.2 Microsoft SQL Server 17](#_Toc20943212)

[3.1.3 PostgreSQL 18](#_Toc20943213)

[3.1.4 MongoDB 18](#_Toc20943214)

[**3.2 Выбор языка программирования** 20](#_Toc20943215)

[**3.3 Выбор сборщика** 20](#_Toc20943216)

[**3.4 Интерфейс программы** 21](#_Toc20943217)

[**Заключение** 23](#_Toc20943218)

[**Список использованной литературы** 24](#_Toc20943219)

# **Введение**

Получить прибыль (выгодно реализовать как можно больше товаров и услуг) — конечная цель любого бизнеса. Чтобы достичь этой цели, нужно задействовать все инструменты, в том числе оптимизировать взаимоотношения с клиентами при помощи стратегии CRM.

CRM расшифровывается как Customer Relationship Management, то есть «управление отношениями с клиентами». Действительно, одна из самых главных функций CRM-программ — это учет клиентов и сделок.

Клиенты — один из важнейших активов любой компании. В условиях жесткой конкуренции, когда рынок насыщен товарами/услугами, примерно одинаковыми по цене и качеству, грамотная коммуникационная стратегия поможет удержать потребителей, повысить их лояльность. Владея полными и актуальными данными по каждому заказчику, можно организовать эффективную поддержку покупателя и управлять воронкой продаж. А на основе анализа накопленной информации можно улучшить внутренние бизнес-процессы и определить перспективные направления развития.

Подобные системы умеют автоматически выполнять часть вашей работы: формирует документы по шаблону, ставит задачи менеджерам на каждом этапе сделки, отправляет sms клиентам, в онлайн-режиме создает наглядные отчеты по всем бизнес-показателям — от суммы сделок до количества выполненных звонков. Все это помогает более эффективно выстраивать диалог с покупателем, поддерживать его лояльность и в итоге продавать ему больше и чаще. В 2015 году консалтинговая компания Capterra провела опрос среди 500 компаний, и выяснилось, что после внедрения CRM их прибыль выросла от 25 до 35%. Неудивительно, что CRM-системы стали настолько востребованы. По данным исследования, проведенного компанией Gartner в 2015 году, две трети зарубежных компаний уже предпочитают использование этих программ.

Разработать базу данных информационной системы дистрибьютора товаров оптовым клиентам для работы с поставщиками, которая является частью базы данных в CRM системе, и написать программу для взаимодействия с базой данных.

Программное обеспечение должно включать в себя функционал по просмотру и изменению базы данных. В модели данных предусмотреть следующие сущности предметной области: поставщики, адреса и поставляемые бренды.

# **1. Аналитический раздел**

## **1.1 Теоретические основы разработки баз данных**

Стержневые идеи современных информационных технологий базируются на концепции *баз данных.*

Согласно этой концепции, основой информационных технологий являются *данные,* которые должны быть организованы в базы данных в целях адекватного отображения изменяющегося реального мира и удовлетворения информационных потребностей пользователей.

Одним из важнейших понятий в теории баз данных является понятие *информации.* Под *информацией* понимаются любые сведения о каком-либо событии, процессе, объекте. К информации может относиться все, что может интересовать пользователя любого уровня.

*Данные -* это информация, представленная в определенном виде, позволяющем автоматизировать ее сбор, хранение и дальнейшую обработку человеком или информационным средством. Для компьютерных технологий данные - это информация в дискретном, фиксированном виде, удобная для хранения, обработки на ЭВМ, а также для передачи по каналам связи.

*База данных* (БД) - именованная совокупность данных, отражающая состояние объектов и их отношений в рассматриваемой предметной области, или иначе БД - это совокупность взаимосвязанных данных при такой минимальной избыточности, которая допускает их использование оптимальным образом для одного или нескольких приложений в определенной предметной области. БД состоит из множества связанных файлов.

*Система управления базами данных* (СУБД) - совокупность языковых и программных средств, предназначенных для создания, ведения и совместного использования БД многими пользователями.

*Автоматизированная информационная система* (АИС) - это система, реализующая автоматизированный сбор, обработку, манипулирование данными, функционирующая на основе ЭВМ и других технических средств и включающая соответствующее программное обеспечение (ПО) и персонал. В дальнейшем в этом качестве будет использоваться термин *информационная система* (ИС), который подразумевает понятие автоматизированная.

ИС может функционировать самостоятельно или служить компонентом более сложной системы.

Каждая ИС в зависимости от ее назначения имеет дело с той или иной частью реального мира, которую принято называть *предметной областью* (ПрО) *системы.* Выявление ПрО - это необходимый начальный этап разработки любой ИС. Именно на этом этапе определяются информационные потребности всей совокупности пользователей будущей системы, которые, в свою очередь, предопределяют содержание ее базы данных. По области применения ИС можно разделить на системы, используемые в образовании, производстве, бизнесе, науке и других областях.

В большинстве случаев прибегают к разбиению всего множества объектов ПрО на группы объектов, однородных по структуре, обладающих одинаковыми свойствами, например. В каждом фрагменте ПрО выделяется система объектов и связи между объектами, выделяются процессы, происходящие в этом фрагменте, а также конечное число пользователей.

*Банк данных* (БнД) является разновидностью ИС. БнД - это система специальным образом организованных данных: баз данных, программных, технических, языковых, организационно-методических средств, предназначенных для обеспечения централизованного накопления и коллективного многоцелевого использования данных.

Под *задачами обработки данных* обычно понимается специальный класс решаемых на ЭВМ задач, связанных с видом, хранением, сортировкой, отбором по заданному условию и группировкой записей однородной структуры.

При этом для пользователя предусматривается генерация различных отчетов, как правило, табличной формы.

Отдельные программы или комплекс программ, реализующие автоматизацию решения прикладных задач обработки данных, называются *приложениями.* Поскольку одни и те же данные могут использоваться для решения многих задач, то и приложений к одной и той же базе данных может быть много. Приложения, созданные средствами СУБД, относят к *приложениям СУБД.* Приложения, созданные вне среды СУБД с помощью систем программирования, использующих средства доступа к БД, к примеру, Delphi или C++Builder, называют *внешними приложениями.* Все приложения, работающие с одной и той же базой данных, должны функционировать корректно, не мешать друг другу и учитывать все изменения, которые вносятся другими приложениями. Такая координация работы приложений осуществляется СУБД.

Проектируемая БД должна обладать определенными свойствами. Назовем основные свойства БД.

*Целостность.* В каждый момент времени существования БД сведения, содержащиеся в ней, должны быть непротиворечивы. Целостность БД достигается вследствие введения ограничений целостности, в частности, к ним относятся ограничения, связанные с нормализацией БД. Желательно отслеживать диапазон допустимых значений, соотношения между значениями в полях, особенности написания формата. Существуют ограничения, работающие только при удалении записей. Например, нельзя удалять запись, связанную с другой неудаляемой записью.

*Восстанавливаемость.* Данное свойство предполагает возможность восстановления БД после сбоя системы или отдельных видов порчи системы. Сюда относится проверка наличия файлов, составляющих приложение. В основном свойство восстанавливаемости обеспечивается дублированием БД и использованием техники повышенной надежности.

*Безопасность.* Безопасность БД предполагает защиту данных от преднамеренного и непреднамеренного доступа, модификации или разрушения. Применяется запрещение несанкционированного доступа, защита от копирования и криптографическая защита. Также необходимы и чисто административные меры, например ограничение доступа к носителям информации.

*Эффективность.* Свойство эффективности обычно понимается как:

- минимальное время реакции на запрос пользователя;

- минимальные потребности в памяти;

- сочетание этих параметров.

*Предельные размеры и эксплуатационные ограничения.* Предельные размеры, а также другие ограничения, накладываемые эксплуатацией данной БД, могут существенно повлиять на проектное решение.

Создание баз данных, поддержка их в целостном, непротиворечивом состоянии, обеспечение безопасности их использования и сохранности информации вплоть до восстановления ее после различных сбоев, предоставление различных информационных услуг пользователям и многое другое обеспечивается системами управления баз данных, которым посвящен следующий подраздел книги.

*Модель данных -* это некоторая абстракция, в которой отражаются самые важные аспекты функционирования выделенной предметной области, а второстепенные - игнорируются. Модель данных включает в себя набор понятий для описания данных, связей между ними и ограничений, накладываемых на данные. В модели данных различают три главные составляющие:

- структурную часть, определяющую правила порождения допустимых для данной СУБД видов структур данных;

- управляющую часть, определяющую возможные операции над такими структурами;

- классы ограничений целостности данных, которые могут быть реализованы средствами этой системы.

Каждая СУБД поддерживает ту или иную модель данных. Необходимо отметить, что понятие модели данных фактически вошло в обиход специалистов в области БД только вместе с появлением реляционного подхода. Все ранние системы не основывались на каких-либо абстрактных моделях. Абстрактные представления ранних систем появились позже на основе анализа и выявления общих признаков у различных систем.

Ранние (корреляционные) СУБД активно использовались в течение многих лет. В свое время в подобные системы многие компании инвестировали большие средства. Они применялись дольше, чем используется какая-либо из реляционных СУБД, а некоторые из ранних систем используются даже в наше время. За время их существования накоплены громадные базы данных, и одной из актуальных проблем информационных систем является использование этих систем совместно с современными системами.

Итак, по существу модель данных, поддерживаемая механизмами СУБД, полностью определяет множество конкретных баз данных, которые могут быть созданы средствами этой системы, а также способы модификации состояния БД с целью отображения тех изменений, которые происходят в предметной области.

## **1.2 Анализ предметной области**

Для создания эффективной системы необходима хорошо спроектированная база данных, содержащая подробную информацию обо всех участниках

Можно выделить следующие основные сущности:

* Поставщики (юридические лица) – для предоставления основной информации о поставщиках как о юр. лицах;
* Наименования юридических лиц - для предоставления полной информации об юридических лицах;
* Сведения о регистрации юридических лиц – для предоставления информации о регистрации юр. лиц в налоговом органе;
* Бренды – для предоставления основной информации по поставляемым брендам;
* Бренды-конкуренты – характеризует конкурентные отношения поставляемых брендов;
* Известность брендов – для хранения сведений об узнаваемости бренда среди клиентов в виде статистики по опросам качества продукции;
* Профили потребителей бренда в РФ – для хранения сведений о целевом потребителе конкретного бренда в РФ;
* Контактные данные бренда – содержит информацию о контактных данных по конкретному бренду;
* Потенциальные рынки брендов в РФ – для получения статистической информации по продажам конкретного бренда в РФ;
* Сезоны – для предоставления информации по запланированным сезонным мероприятиям (показы, презентации, снятие с продажи) для каждого бренда;

Так же понадобится информация по адресам поставщиков. Для адресов можно выделить следующие сущности:

* Страны
* Регионы
* Населённые пункты
* Районы
* Улицы
* Адреса

# **2. Конструкторский раздел**

## **2.1 Сущности модели базы данных**

Фундаментальными реалиями моделирования являются данные с их свойствами и связи между ними. Главными элементами модели данных являются ее сущности.

### 2.1.1 Информация по аукциону (AuctionInfo)

|  |  |  |
| --- | --- | --- |
| *Атрибут* | *Тип* | *Описание* |
| Id | Int | PK таблицы |
| AuctionNumber | String | Номер аукциона |
| Date | String | Дата проведения аукциона |
| Status | String | Статус аукциона (проверен/не проверен) |

### 2.1.2 Аукционы (Auction)

|  |  |  |
| --- | --- | --- |
| *Атрибут* | *Тип* | *Описание* |
| Id | Int | PK таблицы |
| AuctionInfoId | Int | FK на таблицу информации по аукциону |

### 2.1.3 Дистрибьюторы (Distributor)

|  |  |  |
| --- | --- | --- |
| *Атрибут* | *Тип* | *Описание* |
| Id | Int | PK таблицы |
| Name | String | Имя дистрибьютора |

### 2.1.4 Информация по отгрузкам (ShipmentInfo)

|  |  |  |
| --- | --- | --- |
| *Атрибут* | *Тип* | *Описание* |
| Id | Int | PK таблицы |
| Date | String | Дата отгрузки |
| DistributorId | Int | FK на таблицу дистрибьюторов |
| Status | String | Статус отгрузки (отгружено / не отгружено) |

### 2.1.5 Отгрузка (Shipment)

|  |  |  |
| --- | --- | --- |
| *Атрибут* | *Тип* | *Описание* |
| Id | Int | PK таблицы |
| ShipmentInfoId | Int | FK на таблицу информации по отгрузкам |

### 2.1.6 Информация о заказе (OrderInfoId)

|  |  |  |
| --- | --- | --- |
| *Атрибут* | *Тип* | *Описание* |
| Id | Int | PK таблицы |
| Date | String | Дата заказа |
| PreShipmentDate | String | Дата предположительной отгрузки |
| CustomerName | String | Название заказчика |
| CustomerLocationArea | String | Область расположения заказчика |
| CustomerCity | String | Город заказчика |
| Status | String | Статус заказа (в процессе/ закрыт) |

### 2.1.7 Препарат (Preparation)

|  |  |  |
| --- | --- | --- |
| *Атрибут* | *Тип* | *Описание* |
| Id | Int | PK таблицы |
| OrderInfoId | Int | FK на таблицу информации о заказе |
| Name | String | Название препарата |
| Amount | Int | Количество товара |
| ExpirationDate | String | Дата истечения срока годности |
| Total | Double | Общая цена без НДС |
| TotalVAT | Double | Общая цена с НДС |
| PaymentDate | String | Дата оплаты товара |

### 2.1.8 Заказ (Order)

|  |  |  |
| --- | --- | --- |
| *Атрибут* | *Тип* | *Описание* |
| Id | Int | PK таблицы |
| OrderInfoId | Int | FK на таблицу информации о заказе |
| AuctionId | Int | FK на таблицу аукционов |
| ShipmentId | Int | FK на таблицу отгрузок |

### 2.1.9 Препараты на складе (PreparationsBase)

|  |  |  |
| --- | --- | --- |
| *Атрибут* | *Тип* | *Описание* |
| Id | Int | PK таблицы |
| Name | String | Название препарата |
| Price | Double | Цена за 1 ед. товара |
| Amount | Int | Количество |
| PackageType | String | Тип упаковки |
| Unit | String | Единица измерения |
| ProductionCode | String | Код продукции |
| Weight | Double | Вес (т.) |

## **2.2 Реляционная модель базы данных**

Реляционный подход обозначает определенную идеологию создания баз данных. Настойчивое желание пользователей оперировать более крупными объектами, чем элементы данных, предопределило ее появление и способствовало тому, что эта идеология довольно быстро завоевала мир. На скорость распространения идей реляционного подхода значительное влияние в основном оказали два фактора. Создал Кодд - IBM

Во-первых, БД представляется на внешнем, не зависящем от структуры ЭВМ уровне в виде совокупности двумерных таблиц, повседневно встречающихся в человеческой практике. Работа с таблицами привычна и понятна каждому пользователю. При этом весьма важно, что поиск и обработка информации, хранящейся в таблицах, не зависит от организации хранения данных в памяти ЭВМ, что значительно упрощает взаимодействие пользователя с банком данных и существенно повышает производительность его труда.

Во-вторых, манипулирование данными реляционной базы данных, которая с математической точки зрения представляет собой конечный набор конечных отношений различной арности между заранее определенным множеством элементарных данных, осуществляется в соответствии со специально разработанной для этой цели реляционной теорией. Над отношениями модели можно осуществлять различные алгебраические операции. Теория РБД как раз и определяет, какие операции и каким образом необходимо выполнять над отношениями, чтобы достичь заданной цели.

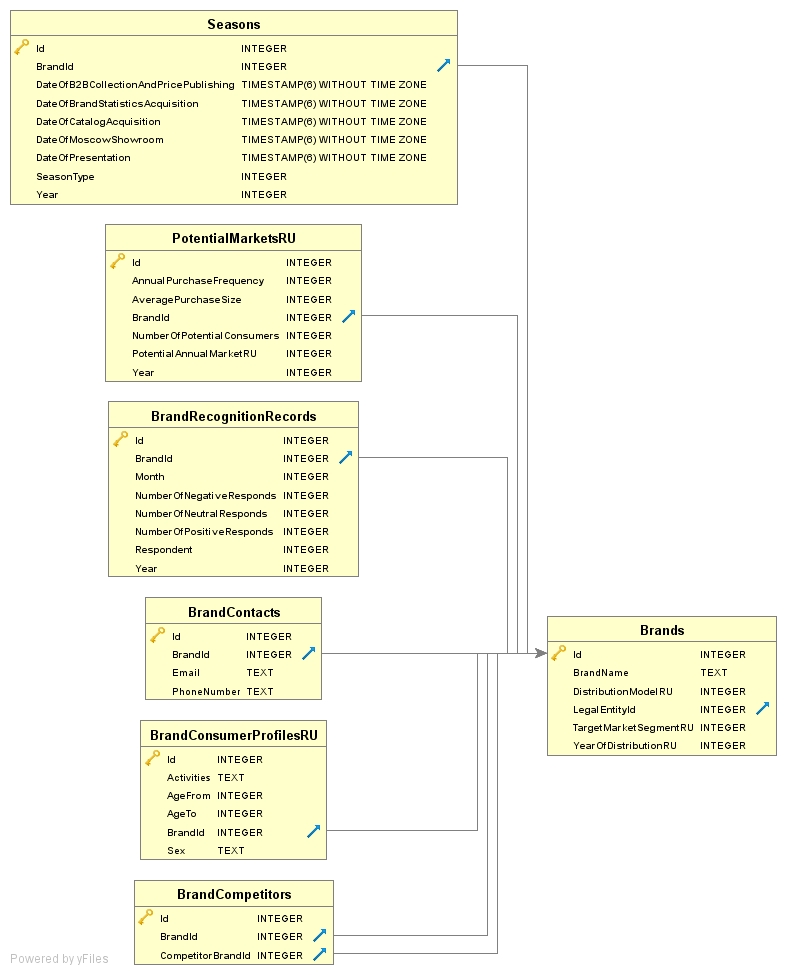
В настоящее время реляционный подход к построению информационных систем является наиболее распространенным. К числу достоинств реляционного подхода можно отнести:

- наличие небольшого набора абстракций, которые позволяют сравнительно просто моделировать большую часть распространенных предметных областей и допускают точные формальные определения, оставаясь интуитивно понятными;

- наличие простого и в то же время мощного математического аппарата, опирающегося главным образом на теорию множеств и математическую логику и обеспечивающего теоретический базис реляционного подхода к организации БД;

- возможность ненавигационного манипулирования данными без необходимости знания конкретной физической организации баз данных во внешней памяти.

В результате преобразования сущностей модели была построена следующая реляционная модель базы данных. (рисунок 1)

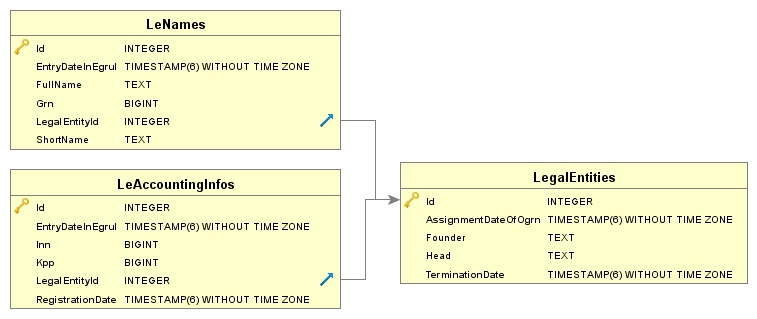


**Рисунок 1** - реляционная модель базы данных

## **2.3 Структура базы данных**

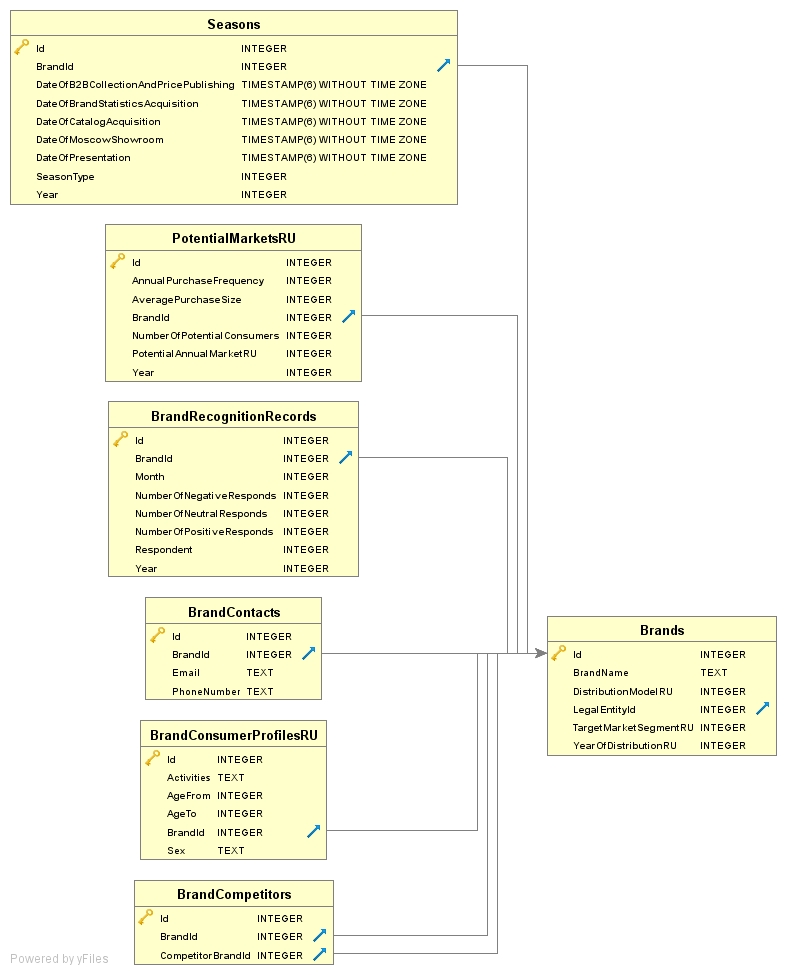
Таблицы, описывающие основные сущности модели базы данных:

Таблица «Юридические лица» и таблицы, связанные с ней, содержат информацию о поставщиках:



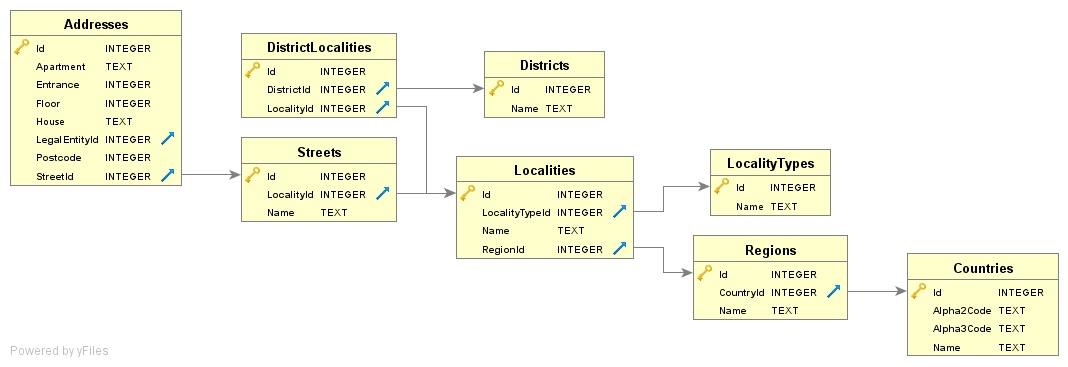
**Рисунок 2** - таблицы юридических лиц

Таблица «Бренды» и таблицы, связанные с ней, содержат информацию о поставляемых брендах каждым юридическим лицом



**Рисунок 3** - таблицы брендов

Таблица «Адреса» и таблицы, связанные с ней, содержат информацию об адресах, по которым зарегистрированы поставщики



**Рисунок 4** - таблицы адресов

## **2.4 Требования к API**

|  |  |  |
| --- | --- | --- |
| **Тип запроса** | **URL** | **Действие** |
| GET | /api/{TableName} | Получение информации из таблицы |
| GET | /api/{TableName}/?  from={FromVal}&  limit={LimitVal}& sortby={SortOrder}{SortVal} | Получение информации из таблицы согласно параметрам отображения:  1.Начальный номер 2.Количество элементов 3.Порядок сортировки  4.Атрибут сортировки |
| POST | /api/{TableName} | Создание новой записи в таблице |
| PUT | /api/{TableName}/{Id} | Изменение записи в таблице по указанному Id |
| DELETE | /api/{TableName}/{Id} | Пользователи данного форума |

# **3. Технологический раздел**

## **3.1 Выбор СУБД**

### 3.1.1 MySQL

MySQL - одна из самых популярных баз данных для веб-приложений. Фактически, является стандартом de facto для веб-серверов, которые работают под управлением операционной системы Linx. MySQL - это бесплатный пакет программ, однако новые версии выходят постоянно, расширяя функционал и улучшая безопасность. Существуют специальные платные версии, предназначенные для коммерческого использования. В бесплатной версии наибольший упор делается на скорость и надежность, а не на полноту функционала, который может стать и достоинством и недостатком - в зависимости от области внедрения.

Разработку и поддержку MySQL осуществляет корпорация Oracle, получившая права на торговую марку вместе с поглощённой Sun Microsystems, которая ранее приобрела шведскую компанию MySQL AB. Продукт распространяется как под GNU General Public License, так и под собственной коммерческой лицензией. Помимо этого, разработчики создают функциональность по заказу лицензионных пользователей. Именно благодаря такому заказу почти в самых ранних версиях появился механизм репликации.

Эта СУБД позволяет выбирать различные движки для системы хранения, которые позволяют менять функционал инструмента и выполнять обработку данных, хранящихся в различных типах таблиц. Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать как таблицы типа MyISAM, поддерживающие полнотекстовый поиск, так и таблицы InnoDB, поддерживающие транзакции на уровне отдельных записей. Более того, СУБД MySQL поставляется со специальным типом таблиц EXAMPLE, демонстрирующим принципы создания новых типов таблиц. Благодаря открытой архитектуре и GPL-лицензированию, в СУБД MySQL постоянно появляются новые типы таблиц. Она также имеет простой в использовании интерфейс, и пакетные команды, которые позволяют удобно обрабатывать огромные объемы данных. Система невероятно надежна и не стремится подчинить себе все доступные аппаратные ресурсы.

**Достоинства**

* Распространяется бесплатно
* Прекрасно документирована
* Предлагает много функций, даже в бесплатной версии
* Пакет MySQL включен в стандартные репозитории наиболее распространённых дистрибутивов операционной системы Linux, что позволяет устанавливать её элементарно
* Поддерживает набор пользовательских интерфейсов
* Может работать с другими базами данных, включая DB2 и Oracle.

**Недостатки**

* Придётся потратить много времени и усилий, чтобы заставить MySQL выполнять несложные задачи, хотя другие системы делают это автоматически, например: создавать инкрементные резервные копии.
* Отсутствует встроенная поддержка XML или OLAP.
* Для бесплатной версии доступна только платная поддержка.

Идеально подходит для: организаций, которым требуется надежный инструмент управления базами данных, но бесплатный.

### 3.1.2 Microsoft SQL Server

Ещё одной из популярных СУБД является программный продукт Microsoft SQL-сервер. Это система управления базами данных, движок которой работает на облачных серверах, а также локальных серверах, причем можно комбинировать типы применяемых серверов одновременно. Вскоре после выпуска Microsoft SQL сервер 2016, Microsoft адаптировала продукт для операционной системы Linux, а на Windows-платформе он работал изначально.

Одной из уникальных особенностей версии 2016 года является temporal data support (временная поддержка данных), которая позволяет отслеживать изменения данных с течением времени. Последняя версия Microsoft SQL-сервер поддерживает dynamic data masking (динамическую маскировку данных), которая гарантирует, что только авторизованные пользователи будут видеть конфиденциальные данные.

**Достоинства**

* Продукт очень прост в использовании
* Текущая версия работает быстро и стабильно
* Движок предоставляет возможность регулировать и отслеживать уровни производительности, которые помогают снизить использование ресурсов.
* Вы сможете получить доступ к визуализации на мобильных устройствах.
* Он очень хорошо взаимодействует с другими продуктами Microsoft.

**Недостатки**

* Цена для юридических лиц оказывается неприемлемой для большей части организаций.
* Даже при тщательной настройке производительности корпорация SQL Server способен занять все доступные ресурсы.
* Сообщается о проблемах с использованием службы интеграции для импорта файлов.

Идеально подходит для: крупных организаций, которые уже используют ряд продуктов Microsoft.

### 3.1.3 PostgreSQL

PostgreSQL является одним из нескольких бесплатных популярных вариантов СУБД, часто используется для ведения баз данных веб-сайтов. Это была одна из первых разработанных систем управления базами данных, поэтому в настоящее время она хорошо развита, и позволяет пользователям управлять как структурированными, так и неструктурированными данными. Может быть использован на большинстве основных платформ, включая Linux. Прекрасно справляется с задачами импорта информации из других типов баз данных с помощью собственного инструментария.

Движок БД может быть размещен в ряде сред, в том числе виртуальных, физических и облачных. Самая свежая версия, PostgreSQL 9.5, предлагает обработку больших объемов данных и увеличение числа одновременно работающих пользователей. Безопасность была улучшена благодаря поддержке DBMS\_SESSION.

**Достоинства**

* Является масштабируемым и способен обрабатывать терабайты данных.
* Поддерживает формат json.
* Существует множество предопределенных функций.
* Доступен ряд интерфейсов.

**Недостатки**

* Документация туманна, поэтому, возможно, ответы на некоторые вопросы придется искать в интернете.
* Конфигурация может смутить неподготовленного пользователя.
* Скорость работы может падать во время проведения пакетных операций или выполнения запросов чтения.

Идеально подходит для организаций с ограниченным бюджетом, но квалифицированными специалистами, когда требуется возможность выбрать свой интерфейс и использовать json.

### 3.1.4 MongoDB

Бесплатная база данных, которая имеет коммерческую версию - MongoDB, она предназначена для приложений, которые используют как структурированные, так и неструктурированные данные. Ядро является очень гибким и работает при подключении базы данных к приложениям через драйверы MongoDB. Существует широкий выбор доступных драйверов, поэтому легко найти драйвер, который будет работать с требуемым языком программирования.

Поскольку изначально система MongoDB не была разработана для обработки моделей реляционных данных (хотя может это выполнять), могут возникнуть проблемы производительности, если вы попытаетесь использовать её таким образом. Однако, движок предназначен для обработки различных данных, которые нельзя отнести к реляционным, и может хорошо справляться там, где другие движки работают медленно или бессильны.

MongoDB 3.2 - это последняя версия, и она имеет новую подключаемую систему движков хранения. Документы могут быть проверены в процессе обновления или выполнения вставок, а функции текстового поиска были улучшены. Новая способность частичного индексирования может привести к более высокой производительности, уменьшая размер индексов.

**Достоинства**

* Скорость и простота в использовании
* Движок поддерживает json и другие традиционные документы NoSQL.
* Данные любой структуры могут быть сохранены/прочитаны быстро и легко.

**Недостатки**

* SQL не используется в качестве языка запросов.
* Инструменты для перевода SQL-запросов в MongoDB доступны, но их следует рассматривать именно как дополнение.
* Программа установки может занять много времени.

Подходит для организаций, работающих с разнородными данными, которые тяжело поддаются классификации. Для внедрения потребуются высококлассные специалисты.

Исходя из всех достоинств и недостатков, была выбрана СУБД PostgreSQL

## **3.2 Выбор языка программирования**

Выбор языка программирования и среды разработки для данного проекта обуславливается его спецификой. Во-первых, данный проект является частью большой СRM системы. Во-вторых, проект должен быть не требователен к ПК, на которой он выполняется. Чтобы проект мог отвечать всем этим условиям, выбор пал на разработку frontend данной веб-системы на фреймворке Angular. В качестве языка программирования был выбран язык TypeScript – язык, разработанный корпорацией Microsoft на базе языка JavaScript. В языке реализован синтаксис для комфортной разработки веб-приложений в стиле ООП.

Для разработки backend составляющей данной системы был выбран язык C# на фреймворке ASP.NET Core.

## **3.3 Выбор сборщика**

Webpack - это статический модульный сборщик для приложений на JavaScript. Приложения, написанные на JavaScript, постоянно усложняются, поэтому хорошим выходом является использование сборщика. Подобные инструменты позволяют разработчикам упаковывать, компилировать и в целом организовывать все ресурсы, необходимые для проекта. Можно использовать не только сторонние библиотеки, но и собственные файлы. Подобная модульная система позволяет добиться лучшей организации проекта, так как получается, что он разбит на небольшие модули.

Webpack на данный момент является одним из самых мощных подобных инструментов. Он имеет открытый исходный код и позволяет решать множество разных задач..

**Достоинства**

* Отлично подходит для работы с одностраничными приложениями.
* Продвинутое разделение кода.
* Один из наиболее популярных инструментов JS-разработки.

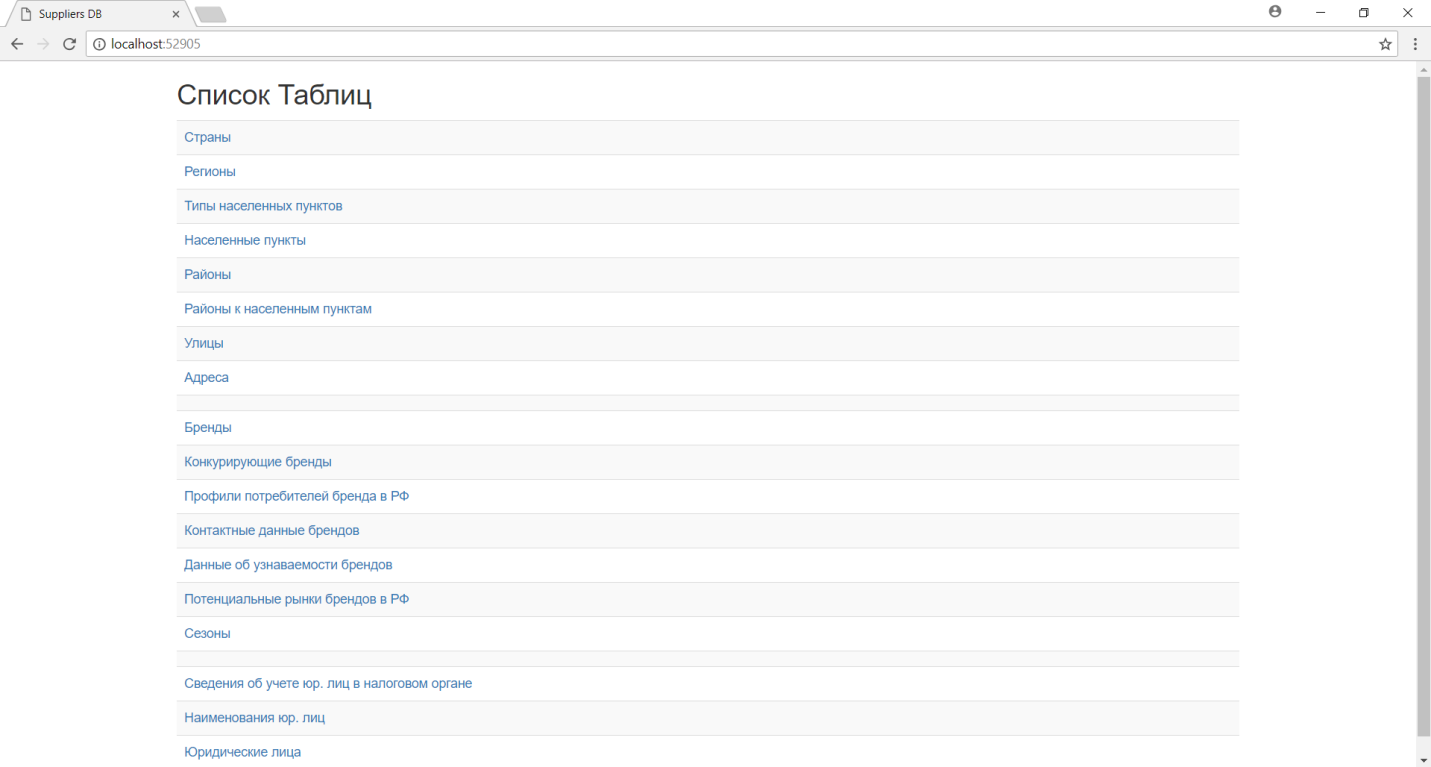
**Недостатки**

* Вначале немного сложно разобраться в его работе, часть документации устарела из-за большого количества изменений.

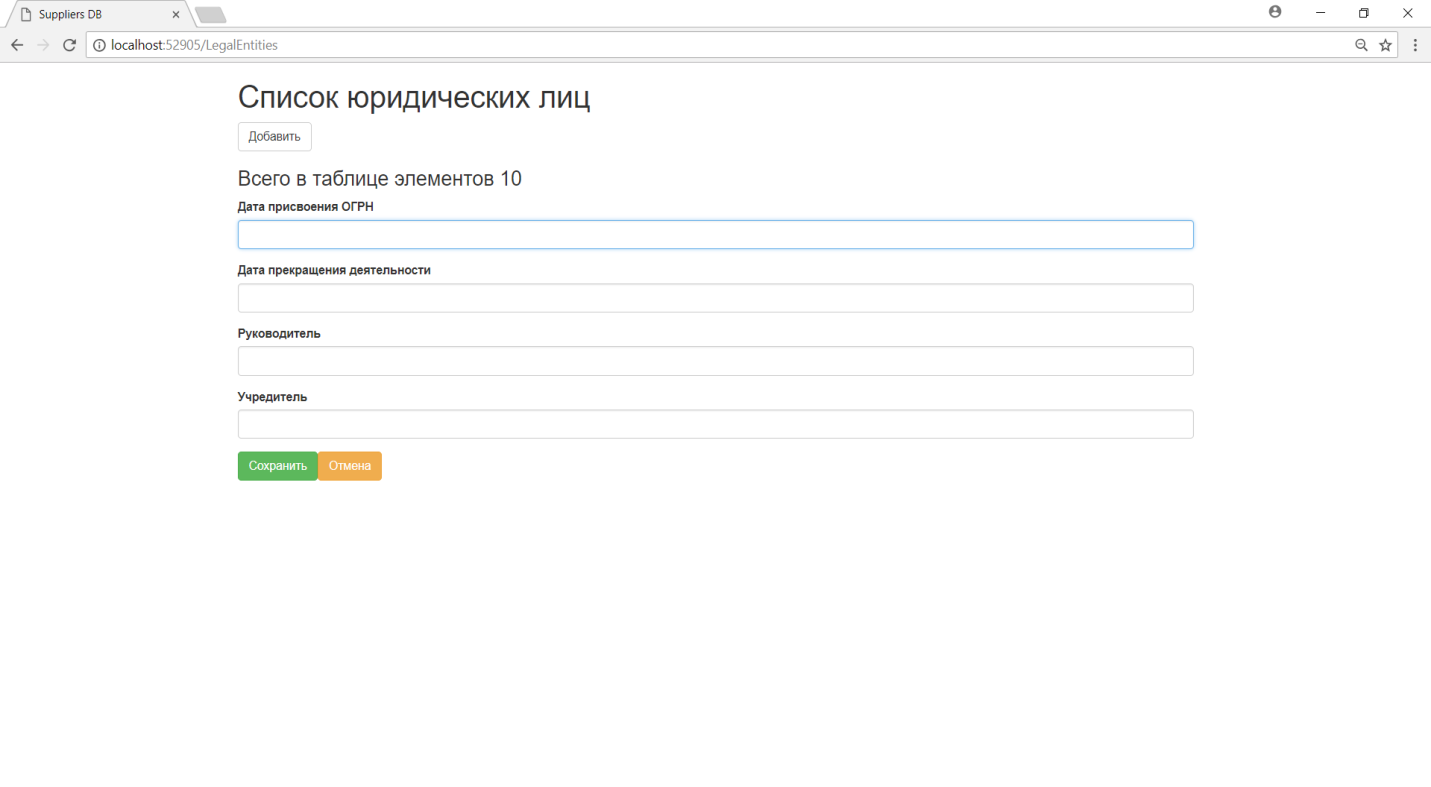
Учитывая все плюсы и минусы, Webpack был выбран в качестве сборщика frontend.

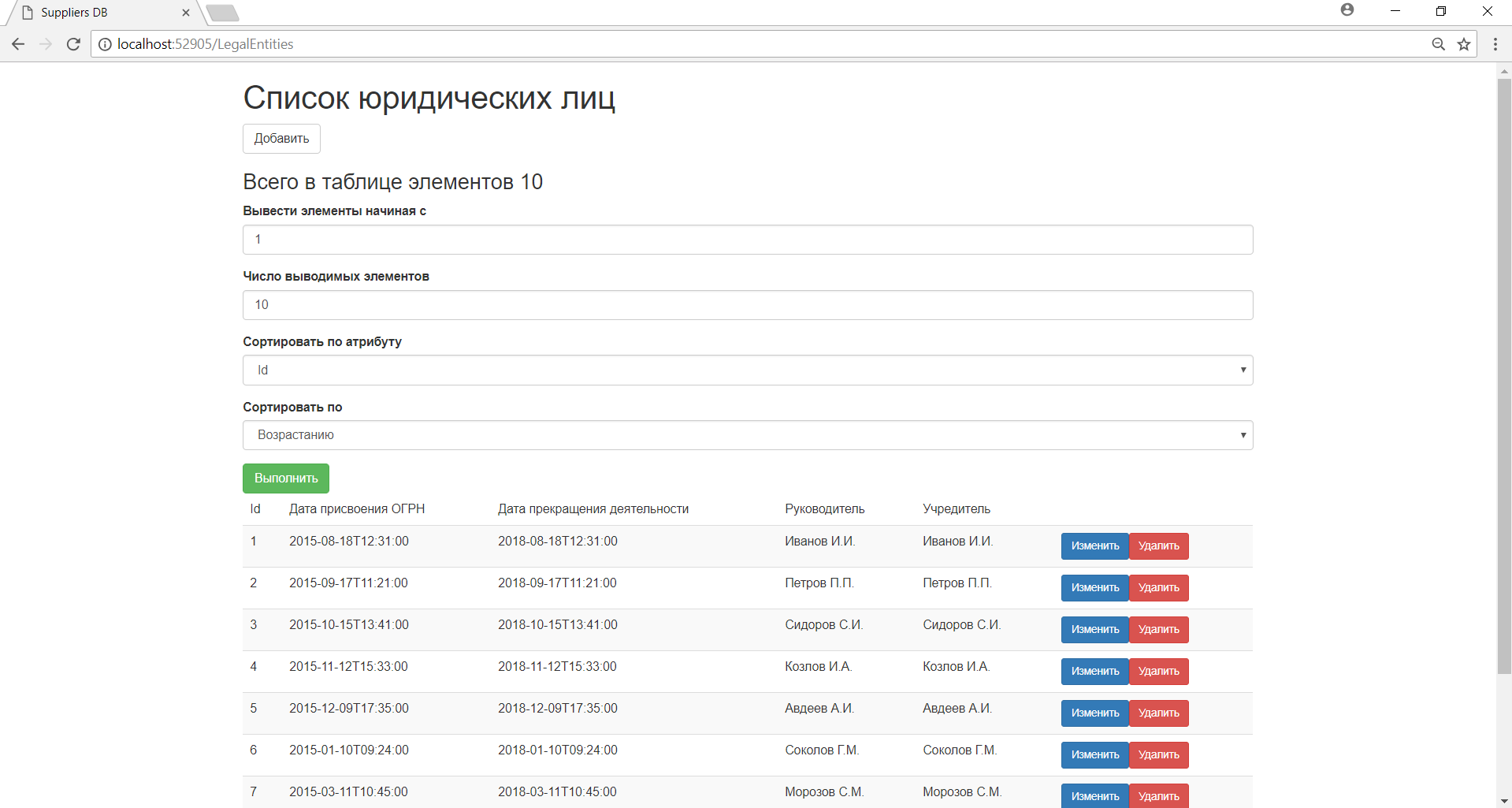
## **3.4 Интерфейс программы**

Интерфейс клиента данного приложения представляет собой веб-страницу, содержащую список всех таблиц для работы с поставщиками. Предусмотрена возможность вывода на экран элементов таблиц с заданными параметрами вывода (диапазон идентификаторов, порядок сортировки и атрибут сортировки). Также предусмотрено изменение атрибутов отдельных элементов таблиц, их удаление и добавление новых. Пример работы программы представлен на рисунках 5, 6 и 7.



**Рисунок 5** - вывод списка таблиц БД

**Рисунок 6** - форма добавления новой записи в БД

****

**Рисунок 7** - вывод на экран списка юридических лиц

# **Заключение**

В данном курсовом проекте были успешно реализованы все поставленные задачи. Были подробно рассмотрены: задача курсового проекта, а также порядок выполнения задания.

Разработана база данных, реализовано API для базы данных на двух языках программирования, приложение готово к развертыванию.

Средства эффективного хранения и выборки больших объемов информации внесли огромный вклад в успешное развитие Интернета. Обычно для хранения информации используются базы данных. Работа таких известных сайтов, как Yahoo, Amazon и Ebay, в значительной степени зависит от надежности баз данных, хранящих громадные объемы информации. Конечно, поддержка баз данных ориентирована не только на интересы гигантских корпораций - в распоряжении web-программистов имеется несколько мощных реализаций баз данных.

Правильная организация базы данных обеспечивает более быстрые и гибкие возможности выборки данных. Она существенно упрощает реализацию средств поиска и сортировки, а проблемы прав доступа к информации решаются при помощи средств контроля за привилегиями, присутствующими во многих системах управления базами данных (СУБД).

# **Список использованной литературы**

1. В.И. Ключко “Методическое указание к выполнению курсовой работы”, Краснодар, Издательство КубГТУ, 1997
2. М.П. Малыхина “Базы данных: основы, проектирование, использование”
3. Т.М. Карпова “Базы данных: модели, разработка, реализация”, издательство “Питер”, 2001
4. Кандзюба С. П. “Delphi 6/7. Базы данных и приложения. Лекции и упражнения”. – СПб.: ООО «ДиаСофтЮП», 2002. – 576.
5. Официальный сайт фреймворка Angular <https://angular.io/>
6. Библиотека оформления веб-интерфейса Angular Material <https://material.angular.io/>