МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ИНДИВИДУАЛЬНОЕ ДОМАШНЕЕ ЗАДАНИЕ

по дисциплине «Разработка программного обеспечения информационных систем»

Тема: Сопоставление разнородных наборов открытых (гео)данных - самые аварийные улицы

Студент гр. 5303 _____ Басин Д.Д.

Студент гр. 5303 _____ Губа Д.А.

Студент гр. 5303 _____ Кадыров Р.Р.

Преподаватель _____ Заславский М.М.

Санкт-Петербург
2018

ЗАДАНИЕ

НА ИНДИВИДУАЛЬНОЕ ДОМАШНЕЕ ЗАДАНИЕ

Студенты Басин Д.Д., Губа Д.А., Кадыров Р.Р.

Группа 5303

Тема проекта: Сопоставление разнородных наборов открытых (гео)данных - самые аварийные улицы

Исходные данные:

Проект должен быть разработан с использованием базы данных MongoDB

Содержание пояснительной записки:

Содержание, Введение, Качественные требования к решению, Сценарии использования, Модель данных, Разработанное прилоежение, Заключение, Список использованных источников.

Предполагаемый объем пояснительной записки:

Не менее 15 страниц.

Дата выдачи задания: 13.09.2018

Дата сдачи реферата: 27.12.2018

Дата защиты реферата: 27.12.2018

Студент	_____	Басин Д.Д.
Студент	_____	Губа Д.А.
Студентка	_____	Кадыров Р.Р.
Преподаватель	_____	Заславский М.М.

АННОТАЦИЯ

В индивидуальном домашнем задании реализован веб-сервис на основе СУБД MongoDB, с помощью которого можно определить самые аварийные улицы.

SUMMARY

In the individual homework implemented a web service based on MongoDB DBMS, with which you can determine the most emergency streets.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В настоящее время в России происходит большое количество аварий. Далеко не каждый житель страны знает, какие улицы и места в их городе наиболее опасны.

Целью проекта является разработка приложения, с помощью которого можно определить самые аварийные улицы для заданного района и получить краткую информации о ней.

В проекте разработано веб-приложение на основе СУБД MongoDB.

1. КАЧЕСТВЕННЫЕ ТРЕБОВАНИЯ К РЕШЕНИЮ

Необходимо разработать веб-приложение, позволяющее узнать самые аварийные улицы района.

Основные функции:

- Выбор необходимого района.
- Просмотр информации про выбранную аварию на карте.
- Просмотр статистики по загруженным данным.
- Импорт и экспорт данных.

2. СЦЕНАРИИ ИСПОЛЬЗОВАНИЯ

2.1. Сценарии использования для задачи импорта, представления, анализа и экспорта данных

Для импорта данных Пользователь должен нажать на кнопку «Uploader», в ней выбрать кнопку «Choose», выбрать файл формата .json, после чего нажать на «Upload». В файле должна быть структура, которая описана в пункте 3.1.

Сценарии использования

0. Сценарий использования - "Выбор типа работы":

Действующее лицо: Пользователь.

Основной сценарий:

1) Пользователь нажимает на "Stats".
2) Система выдает страницу для запроса отчета.
3) Пользователь выбирает нужный запрос.
4) Система выдает график, в зависимости от типа запроса.

Альтернативный сценарий №1:

1) Пользователь нажимает на "Map".
2) Начинается сценарий использования – «Выбор Города/Края/Области».

1. Сценарий использования - «Выбор Города/Края/Области»:

Действующее лицо: Пользователь.

Основной сценарий:

1) Пользователь нажимает на select-box: “Край / Область”.
2) Система выдает выпадающий список областей.
3) Пользователь выбирает нужный пункт.
4) Система сворачивает выпадающий список областей.
5) Система отображает на select-box выбранный пункт.
6) Пользователь нажимает кнопку «Применить».
7) Система производит фильтрацию по выбранному пункту на карте.

Альтернативный сценарий №1:

1) Пользователь не нажимает кнопку «Применить», а нажимает на select-box: “Год”.

2) Начинается сценарий использования – «Выбор Года».

Альтернативный сценарий №2:

1) Пользователь не нажимает кнопку «Применить», а нажимает на select-box: “Тип аварии”.

2) Начинается сценарий использования – «Выбор Типа аварии».

2. Сценарий использования - «Выбор Года»:

Действующее лицо: Пользователь.

Основной сценарий:

1) Пользователь нажимает на select-box: “Год”.
2) Система выдает выпадающий список доступных лет.
3) Пользователь выбирает нужный пункт.
4) Система сворачивает выпадающий список областей.
5) Система отображает на select-box выбранный пункт.
6) Пользователь нажимает кнопку «Применить».
7) Система производит фильтрацию по выбранному пункту на карте.

Альтернативный сценарий №1:

1) Пользователь не нажимает кнопку «Применить», а нажимает на select-box: “Край / Область”.

2) Начинается сценарий использования – «Выбор Города/Края/Области».

Альтернативный сценарий №2:

1) Пользователь не нажимает кнопку «Применить», а нажимает на select-box: “Тип аварии”.

2) Начинается сценарий использования – «Выбор Типа аварии».

3. Сценарий использования - «Выбор Типа аварии»:

Действующее лицо: Пользователь.

Основной сценарий:

1) Пользователь нажимает на select-box: “Тип аварии”.
2) Система выдает выпадающий список доступных лет.
3) Пользователь выбирает нужный пункт.
4) Система сворачивает выпадающий список областей.
5) Система отображает на select-box выбранный пункт.
6) Пользователь нажимает кнопку «Применить».
7) Система производит фильтрацию по выбранному пункту на карте.

Альтернативный сценарий №1:

1) Пользователь не нажимает кнопку «Применить», а нажимает на select-box: “Край / Область”.
2) Начинается сценарий использования – «Выбор Города/Края/Области».

Альтернативный сценарий №2:

1) Пользователь не нажимает кнопку «Применить», а нажимает на select-box: “Год”.
2) Начинается сценарий использования – «Выбор Года».

4. Сценарий использования - «Маркер на карте»:

Действующее лицо: Пользователь.

Основной сценарий:

1) Пользователь нажимает на активный маркер в области карты.
2) Система выдает всплывающее pop-up окно «Детали аварии».
3) Пользователь ознакамливается с деталями аварии.
4) Пользователь закрывает pop-up окно «Детали аварии».

Альтернативный сценарий:

1) Пользователь не закрывает pop-up окно «Детали аварии», а нажимает «Полный отчет».
2) Система формирует и выдает pop-up окно «Полный отчет по аварии».
3) Пользователь ознакамливается с полным отчетом по аварии.
4) Пользователь закрывает pop-up окно «Полный отчет по аварии».
5) Переход на шаг 3 «Основного сценария».

Для экспорта данных Пользователь должен нажать на кнопку «Uploader», в ней выбрать кнопку «Download dump». Скачается zip архив. В нем .bson и .json для каждой коллекции в базе, по ним можно восстановить базу командой mongorestore.

2.2. Вывод

Для решения преобладают операции чтения, так как для Пользователя реализован поиск аварийных улиц, то есть поиск и вывод данных, а не их добавление.

3. МОДЕЛЬ ДАННЫХ

3.1. Нереляционная модель данных

Графическое описание модели данных

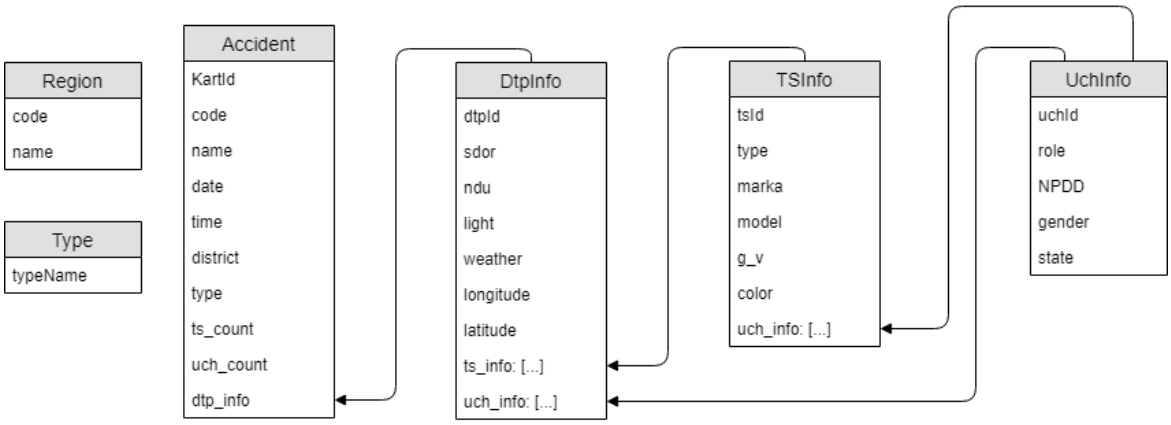

Рис. 1. - Графическое описание модели данных

Описание модели данных

1. Коллекция регионов "regions"

Содержит документы типа Region

- name (string) - название региона
- code (string) - код региона

2. Коллекция типов происшествий "types"

Содержит документы типа Type

- typeName (string) - название типа

3. Коллекция происшествий "accidents"

Содержит документы типа Accident

- code (string) - код региона, в котором произошло происшествие
- name (string) - название региона, в котором произошло происшествие
- KartId (string) - идентификатор происшествия на сайте ГИБДД
- date (string) - дата происшествия
- time (string) - время происшествия
- district (string) - район происшествия
- ts_count (int) - количество ТС, участвующих в происшествии
- uch_count (int) - количество лиц, участвующих в происшествии
- dtp_info (DtpInfo) - подробная информация о происшествии

Описание документа DtpInfo

- sdor (string[]) - участок дороги, на ктором произошло присшествие
- ndu (string[]) - нарушения дорожных условий
- light (string) - освещение место происшествия
- weather (string) - погодные условия во время происшествия
- longitude (string) - долгота координат происшествия
- latitude (string) - широта координат происшествия
- ts_info (TSInfo[]) - список с описанием ТС, участвующих в происшествии
- uch_info (UchInfo[]) - список с описанием лиц, участвующих в происшествии вне ТС

Описание документа TSInfo

- type (string) - тип ТС
- g_v (int) - год выпуска ТС
- model (string) - модель ТС
- marka (string) - марка производителя ТС
- color (string) - цвет ТС
- uch_info (UchInfo[]) - список с описанием лиц, участвующих в происшествии в данном ТС

Описание документа UchInfo

- role (string) - роль лица, как участника движения
- NPDD (string[]) - нарушение ПДД
- gender (string) - пол
- state (string) - состояние в результате происшествия

Оценка объема хранимой информации

Коллекция регионов

В данной коллекции может храниться до 85 регионов РФ.

Размер одного объекта = 12 байт (_id) + 25 байт (name) + 2 байта (code) = 39 байт

Размер коллекции = 85 * 39 = 3 315 байт

Коллекция типов происшествий

Количество элементов в данной коллекции сложно предсказать, так как типы могут пополняться.

На момент разработки проекта выделены 25 типов происшествий

Средний размер названия происшествия = 30 символов

Резмер одного объекта = 12 байт (_id) + 30 байт (typeName) = 42 байта

Размер коллекции = 25 * 42 = 1 050 байт

Коллекция происшествий

Максимальный размер данной коллекции очень сложно, т.к. документы имеют большую вложенность и много полей, содержащих массивы данных.

Средний размер документа UchInfo

> 12 байт (_id) + 11 байт (role) + 40 байт (NPDD) + 7 байт (gender) + 50 байт (state) = 120 байт

Средний размер документа TSInfo

> 12 байт (_id) + 25 байт (type) + 10 байт (marka) + 4 байта (g_v) + 8 байт (color) + 21 байт (model) + 2*120 байт (uch_info исходя из рассчета 2 участника/ТС) = 320 байт

Средний размер документа DtpInfo

> 12 байт (_id) + 50 байт (ndu) + 40 байт (sdor) + 2 * 7 байт (координаты) + 14 байт (weather) + 30 байт (light) + 3*320 байт (ts_info исходя из рассчета 3 ТС/ДТП) + 1*120(uch_info исходя из рассчета 1 участник без ТС/ДТП) = 1240 байт

Средний размер документа Accident

> 12 байт (_id) + 25 байт (name) + 2 байта (code) + 9 байт (KartId) + 10 байт (date) + 5 байт (time) + 15 байт (district) + 30 байт (typeName) + 2 * 1 байта (количества) + 1240 байт (dtp_info) = 1350 байт

Среднее количество ДТП в регионе за 1 год = 2000

Размер коллекции для 5 лет (2014-2018 года) для 85 регионов: 85 * 5 * 2000 * 1350 = 1 147 500 000 байт (примерно 1 Гб)

Коллекция "region":
name + code = 48 байт

Коллекция "types":
typeName = 24 байта

Коллекция "accidents":
code + name + kartID + date + time + district + ts_count + uch_count +
[N] dtp_info: [L] sdor + [F] ndu + light + longitude + latitude +
[K] ts_info: type + g_v + model + marka + color +
[P] uch_info: role + [S] NPDD + gender + state

Примерная формула для расчета: ((((P * (24 + S * 24 + 24 + 24))+K * (24+4+24+24+24))+N * (L * 24+F * 24+24+24+24))+24+24+24+24+24+24+4+4)

Примеры запросов

Запрос для получения всех регионов	db.regions.find({})
Запрос для получения всех типов	db.type.find({})
Запрос для добавления происшествия	db.regions.insertOne({ "code": "84", "name": "Республика Алтай", "KartId": 208207038, "date": "19.12.2017", "time": "21:25", "district": "Майминский район", "type": "Наезд на пешехода", "ts_count": 1,

```
"uch_count": 2,
"dtp_info": {
 "ndu": [
  "Отсутствие тротуаров (пешеходных дорожек)",
  "Отсутствие освещения"
 ],
 "sdor": [
  "Нерегулируемый перекрёсток неравнозначных улиц (дорог)"
 ],
 "light": "В темное время суток, освещение отсутствует",
 "weater": "Ясно",
 "longitude": "85.8431",
 "latitude": "51.8858",
 "ts_info": [
  {
   "type": "Цистерны",
   "marka": "КАМАЗ",
   "model": "Прочие модели КАМАЗ",
   "color": "Синий",
   "g_v": "1991",
   "uch_info": [
    {
     "role": "Водитель",
     "NPDD": [
      "Нет нарушений"
     ],
     "gender": "Мужской",
     "state": "Не пострадал"
    }
   ]
  }
```

```
            ],
            "uch_info": [
              {
                "role": "Пешеход",
                "NPDD": [
                  "Нахождение на проезжей части без цели её перехода"
                ],
                "gender": "Мужской",
                "state": "Раненый, находящийся (находившийся) на стационарном лечении"
              }
            ]
          }
        }
      })
```


3.2. Реляционная модель данных

Графическое описание модели данных

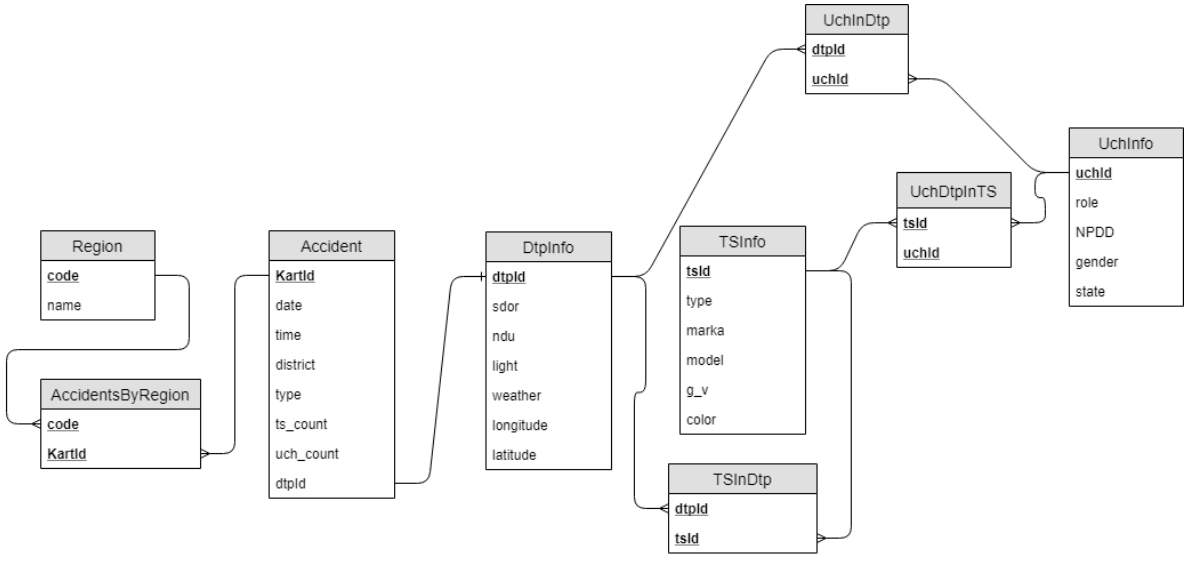

Рис. 2. - Графическое описание модели данных

Оценка объема хранимой информации

int ~ 4 байта

string ~ 24 байта

UchInfo:
uchId (int) + role + [S] NPDD + gender + state

[P] UchDtpinTS:
tsId (int) + uchId +
+ TSInfo:
tsId + type + marka + model + g_v + color

[M1] TSInDtp:
dtpId (int) + tsId

[M2] UchInDtp:
dtpId + uchId

[N] DtpInfo:
dtpId + [L] sdor + [F] ndu + light + longitude + latitude

Accident:
code + name + kartID + date + time + district + ts_count + uch_count

[A] AccidentByRegion:
code + KartId

Region:
code + name

Итого:
48 + A * (24+24) + (24+24+24+24+24+24+4+4) + N * (4+L * 24+F * 24+24+24+24) + M1 * (4+4) + M2 * (4+4) + P * (4+24+4+24+24+24+24+24) + (4+24+ S * 24+24+24)

Результат вычислений для предоставленной БД: 1 332 000 000 байт (больше 1 Гб).

Примеры запросов

Запрос на получение всех регионов	SELECT * FROM Region
Запрос на получение всех типов	SELECT * FROM Type
Запрос на получение происшествий по региону	SELECT * FROM Accident AS acc WHERE region.code = acc.code
Запрос на получение общей информации по происшествию	SELECT * FROM DtpInfo AS dtp WHERE dtp.dtpId = acc.dtp_info
Запрос на получение информации о транспортных средствах по происшествию	SELECT * FROM TSInfo AS ts WHERE ts.tsId IN (SELECT ts_info FROM DtpInfo WHERE dtp.dtpId = acc.dtp_info)
Запрос на получение информации об участниках по происшествию	SELECT * FROM UchInfo AS uch WHERE uch.uchId IN (SELECT uch_info FROM DtpInfo WHERE dtp.dtpId = acc.dtp_info) OR IN (SELECT uch_info FROM TSInfo AS ts WHERE ts.tsId IN (SELECT ts_info FROM DtpInfo WHERE dtp.dtpId = acc.dtp_info)

3.5. Выводы

Для рассматриваемой задачи NoSQL модель данных предпочтительнее SQL модели в плане сложности запросов к БД и скорости их выполнения, причем затрачиваемое количество памяти будет меньше. Это связано с тем, что появляется большое дублирование полей для связи таблиц.

4. РАЗРАБОТАННОЕ ПРИЛОЖЕНИЕ

4.1. Краткое описание

Разработанное приложение осуществляет аварийных происшествий по (гео)данным. Приложение состоит из страниц:

1. Главная страница. На главной странице пользователь может выбрать район из загруженных регионов, год, вид происшествия. Пользователь может увидеть отображение аварий на карте. Также может включить «Heat map».
2. Страница статистики. Пользователь может выбрать вид статистики:
    - Количество смертей в ДТП в зависимости от количества транспортных средств в происшествии.
    - Количество смертей в ДТП по регионам.
    - Соотношение смертей в ДТП мужчин/женщин.

    А также выбрать район и город, по которым необходимо получить отчет.
3. Страница загрузок. Пользователь может загрузить свои данные в систему, либо скачать загруженные данные на устройство.

4.2. Использованные технологии

При написании приложения использовались следующие технологии:

- Java 8.
- Angular 7.
- PrimeNG.
- Google API.
- ChartsJS.
- Spring Boot.

Для сборки использовался Maven.

4.3. Ссылки на Приложение

Исходный код приложения и инструкция по установке находятся по ссылке:

https://github.com/moevm/nosql2018-worst_accidents

ЗАКЛЮЧЕНИЕ

Разработано приложение для эффективного поиска аварийных происшествий с использованием базы данных MongoDB. Приложение работает полностью корректно с использованием актуальных технологий.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Java Platform Standard Edition 8 Documentation URL: https://docs.oracle.com/javase/8/docs/ (дата обращения: 21.10.2018).
2. Core Technologies // Spring Framework Documentation URL: https://docs.spring.io/spring/docs/current/spring-framework-reference/core.html#spring-core (дата обращения: 03.11.2018).
3. Spring Boot Reference Guide URL: https://docs.spring.io/spring-boot/docs/current/reference/html/ (дата обращения: 17.10.2018).

ПРИЛОЖЕНИЕ А. ДОКУМЕНТАЦИЯ ПО СБОРКЕ И РАЗВЕРТЫВАНИЮ ПРИЛОЖЕНИЯ

Загрузка данных

Можно импортировать данные с https://opengovdata.ru/dataset/statgibddru

Данные, созданные с помощью анализатора:

https://github.com/Shorstko/GibddStat

Импорт файлов данных .json один за другим

Можно создать дамп БД (mongodump) и скачать его, а затем восстановить его с помощью mongorestore.

Развертывание

Запустить «mvn clean install» в родительской папке.

Требуемая версия Maven выше, чем 3.1.0

Исполняемый файл jar будет сгенерирован в каталоге ./nosql-server/target

Запуск

Уже развернутый файл - папка «Исполняемый файл»

Выполнить nosql-server-0.0.1-SNAPSHOT.jar с помощью JRE с помощью команды: java -jar nosql-server-0.0.1-SNAPSHOT.jar

ПРИЛОЖЕНИЕ В. ИНСТРУКЦИЯ ДЛЯ ПОЛЬЗОВАТЕЛЯ

Главный экран

При входе на сайт отображается главная страница, на которой пользователь должен выбрать интересующий его регион, год и тип аварии.

Пользователь может посмотреть полученные данные на карте. Если пользователю интересна «Heat map», то он может выбрать соответствующую кнопку.

Страница статистики

Пользователь при необходимости получить статистику по интересующему ему региону переходит на эту страницу. Пользователь может выбрать вид статистики:

- Количество смертей в ДТП в зависимости от количества транспортных средств в происшествии.
- Количество смертей в ДТП по регионам.
- Соотношение смертей в ДТП мужчин/женщин.

А также выбрать район и город, по которым необходимо получить отчет.

Страница загрузок

Если пользователь хочет скачать имеющиеся данные, либо загрузить свои, то он переходит в этот раздел. Здесь пользователь может выбрать кнопкой «Choose» необходимые объекты для загрузки и загрузить их. Кнопкой «Download dump» пользователь может скачать данные.

Навигация по страницам приложения осуществляется с помощью кнопок меню в шапке страницы.

ПРИЛОЖЕНИЕ С. СНИМКИ ЭКРАНА ПРИЛОЖЕНИЯ

На рисунках 3-10 изображены снимки экрана приложения.

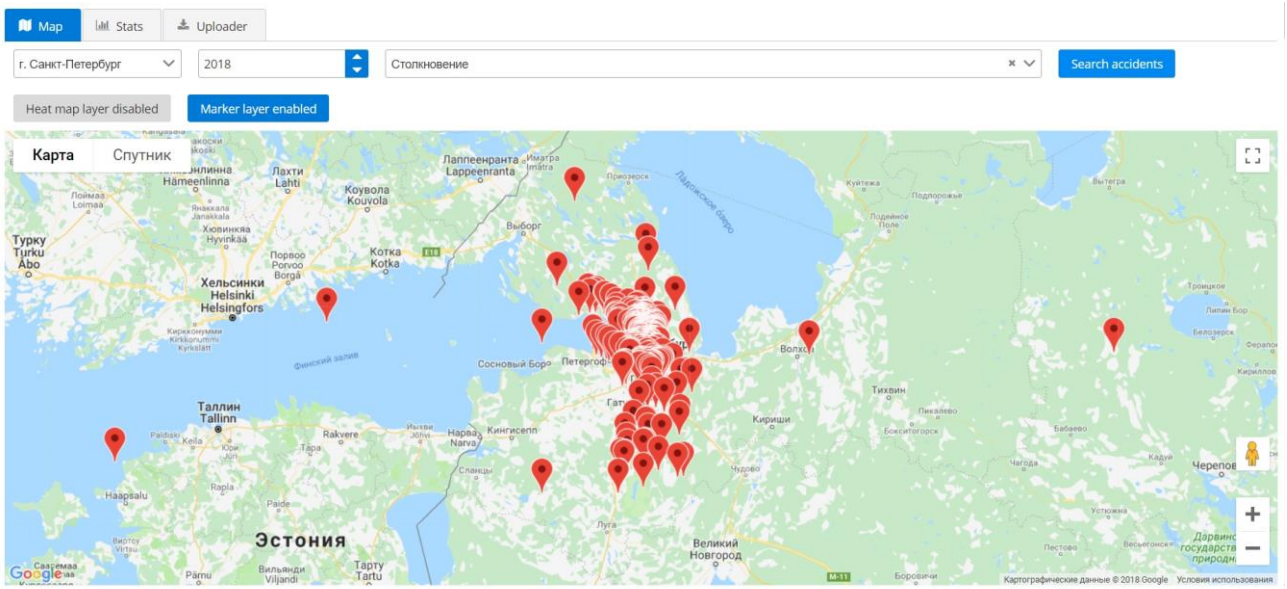

Рисунок 3 – Главная страница

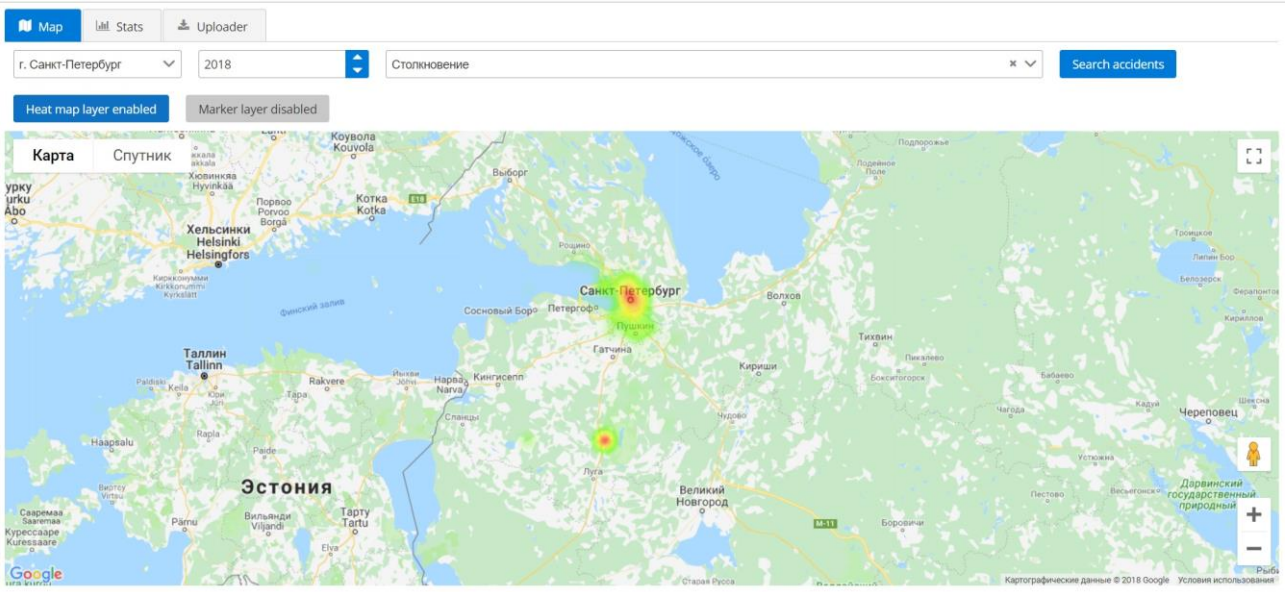

Рисунок 4 – Главная страница. Режим «Heat map»

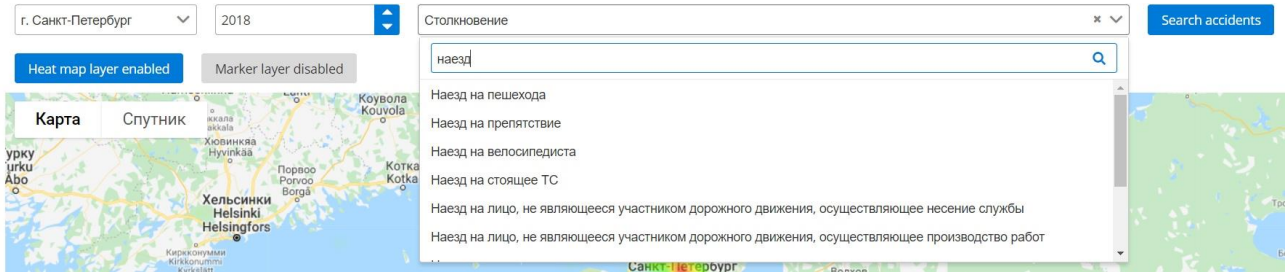

Рисунок 5 – Главная страница. Выбор типа происшествия

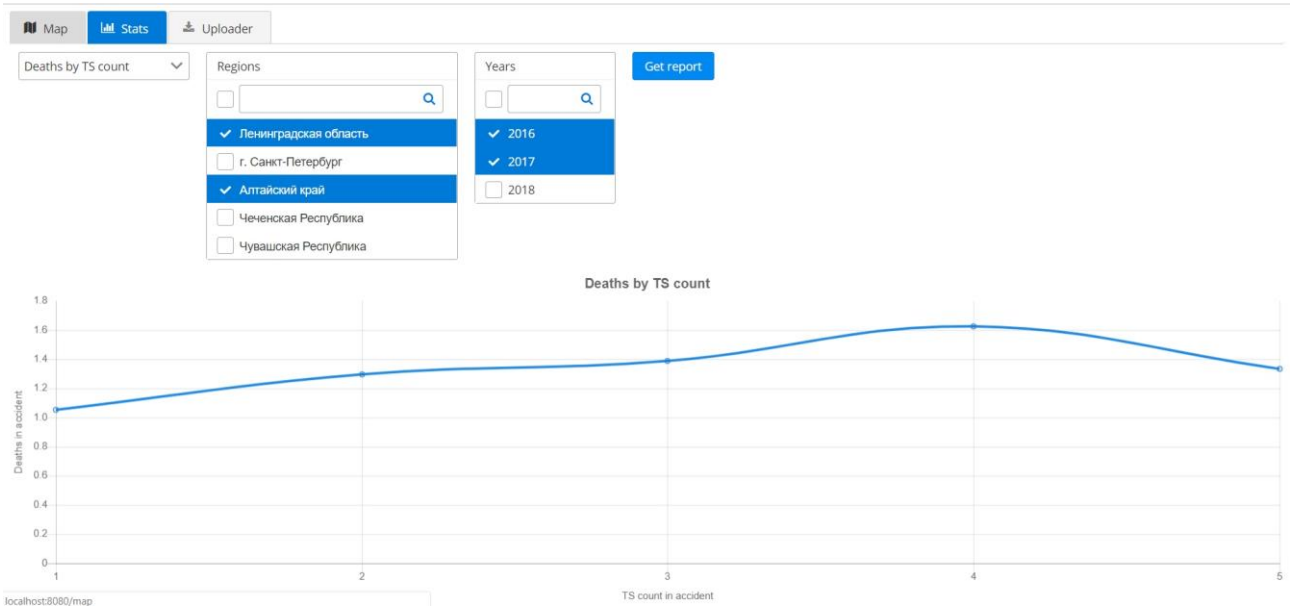

Рисунок 6 – Страница статистики. Смерти в ДТП от количества транспортных средств, участвующих в нем

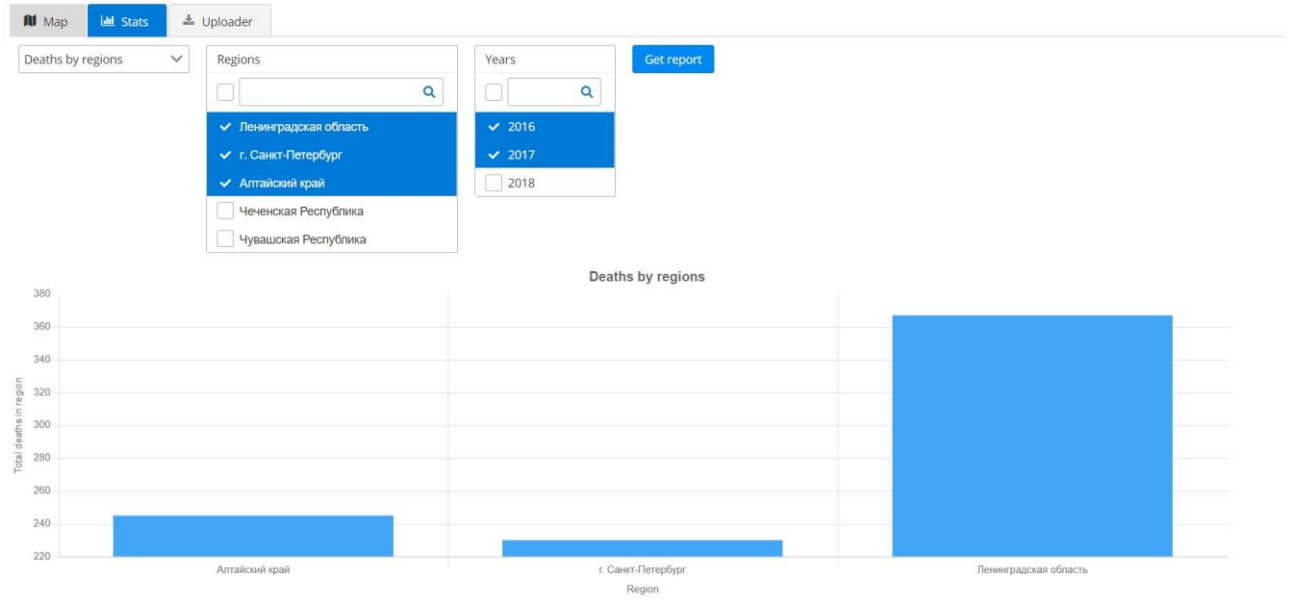

Рисунок 7 – Страница статистики. Смерти в ДТП от региона

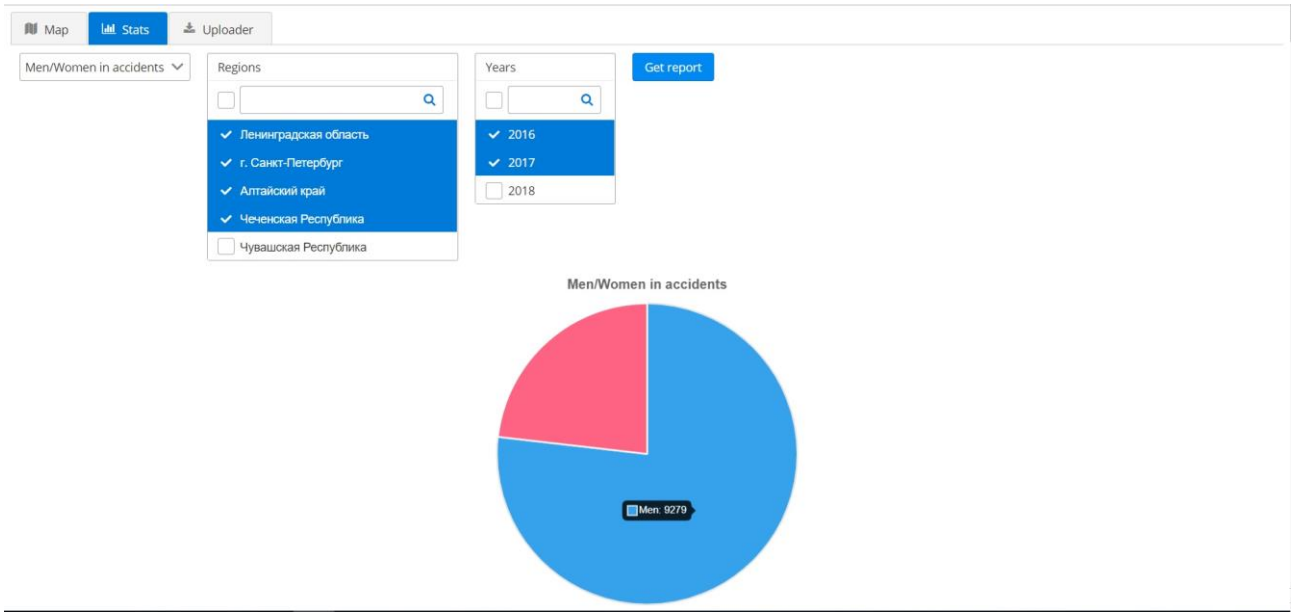

Рисунок 8 – Страница статистики. Соотношение смертей женщин/мужчин

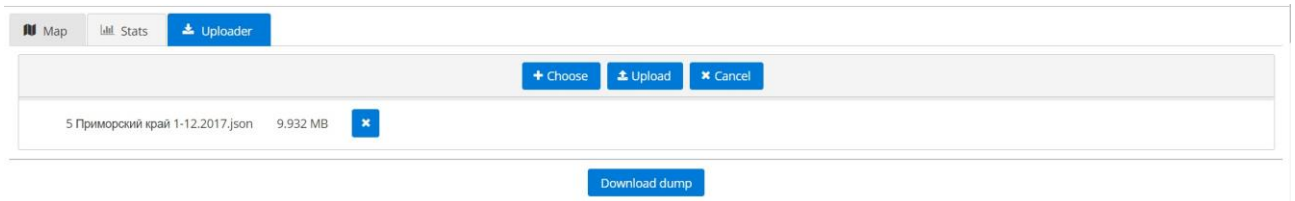

Рисунок 9 – Страница загрузки. Загрузка.

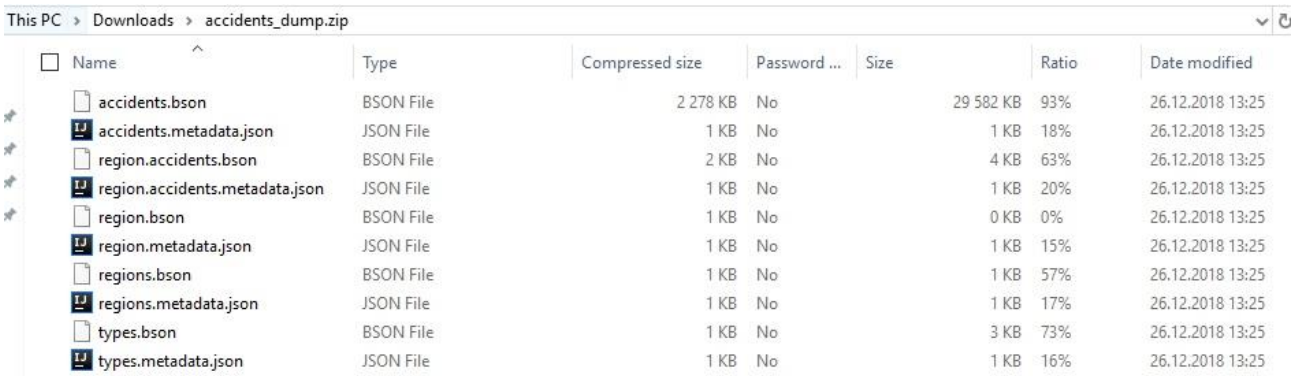

Рисунок 10 – Страница загрузки. Скаченные файлы.