# Веб-парсинг и анализ данных с использованием Python

Разработка приложения, которое включает в себя парсинг веб-страниц, управление базами данных и анализ данных. Проект заключается в получении данных о Covid-19 из Интернета и хранении их в базе данных. Затем данные необходимо проанализировать, создать диаграммы и графики.

Необходимо подготовить письменный отчет, в котором документируется ваш анализ и сделанные выводы.

# Технические требования

Проект должен состоять из модулей, которые при объединении позволяют пользователю собирать / сохранять / анализировать данные.

## Сбор веб-данных

* Соберите данные с <https://www.worldometers.info/coronavirus/>
* Сохраняйте собранные данные в файлах в форматах JSON и / или CSV.
* Вы должны собрать данные по всем 217 странам за последние 6 дней.

Это означает, что вам нужно будет парсить веб-сайт как минимум дважды. При каждом парсинге будут извлекаться и сохраняться данные за текущий день и два предыдущих дня.

Например, если вы начнете собирать данные в среду, у вас будут данные за среду и последние два дня (вторник и понедельник). Тогда вам придется подождать до субботы, чтобы снова начать. Таким образом, у вас есть данные за 6 дней (с понедельника по субботу).

## 1.2. Архивировать данные в базе данных

* Архивируйте собранные в части 1.1. данные и данные, хранящиеся в файле country\_neighbour\_dist\_file.json (он предоставлен) в базе данных.
* Схема для одной из ваших таблиц приведена ниже (также может быть извлечена с веб-сайта). Можете создать и добавить другие поля, если считаете, что это необходимо.

## 1.3. Анализ данных

* После того, как ваши данные будут сохранены в базе данных, вам необходимо выполнить анализ данных, используя pandas для извлеченных (и сохраненных) данных. Процесс выглядит следующим образом:
* Пользователь (я) входит / выбирает страну
* Сравните и проанализируйте эволюцию (ключевых показателей) за 3 дня для рассматриваемой страны.
* сравните извлеченные данные (по ключевым числам) для рассматриваемой страны с соседом с самой длинной границей (всеми его соседями).
* Эта часть должна быть модульной программой, а не скучей операторов Python.
* Программа должна включать функции построения графиков.
* Не требуется, но хорошо, если вы используете графический интерфейс Python

## 1.4. Построение данных

Построить 3 типа данных из базы данных

1. Создайте гистограмму, сравнивающую разницу между текущим числом погибших на 1 миллион населения на сегодняшний день и два дня назад для рассматриваемой страны и не более 5 ее соседей.
2. Создайте линию тренда для сравнения трех данных (сегодня, вчера и два дня назад) с эволюцией для нишевых стран.
3. Создайте гистограмму для нишевых стран, сравнивая два набора данных: Общее количество восстановленных и Общее количество тестов. Убедитесь, что вы сравниваете сопоставимые: создайте общее количество восстановленных по отношению к генеральной совокупности.

* Комментарии к коду - нужны
* Использовать обработку исключений (try / except)
* Разделите свои проекты на модули
* Текстовый файл Readme с кратким описанием содержания проекта и того, как запустить вашу программу

**Заголовки основных таблиц**

Заголовки таблиц приведены в списке:

['rank', 'Country,Other', 'TotalCases', 'NewCases', 'TotalDeaths', 'NewDeaths', 'TotalRecovered', 'NewRecovered', 'ActiveCases', 'Serious,Critical', 'Tot Cases/1M pop', 'Deaths/1M pop', 'TotalTests', 'Tests/1M pop', 'Population', 'Continent', '1 Caseevery X ppl', '1 Deathevery X ppl', '1 Testevery X ppl']

**Страны и соседние с ними страны**

**country\_neighbour\_dist\_file.json** файл

**При проведении тестов лучше очистить данные с веб-сайта (один раз), а затем сохранить их в локальном (html) файле. Последующие тесты на странице должны быть направлены на локальную страницу, а не на веб-сайт.**