**Python. Домашнее задание 3.**

1. **Описание структуры данных**

Основными параметрами погоды являются температура, осадки, ветер и атмосферное давление. Однако в базе данных должна быть возможность беспрепятственного добавления дополнительных параметров прогноза погоды. Возможность хранения информации о погоде не только в одном (основном) месте, но и в других произвольных городах может расширить круг пользователей.

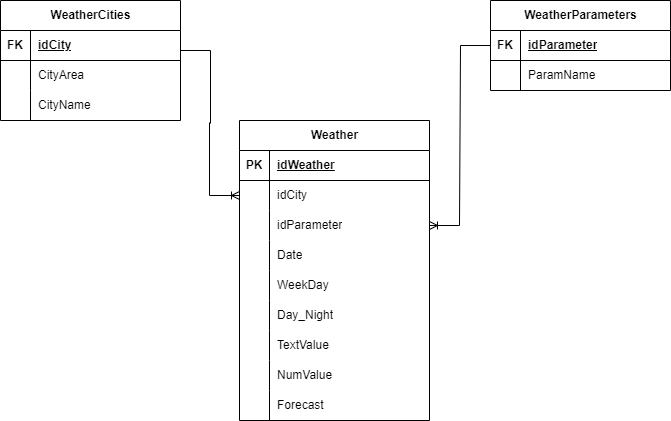
Ввиду того, что точность прогноза погоды обратно пропорциональна сроку прогноза, предлагается ограничить период прогнозирования одной неделей с детализацией по дню/ночи.

После наступления даты прогноза погода данные о прогнозе погоды не удаляются из базы, а помечаются как исторические и могут быть использованы в будущем для анализа и/или построения графиков.

Реализация базы данных по хранению информации о прогнозе погоды предлагается в следующем виде.

База данных состоит из трёх таблиц:

* WeatherCities. Хранятся записи о регионах и городах.
  + idCity – внешний ключ
  + CityArea – облать
  + CityName- название города
* WeatherParameters. Хранятся записи о параметрах прогноза погоды
  + idParameter – внешний ключ
  + ParamName – название параметра
* Weather. Хранятся записи о погоде
  + idCity – ключ к таблице WeatherCities
  + idParameter – ключ к таблице WeatherParameters
  + Date – дата записи о погоде
  + WeekDay – день недели
  + Day\_Night – ночь или день
  + TextValue – текстовое значение параметра погоды (облачность, осадки, направление верта)
  + NumValue – числовое значение параметра погоды (температура, вероятность осадков, скорость ветра, атмосферное давление)
  + Forecast – прогнозное или историческое значение (булево)



1. **Парсер**

Данные о прогнозе погоды получаются с сайта РосГидроМетЦентра ([www.meteoinfo.ru](http://www.meteoinfo.ru)). Для работы парсера используются фреймворки requests, BeautifulSoup, Pandas, string, re. Т.к. часть информации в формате хранения в таблице базы данных находится в параметрах тегов html, наряду в библиотекой BeautifulSoup предлагается использование регулярных выражений.

При помощи парсера есть возможность получить прогноз погоды по городу и региону, который вводится пользователем.

С сайта получаются следующие данные о прогнозе погоды:

* Даты (день, месяц). Прогноз на неделю с детализацией день/ночь.
* Дни недели (понедельник – воскресенье)
* Облачность, осадки (текст)
* Температура (число)
* Вероятность осадков (число)
* Направление ветра (текст)
* Скорость ветра (число)
* Атмосферное давление (число)

Из полученных и обработанных данных формируется словарь Python, из которого создаётся DataFrame Pandas. Данный DataFrame можно использовать для добавления полученных с сайта данных в базу данных.

1. **Ресурсы**

Поскольку потенциальных пользователей системы может быть несколько, то простой вариант базы данных sqlite (возможностей которого вполне хватило бы для реализации идеи) не подойдёт. Необходимо развернуть серверную СУБД (например, PostgreSQL) на корпоративном сервере с выходом в сеть, либо арендовать облачный вариант (например, на pythonanywhere.com). На этом же ресурсе необходимо установить python и запускать скрипт парсера. В зависимости от способа выдачи информации о прогнозе погоды необходимо разработать приложение для работы с пользователем на Flask (если надо выдавать результат в браузер в виде html-страницы) или на FastAPI (если надо передавать информацию, собранную парсером, иному приложению корпорации).