Согласно заданию написана программа program.py, использующая модуль class\_fstr.py в котором описан нужный класс работы с базой данных.

Сама программа делает:

Часть 1:

1. Создание исходной базы данных – запуск программы sql/init\_database.sql . В исходную программу внесено изменение – длина фотографии существенно уменьшена, до первых байт. Видимо в исходных даненых было 2 фото, а после уменьшения осталось только 1, но это и к лучшему – получается тестирование программы на неполностью внесённых в базу данных предыдущим пользователем.

2. Корректировка базы данных, добавление поля Status – запуск программы sql/init\_database\_add\_status\_field.sql

3. Добавление ещё одного пользователя с 1 фотографией (оба с id=2) – запуск программы sql/init\_database\_new\_str.sql

Часть 2:

Формирование данных, для последующего внесения в базу – в качестве нового пользователя используется шаблон от имеющейся записи с id=1, в котором единственное изменение – список собственно фотографий и из названий. Список фоток из 3 штук, все фото идентичны прочитанному из базы с id=1, но разные названия. Количество фоток можно поставить любым.

Можно опробовать вариант сбоя при загрузке фото – в закомментаренной строке стоит тот же список фотографий, но с отсутствием самой фотки2 она не грузится и как следствие не грузится и фото3. В результате, при следующей загрузке данных, для новой порции фоток max(id) имеющихся берётся из базы данных перевалов а не самих фотографий.

Часть 3:

Собственно загрузка данных – запуск процедуры dd.POST(submitData)

Так как в исходном примере было дано время загрузки фотографий позже времени загрузки перевалов, то моя программа делает то же самое. Сперва грузятся данные пользователя, потом длинные данные по фотографиям. С помощью исходных данных в части 2 можно смоделировать вариант недосылки части фоток, анонсированных при исходной передаче основной части данных.

Часть 4:

Проверка того, что в результате получилось в базе данных.

Собственно программа описания класса class\_fstr.py

В ней основные модули, это подсоединение к базе данных, и требуемый по заданию модуль записи данных в базу.

Помимо этого. Для работы основного модуля, и для формирования исходных данных и проверки качества данных после записи, сделаны несколько модулей запуска заранее написанного SQL из файла, или произвольного SQL внутри программы, а так же чтения получившихся после таких вычисления данных – как в виде сырого списка, так и в виде NumPy таблицы.

Помимо этого, программа для подсоединения к базе использует файл .env, который должен содержать данные:

FSTR\_DB\_HOST="localhost"

FSTR\_DB\_PORT=5432

FSTR\_DB\_LOGIN="postgres"

FSTR\_DB\_PASS="..."