Отчет по практической задаче.

Выполнил: Попков И. А.

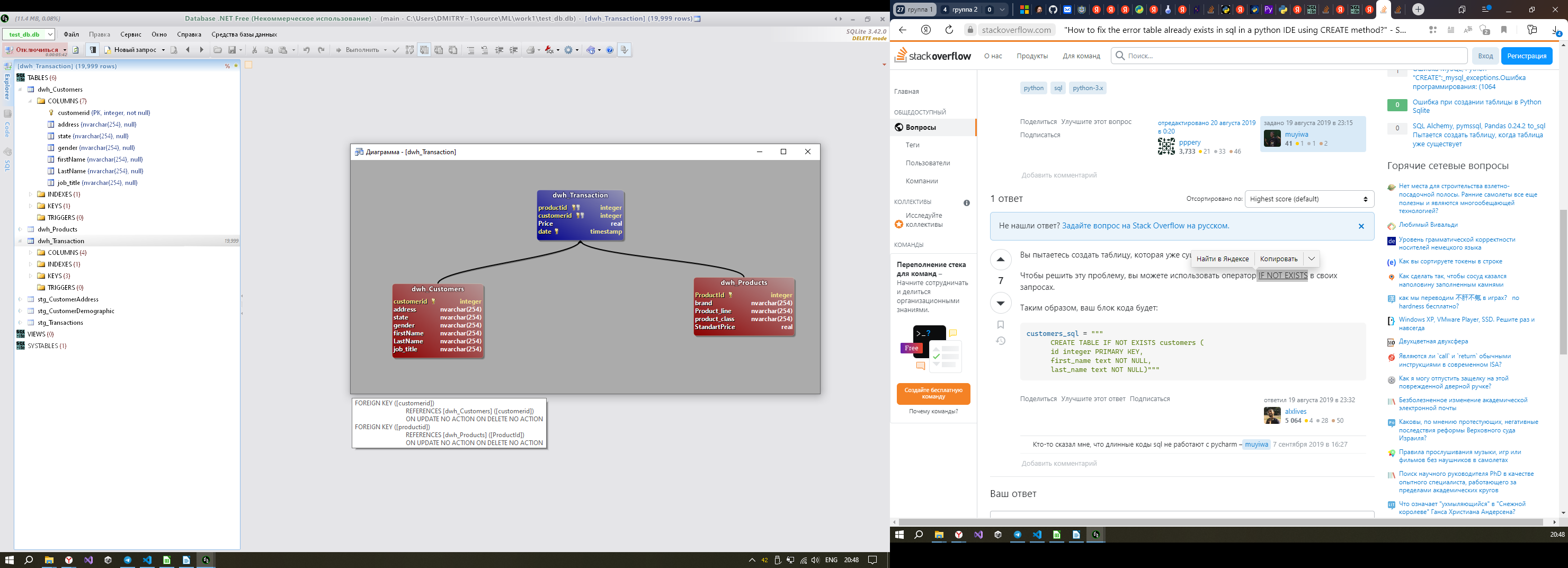
1. Описание кейса

Задачей проекта является разработка системы для выявления наиболее подверженных износу компонентов автомобилей с целью повышения эффективности диагностики и обслуживания транспортных средств. Эта система предназначена для облегчения обнаружения потенциальных дефектов и позволяет оперативно предпринимать меры по их устранению, что способствует обеспечению безопасности и надежности автомобилей.

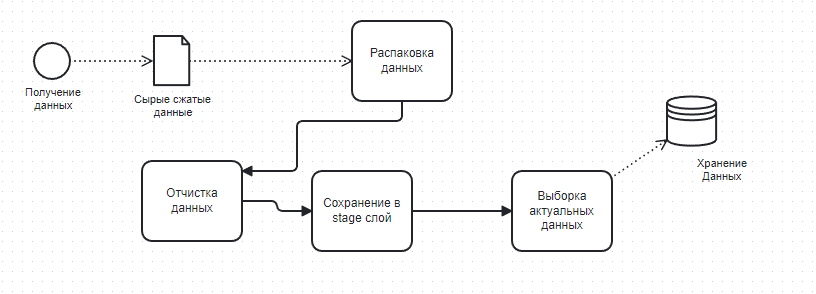
Структура хранения данных включает 3 сущности

* + Товары — таблица содержит данные о проданных деталях.
  + Покупатели — таблица содержит данные о владельцах.
  + Транзакции — таблица содержит сводную информацию о взаимодействиях покупателей и товаров.

1. ER-Диаграмма



1. Бизнес-процесс – BPMN-нотация



1. Архитектура

Архитектура пайплайна представляет собой последовательную обработку исходных файлов формата (excel) с последующей записью данных в базу (SqlLight). Весь код обработки данных написан на языке python и sql. Раз в день запускается скрипт и в базу добавляются новые данные из нового файла.

Сама база данных представляет собой двухуровневое хранение данных

* слой сырых данных
* слой хранилища актуальных данных.

Отношения сущностей в базе представлено в 3НФ.

1. Выбор СУБД

При анализе исходных данных были сделаны выводы, что табличную структуру хранить выгоднее в реляционной базе. Так же, при недостатке вычислительных ресурсов можно воспользоваться денормализаций и привести отношение сущностей к 2НФ. В качестве СУБД выбрана SqlLight как наиболее не требовательная к ресурсам.

1. Выбор СХД

Область временного хранения данных (Staging Area) – предназначена для временного хранения данных, извлеченных из систем-источников; является промежуточным слоем между операционными системами компании и хранилищем данных;

Поэтому, нам удобнее всего использовать файловую СХД. Дальнейшее преобразование и запись данных будет осуществляться в реляционной базе SQLight

1. Перечисление алгоритмов и методов анализа и обработки, предлагаемых к использованию в решении кейса

Для обработки исходных данных используется язык python. Дальнейшая трансформация и обновление данных происходит на стороне базы данных с помощью запросов SQL.

1. Описание модели угроз

Перечень угроз:

* Несанкционированный доступ в базу.

База расположена во внутреннем контуре с ограниченным доступом.

* Права доступа к хранимой информации. Это ограниченный круг лиц, кто может взаимодействовать с этой информацией.
* Защита от внешних атак таких как sqlinjection.

Исходный код модуля.

import sqlite3

from zipfile import ZipFile

import pandas as pd

def ExecuteSqlFromFile(sqlFile, database):

    with open(sqlFile) as sqlQ:

        sqlQueru = sqlQ.read()

        print(sqlQueru)

        with sqlite3.connect(database) as conn:

            conn.executescript(sqlQueru)

            conn.commit()

if \_\_name\_\_ == "\_\_main\_\_":

    dataFile = '99Bikers\_Raw\_data.xlsx'

    archivePath = 'archive.zip'

    resultDataBase = 'test\_db.db'

    archive = ZipFile(archivePath)

    xclfile = archive.open(dataFile)

    ExecuteSqlFromFile('MilovanovSql\\C\_stg\_CustomerAddress.sql',resultDataBase)

    ExecuteSqlFromFile('MilovanovSql\\C\_stg\_CustomerDemographic.sql',resultDataBase)

    ExecuteSqlFromFile('MilovanovSql\\C\_stg\_Transactions.sql',resultDataBase)

    ExecuteSqlFromFile('MilovanovSql\\C\_dwh\_Products.sql',resultDataBase)

    ExecuteSqlFromFile('MilovanovSql\\C\_dwh\_Customers.sql',resultDataBase)

    ExecuteSqlFromFile('MilovanovSql\\C\_dwh\_Transaction.sql',resultDataBase)

    Transactionslist = pd.read\_excel(xclfile,sheet\_name='Transactions')

    with sqlite3.connect(resultDataBase) as conn:

        Transactionslist.to\_sql('stg\_Transactions', conn, if\_exists='append')

    CustomerDemographiclist = pd.read\_excel(xclfile,sheet\_name='CustomerDemographic')

    CustomerDemographiclist = CustomerDemographiclist.drop('default',axis=1)

    with sqlite3.connect(resultDataBase) as conn:

        CustomerDemographiclist.to\_sql('stg\_CustomerDemographic', conn, if\_exists='append')

    CustomerAddresslist = pd.read\_excel(xclfile,sheet\_name='CustomerAddress')

    with sqlite3.connect(resultDataBase) as conn:

        CustomerAddresslist.to\_sql('stg\_CustomerAddress', conn, if\_exists='append')

    ExecuteSqlFromFile('MilovanovSql\\I\_dwh\_Products.sql',resultDataBase)

    ExecuteSqlFromFile('MilovanovSql\\I\_dwh\_Customers.sql',resultDataBase)

    ExecuteSqlFromFile('MilovanovSql\\I\_dwh\_Transaction.sql',resultDataBase)