Домашнее задание
по теме «Работа с РСУБД. PostgreSQL / MySQL + Python. pySpark фреймворк»

Формулировка задания

Установить СУБД и фреймворк pySpark. Решить 2 задачи на python, где необходимо применить навыки работы с реляционными базами данных PostgreSQL или MySQL и библиотеками python psycopg2, pymysql, sqlalchemy, pySpark.
Результирующий код должен быть читаемым, с единой системой отступов и адекватными названиями переменных.

Описание плана работы

Перед выполнением задания необходимо:

1. Установить базу данных PostgreSQL или MySQL Community (под администратором системы):

- для Windows (инсталлятор и документация) : https://postgrespro.ru/windows
- для Linux (инсталлятор и документация) : https://www.postgresql.org/download/
- для Windows (инсталлятор и документация) : https://dev.mysql.com/downloads/installer/
- для Linux (инсталлятор и документация) : https://dev.mysql.com/doc/mysql-installation-excerpt/8.0/en/linux-installation.html

2. Установить и настроить pySpark фреймворк.Обратите внимание, что для PySpark требуется Java 8 (кроме версий до 8u371), 11 или 17 с правильно настроенным JAVA_HOME. Инструкции по установке:

- Официальная документация: https://spark.apache.org/docs/latest/api/python/getting_started/install.html
- Настройка переменных окружения под Windows: https://newtechaudit.ru/instrukcziya-k-primeneniyu-kak-ustanovit-pyspark-na-windows-i-sdelat-v-nyom-word2vec/
- Set JAVA_HOME Variable in Windows, Mac OS X, and Linux: https://www.baeldung.com/java-home-on-windows-mac-os-x-linux

После установки и настройки загрузить библиотеки psycopg2, pymysql, sqlalchemy. Решить 2 задачи на python, где необходимо применить навыки работы с реляционными базами данных PostgreSQL или MySQL и библиотеками python psycopg2, pymysql, sqlalchemy, pySpark.

Желательна реализация в файлах py. Сохранить задачи (сделать коммиты для каждой) в локальном git и опубликовать в удаленном репозитории.

Для отчета по работе выполнить задание в файле.py или .ipynb. Сделать снимки экрана корректного выполнения программы в IDE.

Задача 1 Создание базы данных из файлов и работа с реляционными СУБД

Данные для задачи загрузить из дополнительных материалов или по ссылке: https://www.kaggle.com/datasets/dillonmyrick/bike-store-sample-database

Выбрать СУБД PostgreSQL или MySQL. Загрузить файлы базы данных как таблицы в выбранную СУБД. При загрузке таблиц обратить внимание на отношения между таблицами: первичные ключи (Primary Key) и внешние ключи (Foreign Key).

Отношения между таблицами должны соответствовать Рис. 1.

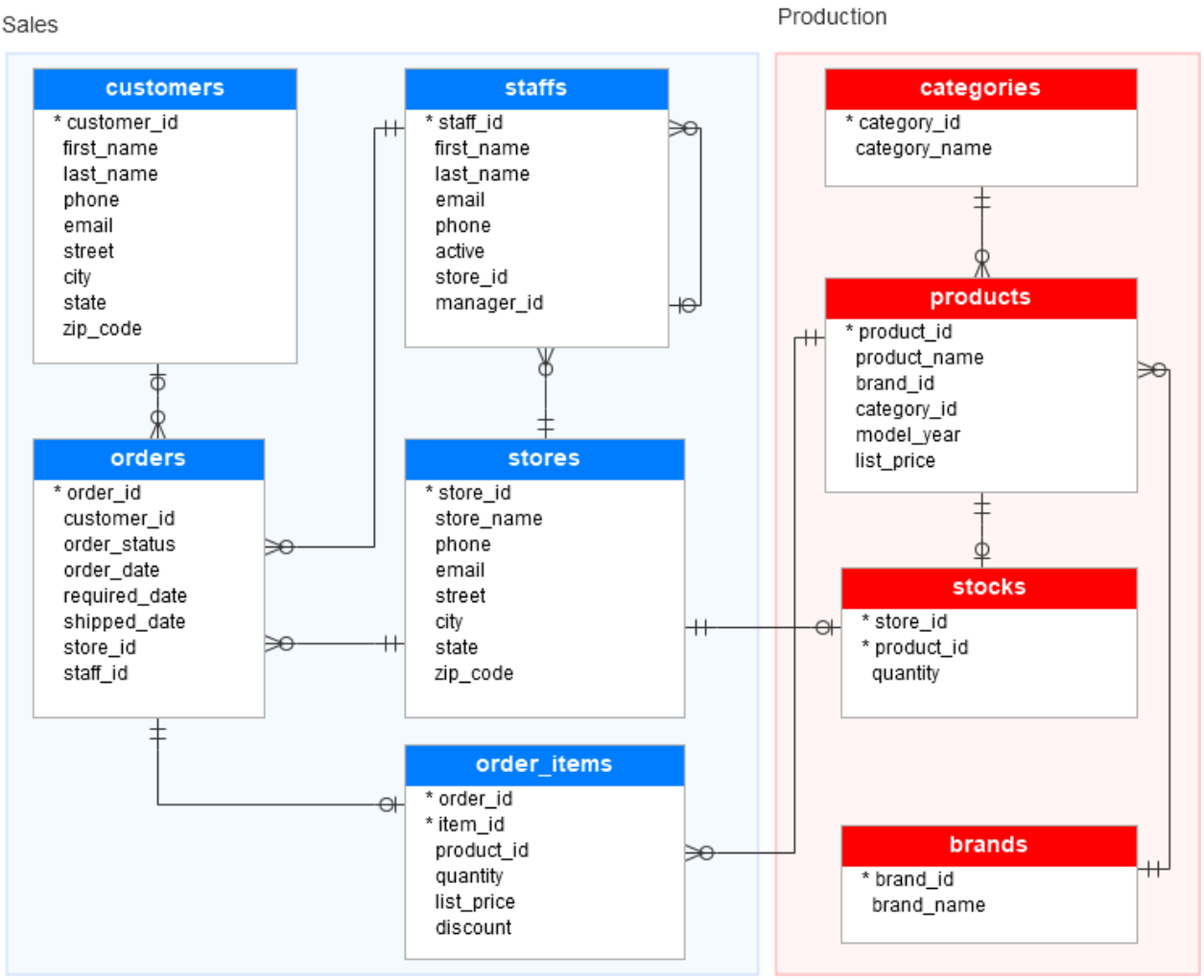

Рис. 1 Схема базы данных магазина Велосипедов

После загрузки данных в таблицу проверить корректность типов данных и значений. Выгрузить dump базы данных:

Для PostgreSQL: https://www.dmosk.ru/miniinstruktions.php?mini=postgresql-dump

Для MySQL: https://www.dmosk.ru/miniinstruktions.php?mini=mysql-dump

Выполнить следующие запросы к созданной базе:

1) Напишите запрос, чтобы получить все названия продуктов и соответствующие им торговые марки (brand).
2) Напишите запрос, чтобы найти всех активных сотрудников и наименования магазинов, в которых они работают.
3) Напишите запрос, чтобы перечислить всех покупателей выбранного магазина с указанием их полных имен, электронной почты и номера телефона.
4) Напишите запрос для подсчета количества продуктов в каждой категории.
5) Напишите запрос, чтобы указать общее количество заказов для каждого клиента.
6) Напишите запрос, в котором будет указана информация о полном имени и общем количестве заказов клиентов, которые хотя бы 1 раз сделали заказ.

Входные данные задачи:

- Таблицы, выгруженные из Bike Store Relational Database
- СУБД для выбора (установленная и настроенная): PostgreSQL или MySQL
- Структура данных: файлы базы данных в формате CSV

Выходные данные задачи:

- Схема данных в формате SQL для таблиц (выгрузить dump из БД): customers, staffs, orders, stores, order_items, categories, products, stocks, brands
- Код на python выполнения работы
- Отчет со снимками экрана хода работы
- SQL-запросы к базе данных 1-6

Задача 2 Работа с pySpark

Выполнить задание из Задачи 1 с помощью фреймворка pySpark. Данные для задачи загрузить из дополнительных материалов или по ссылке: https://www.kaggle.com/datasets/dillonmyrick/bike-store-sample-database

Загрузить данные в pySpark. Проверить корректность типов данных и значений.

Выполнить следующие запросы к данным:

1) Напишите запрос для расчета общего объема продаж по каждому продукту (с учетом количества продукта, его цены по прейскуранту и скидки).
2) Напишите запрос с расчетом количества заказов по каждому статусу заказа.
3) Напишите запрос для расчета общей суммы продаж за каждый месяц.
4) Напишите запрос, чтобы найти топ 5 клиентов, которые потратили больше всего денег.

Входные данные задачи:

- Таблицы, выгруженные из Bike Store Relational Database

- Фреймворк pySpark с настроенными переменными JAVA_HOME, SPARK_HOME

- Структура данных: файлы базы данных в формате CSV

Выходные данные задачи:

- Отчет со снимками экрана хода работы
- Код на python выполнения работы
- SQL-запросы к базе данных 1-4

Перечень необходимых инструментов

- Python
- venv
- psycopg2
- pymysql
- sqlalchemy
- pySpark 3.5.1+
- PostgreSQL 14+
- MySQL Community 8+
- Jupiter Notebook
- IDE VS Code
- GigaIDE

Форма предоставления результата

1. В поле ссылки загрузить ссылку на удаленный репозиторий с доступом для наставника.
2. В поле файла загрузить архив с папкой, в которой разместить отчет со скриншотами по заданиям и решение задач 1-2. Решения должны быть представлены в формате .ipynb или .py.
3. Скрипты запросов к базе данных прописать в коде или вынести в отдельный файл.

Шкала оценивания

- **1.0 – отлично**

Решены 2 задачи. Учтены все требования, код структурирован. Продемонстрирован навык работы с реляционными базами данных. Все запросы написаны на SQL или с применением SqlAlchemy или pySpark и отрабатывают корректно.

- **0.7–0.9 – хорошо**

Решены 2 задачи или 1 задача решена на высоком уровне. Код соответствует большинству требований, но может содержать небольшие ошибки или недочеты. Продемонстрирован навык работы с реляционными базами данных. Все запросы написаны на SQL или с применением SqlAlchemy или pySpark и отрабатывают корректно с небольшими недочетами. Есть к чему стремиться в изучении темы.

- **0.5–0.6 – удовлетворительно**

Решены 2 задачи или 1 задача с множеством замечаний. Код соответствует большинству требований, но может содержать значительные ошибки или недочеты. Требуется дополнительное изучение темы и исправление программы.

- **Менее 0.5 – задание не выполнено**

Задание выполнено на очень низком уровне или решено 0 задач. Требуется дополнительное изучение темы и решение заданий для практического изучения темы.