Курсовой проект для студентов учебного центра Neoflex по направлению «Data-Engineering (BDS)»

Задание №2

В предыдущем задании вы занимались восстановлением потерянных данных и сломавшегося функционала расчёта витрины 101-й формы в реляционной БД (Oracle / PostgreSQL).

Во время общения с архитектором вы узнаёте, что банк решил мигрировать свой DWH в Hadoop, а ETL-процессы перевести на Spark. В виду этого необходимо настроить среду для исследования возможностей новых инструментов и того как в них можно переложить существующий ETL-функционал.

**Задача 2.2**

Вы проверили работоспособность нового ETL-инструмента (Spark) и теперь с его помощью необходимо разработать один из типовых процессов для банковского хранилища - построение зеркал.

Зеркало - это таблица в сыром или детальном слое, в которой содержится последнее (актуальное) состояние каждой еë сущности (записи/строки). То есть, если 3 раза запускался процесс загрузки, который выделял дельту и в каждой такой дельте была одна и так же сущность (какой-то один счëт), у которой менялись вторичные параметры - то в зеркале должна остаться одна строка с последним состоянием, каждого из вторичных параметров.

Для того что бы такой процесс был возможным - у таблицы обязательно должен быть уникальный ключ (id или номер счëта, например).

Данный механизм необходимо сделать универсальным - в него должны передаваться: путь где хранятся дельты, наименование таблицы и список полей, являющимся уникальным ключём. Согласовано, что дельты внутри указанного пути (основной директории) всегда хранятся в виде вложенных директорий с наименованием в порядке возрастания (например: 1000, 1001…..) – это тоже нужно заложить в универсальный механизм.

Для демонстрации работоспособности вашего приложения нужно будет обработать 4 дельты с изменениями некоторых параметров в справочнике счетов. Каждая дельта - это csv- файл внутри директории с наименованием id- дельты (id сессии загрузки).

Итоговый результат должен быть сохранëн в csv- файл в директории «mirr\_md\_account\_d».

Важно добавить в этот механизм процесс логирования, что бы не обрабатывать повторно, ранее загруженные дельты - это будет экономить время и ресурсы сервера.

Логи могут быть в виде отдельной директории с файлами того формата, что вы сочтëте для себя удобным (csv / parquet / json) или в реляционной БД (во вложении файл со скриптами установки PostgresSQL в Ubuntu). В них должна содержатся информация с датой-время старта и завершения загрузки каждой из дельт (то есть запоминать еë id). Так же в логах должно сохраняться наименование обновляемой таблицы.

Примечания:

* Полезная статья о работе с дельтами в Oracle, но сам процесс можно считать универсальным:  
  <https://danischnider.wordpress.com/2016/10/08/delta-detection-in-oracle-sql/>
* Если вы принципиально заходите применять в Spark такие операции, как MERGE и UPDATE, то вам понадобиться самостоятельно изучить работы с форматом «delta» и методику работы с данными через «deltaTable». Полезные ссылки:
  + <https://docs.delta.io/latest/api/python/index.html#delta.tables.DeltaTable.merge>
  + <https://www.confessionsofadataguy.com/introduction-to-delta-lake-on-apache-spark-for-data-engineers/>

Требования к демонстрации работы:

* Покажите как вы в приложении указываете путь до директрии с дельтами, наименование таблицы и список полей являющихся ключами (для таблицы счетов, оно одно - id счëта) ;
* Запустите приложение и покажите как выводится (через df.show()) промежуточное состояние зеркала и конечное. Откройте финальный csv-файл, который сохранился в директорию "md\_account";
* Покажите какая информация записалась в логи;
* Создайте сами ещё одну дельту (то есть ещё одну директорию с ещё одним csv-файликом, где вы укажите существующий счёт, но измените его вторичный параметр – client\_id / branch\_id / open\_in\_internet);
* Запустите ваше Spark-приложение повторно и покажите, что, опираясь на логи оно не будет повторно грузить все дельты и применит только вашу новую дельту;
* Покажите финальный csv-файл с обновлённым значением у счёта;
* Записать видео с экрана компьютера, в котором вы демонстрируйте и комментируете в слух, то что вы делаете / уже разработали;
* Это видео загрузите к себе на облако (гугл-диск, яндекс-диск и т.д.) и предоставьте доступ по ссылке;
* Приложите в домашнее задание внутри архива текстовый файл с ссылкой на ваше видео и все скрипты решения.