GeekBrains

BigData. Введение в экосистему Hadoop.

Урок 4. Hive

На уроке рассматриваем основные подход в работе с данными Hadoop через SQL. Что такое Hive и основные его принципы.

Оглавление

Теоретическая часть

Hive

Apache Hive - программное обеспечение, используемое хранилищами данных. Облегчает использование запросов и управление большими объемами данных, находящихся в распределенных хранилищах. Hive предоставляет механизм проектирования структур для этих данных и позволяет создавать запросы с использованием SQL -подобного языка, называемым HiveQL. В то же время этот язык позволяет программистам использовать их собственные запросы, когда неудобно или неэффективно использовать логику в HiveQL.

Возможности

- Работа с данными используя SQL-подобный язык запросов;
- Поддержка различных форматов хранения данных;
- Работа напрямую с HDFS и Apache HBase;
- Выполнение запросов через Apache Tez, Apache Spark или MapReduce.

HiveQL

Apache Hive поддерживает язык запросов Hive Query Language, который основан на языке SQL, но не имеет полной поддержки стандарта SQL-92. HiveQL имеет функции для работы с форматами XML и JSON, поддержку нескалярных типов данных, таких как массивы, структуры, ассоциативные массивы, поддерживает широкий набор агрегирующих функций, определяемые пользователем функции (User Defined Functions), блокировки.

Hive MetaStore

Хранит метаданные для таблиц Hive — схему на чтение (schema-on-read) , расположение, информацию о столбцах в таблице, типы данных, ACL и тд.

Практическая часть

Заходим в hive, для этого вводим в консоли команду:

```
hive
```

Далее создаём таблицу скриптом:

```
create table lesson4.customers
(customer_id string,
customer_unique_id string,
customer_zip_code_prefix string,
customer_city string,
customer_state string)
ROW FORMAT serde 'org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe'
WITH SERDEPROPERTIES ('field.delim' = ',')
location "/user/hive/warehouse/lesson4.db/customers"
tblproperties ("skip.header.line.count"="1");
```

Важно: в location указываем путь до папки, а не до конкретного файла. В параметре field.delim указывается разделитель, а в ROW FORMAT формат файла.

Далее выполняем команду:

```
drop table lesson4.customers;
```

И познаем разницу между managed tables и external table, увидев что все файлы в HDFS удалились.

Для создания таблицы с партициорованием используем скрипт:

```
create external table lesson4_10.customers
(customer_id string,
customer_unique_id string,
customer_zip_code_prefix string,
customer_city string,
```

```
customer_state string)

partitioned by ( p_date string )

ROW FORMAT serde 'org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe'

WITH SERDEPROPERTIES ('field.delim' = ',')

location "/user/hive/warehouse/lesson4_10.db/customers"

tblproperties ("skip.header.line.count"="1");
```

Добавляется partitioned by (p_date string) и вложенные папки с названием p_date=YYYYMMDD.

Пытаемся изменить формат данных и убеждаемся, что сами данные не изменяются:

```
ALTER TABLE lesson4.customers change customer_id customer_id int;
```

Пробуем команду, которая обновляется meta-store:

```
msck REPAIR TABLE lesson4.customers;
```

Домашнее задание

Скачать любой датасет из списка ниже с сайта Kaggle.com (достаточно большой)

Загрузить этот датасет в HDFS в свою домашнюю папку

Создать собственную базу данных в HIVE

Создать таблицы внутри базы данных с использованием всех загруженных файлов. Один файл – одна таблица.

Сделать любой отчет по загруженным данным используя групповые и агрегатные функции.

Сделать любой отчет по загруженным данным используя JOIN.

Задачи со * предназначены для продвинутых учеников, которым мало сделать обычное ДЗ.

Используемая литература

Для подготовки данного методического пособия были использованы следующие ресурсы:

1. https://ru.bmstu.wiki/Apache_Hive
2. https://ru.wikipedia.org/wiki/Apache_Hive