BIGDATA TEAM

HW #04: Hive

Deadline: 15.07.2019, 08:00

---

1. Описание задания.

В данном ДЗ нужно решить **3 задачи**. Решение надо выполнить с помощью Hive. Задача 1 - общая для всех. В задачах 2,3 вариант определяются ID (см. таблицу с оценками) с помощью следующей формулы:

```
ID % 3 + 1
```

2. Критерии оценивания.

Балл за задачу складывается из:

- **60%** - правильное решение задачи

- **20%** - поддерживаемость и читаемость кода (Clean Code, см. например Google Python Style Guide)
- **20%** - эффективность решения

Штрафы:
- **10%** за несоответствие правилам оформления задания
- **30%** за просрочку дедлайн

Веса задач:
1. 33.3%
2. 33.3%
3. 33.3%

3. Описание данных

3.1. Логи запросов пользователей новостных сайтов.

user_logs:
- Путь на кластере: полный датасет - **`/data/user_logs/user_logs_M`**
- Семпл (для тестирования): **`/data/user_logs/user_logs_S`**
- Формат: текст
- В каждой строке находятся следующие поля, разделенные знаком табуляции (**иногда не одним**):
    1. STRING - ip-адрес, с которого пришел запрос,
    2. STRING (TIMESTAMP) - время запроса,
    3. STRING - пришедший с ip-адреса http-запрос,
    4. INT - размер переданной клиенту страницы в байтах,
    5. INT - http-статус запроса.
    6. STRING - User Agent, информация о клиентском приложении, с которого осуществлялся запрос на сервер, в том числе, информация о браузере.

Пример:

```
135.124.143.193                20150601013300
http://newsru.com/4712386      235   412   Firefox/5.0 (compatible; MSIE
9.0; Windows NT 6.1; Win64; x64; Trident/5.0)n
```

Важно:
- разделитель между IP и временем запроса состоит из 3 символов табуляции;
- Будем считать, что браузере содержится в начале 6-ого поля лога - символы с нулевой позиции до позиции первого пробельного символа.
    - пример User Agent:

- `Chrome/5.0 (compatible; MSIE 9.0; Windows NT 8.0; WOW64; Trident/5.0; .NET CLR 2.7.40781; .NET4.0E; en-SG)`
- тогда браузером будет: `Chrome/5.0`

Подсказка:

- поскольку нас не интересует оставшаяся часть User Agent, то получить тип браузера пользователя можно с помощью правильного регулярного выражения в период чтения logs_raw.

3.2. Информация о пользователях.

user_data:

- Путь на кластере: полный датасет - **/data/user_logs/user_data_M**
- Семпл (для тестирования): **/data/user_logs/user_data_S**
- Формат: текст
- В каждой строке находятся следующие поля, разделенные знаком табуляции:
  1. STRING - IP-адрес, с которого пользователь выходит в интернет;
  2. STRING - браузер пользователя;
  3. STRING - пол (male / female);
  4. INT - возраст.

Пример:

```
197.72.248.141    Opera/12.0  male  30
```

3.3. Геобаза - информация о соответствии ip-адресов регионам.

ip_data:

- Путь на кластере: полный датасет - **/data/user_logs/ip_data_M**
- Семпл (для тестирования): **/data/user_logs/ip_data_S**
- Формат: текст
- В каждой строке находятся следующие поля, разделенные знаком табуляции:
  1. STRING - IP-адрес;
  2. STRING - регион.

Пример:

```
33.49.147.163     Kemerovo Oblast
197.72.248.141    Belgorod Oblast
135.124.143.193   Krasnoyarsk Krai
...
```

4. Задача #1[1]: создание таблиц в Hive.

Создайте внешние (EXTERNAL) таблицы по исходным данным:

1. **logs_raw** - логи пользователей;
2. **users** - таблица с информацией о пользователях;
3. **ip_regions** - таблица с IP и регионами;

Из таблицы логов перенесите данные в другую таблицу, партицированную по датам – одна партиция на каждый день:

4. **logs** - партиционированная таблица с логами.

Условия:

1. Название базы данных должно иметь вид **mf_<username>**, например **mf_dral**;
2. Таблицы должны называться ровно так, как указано в описании задачи.
3. Сериализация и десериализация данных должна осуществляться с использованием регулярных выражений, см.:
    - **org.apache.hadoop.hive.serde2.RegexSerDe**

Проверить правильность создания таблиц можно с помощью простых SELECT-запросов:

```
SELECT * FROM <table> LIMIT 10
```

Рекомендации:

- предлагается начать с простых таблиц, а потом двигаться к сложным, например: ip_regions → users → logs_raw → logs;
- для создания таблиц ip_regions и users рекомендуется воспользоваться следующей конструкцией:
    - `ROW FORMAT delimited`
    - Документация по полям, разделяющим колонки доступны в документации по адресу. Вам необходимо найти способ указать разделить `<tab>` вместо `<space>` (пробела).

Подсказки как сделать партиционированную таблицу logs:

1. Чтобы выделить день в формате “YYYYMMDD” достаточно воспользоваться функцией для работы со строками SUBSTR.
2. Посчитайте, сколько уникальных (DISTINCT) дней в “сырых” логах (logs_raw). Это число должно получиться более 100 на датасете размера “_M”.

[1] Внутренний ID задачи (для проверяющих) - 411

3. Используйте это число, чтобы задать переменную окружения Hive, которая позволит запустить динамическое создание партиций[2]:
   - `set hive.exec.max.dynamic.partitions.pernode=***;`
4. После этого можно написать запрос:
   - `INSERT OVERWRITE TABLE logs PARTITION(date) SELECT ... FROM logs_raw`

На партиционированной таблице `logs` и нужно будет выполнять запросы в следующих задачах.

5.1. Задача #2 (вариант 1)[3]: горячий денек.

Напишите запрос, который считает какое количество посещений новостных сайтов было в разрезе дней. Полученные результаты отсортируйте (**ORDER BY**) по убыванию популярности. На экран выведите TOP-10 самых “горячих” дней с точки зрения нагрузки на инфраструктуру новостных сервисов.

Пример вывода:

```
20140308 96
20140409 94
20140318 89
...
```

5.2. Задача #2 (вариант 2)[4]: поиск аномалий в работе HTTP.

Напишите запрос, который считает какое количество **различных** HTTP-статусов возвращали новостные сайтов в разрезе дней. Полученные результаты отсортируйте (**ORDER BY**) по убыванию популярности. На экран выведите TOP-10 самых “подозрительных” дней.

Пример вывода:

```
20140207 46
```

[2] Подробную документацию по dynamic partitioning см. здесь: https://cwiki.apache.org/confluence/display/Hive/LanguageManual+DML#LanguageManualDML-DynamicPartitionInserts

[3] Внутренний ID задачи (для проверяющих) - 421

[4] Внутренний ID задачи (для проверяющих) - 422

```
20140126 44
20140112 42
...
```

5.3. Задача #2 (вариант 3)[5]: Market Research.

Постройте гистограмму использования браузеров пользователями новостных сайтов на основе таблицы logs. Полученные результаты отсортируйте (**ORDER BY**) по убыванию популярности. На экран выведите TOP-10 самых популярных браузеров.

Пример вывода:

```
Firefox/5.0 25
Opera/5.0 21
...
```

6.1. Задача #3 (вариант 1)[6]: Male vs Female popularity per region.

Напишите запрос, считающий популярность новостных ресурсов в регионах среди мужчин и женщин. Выведите произвольные 10 записей (LIMIT 10) в формате:

- регион <tab> посещаемость мужчинами <tab> посещаемость женщинами

Пример вывода:

```
Tver 66968157 29097223
Voronezh 60445347 26333509
...
```

Подсказки:

- для решения задачи рекомендуется воспользоваться оператором IF, примеры его использования см. в официальной документации Hive (см. здесь) или в слайдах занятия.
- для решения этой задачи нужно сделать join трех таблиц. Сложность заключается в том, что: по умолчанию, из-за небольшого объема данных Hive преобразует этот запрос в Map-Side Join, НО у него **может** не хватит оперативной памяти, чтобы выполнить эту задачу, поэтому:

[5] Внутренний ID задачи (для проверяющих) - 423
[6] Внутренний ID задачи (для проверяющих) - 441

1. Нужно отключить авто-конвертацию join в оптимизированный вид join. см. опцию:
   - `set hive.auto.convert.join`
2. По умолчанию, опять же из-за небольшого объема данных, Hive попробует запустить все вычисления в рамках Reduce-Side Join на одном редьюсере. Чтобы этого избежать необходимо изменить число редьюсеров с помощью флага:
   - `set mapreduce.job.reduces` (укажите например 8 редьюсеров, чтобы дождаться результатов вычислений в течение **30 минут**).

6.2. Задача #3 (вариант 2)[7]: identify browser sex.

Напишите запрос, который считает число употреблений браузера мужчинами и женщинами. Группируем браузеры из таблицы **`logs`**. Выведите **произвольные** 10 записей (LIMIT 10) в формате:

- браузер `<tab>` посещаемость мужчинами `<tab>` посещаемость женщинами

Пример вывода:

```
Firefox/5.0 1419872 621124
Opera/5.0 1426114 623333
...
```

Подсказки:

- для решения задачи рекомендуется воспользоваться оператором IF, примеры его использования см. в официальной документации Hive (см. здесь) или в слайдах занятия.
- для решения этой задачи нужно сделать join двух таблиц. Сложность заключается в том, что: по умолчанию, из-за небольшого объема данных Hive преобразует этот запрос в Map-Side Join, НО у него **может** не хватит оперативной памяти, чтобы выполнить эту задачу, поэтому:
  1. Нужно отключить авто-конвертацию join в оптимизированный вид join. см. опцию:
     - `set hive.auto.convert.join`
  2. По умолчанию, опять же из-за небольшого объема данных, Hive попробует запустить все вычисления на одном редьюсере (если вы воспользовались первой рекомендацией, то это будет Reduce-Side Join). Чтобы этого избежать необходимо изменить число редьюсеров с помощью флага:

[7] Внутренний ID задачи (для проверяющих) - 442

- set mapreduce.job.reduces

6.3. Задача #3 (вариант 3)[8]: identify gender-related HTTP errors.

Напишите запрос, который считает сколько раз мужчины и женщины сталкивались с разными кодами возврата HTTP. Выведите **произвольные** 10 записей (LIMIT 10) в формате:

- HTTP-код `<tab>` встретилось у мужчин `<tab>` встретилось у женщин

Пример вывода:

```
511 90675090 39459549
412 87782696 38146030
...
```

Важно: не стоит беспокоиться, что в выводе будут **несуществующие** HTTP-коды возврата, это результат обфусцирования данных.

Подсказки:

- для решения задачи рекомендуется воспользоваться оператором IF, примеры его использования см. в официальной документации Hive (см. здесь) или в слайдах занятия.
- для решения этой задачи нужно сделать join двух таблиц. Сложность заключается в том, что: по умолчанию, из-за небольшого объема данных Hive преобразует этот запрос в Map-Side Join, НО у него **может** не хватит оперативной памяти, чтобы выполнить эту задачу, поэтому:
  1. Нужно отключить авто-конвертацию join в оптимизированный вид join. см. опцию:
     - `set hive.auto.convert.join`
  2. По умолчанию, опять же из-за небольшого объема данных, Hive попробует запустить все вычисления на одном редьюсере (если вы воспользовались первой рекомендацией, то это будет Reduce-Side Join). Чтобы этого избежать необходимо изменить число редьюсеров с помощью флага:
     - set mapreduce.job.reduces

7. Сроки сдачи и правила оформления задания.

[8] Внутренний ID задачи (для проверяющих) - 443

Deadline: 15.07.2019, 08:00

Оформление задания:

- Код задания (Short name): **HW4:Hive**.
- Решения задач должны содержаться в одной папке.
- HQL-скрипты для запуска решений следует называть по номеру задачи и варианта **task_<#task>_<#variant>.hql**:
    - например решение задачи #2 для 3го варианта должно называться task_2_3.hql и его можно запустить с помощью команды:
        - `hive -f task_2_3.hql`
    - скрипт выводит на экран (STDOUT) указанное в задание число строк в нужном формате
- **Вывод STDOUT задач просьба написать в соответствующие файлы, например task_2_3.out.**
- Выполненное ДЗ запакуйте в архив **MF2019Q2_<фамилия>_HW#.zip**, к примеру -- **MF2019Q2_Ivanov_HW3.zip**. Например, ваше решение лежит в папке my_solution_folder, тогда чтобы на Linux и Mac OS создать архив под названием hw.zip и пожать его с помощью zip выполните команду[9]:
    - zip -r hw.zip my_solution_folder/

  На Windows 7/8/10: необходимо нажать правую кнопку мыши на директорию my_solution_folder/, выбрать в открывшемся меню "Отправить >", затем "Сжатая ZIP-папка". Теперь можно переименовать архив.
- Присылайте выполненное задание на почту bigdata_mf2019q2@bigdatateam.org с темой письма “Short name. ФИО.”. Например: “**HW4:Hive**. Иванов Иван Иванович.”
- Перед отправкой письма, оставьте, пожалуйста, отзыв о домашнем задании по сссылке: http://rebrand.ly/mf2019q2_feedback_hw04. Это позволит нам скорректировать учебную нагрузку по следующим заданиям (в зависимости от того, сколько часов уходит на решение ДЗ), а также ответить на интересующие вопросы.

Любые вопросы / комментарии / предложения можно писать:

- в телеграм-канале: http://rebrand.ly/mf2019q2_telegram_join
- На почту: bigdata_mf2019q2@bigdatateam.org

Всем удачи!

8. Дорешка.

[9] Флаг -r значит, что будет совершен рекурсивный обход по структуре директории

Решения после получения фидбека на решение ДЗ можно улучшить. Разрешается одна досылка в течение 1й недели после окончания дедлайна за ДЗ. Соответственно, фидбек за дорешенные ДЗ вы получите в течение 24 часов после окончания deadline следующего ДЗ.

Дорешивать неработающие задания - нельзя. Это позволит исправить дисбаланс между

присланными **работающими** заданиями **после** deadline

VS

присланными **НЕработающими** заданиями **до** deadline