Скачать любой датасет из списка ниже

https://www.kaggle.com/shuyangli94/food-com-recipes-and-user-interactions <https://www.kaggle.com/datasnaek/youtube-new>

<https://www.kaggle.com/akhilv11/border-crossing-entry-data>

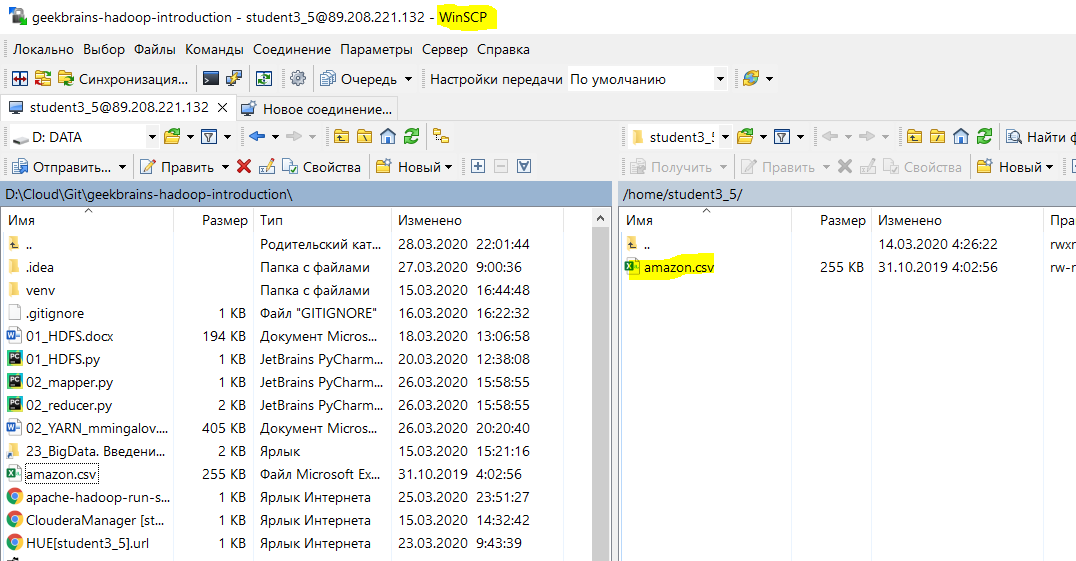
https://www.kaggle.com/tristan581/17k-apple-app-store-strategy-games https://www.kaggle.com/gustavomodelli/forest-fires-in-brazil

Качаем amazon.csv

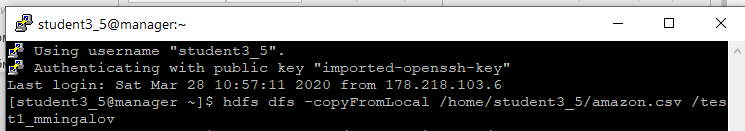
Загрузить этот датасет в HDFS в свою домашнюю папку

Сперва закачиваем на локальную машину кластера.

Для этой операции используем WinSCP.



Далее копируем в HDFS через Putty

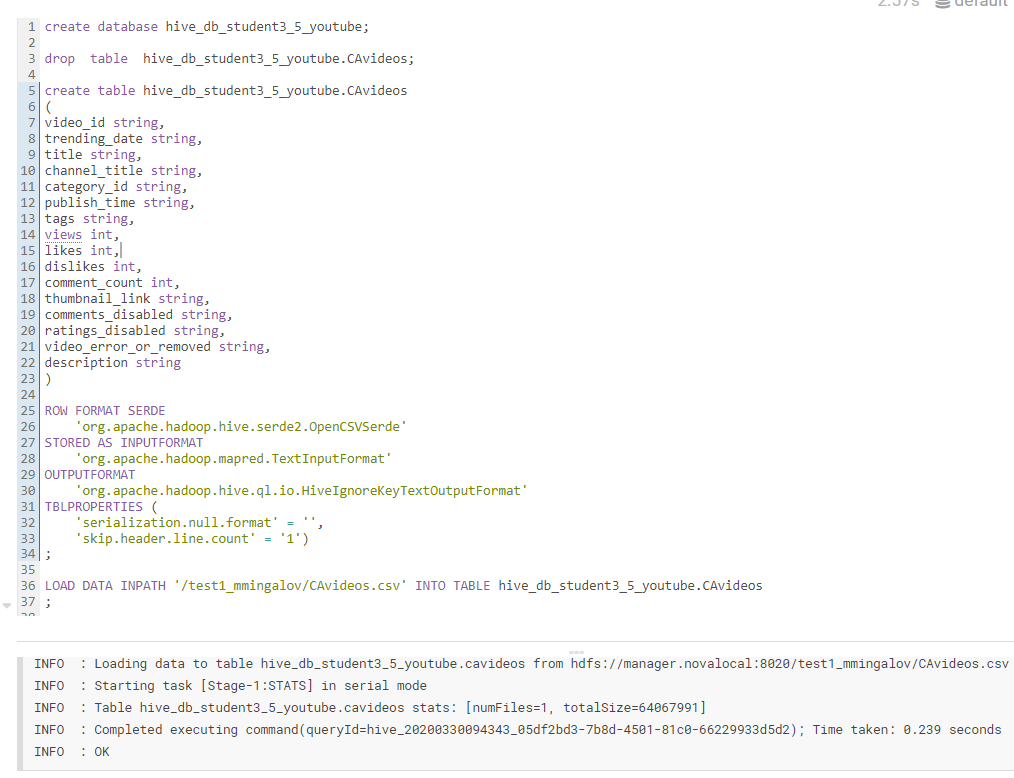


Повторим те же действия для 4-х датасетов YOUTUBE-NEW (CA, GB, RU, US)

Создать собственную базу данных в HIVE. Желательно чтобы имя базы содержало номер вашего пользователя.

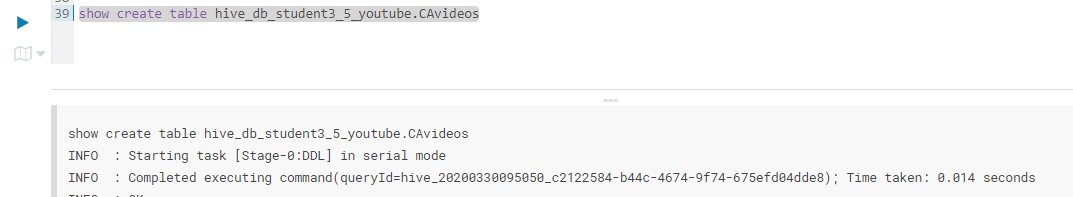
Открываем HUE

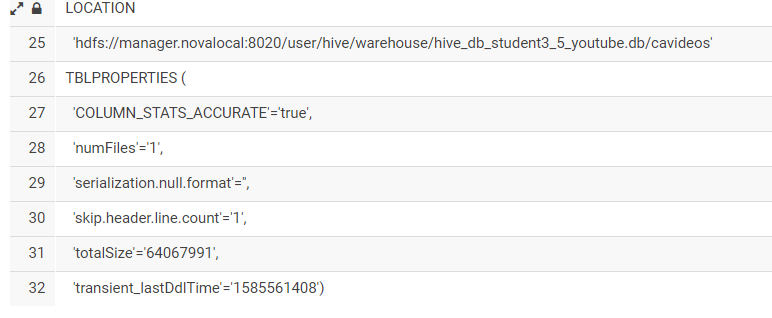
Создадим БД и простую таблицу в ней, положив в нее данные из датасета CSV



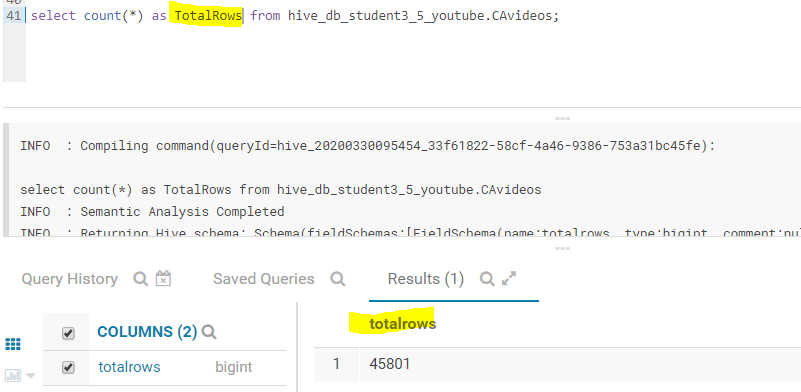
Посмотрим как в действительности была создана таблица:

show create table hive\_db\_student3\_5\_youtube.CAvideos





Подсчитаем кол-во строк в ней:



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

Создадим EXTERNAL таблицы.

Пример конструкции на создание

-- EXTERNAL

create external table hive\_db\_student3\_5\_youtube.GBvideos\_ext

(

video\_id string,

trending\_date string,

title string,

channel\_title string,

category\_id string,

publish\_time string,

tags string,

views int,

likes int,

dislikes int,

comment\_count int,

thumbnail\_link string,

comments\_disabled string,

ratings\_disabled string,

video\_error\_or\_removed string,

description string

)

ROW FORMAT SERDE

'org.apache.hadoop.hive.serde2.OpenCSVSerde'

STORED AS INPUTFORMAT

'org.apache.hadoop.mapred.TextInputFormat'

OUTPUTFORMAT

'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'

TBLPROPERTIES (

'serialization.null.format' = '',

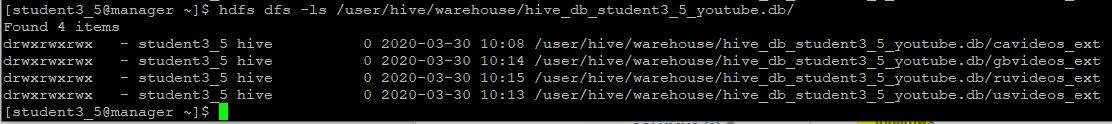
'skip.header.line.count' = '1')

;

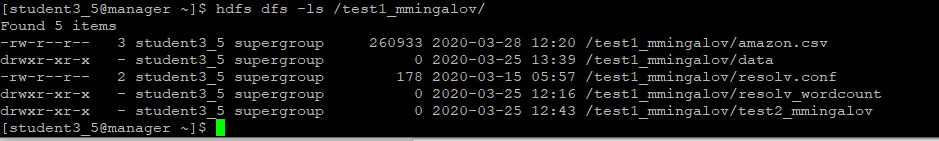
LOAD DATA INPATH '/test1\_mmingalov/GBvideos.csv' INTO TABLE hive\_db\_student3\_5\_youtube.GBvideos\_ext

;

Проверим содержимое папки, где они были созданы:

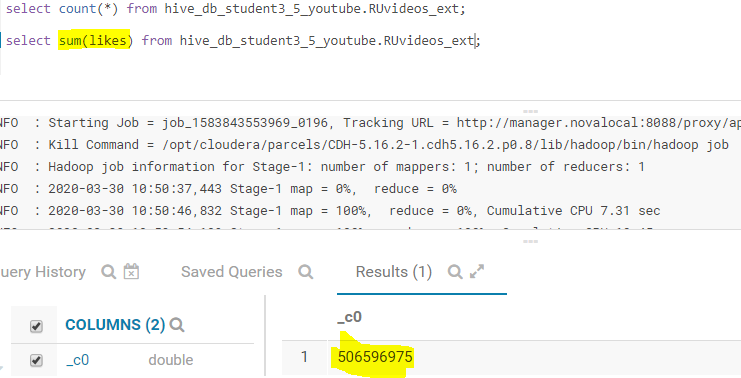


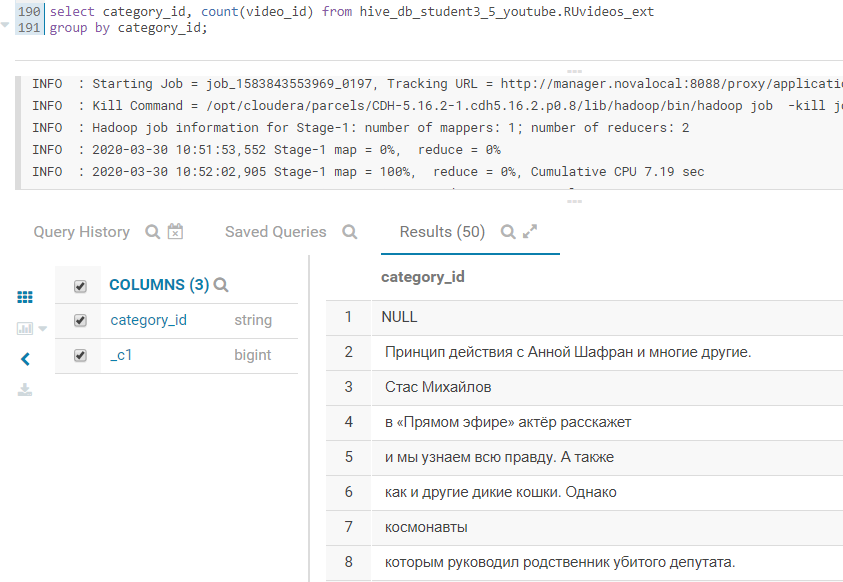
Следует отметить, что при создании таблиц исходные CSV файлы исчезают из директории:



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

ВЫПОЛНИМ разные выражения





Сделать любой отчет по загруженным данным используя JOIN.

Отберем максимальное кол-во лайков по тем видео датасета RU, которые так же присутствуют в датасете US

select RU.video\_id, max(RU.likes)

from hive\_db\_student3\_5\_youtube.RUvideos\_ext RU, hive\_db\_student3\_5\_youtube.USvideos\_ext US where RU.video\_id = US.video\_id

group by RU.video\_id;

INFO : Compiling command(queryId=hive\_20200330113131\_24e6e9bf-d797-4f51-b79a-356809a16ee5):

select RU.video\_id, max(RU.likes)

from hive\_db\_student3\_5\_youtube.RUvideos\_ext RU, hive\_db\_student3\_5\_youtube.USvideos\_ext US where RU.video\_id = US.video\_id

group by RU.video\_id

INFO : Semantic Analysis Completed

INFO : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:ru.video\_id, type:string, comment:null), FieldSchema(name:\_c1, type:string, comment:null)], properties:null)

INFO : Completed compiling command(queryId=hive\_20200330113131\_24e6e9bf-d797-4f51-b79a-356809a16ee5); Time taken: 0.312 seconds

INFO : Executing command(queryId=hive\_20200330113131\_24e6e9bf-d797-4f51-b79a-356809a16ee5):

select RU.video\_id, max(RU.likes)

from hive\_db\_student3\_5\_youtube.RUvideos\_ext RU, hive\_db\_student3\_5\_youtube.USvideos\_ext US where RU.video\_id = US.video\_id

group by RU.video\_id

INFO : Query ID = hive\_20200330113131\_24e6e9bf-d797-4f51-b79a-356809a16ee5

INFO : Total jobs = 2

INFO : Starting task [Stage-6:CONDITIONAL] in serial mode

INFO : Stage-1 is selected by condition resolver.

INFO : Launching Job 1 out of 2

INFO : Starting task [Stage-1:MAPRED] in serial mode

INFO : Number of reduce tasks not specified. Estimated from input data size: 3

INFO : In order to change the average load for a reducer (in bytes):

INFO : set hive.exec.reducers.bytes.per.reducer=<number>

INFO : In order to limit the maximum number of reducers:

INFO : set hive.exec.reducers.max=<number>

INFO : In order to set a constant number of reducers:

INFO : set mapreduce.job.reduces=<number>

INFO : number of splits:2

INFO : Submitting tokens for job: job\_1583843553969\_0202

INFO : The url to track the job: http://manager.novalocal:8088/proxy/application\_1583843553969\_0202/

INFO : Starting Job = job\_1583843553969\_0202, Tracking URL = http://manager.novalocal:8088/proxy/application\_1583843553969\_0202/

INFO : Kill Command = /opt/cloudera/parcels/CDH-5.16.2-1.cdh5.16.2.p0.8/lib/hadoop/bin/hadoop job -kill job\_1583843553969\_0202

INFO : Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 3

INFO : 2020-03-30 11:31:42,671 Stage-1 map = 0%, reduce = 0%

INFO : 2020-03-30 11:31:51,983 Stage-1 map = 50%, reduce = 0%, Cumulative CPU 7.81 sec

INFO : 2020-03-30 11:31:56,110 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 18.98 sec

INFO : 2020-03-30 11:32:05,387 Stage-1 map = 100%, reduce = 67%, Cumulative CPU 31.24 sec

INFO : 2020-03-30 11:32:11,619 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 38.34 sec

INFO : MapReduce Total cumulative CPU time: 38 seconds 340 msec

INFO : Ended Job = job\_1583843553969\_0202

INFO : Launching Job 2 out of 2

INFO : Starting task [Stage-2:MAPRED] in serial mode

INFO : Number of reduce tasks not specified. Estimated from input data size: 1

INFO : In order to change the average load for a reducer (in bytes):

INFO : set hive.exec.reducers.bytes.per.reducer=<number>

INFO : In order to limit the maximum number of reducers:

INFO : set hive.exec.reducers.max=<number>

INFO : In order to set a constant number of reducers:

INFO : set mapreduce.job.reduces=<number>

INFO : number of splits:1

INFO : Submitting tokens for job: job\_1583843553969\_0203

INFO : The url to track the job: http://manager.novalocal:8088/proxy/application\_1583843553969\_0203/

INFO : Starting Job = job\_1583843553969\_0203, Tracking URL = http://manager.novalocal:8088/proxy/application\_1583843553969\_0203/

INFO : Kill Command = /opt/cloudera/parcels/CDH-5.16.2-1.cdh5.16.2.p0.8/lib/hadoop/bin/hadoop job -kill job\_1583843553969\_0203

INFO : Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1

INFO : 2020-03-30 11:32:23,112 Stage-2 map = 0%, reduce = 0%

INFO : 2020-03-30 11:32:29,554 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 2.87 sec

INFO : 2020-03-30 11:32:36,847 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 5.65 sec

INFO : MapReduce Total cumulative CPU time: 5 seconds 650 msec

INFO : Ended Job = job\_1583843553969\_0203

INFO : MapReduce Jobs Launched:

INFO : Stage-Stage-1: Map: 2 Reduce: 3 Cumulative CPU: 38.34 sec HDFS Read: 139053119 HDFS Write: 7611 SUCCESS

INFO : Stage-Stage-2: Map: 1 Reduce: 1 Cumulative CPU: 5.65 sec HDFS Read: 12971 HDFS Write: 3793 SUCCESS

INFO : Total MapReduce CPU Time Spent: 43 seconds 990 msec

INFO : Completed executing command(queryId=hive\_20200330113131\_24e6e9bf-d797-4f51-b79a-356809a16ee5); Time taken: 62.624 seconds

INFO : OK

**РЕЗУЛЬТАТ**



[Продвинутый вариант] Сделать все вышеперечисленное с использованием JSON SerDe. Подсказка: см в сторону команды «ADD JAR»

**НЕ ДЕЛАЛ**