Big Data bootcamp, Hive: hands on

# General tips

Use official documentation before googling: <https://cwiki.apache.org/confluence/display/Hive/LanguageManual+DML>

<https://cwiki.apache.org/confluence/display/Hive/LanguageManual+UDF>

In console: beeline -u jdbc:hive2://localhost:10000

# Report

A file in sensible format (.docx and .pdf are fine) with all the steps you have taken and answers to questions. Include exact commands you used (text is OK for most points, when you should take screenshot it’s said so). If you had some problems, describe what the problem was, what you tried and what the solution was. Push report to git.

# Task

1. Get data from https://drive.google.com/file/d/1AOytyTA2MYYQOs6x9Q5M3n24gkbU8Jet/view

*These data are LV wikipipedia TOP 1000 articles by month, each file should contain one month and have header*

1. Clean data if necessary. Probably you will realize some cleaning is necessary few steps later. It’s ok. Come back and do some cleaning. It’s ok to do cleaning by hand. Extra points if you do it with script.
2. Put all data files on hdfs in directory /user/cloudera/wiki
3. Create database bootcamp

Create database bootcamp;

Use bootcamp;

1. Create external table e\_wiki in database bootcamp, location: /user/cloudera/wiki

CREATE EXTERNAL TABLE IF NOT EXISTS e\_wiki(

project STRING, access STRING, year STRING, month STRING, day STRING, articles\_\_article STRING, articles\_\_views STRING, articles\_\_rank STRING)

COMMENT 'Bootcamp external table e\_wiki'

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

STORED AS TEXTFILE

LOCATION '/user/cloudera/wiki'

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

1. How can you be sure all the data from files are present in table? Think of some way how to do this check. Answer “it should be after my commands” is not enough, becasuse sometimes things break unexpectedly and sometimes people make a mistake. One of the simplest ways how to do it is check row count and number of items in new table.

We can check it different ways – the easiest way is to open table in Hue and check field samples. I would use SELECT \* FROM e\_wiki limit 3; to check the results. Then, if something is wrong I’d go to files storage in Hue and check source DB integrity.

1. Create internal table t\_wiki and fill it with all the data from table e\_wiki and:
   1. Storage format ORC
   2. Schema:
      1. Year:int
      2. Month:int
      3. Article: string
      4. Views: int
      5. Rank: int

CREATE TABLE t\_wiki(

year INT,

month INT,

articles\_\_views INT,

articles\_\_article STRING,

articles\_\_rank INT

)

STORED AS ORC;

INSERT INTO TABLE t\_wiki SELECT

CAST(regexp\_replace(year,'"', '') as INT) as year,

CAST(regexp\_replace(month,'"', '') as INT) as month,

CAST(regexp\_replace(articles\_\_views,'"', '') as INT) as articles\_\_views,

articles\_\_article,

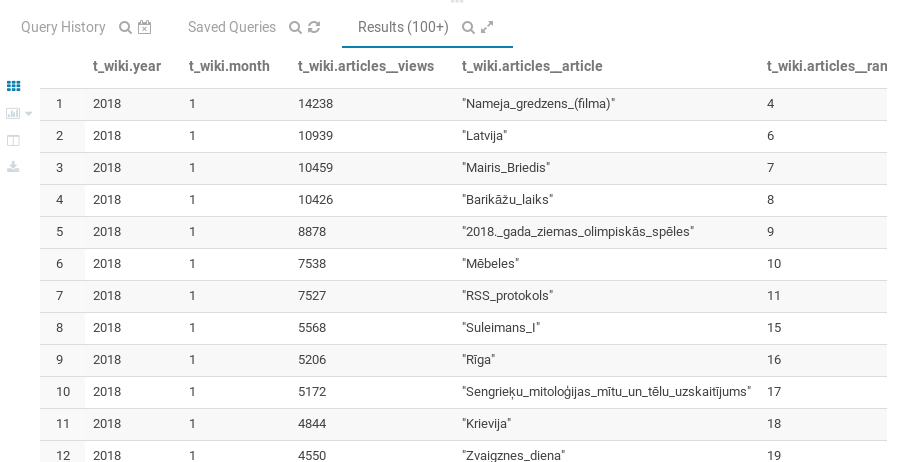
CAST(regexp\_replace(articles\_\_rank,'"', '') as INT) as articles\_\_rank

FROM e\_wiki;

1. Open one of the files in the new table. Are there any differences in directory structure? Why? How the file content differs (screenshot for this point with file content) from data you have on your laptop?

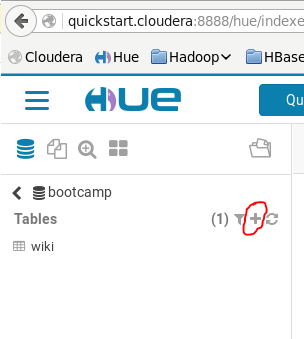
The difference is we don’t have CSV file in the internal ORC table and structure differs. The data is in column divided text file on the laptop, but there’s no columns separator between data columns on internal ORC DB.

Internal DB ORC:



File on laptop:



1. Try creating a table trough the HUE – easy way!
   1. Click on + sign in Hue  
      
   2. Work your way trough GUI
   3. What are advantages and disadvantages of this approach? Can you think about one use case for each?

Advantages are easy-to-use interface and quick access to specific table, field, field type, visualizng table, visualizing file system, without writing additional lines of code in terminal. The disadvantages are query latency and overall query speed, also dependence to the interface is disadvantage, because you are put „in the box”. Using HUE also takes additional RAM for generating webpage through browser, it may have real inpact on project speed in VM.

1. Write a queries to get to know following. If you notice any weird answers, point them out and fix them if it can be done in step 2
   1. How many records are in table t\_wiki ?  
      select count(\*) from t\_wiki;  
      5899
   2. What’s the biggest view count in one month?  
      SELECT Max(articles\_\_views) from t\_wiki group by month;  
      14238 , 3803, 10180, 12727, 92344, 20484, 20484
   3. What is the top article in all months?   
      select \* from t\_wiki order by articles\_\_views desc limit 1000;  
      2018.\_gada\_ziemas\_olimpisko\_spēļu\_medaļu\_tabula - 13857  
      What’s the top real article in all months? (Sākumlapa isn’t a real article. Special:<something> isn’t a real article)  
      2018.\_gada\_ziemas\_olimpisko\_spēļu\_medaļu\_tabula - 13857
   4. What’s the second biggest view count?  
      13252
   5. How many records there are for each month?   
      select count(\*) from t\_wiki group by month;  
      979, 980, 5, 986, 988, 977, 977
   6. How many real articles there are for each month?  
      979, 980, 5, 986, 988, 977, 977
   7. Which article reached TOP 1000 in most months?  
      Latvija\_2018.\_gada\_ziemas\_olimpiskajās\_spēlēs
   8. What’s the average view count to reach TOP 1000?  
      select avg(articles\_\_views) from t\_wiki limit 1000;  
      954.97674813306173
   9. (for top achievers) Is there any correllation between page ranks between months? (hint: Hive udf exists for that, but you might rearrange data before. It’s OK to create a temporary table or a view for this)  
      created new file ranks.csv from all previous files.  
      put in HDFS  
      then I created new table

CREATE EXTERNAL TABLE ranks(

mon1 INT, mon2 INT, mon3 INT, mon4 INT, mon5 INT, mon6 INT, mon7 INT)

COMMENT 'Bootcamp external table e\_wiki'

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

STORED AS TEXTFILE

LOCATION '/user/cloudera/wiki/ranks/;

Then I take correlation between columns, which are now ranks of articles

The biggest correlation result: 0.99999210352138879

The best result: 1.0

This means that ranks between articles have extremely high correlation. See screens below.



