


# Apache Hive and Apache Impala Homework

Realtime and Big Data Analytics

**Fall 2018**



# Homework

Class 8

## Analytics Project

1. Research your project. Each team member should upload a list of 5 papers relevant to your project and a short summary for one paper which includes your thoughts on how the paper is connected to your project.

It is useful to understand the state of the art before you begin a project. To do this, we read recently published papers. IEEE and ACM conferences and journals from 2015 to present are a good source.

Please do not choose papers on a Hadoop technology or other tool. Instead, choose papers related to your project thesis - the paper does not have to be in the same domain as your project. For example, do not choose a paper about Spark MLlib or Hadoop; do choose a paper about using big data in healthcare to solve some problem.

Where do I find a paper?

- Try using [GoogleScholar](#) to find papers.
- Try googling 'IEEE Big Data Analytics' for example - this brings up a bunch of conferences to pick from. Let me know if you have trouble finding a paper.
- Some other places to look for papers:
  - ACM KDD Conference:** <http://www.kdd.org/kdd2016/>
  - ACM DL:** <http://dl.acm.org/>
- Ask the professor if you get stuck.

List all five papers - title, authors, link where you found it - at the top of your document. Below this list, add a summary (a few paragraphs) about one of the papers. This summary should include your thoughts on how the paper relates to your project. Upload to NYU Classes. (In a future homework, you will add the team summaries to the 'Related Work' section of your project paper.)

Coordinate with your teammates to ensure each member reads a different paper. Share what you learn with your teammates.

**Note: The MapReduce, HDFS, and other papers already assigned cannot be used for this assignment.**

**Please provide the following:**

- 5 papers - title, authors, link to paper
- For the summarized paper, provide:
  - Paper title
  - Paper authors
  - Link to the paper
  - Paper abstract
  - Your summary

# Homework

Class 8

## **Analytics Project (continued)**

2. Draw *initial* diagrams using PowerPoint, Visio, etc. to describe your project. Include the software architecture (Big Data tools you'll use), the data flows, and anything else you think is important to show. This is a first draft, you will refine it in the coming weeks. **All team members should upload the diagrams.**
3. Create a list of tasks for your analytics project - you must use the TaskList.xlsx/numbers template in Resources. Assign team members to tasks, and assign a due date to each task. This can be just a simple table showing tasks, team member name, and target completion date. Try to identify milestones – that will help you know if you are on or off track. The next page has some suggested tasks, feel free to add to it. **All team members should upload the schedule.**

# Homework

---

**A. Try out Hive!** (see next bunch of slides for details). Screenshot upload required.

**B. Try out Impala!** (details follow the Hive homework). Screenshot upload required.

## **C. Readings**

1. Please read about Hive: TDG4: pp. 471-475, 478-493, 500-503, 505-507, 510-515

### **Readings (optional):**

1. Reference: “Improving MapReduce Performance in Heterogeneous Environments”, Zaharia, et al., OSDI ‘08, [https://www.usenix.org/legacy/event/osdi08/tech/full\\_papers/zaharia/zaharia\\_html/index.html](https://www.usenix.org/legacy/event/osdi08/tech/full_papers/zaharia/zaharia_html/index.html)

## 1. Create input data for simple hive test:

Paste the following data into a new file named smallWeather1.txt:

```
// Get data ready for Hive tests
$ hdfs dfs -put smallWeather1.txt hiveInput
$ hdfs dfs -ls hiveInput
$ hdfs dfs -cat hiveInput/smallWeather1.txt
```

© 2013-2018 Suzanne McIntosh

# Homework

---

## A. Try out Hive! *(continued)*

### 2. Create a hive external table:

#### *On Dumbo:*

```
$ beeline
beeline> !connect jdbc:hive2://babar.es.its.nyu.edu:10000/
```

#### *On the Quickstart VM:*

```
$ beeline -u jdbc:hive2://quickstart:10000/default -n cloudera -d org.apache.hive.jdbc.HiveDriver
```

---

Then select the database that has already been created to you - the database name is your NetId:

```
hive> use yourNetId;
```

```
hive> show tables;
```

```
hive> create external table w1 (data1 string, year int, data2 string, temperature int, quality tinyint, data3 string)
      row format delimited fields terminated by ','
```

```
      location '/user/yourNetId/hiveInput/';      // Note: for Cloudera VM use: location '/user/cloudera/hiveInput/';
```

```
hive> show tables;
```

```
hive> describe w1;
```

*Note 1: If you issue 'drop table w1;', only the table's metadata is deleted, the data remains. With an internal, or 'managed', Hive table, issuing the 'drop' command drops the metadata **AND the actual data**. Be careful...*

*Note 2: I am showing the prompt as **hive>** for brevity, you will likely see this prompt in your VM:*

*0: jdbc:hive2://quickstart:10000/default>*

# Homework

---

## A. Try out Hive! *(continued)*

### 3. View your data using HiveQL queries:

```
hive> select * from w1;
```

```
hive> select * from w1 limit 2;
```

```
hive> select year from w1;
```

```
hive> select * from w1 where year > 1949;
```

```
hive> select * from w1 where year >= 1949;
```

```
hive> select distinct year from w1;
```

*Note: Notice that a MapReduce job runs this time.*

```
hive> select w.year, w.temp from  
      (select year, max(temperature) as temp from w1 group by year) w;
```

*Note: This last result should look familiar, but here you only had to write one line of code!*

# Homework

---

## A. Try out Hive! *(continued)*

### 4. Create two more Hive tables with slightly different fields but with the same external data source:

```
hive> create external table w2 (data1 string, year int, data2 string, temperature int, quality tinyint, nines int)  
      row format delimited fields terminated by ','  
      location '/user/yourNetId/hiveInput';    // Note: for Cloudera VM use: location '/user/cloudera/hiveInput';
```

```
hive> show tables;  
hive> describe w2;
```

```
hive> select * from w2;  -- What unexpected value do you see?
```

```
hive> create external table w3 (data1 string, year int, data2 string, temperature int, quality tinyint, nines bigint)  
      row format delimited fields terminated by ','  
      location '/user/yourNetId/hiveInput';    // Note: for Cloudera VM use: location '/user/cloudera/hiveInput';
```

```
hive> show tables;  
hive> select * from w3;  -- Is this what you expected to see?
```

```
hive> drop table w2;  
hive> show tables;
```

```
hive> select * from w2;  --Should fail. Why?  
hive> select * from w3;  --We dropped table w2, but data and table w3's metadata are still available.  
hive> select * from w1;  --Table w1 is fine too.
```



# Homework

---

## A. Try out Hive! *(continued)*

### 5. Open a second terminal window and type the following commands

```
hdfs dfs -ls hiveInput    --Look at the file you put into hdfs
```

```
hdfs dfs -cat hiveInput/smallWeather1.txt
```

```
hdfs dfs -cp hiveInput/smallWeather1.txt hiveInput/smallWeather2.txt
```

```
hdfs dfs -ls hiveInput
```

```
hdfs dfs -cp hiveInput/smallWeather1.txt hiveInput/smallWeather3.txt
```

```
hdfs dfs -ls hiveInput    --Verify three files now in hdfs
```

### 6. In your Hive beeline session, Test querying multiple files within the same HDFS directory

```
hive> select * from w3;    --Notice now all three files in hdfs are picked up
```

```
hive> select * from w1;    --Notice now all three files in hdfs are picked up
```

# Homework

---

## B. Try out Impala!

- 1. Open a third terminal window, keep the Hive shell and the Linux terminal window open.**  
*(Keep the Hive shell open, you will need it shortly.)*

From the Linux window, verify that the data you put into HDFS is still there:

```
$ hdfs dfs -cat impalaInput/smallWeather1.txt
```

# Homework

---

## B. Try out Impala! *(continued)*

### 2. Start the Impala shell and create an Impala table:

```
$ impala-shell
```

```
impala> connect compute-1-1;
```

```
impala> hive> use yourNetId;
```

```
impala> show tables; -- you may or may not see the tables you created in Hive listed, you'll see them in a couple of steps
```

```
impala> invalidate metadata; -- force Impala to update its metadata
```

```
impala> show tables; -- you should see the tables you created earlier in Hive
```

```
impala> create database yourNetId_imp;
```

```
impala> use yourNetId_imp;
```

```
impala> show tables; -- you should see no tables listed
```

```
impala> create external table w10 (data1 string, year int, data2 string, temperature int, quality tinyint, data3 string)
```

```
    row format delimited fields terminated by ','
```

```
    location '/user/yourNetId/impalaInput/';    // Note: for Cloudera VM use: location '/user/cloudera/impalaInput/';
```

```
impala> show tables;
```

```
impala> describe w10;
```

# Homework

---

## B. Try out Impala! *(continued)*

### 3. View your data using SQL queries:

```
impala> select * from w10;  
impala> select * from w10 limit 2;
```

```
impala> select year from w10;
```

```
impala> select * from w10 where year > 1949;  
impala> select * from w10 where year >= 1949;
```

```
impala> select distinct year from w10; -- Notice how quickly the result is returned compared to Hive
```

```
impala> create external table w11 (data1 string, year int, data2 string, temperature int, quality tinyint, data3 string)  
    row format delimited fields terminated by ','  
    location '/user/yourNetId/impalaInput/';    // Note: for Cloudera VM use: location '/user/cloudera/impalaInput/';
```

// Run the **select distinct** command in the Impala shell on w1 twice - notice how much quicker it runs the 2<sup>nd</sup> time.

```
impala> select distinct year from w11; (takes about 6 seconds on Dumbo)  
impala> select distinct year from w11; (takes about 1 second, immediate response on Dumbo)
```

// Run the **select distinct** command in the Hive shell on w1 twice - notice it is no faster the 2<sup>nd</sup> time (no optimization).

// It takes about 20 seconds on Dumbo both times.

```
hive> select distinct year from w1;  
hive> select distinct year from w1;
```

// Go back to your Impala window and issue the following:

```
impala> select w.year, w.temp from (select year, max(temperature) as temp from w10 group by year) w;
```

*Note: This last result should look familiar!*

# Homework

---

## B. Try out Impala! *(continued)*

### 4. Create one more Impala table with slightly different fields but with the same external data source:

```
impala> create external table w30 (data1 string, year int, data2 string, temperature int, quality tinyint, nines bigint)  
    row format delimited fields terminated by ','  
    location '/user/yourNetId/impalaInput/';    // Note: for Cloudera VM use: location '/user/cloudera/impalaInput/';
```

```
impala> select * from w30;
```

```
impala> select * from w10;
```

// Notice the different data type of the last field (the tables have different schemas), but we are using the same  
// physical file for both external tables:

```
impala> describe w30;
```

```
impala> describe w10;
```

### 5. Quit Impala using Ctrl-D