ADVANCED HIVE

CKME 134 – BIG DATA ANALYTICS TOOLS

RYERSON UNIVERSITY

SPRING 2015

Instructor: Shaohua Zhang

Course Outline [Updated!]

1. Intro to Big Data
2. Distributed Computing and MapReduce
3. Hadoop Ecosystem
4. Programming Hive
5. Advanced Hive
6. Programming Pig
7. Advanced Pig
8. Hadoop Performance Optimization
9. Hadoop In Action: Building Data Pipelines
9. Location Analytics and Recommender Systems
10. Beyond Hadoop: Spark
11. Beyond Hadoop: Graph Analytics

Recap and Random Things…

Lecture 4 Overview
Excel
Data Visualization
News of the Week

Lecture 4 Overview

- HiveQL Basic Syntax
- Hive Primitive Data Types
- Hive Functions
- Hive Joins (Inner/Outer Joins)

Microsoft Excel

- Tabular data is so easy to work with
  - Excel is still one of the most popular analytics tool
  - Python/R dataframes
- What people use Excel for
  - Pivot Table
  - Visualization
  - Hadoop/SAS connect
    - http://azure.microsoft.com/en-us/documentation/articles/hdinsight-connect-excel-power-query/
    - http://research.microsoft.com/en-us/projects/azure/visualization-with-excel-tools-and-windows-azure.pdf
  - Macro/VBA → automation
  - Machine Learning
    - K-Means Clustering
      - http://www.neilson.co.za/k-means-cluster-analysis-in-microsoft-excel/

Data Visualization

- Excel
- **R/Python**
  - ggplot, matplotlib, … you name it
  - great for analysis and presenting insights
- **Javascript - D3.js**
  - Dynamic and interactive
  - Event-driven (real-time)
  - Integrate seamlessly with applications
- Commercial
  - Zoomdata
  - Qlikview/Tableau
  - OLAP tools → SAS, etc.

Data Science News of the Week

- White house to name United States' first chief data scientist
  - "Data Science: Where are We Going?" - Dr. DJ Patil
  - https://www.youtube.com/watch?v=3_1reLdh5xw
- More DJ Patil talks
  - https://www.youtube.com/watch?v=J_CYKk8q1Ao
  - https://www.youtube.com/watch?v=98NrsLE6ot4

Python/R News

- Microsoft embraces Python, Linux in new big data tools
  - Same company that acquired Revolution Analytics a few weeks ago…
  - https://gigaom.com/2015/02/18/microsoft-embraces-python-linux-in-new-big-data-tools/

DEMO

Bonus Resources

- Hive/Pig Cheatsheet
- IPython Notebook (Basic Python)

Stay Tuned!

Lecture 5 - Outline

1. Complex Data Types
2. Advanced Functions
3. Nested Queries
4. Advanced Joins
5. Apache Sqoop

Data – Geotagged Tweets

ID	DateTime	Latitude	Longitude	Tweet
USER_8d0e8566	2010-03-02T23:00:44	30.387524	-91.109663	Pre-workout prep has begun.
USER_8d0e8566	2010-03-02T23:04:20	30.387524	-91.109663	I really don't like that a college FB player's ON-FIELD production can be negated by a single workout. So proof's NOT in the pudding?
USER_87b48222	2010-03-02T23:23:29	37.530819	-77.475577	@USER_9bb099c2 15 pages??? fuck u mean!!?? damn.
USER_87b48222	2010-03-02T23:43:57	37.530819	-77.475577	@USER_e97d1292 lol do u know that song?
USER_01b8a291	2010-03-03T00:56:16	41.51179	-95.893286	HAHAHA OMG! I just found a baggie of weed that I hid from like four/five years ago!! Hahahaha!!!
USER_2e5f8774	2010-03-03T02:06:15	39.669307	-79.85002	@USER_2b2bd61b light skin free way and shit...lol Look like you sell bean pies
USER_942c68df	2010-03-03T02:21:36	41.220425	-85.861873	These judges are being hard this year.
USER_8d0e8566	2010-03-03T02:28:12	30.387524	-91.109663	@USER_b7cdabe3 People don't dance like that to get the burn anymore. Its frowned upon..LOL.
USER_8d0e8566	2010-03-03T02:29:39	30.399934	-91.121502	RT @USER_9c9e75e2: Officially getting rid of my iPhone with its dysfunctional button this weekend|Get a 9700 #BlackertheBerrytheSweetertheUse
USER_2e5f8774	2010-03-03T02:42:44	39.669307	-79.85002	@USER_7ac8dee6 Hey Cuz...Where u been at?
USER_8d0e8566	2010-03-03T02:43:01	30.393485	-91.110458	RT @USER_9c9e75e2: @USER_8d0e8566 I think that's the move!|Make it happen and we can play Word Mole against each other.
USER_8d0e8566	2010-03-03T02:53:19	30.393485	-91.110458	@USER_b7cdabe3 Oh, okay!
USER_942c68df	2010-03-03T02:55:36	41.234181	-85.812994	@USER_20c15b69 Me too.
USER_8d0e8566	2010-03-03T03:00:37	30.387524	-91.109663	The next 2hrs of tweets are @USER_fe579e73 for gibing me the idea with his #theory tweet
USER_942c68df	2010-03-03T03:14:53	41.234181	-85.812994	@USER_21fe08ea Aww that sucks. If ya dont mind me asking, whats ruining your relationship?
USER_8d0e8566	2010-03-03T03:26:46	30.387524	-91.109663	@USER_fe579e73 did u change ur settings to use twitlonger?
USER_8d0e8566	2010-03-03T03:29:41	30.387524	-91.109663	RT @USER_de057bc2: Twitter is jacked up tonight|Just on iPhones. #BlackertheBerrytheSweetertheUse
USER_8d0e8566	2010-03-03T03:33:47	30.387524	-91.109663	RT @USER_de057bc2: @USER_8d0e8566 EFF YO Blackberry|Sore Loser
USER_8d0e8566	2010-03-03T03:47:43	30.387524	-91.109663	@USER_45b5c066 @USER_2b5b12ff The body nice but that had to be a contest at a Bukket Nekked.
USER_8d0e8566	2010-03-03T03:57:23	30.387524	-91.109663	#PeterWisdom "If u wake up and ur gal or the gal ur in bed with is staring at u,take solace in knowing she'll be sleep when u escape." LOL
USER_87b48222	2010-03-03T03:59:01	37.530819	-77.475577	Where do you those rip away jeans?!! @USER_af454d84 and where can I get some?!
USER_8d0e8566	2010-03-03T04:17:29	30.387524	-91.109663	@USER_b7cdabe3 LOL
USER_8d0e8566	2010-03-03T04:37:07	30.387524	-91.109663	RT @USER_45b5c066: #FamilyGuy Meg and Brian make out. Meg stalks him like Misery << did u just use a shag blog term?? #CLASSIC|Did I?

Lab Data Preparation

```
hive>
    > ------------------------------------------------------------------
    > -- load geo-tagged tweets as external hive table
    >
    > -- Note: you can skip this if you already have
    > -- twitter.full_text_ts table created from lab 4
    > ------------------------------------------------------------------
    >
    > -- create and load tweet data as external table
    > drop table twitter.full_text;
OK
Time taken: 3.176 seconds
hive> create external table twitter.full_text (
    >           id string,
    >           ts string,
    >           lat_lon string,
    >           lat string,
    >           lon string,
    >           tweet string)
    > row format delimited
    > fields terminated by '\t'
    > location '/user/twitter/full_text' ;     -- note: you may have your data in a different hadoop directory and that's fine!
    >
    > -- convert timestamp
    >
    > drop table twitter.full_text_ts;
OK
Time taken: 0.826 seconds
OK
Time taken: 0.742 seconds
hive>
    > create table twitter.full_text_ts as
    > select id, cast(concat(substr(ts,1,10), ' ', substr(ts,12,8)) as timestamp) as ts, lat, lon, tweet
    > from twitter.full_text;
Query ID = root_20150223022929_ee094283-db00-495c-9e15-0e1ec1a90f0c
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1424547612900_0013, Tracking URL = http://sandbox.hortonworks.com:8088/proxy/application_1424547612900_0013/
Kill Command = /usr/hdp/2.2.0.0-2041/hadoop/bin/hadoop job  -kill job_1424547612900_0013
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2015-02-23 02:29:50,598 Stage-1 map = 0%,  reduce = 0%
2015-02-23 02:30:10,254 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 13.51 sec
MapReduce Total cumulative CPU time: 13 seconds 510 msec
Ended Job = job_1424547612900_0013
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to: hdfs://sandbox.hortonworks.com:8020/tmp/hive/root/ea5bec65-2110-4382-b935-7f5cb4009355/hive_2015-02-23_02-29-30_900_8189695155451937898-1/-ext-10001
Moving data to: hdfs://sandbox.hortonworks.com:8020/apps/hive/warehouse/twitter.db/full_text_ts
Table twitter.full_text_ts stats: [numFiles=1, numRows=377616, totalSize=47273124, rawDataSize=46895508]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 13.51 sec   HDFS Read: 57140168 HDFS Write: 47273210 SUCCESS
Total MapReduce CPU Time Spent: 13 seconds 510 msec
OK
```

Hive Complex Data Types

Complex Data Types

Complex Type	*Description*	*Literal Syntax*
Array	*Ordered sequences of the same type that are indexable using zero-based integers*	*array(lat,lon)* *[-48.01234, 93.44444]* *location[0] → -48.01234*
Map	*An unordered collection of key-value tuples.*	*map('lat', lat, 'lon', lon)* *{'lat': -48.01234, 'lon': 93.44444 }* *location['lat'] → -48.01234*
Struct	*More structured data type, like a table. Fields can be accessed using the "dot" notation*	*struct(lat, lon)* *{'lat': -48.01234, 'lon': 93.44444 }* *location.lat → -48.01234*

DEMO

```
hive>
    > -- create a temporary table schema
    > drop table twitter.full_text_ts_complex_tmp;
OK
Time taken: 0.483 seconds
hive> create external table twitter.full_text_ts_complex_tmp (
    >                       id string,
    >                       ts timestamp,
    >                       lat float,
    >                       lon float,
    >                       tweet string,
    >                       location_array string,
    >                       location_map string,
    >                       tweet_struct string
    > )
    > row format delimited
    > fields terminated by '\t'
    > stored as textfile
    > location '/user/twitter/full_text_ts_complex';
OK
Time taken: 0.2 seconds
hive>
    > -- load transformed data into the temp table
    > insert overwrite table twitter.full_text_ts_complex_tmp
    > select id, ts, lat, lon, tweet,
    >        concat(lat,',',lon) as location_array,
    >        concat('lat:', lat, ',', 'lon:', lon) as location_map,
    >        concat(regexp_extract(lower(tweet), '(.*)@user_(\\S{8})([:| ])(.*)',2), ',', length(tweet)) as tweet_struct
    > from twitter.full_text_ts;
Query ID = root_20150223024040_c5b3b0bd-54fb-44fa-a498-415f0c32e46b
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1424547612900_0014, Tracking URL = http://sandbox.hortonworks.com:8088/proxy/application_1424547612900_0014/
Kill Command = /usr/hdp/2.2.0.0-2041/hadoop/bin/hadoop job  -kill job_1424547612900_0014
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2015-02-23 02:40:13,852 Stage-1 map = 0%,  reduce = 0%
2015-02-23 02:40:52,003 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 33.0 sec
MapReduce Total cumulative CPU time: 33 seconds 0 msec
Ended Job = job_1424547612900_0014
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to: hdfs://sandbox.hortonworks.com:8020/tmp/hive/root/ea5bec65-2110-4382-b935-7f5cb4009355/hive_2015-02-23_02-40-00_396_1047887830264700745-1/-ext-10000
Loading data to table twitter.full_text_ts_complex_tmp
Moved: 'hdfs://sandbox.hortonworks.com:8020/user/twitter/full_text_ts_complex/000000_0' to trash at: hdfs://sandbox.hortonworks.com:8020/user/root/.Trash/Current
Table twitter.full_text_ts_complex_tmp stats: [numFiles=1, numRows=377616, totalSize=69217207, rawDataSize=68839591]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 33.93 sec   HDFS Read: 47273366 HDFS Write: 69217305 SUCCESS
Total MapReduce CPU Time Spent: 33 seconds 930 msec
OK
Time taken: 54.859 seconds
hive>
    >
    > select * from twitter.full_text_ts_complex_tmp limit 3;
OK
USER_79321756   2010-03-03 04:15:26     47.528137       -122.197914     RT @USER_2ff4faca: IF SHE DO IT 1 MORE TIME......IMA KNOCK HER DAMN KOOFIE OFF.....ON MY MOMMA
-122.197916     lat:47.528139,lon:-122.197916   2ff4faca,119
USER_79321756   2010-03-03 04:55:32     47.528137       -122.197914     @USER_77a4822d @USER_2ff4faca okay:) lol. Saying ok to both of yall about to different things!
t:47.528139,lon:-122.197916     2ff4faca,96
USER_79321756   2010-03-03 05:13:34     47.528137       -122.197914     RT @USER_5d4d777a: YOURE A FAG FOR GETTING IN THE MIDDLE OF THIS @USER_ab059bdc WHO THE FUCK A
```

DEMO


```
hive> -- Reload the temp file using complex types instead of strings
    > -- NOTE: you specify the complex type when you create the table schema
    > drop table twitter.full_text_ts_complex;
OK
Time taken: 0.707 seconds
hive> create external table twitter.full_text_ts_complex (
    >                       id                  string,
    >                       ts                  timestamp,
    >                       lat                 float,
    >                       lon                 float,
    >                       tweet               string,
    >                       location_array      array<float>,
    >                       location_map        map<string, string>,
    >                       tweet_struct        struct<mention:string, size:int>
    > )
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY '\t'
    > COLLECTION ITEMS TERMINATED BY ','
    > MAP KEYS TERMINATED BY ':'
    > location '/user/twitter/full_text_ts_complex';
OK
Time taken: 0.462 seconds
hive>
    > select * from twitter.full_text_ts_complex limit 3;
OK
USER_79321756   2010-03-03 04:15:26      47.528137       -122.197914     RT @USER_2ff4faca: IF SHE DO IT 1 MORE TIME......IMA KNOCK HER DAMN KOOFIE OFF.
,-122.197914]   {"lat":"47.528139","lon":"-122.197916"} {"mention":"2ff4faca","size":119}
USER_79321756   2010-03-03 04:55:32      47.528137       -122.197914     @USER_77a4822d @USER_2ff4faca okay:) lol. Saying ok to both of yall about to di
lat":"47.528139","lon":"-122.197916"}   {"mention":"2ff4faca","size":96}
USER_79321756   2010-03-03 05:13:34      47.528137       -122.197914     RT @USER_5d4d777a: YOURE A FAG FOR GETTING IN THE MIDDLE OF THIS @USER_ab059bdc
t;Lol! Dayum! Aye!      [47.528137,-122.197914] {"lat":"47.528139","lon":"-122.197916"} {"mention":"ab059bdc","size":148}
Time taken: 0.2 seconds, Fetched: 3 row(s)
```

list map struct

Collection Functions

- size(Array<T>)
- array_contains(Array<T>)
- sort_array(Array<T>)
- size(Map<k,v>)
- map_keys(Map<K,V>)
- map_values(Map<K,V>)

DEMO

```
hive> ------------------------------------------------
    > -- Hive Collection Functions
    > ------------------------------------------------
    >
    > -- Create complex type directly using map(), array(), struct() functions
    >
    > select id, ts, lat, lon,
    >        array(lat, lon) as location_array,
    >        map('lat', lat, 'lon', lon)  as location_map,
    >        named_struct('lat', lat, 'lon',lon) as location_struct
    > from twitter.full_text_ts
    > limit 10;
OK
USER_79321756   2010-03-03 04:15:26     47.528139       -122.197916     ["47.528139","-122.197916"]     {"lat":"47.528139","lon":"-122.197916"} {"lat":"47.528139","lon":"-122.197916"}
USER_79321756   2010-03-03 04:55:32     47.528139       -122.197916     ["47.528139","-122.197916"]     {"lat":"47.528139","lon":"-122.197916"} {"lat":"47.528139","lon":"-122.197916"}
USER_79321756   2010-03-03 05:13:34     47.528139       -122.197916     ["47.528139","-122.197916"]     {"lat":"47.528139","lon":"-122.197916"} {"lat":"47.528139","lon":"-122.197916"}
USER_79321756   2010-03-03 05:28:02     47.528139       -122.197916     ["47.528139","-122.197916"]     {"lat":"47.528139","lon":"-122.197916"} {"lat":"47.528139","lon":"-122.197916"}
USER_79321756   2010-03-03 05:56:13     47.528139       -122.197916     ["47.528139","-122.197916"]     {"lat":"47.528139","lon":"-122.197916"} {"lat":"47.528139","lon":"-122.197916"}
USER_79321756   2010-03-03 16:52:44     47.528139       -122.197916     ["47.528139","-122.197916"]     {"lat":"47.528139","lon":"-122.197916"} {"lat":"47.528139","lon":"-122.197916"}
USER_79321756   2010-03-03 16:57:24     47.528139       -122.197916     ["47.528139","-122.197916"]     {"lat":"47.528139","lon":"-122.197916"} {"lat":"47.528139","lon":"-122.197916"}
USER_79321756   2010-03-03 20:20:40     47.528139       -122.197916     ["47.528139","-122.197916"]     {"lat":"47.528139","lon":"-122.197916"} {"lat":"47.528139","lon":"-122.197916"}
USER_79321756   2010-03-03 23:23:33     47.528139       -122.197916     ["47.528139","-122.197916"]     {"lat":"47.528139","lon":"-122.197916"} {"lat":"47.528139","lon":"-122.197916"}
USER_79321756   2010-03-03 23:37:36     47.528139       -122.197916     ["47.528139","-122.197916"]     {"lat":"47.528139","lon":"-122.197916"} {"lat":"47.528139","lon":"-122.197916"}
Time taken: 0.171 seconds, Fetched: 10 row(s)
hive>
    >
    > -- Work with collection functions
    >    -- extract element from arrays/maps using indexing
    >    -- extract element from struct using 'dot' notation
    > select location_array[0] as lat,
    >        location_map['lon'] as lon,
    >        tweet_struct.mention as mention,
    >        tweet_struct.size as tweet_length
    > from twitter.full_text_ts_complex
    > limit 5;
OK
47.528137       -122.197916     2ff4faca        119
47.528137       -122.197916     2ff4faca        96
47.528137       -122.197916     ab059bdc        148
47.528137       -122.197916     77a4822d        89
47.528137       -122.197916             82
Time taken: 0.201 seconds, Fetched: 5 row(s)
hive>
    > -- Work with collection functions
    >    -- extract all keys/values from maps
    >    -- get number of elements in arrays/maps
    > select size(location_array), sort_array(location_array),
    >        size(location_map), map_keys(location_map), map_values(location_map)
    > from twitter.full_text_ts_complex
    > limit 5;
OK
2       [-122.197914,47.528137] 2       ["lat","lon"]   ["47.528139","-122.197916"]
2       [-122.197914,47.528137] 2       ["lat","lon"]   ["47.528139","-122.197916"]
2       [-122.197914,47.528137] 2       ["lat","lon"]   ["47.528139","-122.197916"]
2       [-122.197914,47.528137] 2       ["lat","lon"]   ["47.528139","-122.197916"]
2       [-122.197914,47.528137] 2       ["lat","lon"]   ["47.528139","-122.197916"]
Time taken: 0.191 seconds, Fetched: 5 row(s)
```

Hive Advanced Functions

String Functions

String Func	Description	Syntax
split	Split strings around regex pattern	split(string str, string pat) split('big data tools ckme134', ' ')
sentences	Tokenizes a string of natural language text into words and sentences, where each sentence is broken at the appropriate sentence boundary and returned as an array of words	sentences('Hello there! How are you?') returns (("Hello", "there"), ("How", "are", "you"))
ngrams	Returns the top-k N-grams from a set of tokenized sentences, such as those returned by the sentences()	SELECT explode(ngrams(sentences(lower(val)), 2, 10)) AS x FROM kafka; {"ngram":[of","the],"estfrequency":23.0} {"ngram":[on","the],"estfrequency":20.0} {"ngram":[in","the],"estfrequency":18.0} {"ngram":[he","was],"estfrequency":17.0} {"ngram":[at","the],"estfrequency":17.0}
context_ngrams	Returns the top-k N-grams from a set of tokenized sentences, given a string of "context"	

DEMO

sentences() function


```
hive> -- sentences function
    > select sentences(tweet)
    > from twitter.full_text_ts
    > limit 10;
OK
[["RT","USER","2ff4faca","IF","SHE","DO","IT","1","MORE","TIME","IMA","KNOCK","HER","DAMN","KOOFIE","OFF","ON","MY","MOMMA","gt","
[["USER","77a4822d","USER","2ff4faca","okay","lol"],["Saying","ok","to","both","of","yall","about","to","different","things"],[]]
[["RT","USER","5d4d777a","YOURE","A","FAG","FOR","GETTING","IN","THE","MIDDLE","OF","THIS","USER_ab059bdc","WHO","THE","FUCK","ARE
[["USER","77a4822d","yea","ok","well","answer","that","cheap","as","Sweden","phone","you","came","up","on","when","I","call"]]
[["A","sprite","can","disappear","in","her","mouth","lil","kim","hmmmmm","the","can","not","the","bottle","right"]]
[["Lmao"],["I","still","get","txt","when","AJ","tweets","before","they","even","post","mistake","ha"],["And","the","one","I","just
[["Alright","twitters","tryna","take","me","over"]]
[["Just","got","to","work"],["Got","my","pizza","bagel","and","my","raspberry","iced","tea"],["Pulling","up","my","systems","inter
[["Just","got","a","txt","from","my","cousin"],["Yes"],["So","happy","for","you","USER_a9fe21e9","let's","get","it"]]
[["Why","is","this","woman","in","the","bathroom","everytime","I'm","in","the","bathroom"],["Stinkn","up","allll","the","stalls"],
Time taken: 0.192 seconds, Fetched: 10 row(s)
```

ngrams() function

DEMO

return top 10 bi-grams (2grams)

```
hive> -- ngrams function
    > select ngrams(sentences(tweet), 2, 10)
    > from twitter.full_text_ts
    > limit 50;
Query ID = root_20150223034242_b9fca998-d851-441d-92a7-6975145b7c3f
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1424547612900_0025, Tracking URL = http://sandbox.hortonworks.com:8088/proxy/application_1424547612900_0025/
Kill Command = /usr/hdp/2.2.0.0-2041/hadoop/bin/hadoop job  -kill job_1424547612900_0025
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2015-02-23 03:42:39,208 Stage-1 map = 0%,  reduce = 0%
2015-02-23 03:43:01,436 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 15.21 sec
2015-02-23 03:43:11,365 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 17.85 sec
MapReduce Total cumulative CPU time: 17 seconds 850 msec
Ended Job = job_1424547612900_0025
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 17.85 sec   HDFS Read: 47273366 HDFS Write: 140 SUCCESS
Total MapReduce CPU Time Spent: 17 seconds 850 msec
OK
[{"ngram":["RT","USER"],"estfrequency":48781.0},{"ngram":["in","the"],"estfrequency":7327.0},{"ngram":["I","was"],"estfrequency":4764.0},{"ngr
,"estfrequency":4408.0},{"ngram":["to","the"],"estfrequency":4132.0},{"ngram":["to","be"],"estfrequency":3983.0},{"ngram":["I","don't"],"estfr
},{"ngram":["I","need"],"estfrequency":3221.0}]
```

DEMO

ngrams() function


```
hive> select explode(ngrams(sentences(tweet), 2, 10))
    > from twitter.full_text_ts
    > limit 50;
Query ID = root_20150223034444_f2282674-b11e-4c31-b0f0-d2dcf6a2b3c4
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1424547612900_0026, Tracking URL = http://sandbox.hortonworks.com:8088/proxy/application_1424547612900_0026/
Kill Command = /usr/hdp/2.2.0.0-2041/hadoop/bin/hadoop job  -kill job_1424547612900_0026
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2015-02-23 03:45:04,866 Stage-1 map = 0%,  reduce = 0%
2015-02-23 03:45:27,391 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 15.33 sec
2015-02-23 03:45:36,175 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 17.46 sec
MapReduce Total cumulative CPU time: 17 seconds 460 msec
Ended Job = job_1424547612900_0026
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 17.46 sec   HDFS Read: 47273366 HDFS Write: 140 SUCCESS
Total MapReduce CPU Time Spent: 17 seconds 460 msec
OK
{"ngram":["RT","USER"],"estfrequency":48781.0}
{"ngram":["in","the"],"estfrequency":7327.0}
{"ngram":["I","was"],"estfrequency":4764.0}
{"ngram":["lt","lt"],"estfrequency":4669.0}
{"ngram":["on","the"],"estfrequency":4408.0}
{"ngram":["to","the"],"estfrequency":4132.0}
{"ngram":["to","be"],"estfrequency":3983.0}
{"ngram":["I","don't"],"estfrequency":3945.0}
{"ngram":["to","get"],"estfrequency":3506.0}
{"ngram":["I","need"],"estfrequency":3221.0}
Time taken: 46.857 seconds, Fetched: 10 row(s)
```

explode() helps transpose the output n-gram LIST into separate rows

DEMO

context_ngrams() function

most popular word after bigram 'I need'

```
hive> -- context_ngrams function
    >
    > select explode(context_ngrams(sentences(tweet), array('I', 'need', null), 10))
    > from twitter.full_text_ts
    > limit 50;
Query ID = root_20150223034848_f36335d4-7140-4c3d-87da-dab3c41ae0e7
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1424547612900_0027, Tracking URL = http://sandbox.hortonworks.com:8088/proxy/application_1424547612900_0027/
Kill Command = /usr/hdp/2.2.0.0-2041/hadoop/bin/hadoop job  -kill job_1424547612900_0027
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2015-02-23 03:48:38,944 Stage-1 map = 0%,  reduce = 0%
2015-02-23 03:48:55,696 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 10.65 sec
2015-02-23 03:49:06,843 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 13.15 sec
MapReduce Total cumulative CPU time: 13 seconds 150 msec
Ended Job = job_1424547612900_0027
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 13.15 sec   HDFS Read: 47273366 HDFS Write: 88 SUCCESS
Total MapReduce CPU Time Spent: 13 seconds 150 msec
OK
{"ngram":["to"],"estfrequency":999.0}
{"ngram":["a"],"estfrequency":687.0}
{"ngram":["some"],"estfrequency":202.0}
{"ngram":["2"],"estfrequency":97.0}
{"ngram":["my"],"estfrequency":92.0}
{"ngram":["that"],"estfrequency":58.0}
{"ngram":["you"],"estfrequency":51.0}
{"ngram":["it"],"estfrequency":50.0}
{"ngram":["more"],"estfrequency":50.0}
{"ngram":["is"],"estfrequency":42.0}
Time taken: 42.972 seconds, Fetched: 10 row(s)
```

DEMO

ngrams() function

25

```
hive> -- context_ngrams function
    > select explode(context_ngrams(sentences(tweet), array('I', 'need', null, null, null), 10))
    > from twitter.full_text_ts
    > limit 50;
Query ID = root_20150223034949_0be10ba9-37e6-4abb-94ca-b41481f96af3
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1424547612900_0028, Tracking URL = http://sandbox.hortonworks.com:8088/proxy/application_1424547612900_0028/
Kill Command = /usr/hdp/2.2.0.0-2041/hadoop/bin/hadoop job  -kill job_1424547612900_0028
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2015-02-23 03:49:56,116 Stage-1 map = 0%,  reduce = 0%
2015-02-23 03:50:13,935 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 11.02 sec
2015-02-23 03:50:23,830 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 13.55 sec
MapReduce Total cumulative CPU time: 13 seconds 550 msec
Ended Job = job_1424547612900_0028
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 13.55 sec   HDFS Read: 47273366 HDFS Write: 156 SUCCESS
Total MapReduce CPU Time Spent: 13 seconds 550 msec
OK
{"ngram":["to","go","to"],"estfrequency":35.0}
{"ngram":["to","get","my"],"estfrequency":21.0}
{"ngram":["to","get","up"],"estfrequency":15.0}
{"ngram":["something","to","do"],"estfrequency":13.0}
{"ngram":["to","find","a"],"estfrequency":12.0}
{"ngram":["to","talk","to"],"estfrequency":11.0}
{"ngram":["to","get","a"],"estfrequency":9.0}
{"ngram":["to","get","back"],"estfrequency":9.0}
{"ngram":["my","hair","done"],"estfrequency":8.0}
{"ngram":["to","get","on"],"estfrequency":8.0}
Time taken: 42.152 seconds, Fetched: 10 row(s)
```

Aggregation Functions (UDAF)

- count(*), count(distinct)
- sum, avg
- min, max
- percentile
- histogram_numeric
- collect_set
- collect_list

percentile_approx() function

DEMO

- Find twitter users from north west part of U.S.


```
hive>
    > -- PERCENTILE_APPROX function (works with DOUBLE type)
    >    -- Find twitter users from north west part of U.S.
    >    -- You can visualize it using the map tool: http://www.darrinward.com/lat-long/?id=461435
    >
    > select percentile_approx(cast(lat as double), array(0.9))
    > from twitter.full_text_ts_complex;   --  41.79976907219686
    >
    >
    > select percentile_approx(cast(lon as double), array(0.1))
    > from twitter.full_text_ts_complex;   --  -117.06394155417728
    >
    > select distinct lat, lon
    > from twitter.full_text_ts_complex
    > where cast(lat as double) >= 41.79976907219686 AND
    >       cast(lon as double) <= -117.06394155417728
    > limit 10;
```

percentile_approx() function

- find west most point of U.S.

percentile_approx() function

- *Find twitter users from north west part of U.S.*

histogram_numeric() function

DEMO

-Bucket U.S. into 10x10 grids using histogram_numeric

```
hive> select explode(histogram_numeric(lat, 10)) as hist_lon from twitter.full_text_ts_complex
    > ;
Query ID = root_20150223044646_47310903-b8f5-478a-a5d7-b5380cbc63c2
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1424547612900_0045, Tracking URL = http://sandbox.hortonworks.com:8088/proxy/applicatio
Kill Command = /usr/hdp/2.2.0.0-2041/hadoop/bin/hadoop job  -kill job_1424547612900_0045
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2015-02-23 04:46:22,379 Stage-1 map = 0%,  reduce = 0%
2015-02-23 04:46:29,763 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.21 sec
2015-02-23 04:46:38,742 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 4.53 sec
MapReduce Total cumulative CPU time: 4 seconds 530 msec
Ended Job = job_1424547612900_0045
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 4.53 sec   HDFS Read: 69217439 HDFS Write: 247 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 530 msec
OK
{"x":-25.50731767926898,"y":42.0}
{"x":-7.17137844363848,"y":144.0}
{"x":3.77521472175916,"y":12.0}
{"x":13.004202445348103,"y":12.0}
{"x":18.605831107314756,"y":49.0}
{"x":28.804234052185453,"y":43326.0}
{"x":34.66352003913391,"y":106282.0}
{"x":40.65575122055146,"y":218285.0}
{"x":45.472877604624635,"y":9445.0}
{"x":55.8222710458856,"y":19.0}
Time taken: 24.423 seconds, Fetched: 10 row(s)
```

DEMO

histogram_numeric() function

- Bucket U.S. into 10x10 grids using histogram_numeric

```
hive>
    > select explode(histogram_numeric(lon, 10)) from twitter.full_text_ts_complex;
Query ID = root_20150223044040_ae1ea17e-21f5-4dc4-af52-40553a4d758e
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1424547612900_0043, Tracking URL = http://sandbox.hortonworks.com:8088/proxy/application_142
Kill Command = /usr/hdp/2.2.0.0-2041/hadoop/bin/hadoop job  -kill job_1424547612900_0043
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2015-02-23 04:41:13,523 Stage-1 map = 0%,  reduce = 0%
2015-02-23 04:41:25,032 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 4.67 sec
2015-02-23 04:41:36,514 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 7.12 sec
MapReduce Total cumulative CPU time: 7 seconds 120 msec
Ended Job = job_1424547612900_0043
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 7.12 sec   HDFS Read: 69217439 HDFS Write: 250 SUCCESS
Total MapReduce CPU Time Spent: 7 seconds 120 msec
OK
{"x":-118.24762574661922,"y":46003.0}
{"x":-92.51593363544134,"y":2439.0}
{"x":-79.63134285827478,"y":328782.0}
{"x":-74.59835666349564,"y":87.0}
{"x":-43.182586669921875,"y":9.0}
{"x":-1.7777051369349177,"y":15.0}
{"x":27.917787551879883,"y":33.0}
{"x":46.25266622989736,"y":47.0}
{"x":74.89750475761217,"y":39.0}
{"x":109.85270408347802,"y":162.0}
```

"struct" data type

histogram_numeric() function

DEMO

- *Bucket U.S. into 10x10 grids using histogram_numeric*

extract column from a struct using dot notation

```
hive> select t.hist_lat.x from (select explode(histogram_numeric(lat, 10)) as hist_lat from twitter.full_text_ts_complex) t;
Query ID = root_20150223050000_18dc774b-23e0-4c99-8378-21d015377521
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1424547612900_0049, Tracking URL = http://sandbox.hortonworks.com:8088/proxy/application_1424547612900_0049/
Kill Command = /usr/hdp/2.2.0.0-2041/hadoop/bin/hadoop job  -kill job_1424547612900_0049
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2015-02-23 05:00:59,333 Stage-1 map = 0%,  reduce = 0%
2015-02-23 05:01:09,542 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 4.23 sec
2015-02-23 05:01:16,921 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 5.36 sec
MapReduce Total cumulative CPU time: 5 seconds 360 msec
Ended Job = job_1424547612900_0049
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 5.36 sec   HDFS Read: 69217439 HDFS Write: 183 SUCCESS
Total MapReduce CPU Time Spent: 5 seconds 360 msec
OK
-25.50731767926898
-7.17137844363848
3.77521472175916
13.004202445348103
18.605831107314756
28.804234052185453
34.66352003913391
40.65575122055146
45.472877604624635
55.8222710458856
```

histogram_numeric() function

DEMO

- *Bucket U.S. into 10x10 grids using histogram_numeric*


```
hive> select t.hist_lon.x
    > from (select explode(histogram_numeric(lon, 10)) as hist_lon from twitter.full_text_ts_complex) t;
Query ID = root_20150223044444_d21bcc86-4d62-43f2-87e2-a82bdb297d4c
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1424547612900_0044, Tracking URL = http://sandbox.hortonworks.com:8088/proxy/application_14245476
Kill Command = /usr/hdp/2.2.0.0-2041/hadoop/bin/hadoop job  -kill job_1424547612900_0044
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2015-02-23 04:44:52,449 Stage-1 map = 0%,  reduce = 0%
2015-02-23 04:45:02,744 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 4.23 sec
2015-02-23 04:45:12,755 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 6.45 sec
MapReduce Total cumulative CPU time: 6 seconds 450 msec
Ended Job = job_1424547612900_0044
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 6.45 sec   HDFS Read: 69217439 HDFS Write: 191 SUCCESS
Total MapReduce CPU Time Spent: 6 seconds 450 msec
OK
-118.24762574661922
-92.51593363544134
-79.63134285827478
-74.59835666349564
-43.182586669921875
-1.7777051369349177
27.917787551879883
46.25266622989736
74.89750475761217
109.85270408347802
```

histogram_numeric() function

- *Bucket U.S. into 10x10 grids using histogram_numeric*

DEMO

34

```
hive> select t1.lat, t2.lon
    > from
    > (select t.hist_lat.x as lat from (select explode(histogram_numeric(lat, 10)) as hist_lat from twitter.full_text_ts_complex) t  t1
    > JOIN
    > (select t.hist_lon.x as lon from (select explode(histogram_numeric(lon, 10)) as hist_lon from twitter.full_text_ts_complex) t  t2;
```

cross-join

```
-25.50731767926898
-7.17137844363848
3.77521472175916
13.004202445348103
18.605831107314756
28.804234052185453
34.66352003913391
40.65575122055146
45.472877604624635
55.8222710458856
```

×

```
-118.24762574661922
-92.51593363544134
-79.63134285827478
-74.59835666349564
-43.182586669921875
-1.7777051369349177
27.917787551879883
46.25266622989736
74.89750475761217
109.85270408347802
```

=

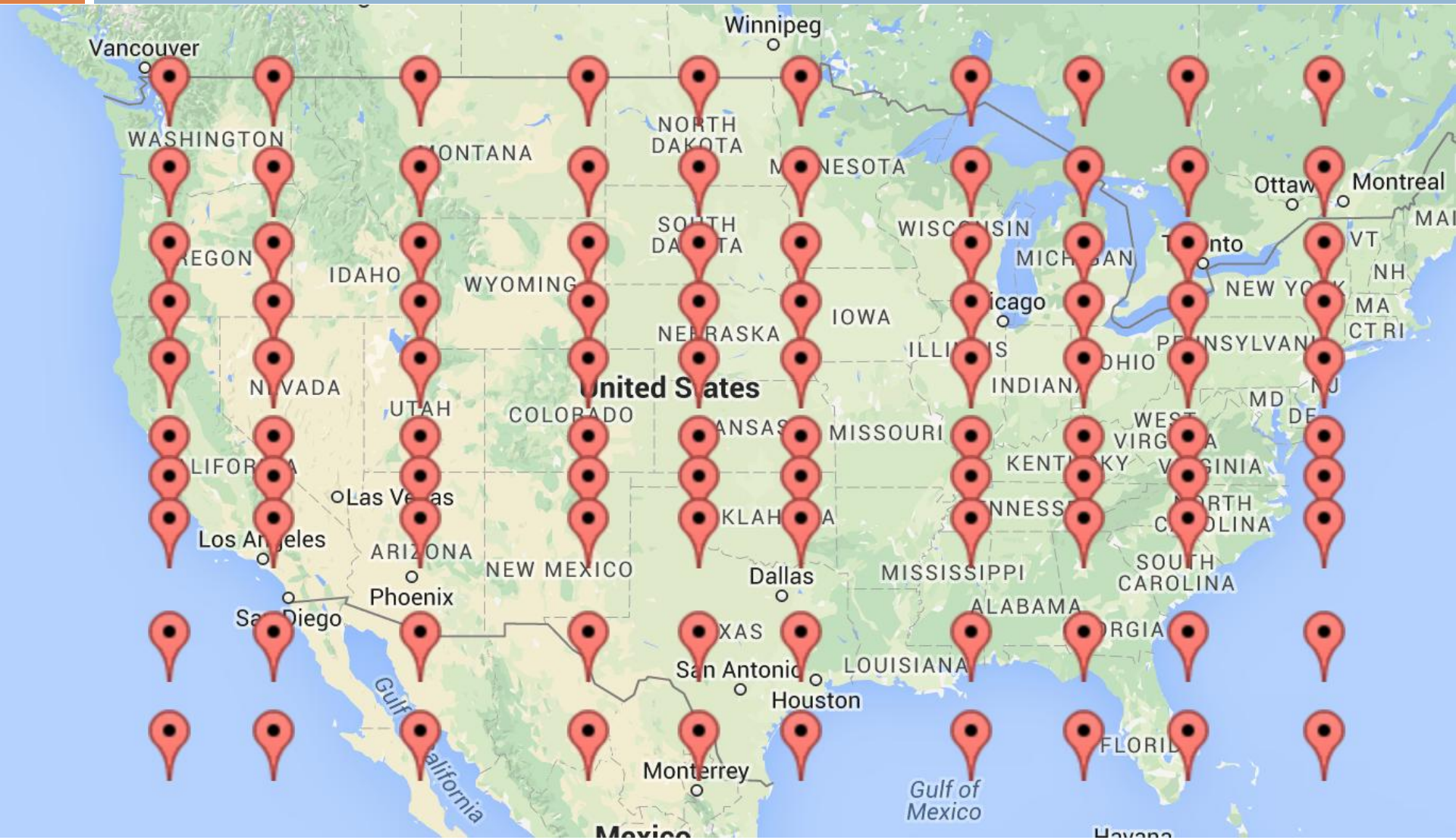
```
OK
-25.50731767926898      -118.24762574661922
-25.50731767926898      -92.51593363544134
-25.50731767926898      -79.63134285827478
-25.50731767926898      -74.59835666349564
-25.50731767926898      -43.182586669921875
-25.50731767926898      -1.7777051369349177
-25.50731767926898      27.917787551879883
-25.50731767926898      46.25266622989736
-25.50731767926898      74.89750475761217
-25.50731767926898      109.85270408347802
-7.17137844363848       -118.24762574661922
-7.17137844363848       -92.51593363544134
-7.17137844363848       -79.63134285827478
-7.17137844363848       -74.59835666349564
-7.17137844363848       -43.182586669921875
-7.17137844363848       -1.7777051369349177
-7.17137844363848       27.917787551879883
-7.17137844363848       46.25266622989736
-7.17137844363848       74.89750475761217
-7.17137844363848       109.85270408347802
3.77521472175916        -118.24762574661922
3.77521472175916        -92.51593363544134
3.77521472175916        -79.63134285827478
3.77521472175916        -74.59835666349564
3.77521472175916        -43.182586669921875
3.77521472175916        -1.7777051369349177
3.77521472175916        27.917787551879883
3.77521472175916        46.25266622989736
3.77521472175916        74.89750475761217
3.77521472175916        109.85270408347802
```

. . .

DEMO


Iceland
Norway
Russia
Canada
United Kingdom
Poland
Germany
Italy
Turkey
Kazakhstan
Mongolia
China
Iran
Iraq
Afghanistan
Pakistan
Libya
Saudi Arabia
India
Thailand
Niger
Chad
Nigeria
Ethiopia
Kenya
Tanzania
Angola
Namibia
Botswana
Madagascar
Indonesia
Mexico
Venezuela
Brazil
Peru
Bolivia
Chile
North Atlantic Ocean
South Atlantic Ocean
Indian Ocean

DEMO


Winnipeg
Vancouver
WASHINGTON
NORTH DAKOTA
IDAHO
WYOMING
IOWA
NEBRASKA
COLORADO
UTAH
NEVADA
MISSOURI
INDIANA
OHIO
Ottawa
Montreal
Las Vegas
Los Angeles
ARIZONA
Phoenix
NEW MEXICO
Dallas
San Antonio
Houston
MISSISSIPPI
ALABAMA
LOUISIANA
SOUTH CAROLINA
Monterrey
Gulf of Mexico

UDTF – Table Generating Functions

- explode()
  - transposes list/map elements into multiple rows
  - usually used with lateral_view
- collect_set
  - transposes multiple rows associated with same key to a list/map
  - usually used with group by

explode() function

DEMO

```
hive>
    > -- explode() function and lateral_view
    >    -- explode() function is often used with lateral_view
    >    -- we extracted twitter mentions from tweets in lab 4. You've probably noticed
    >    -- that it's not optimal soultion because the query we wrote didn't handle multiple
    >    -- mentions. It only extract the very first mention. A better approach is to tokenize
    >    -- the tweet first and then explode the tokens into rows and extract mentions from each token
    >
    > drop table twitter.full_text_ts_complex_1;
OK
Time taken: 0.745 seconds
hive> create table twitter.full_text_ts_complex_1 as
    > select id, ts, location_map, tweet, regexp_extract(lower(tweet_element), '(.*)@user_(\\S{8})([:| ])(.*)',2) as mention
    > from twitter.full_text_ts_complex
    > lateral view explode(split(tweet, '\\s')) tmp as tweet_element
    > where trim(regexp_extract(lower(tweet_element), '(.*)@user_(\\S{8})([:| ])(.*)',2)) != "" ;
Query ID = root_20150223053838_9836b129-baad-44a9-bea4-e05cefff3b12
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1424547612900_0053, Tracking URL = http://sandbox.hortonworks.com:8088/proxy/application_1424547612900_0053/
Kill Command = /usr/hdp/2.2.0.0-2041/hadoop/bin/hadoop job  -kill job_1424547612900_0053
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2015-02-23 05:39:12,682 Stage-1 map = 0%,  reduce = 0%
2015-02-23 05:39:42,921 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 25.09 sec
MapReduce Total cumulative CPU time: 25 seconds 90 msec
Ended Job = job_1424547612900_0053
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to: hdfs://sandbox.hortonworks.com:8020/tmp/hive/root/c09af00e-e578-46c5-9c93-818a7009cf59/hive_2015-02-23_05-38-59_013_5912830725024749079-1/-ex
Moving data to: hdfs://sandbox.hortonworks.com:8020/apps/hive/warehouse/twitter.db/full_text_ts_complex_1
Table twitter.full_text_ts_complex_1 stats: [numFiles=1, numRows=72856, totalSize=13062495, rawDataSize=12989639]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 25.09 sec   HDFS Read: 69217439 HDFS Write: 13062590 SUCCESS
Total MapReduce CPU Time Spent: 25 seconds 90 msec
OK
Time taken: 46.836 seconds
hive>
    > select * from twitter.full_text_ts_complex_1 limit 10;
OK
USER_79321756   2010-03-03 04:15:26     {"lat":"47.528139","lon":"-122.197916"} RT @USER_2ff4faca: IF SHE DO IT 1 MORE TIME......IMA KNOCK HER DAMN KOOFIE OF
f4faca
USER_79321756   2010-03-03 05:13:34     {"lat":"47.528139","lon":"-122.197916"} RT @USER_5d4d777a: YOURE A FAG FOR GETTING IN THE MIDDLE OF THIS @USER_ab059b
!!&gt;&gt;Lol! Dayum! Aye!      5d4d777a
USER_79321756   2010-03-04 01:55:55     {"lat":"47.528139","lon":"-122.197916"} RT @USER_dc5e5498: Drop and give me 50....      dc5e5498
USER_79321756   2010-03-04 06:00:09     {"lat":"47.528139","lon":"-122.197916"} RT @USER_d5d93fec: #letsbereal .. No seriously, #letsbereal&gt;&gt;lol. Don't
USER_79321756   2010-03-04 06:15:01     {"lat":"47.528139","lon":"-122.197916"} RT @USER_d5d93fec: RT @USER_79321756: RT @USER_d5d93fec: Man I don't feel lik
n do this&gt;&gt;Lol. Okay.     d5d93fec
USER_79321756   2010-03-04 06:15:01     {"lat":"47.528139","lon":"-122.197916"} RT @USER_d5d93fec: RT @USER_79321756: RT @USER_d5d93fec: Man I don't feel lik
n do this&gt;&gt;Lol. Okay.     79321756
USER_79321756   2010-03-04 06:15:01     {"lat":"47.528139","lon":"-122.197916"} RT @USER_d5d93fec: RT @USER_79321756: RT @USER_d5d93fec: Man I don't feel lik
n do this&gt;&gt;Lol. Okay.     d5d93fec
USER_79321756   2010-03-04 22:35:47     {"lat":"47.528139","lon":"-122.197916"} RT @USER_620cd4b9: @USER_79321756 I will boo, I'll just jump on her LOL&gt;&g
USER_79321756   2010-03-05 02:10:02     {"lat":"47.528139","lon":"-122.197916"} RT @USER_642c9c1b: RT @USER_9bc2644b: out of line. • Very       642c9c1b
USER_79321756   2010-03-05 02:10:02     {"lat":"47.528139","lon":"-122.197916"} RT @USER_642c9c1b: RT @USER_9bc2644b: out of line. • Very       9bc2644b
Time taken: 0.193 seconds, Fetched: 10 row(s)
```

DEMO

collect_set() function

```
hive> -- collect_set function (UDAF)
    >     -- collect_set() is a UDAF aggregation function.. we run the query at this step
    >     -- from the previous step, we get all the mentions in the tweets but if a user
    >     -- has multiple mentions in the same tweet, they are in different rows.
    >     -- To transpose all the mentions belonging to the same tweet/user, we can use
    >     -- the collect_set and group by to transpose the them into an array of mentions
    >
    > create table twitter.full_text_ts_complex_2 as
    > select id, ts, location_map, tweet, collect_list(mention) as mentions
    > from twitter.full_text_ts_complex_1
    > group by id, ts, location_map, tweet;
FAILED: SemanticException org.apache.hadoop.hive.ql.parse.SemanticException: Table already exists: twitter.full_text_ts_complex_2
hive>
    > describe twitter.full_text_ts_complex_2;
OK
id                      string
ts                      timestamp
location_map            map<string,string>
tweet                   string
mentions                array<string>
Time taken: 0.734 seconds, Fetched: 5 row(s)
hive>
    > select * from twitter.full_text_ts_complex_2
    > where size(mentions) > 5
    > limit 10;
OK
USER_3640e99a   2010-03-05 07:36:03     {"lat":"39.031235","lon":"-77.507424"}  RT @USER_1aa3e63c: RT @USER_fde41415: RT @USER_1a16
_e48989b9: #FollowFriday ? RT    ["1aa3e63c","fde41415","1a16af9f","9a51b022","32f0dfdb","35e60564","e48989b9"]
USER_57de079a   2010-03-05 17:03:13     {"lat":"38.83314","lon":"-77.003375"}   #FF: @USER_815bd484: @USER_e88cb76f: @USER_76a0eec5
_dd8aceae: @USER_a6a19994       ["815bd484","e88cb76f","76a0eec5","60bf045c","6a73e565","dd8aceae"]
USER_770f25de   2010-03-02 22:46:25     {"lat":"40.407929","lon":"-80.017267"}  SCORES: @USER_fdd57211:9pts @USER_23433069:7pts @US
:2pts @USER_e1c2dae6:2pts CONGRATS!      ["fdd57211","23433069","00792fa2","d0d5796b","8e3597ce","5f352e2d","e1c2dae6"]
USER_770f25de   2010-03-05 07:32:10     {"lat":"40.407929","lon":"-80.017267"}  SCORES: @USER_23433069:10pts @USER_fdd57211:6pts @U
979ce:1pt CONGRATS!     ["23433069","fdd57211","f2a30aae","00792fa2","5450ac50","6fb979ce"]
USER_9fe5e5c9   2010-03-05 05:38:34     {"lat":"39.390355","lon":"-76.614869"}  RT @USER_d8abac97: RT @USER_a82c4b6a: RT @USER_5ce3
_7b7d9bda: RT @USER_4fe12f93: ReTweet this tweet if ... ["d8abac97","a82c4b6a","5ce36ebf","20cf3481","4ca89b2b","fde41415","7b7d9bd
USER_de0d2dd1   2010-03-03 11:13:56     {"lat":"47.624279","lon":"-122.353836"} RT: @USER_677188e7: RT @USER_76f30351: RT @USER_550
R_7f63b76e: YG MAU DIPROMOT      ["677188e7","76f30351","5507e635","5fcad3d1","167e34bf","48ecf7d2","7f63b76e"]
USER_de0d2dd1   2010-03-05 11:16:43     {"lat":"47.624279","lon":"-122.353836"} RT: @USER_677188e7: RT @USER_e5bbb68d: RT @USER_fde
 NOW    ["677188e7","e5bbb68d","fde41415","83da799e","7b7d9bda","50c6ff2e"]
USER_de0d2dd1   2010-03-06 07:56:56     {"lat":"47.624279","lon":"-122.353836"} RT: @USER_677188e7: RT @USER_2dc1e7ef: RT @USER_151
saturday? RT    ["677188e7","2dc1e7ef","151642e4","b0c0ec37","e2f2219a","a4522881"]
USER_de0d2dd1   2010-03-06 12:41:48     {"lat":"47.624279","lon":"-122.353836"} RT @USER_677188e7: RT @USER_f5bbeee0: RT @USER_5ce3
 rt cepet       ["677188e7","f5bbeee0","5ce36ebf","5940d700","8be2ad9f","d2640f31"]
Time taken: 0.196 seconds, Fetched: 9 row(s)
```

a list of mentions in a tweet

Hive Nested Queries

DEMO

Nested Queries


```
hive>
    >
    > -- Nested queries
    >    -- *** tweets that have a lot of mentions ***
    >
    > select t.*
    > from (select id, ts, location_map, mentions, size(mentions) as num_mentions
    >       from twitter.full_text_ts_complex_2) t
    > order by t.num_mentions desc
    > limit 10;
Query ID = root_20150223055555_690e3729-f681-40ef-a186-2f960443c634
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1424547612900_0055, Tracking URL = http://sandbox.hortonworks.com:8088/proxy/application_1424547612900_0055/
Kill Command = /usr/hdp/2.2.0.0-2041/hadoop/bin/hadoop job  -kill job_1424547612900_0055
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2015-02-23 05:55:31,842 Stage-1 map = 0%,  reduce = 0%
2015-02-23 05:55:44,155 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 6.58 sec
2015-02-23 05:55:55,023 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 9.01 sec
MapReduce Total cumulative CPU time: 9 seconds 10 msec
Ended Job = job_1424547612900_0055
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 9.01 sec   HDFS Read: 11750668 HDFS Write: 1228 SUCCESS
Total MapReduce CPU Time Spent: 9 seconds 10 msec
OK
USER_9fe5e5c9   2010-03-05 05:38:34     {"lat":"39.390355","lon":"-76.614869"}  ["d8abac97","a82c4b6a","5ce36ebf","20cf3481","4ca89b2b","fde41415","7b7d9bda","4fe12f93"]       8
USER_de0d2dd1   2010-03-03 11:13:56     {"lat":"47.624279","lon":"-122.353836"} ["677188e7","76f30351","5507e635","5fcad3d1","167e34bf","48ecf7d2","7f63b76e"]  7
USER_770f25de   2010-03-02 22:46:25     {"lat":"40.407929","lon":"-80.017267"}  ["fdd57211","23433069","00792fa2","d0d5796b","8e3597ce","5f352e2d","e1c2dae6"]  7
USER_3640e99a   2010-03-05 07:36:03     {"lat":"39.031235","lon":"-77.507424"}  ["1aa3e63c","fde41415","1a16af9f","9a51b022","32f0dfdb","35e60564","e48989b9"]  7
USER_57de079a   2010-03-05 17:03:13     {"lat":"38.83314","lon":"-77.003375"}   ["815bd484","e88cb76f","76a0eec5","60bf045c","6a73e565","dd8aceae"]     6
USER_de0d2dd1   2010-03-06 07:56:56     {"lat":"47.624279","lon":"-122.353836"} ["677188e7","2dc1e7ef","151642e4","b0c0ec37","e2f2219a","a4522881"]     6
USER_de0d2dd1   2010-03-05 11:16:43     {"lat":"47.624279","lon":"-122.353836"} ["677188e7","e5bbb68d","fde41415","83da799e","7b7d9bda","50c6ff2e"]     6
USER_770f25de   2010-03-05 07:32:10     {"lat":"40.407929","lon":"-80.017267"}  ["23433069","fdd57211","f2a30aae","00792fa2","5450ac50","6fb979ce"]     6
USER_de0d2dd1   2010-03-06 12:41:48     {"lat":"47.624279","lon":"-122.353836"} ["677188e7","f5bbeee0","5ce36ebf","5940d700","8be2ad9f","d2640f31"]     6
USER_cd6c53eb   2010-03-04 13:56:05     {"lat":"39.03136","lon":"-77.507377"}   ["ab466b48","cd6c53eb","864aba30","cd6c53eb","864aba30"]        5
Time taken: 37.214 seconds, Fetched: 10 row(s)
```

Sqoop

Sqoop Example

- *check your full_text file*

DEMO

Make sure you have the full_text.txt file on your HDP local. It doesn't matter if your data is in a different folder than what's shown here.

```
[root@sandbox ~]# pwd
/root
[root@sandbox ~]# cd /home/lab
[root@sandbox lab]# cd twitter
-bash: cd: twitter: No such file or directory
[root@sandbox lab]# ll
total 125724
drwxr-xr-x 5 nagios games     4096 Jan 24 05:23 GeoText.2010-10-12
-rw-r--r-- 1 root   root  60973289 Jan 16 21:47 GeoText.2010-10-12.tgz
-rw-r--r-- 1 root   root   4994090 Jan 30 05:02 cities15000.txt
-rw-r--r-- 1 root   root       115 Jan 30 05:03 dayofweek.txt
-rw-r--r-- 1 root   root  57139942 Jan 16 15:34 full_text.txt
-rwxrwxr-- 1 root   root      1027 Jan 23 21:38 sc_reducer.py
-rw-r--r-- 1 root   root   5589917 Jan 30 05:03 shakespeare.txt
-rw-r--r-- 1 root   root     13880 Jan 30 05:03 timeZones.txt
-rwxrwxr-- 1 root   root       537 Jan 23 21:38 wc_mapper.py
[root@sandbox lab]# cd /home/lab
[root@sandbox lab]# ll
total 125724
drwxr-xr-x 5 nagios games     4096 Jan 24 05:23 GeoText.2010-10-12
-rw-r--r-- 1 root   root  60973289 Jan 16 21:47 GeoText.2010-10-12.tgz
-rw-r--r-- 1 root   root   4994090 Jan 30 05:02 cities15000.txt
-rw-r--r-- 1 root   root       115 Jan 30 05:03 dayofweek.txt
-rw-r--r-- 1 root   root  57139942 Jan 16 15:34 full_text.txt
-rwxrwxr-- 1 root   root      1027 Jan 23 21:38 sc_reducer.py
-rw-r--r-- 1 root   root   5589917 Jan 30 05:03 shakespeare.txt
-rw-r--r-- 1 root   root     13880 Jan 30 05:03 timeZones.txt
-rwxrwxr-- 1 root   root       537 Jan 23 21:38 wc_mapper.py
[root@sandbox lab]# cat full_text.txt | head
USER_79321756   2010-03-03T04:15:26      ÜT: 47.528139,-122.197916       47.528139       -122.197916     RT @USER_2ff4faca: IF SHE DO IT 1 MORE TIME..
&gt;haha. #cutthatout
USER_79321756   2010-03-03T04:55:32      ÜT: 47.528139,-122.197916       47.528139       -122.197916     @USER_77a4822d @USER_2ff4faca okay:) lol. Say
USER_79321756   2010-03-03T05:13:34      ÜT: 47.528139,-122.197916       47.528139       -122.197916     RT @USER_5d4d777a: YOURE A FAG FOR GETTING IN
OU ? A FUCKING NOBODY !!!!&gt;&gt;Lol! Dayum! Aye!
USER_79321756   2010-03-03T05:28:02      ÜT: 47.528139,-122.197916       47.528139       -122.197916     @USER_77a4822d yea ok..well answer that cheap
USER_79321756   2010-03-03T05:56:13      ÜT: 47.528139,-122.197916       47.528139       -122.197916     A sprite can disappear in her mouth - lil kim
USER_79321756   2010-03-03T16:52:44      ÜT: 47.528139,-122.197916       47.528139       -122.197916     Lmao! I still get txt when AJ tweets before t
s me dyin! @USER_a5b463b2 what's ur issue!
USER_79321756   2010-03-03T16:57:24      ÜT: 47.528139,-122.197916       47.528139       -122.197916     Alright twitters tryna take me over!
USER_79321756   2010-03-03T20:20:40      ÜT: 47.528139,-122.197916       47.528139       -122.197916     Just got to work. Got my pizza bagel and my r
 not til 2. I just wanna get it done!:D
USER_79321756   2010-03-03T23:23:33      ÜT: 47.528139,-122.197916       47.528139       -122.197916     Just got a txt from my cousin! Yes! So happy
USER_79321756   2010-03-03T23:37:36      ÜT: 47.528139,-122.197916       47.528139       -122.197916     Why is this woman in the bathroom everytime I
```

Sqoop Example DEMO

- load data into mysql table

```
[root@sandbox lab]# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 946
Server version: 5.1.73 Source distribution

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| hive               |
| mysql              |
| ranger             |
| ranger_audit       |
| test               |
+--------------------+
6 rows in set (0.00 sec)

mysql> create database twitter;
Query OK, 1 row affected (0.00 sec)

mysql> create table twitter.full_text_mysql (id varchar(20), ts varchar(20), location varchar(20), lat varchar(20), lon varchar(20), tweet varchar(300));
Query OK, 0 rows affected (0.01 sec)

mysql> LOAD DATA INFILE '/home/lab/full_text.txt' INTO TABLE twitter.full_text_mysql FIELDS TERMINATED BY '\t';
Query OK, 377525 rows affected, 65535 warnings (1.08 sec)
Records: 377525  Deleted: 0  Skipped: 0  Warnings: 375579

mysql> describe twitter.full_text_mysql;
+----------+--------------+------+-----+---------+-------+
| Field    | Type         | Null | Key | Default | Extra |
+----------+--------------+------+-----+---------+-------+
| id       | varchar(20)  | YES  |     | NULL    |       |
| ts       | varchar(20)  | YES  |     | NULL    |       |
| location | varchar(20)  | YES  |     | NULL    |       |
| lat      | varchar(20)  | YES  |     | NULL    |       |
| lon      | varchar(20)  | YES  |     | NULL    |       |
| tweet    | varchar(300) | YES  |     | NULL    |       |
+----------+--------------+------+-----+---------+-------+
6 rows in set (0.00 sec)

mysql> select * from twitter.full_text_mysql limit 3;
+---------------+---------------------+---------------------+-----------+-------------+-----------------------------------------------------------------------------------------------------------------------------------
-----------------------------------+
| id            | ts                  | location            | lat       | lon         | tweet
                                   |
+---------------+---------------------+---------------------+-----------+-------------+-----------------------------------------------------------------------------------------------------------------------------------
-----------------------------------+
| USER_79321756 | 2010-03-03T04:15:26 | ÜT: 47.528139,-122. | 47.528139 | -122.197916 | RT @USER_2ff4faca: IF SHE DO IT 1 MORE TIME.....IMA KNOCK HER DAMN KO(
atout                              |
| USER_79321756 | 2010-03-03T04:55:32 | ÜT: 47.528139,-122. | 47.528139 | -122.197916 | @USER_77a4822d @USER_2ff4faca okay:) lol. Saying ok to both of yall abo
                                   |
| USER_79321756 | 2010-03-03T05:13:34 | ÜT: 47.528139,-122. | 47.528139 | -122.197916 | RT @USER_5d4d777a: YOURE A FAG FOR GETTING IN THE MIDDLE OF THIS @USER_
OBODY !!!!&gt;&gt;Lol! Dayum! Aye! |
```

Load the tweet data into a mysql table. We will sqoop this table to hive in the next step

NOTE: you're in mysql prompt, not in Hive ☺

Sqoop Example

- sqoop import table from mysql to hive

You now have a table named twitter.full_text_mysql in Hive!

DEMO


```
[root@sandbox etc]# sqoop import -m 1 --connect jdbc:mysql://0.0.0.0:3306/twitter --username=root --password=
--table full_text_mysql --columns "id, ts, location" --map-column-hive id=string,ts=string,location=string --h
ive-import --fields-terminated-by '\t'  --hive-table twitter.full_text_mysql  --warehouse-dir /user/twitter/fu
ll_text_mysql
Warning: /usr/hdp/2.2.0.0-2041/accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
15/02/23 07:13:03 INFO sqoop.Sqoop: Running Sqoop version: 1.4.5.2.2.0.0-2041
15/02/23 07:13:03 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider usi
ng -P instead.
15/02/23 07:13:04 INFO manager.SqlManager: Using default fetchSize of 1000
15/02/23 07:13:04 INFO tool.CodeGenTool: Beginning code generation
15/02/23 07:13:04 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `full_text_mysql` AS t LIM
IT 1
15/02/23 07:13:04 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `full_text_mysql` AS t LIM
IT 1
15/02/23 07:13:04 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/hdp/2.2.0.0-2041/hadoop-mapreduce
Note: /tmp/sqoop-root/compile/ba0a7fff224e1d7c23bc80b31fde7bd8/full_text_mysql.java uses or overrides a deprec
ated API.
Note: Recompile with -Xlint:deprecation for details.
15/02/23 07:13:08 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-root/compile/ba0a7fff224e1d7c23bc8
0b31fde7bd8/full_text_mysql.jar
15/02/23 07:13:08 WARN manager.MySQLManager: It looks like you are importing from mysql.
15/02/23 07:13:08 WARN manager.MySQLManager: This transfer can be faster! Use the --direct
15/02/23 07:13:08 WARN manager.MySQLManager: option to exercise a MySQL-specific fast path.
15/02/23 07:13:08 INFO manager.MySQLManager: Setting zero DATETIME behavior to convertToNull (mysql)
15/02/23 07:13:08 INFO mapreduce.ImportJobBase: Beginning import of full_text_mysql
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/hdp/2.2.0.0-2041/hadoop/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/Sta
ticLoggerBinder.class]
```

Lab 5

Lab 5 Exercises

- The script for lab 5 has been uploaded to the BlackBoard
  - *Session5-Advanced-Hive-Script.txt*
- See inline comments for descriptions of each step
  - See demos in this presentation for output of each step

Additional Exercises

- Find the home city of each twitter user in the geo-tagged tweet dataset
- [hint]
  - Use the geonames dataset for city←→ lat/lon mapping
    - The geonames data has been uploaded to the BlackBoard under files section
  - Use replicated/map-side joins to map user location to a list of nearby cities
  - Find the closest city as the city of the tweet location
  - Aggregate the user city location history and find home city (in this case could be the most frequent city)