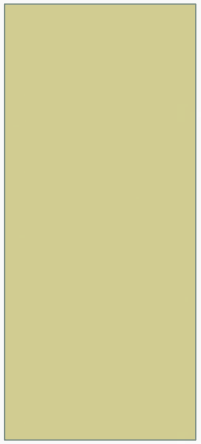
PROCESSING BIG DATA WITH HADOOP

AHMET DEMIRELLI



OUTLINE

- Why is Hadoop?
- What is Hadoop?
- When is Hadoop?
- Where is Hadoop?
- How is Hadoop?

WHY HADOOP?

- Databases can process GBs of data
- TBs is too much for a RDBMS
- Solution
  - Faster processors & More Memory ??
  - Distributed Systems
    - Synchronisation ?
    - Bandwith ?
    - Node Failure ?
    - Job fails ?

WHAT IS HADOOP?

- http://hadoop.apache.org
- Based on papers published by Google
- GFS - http://research.google.com/archive/gfs.html
- MapReduce - http://research.google.com/archive/mapreduce.html
- Implemented by Apache Software Foundation
- A Platform provides both
  - Distributed storage (HDFS)
  - Distributed computation (MapReduce)

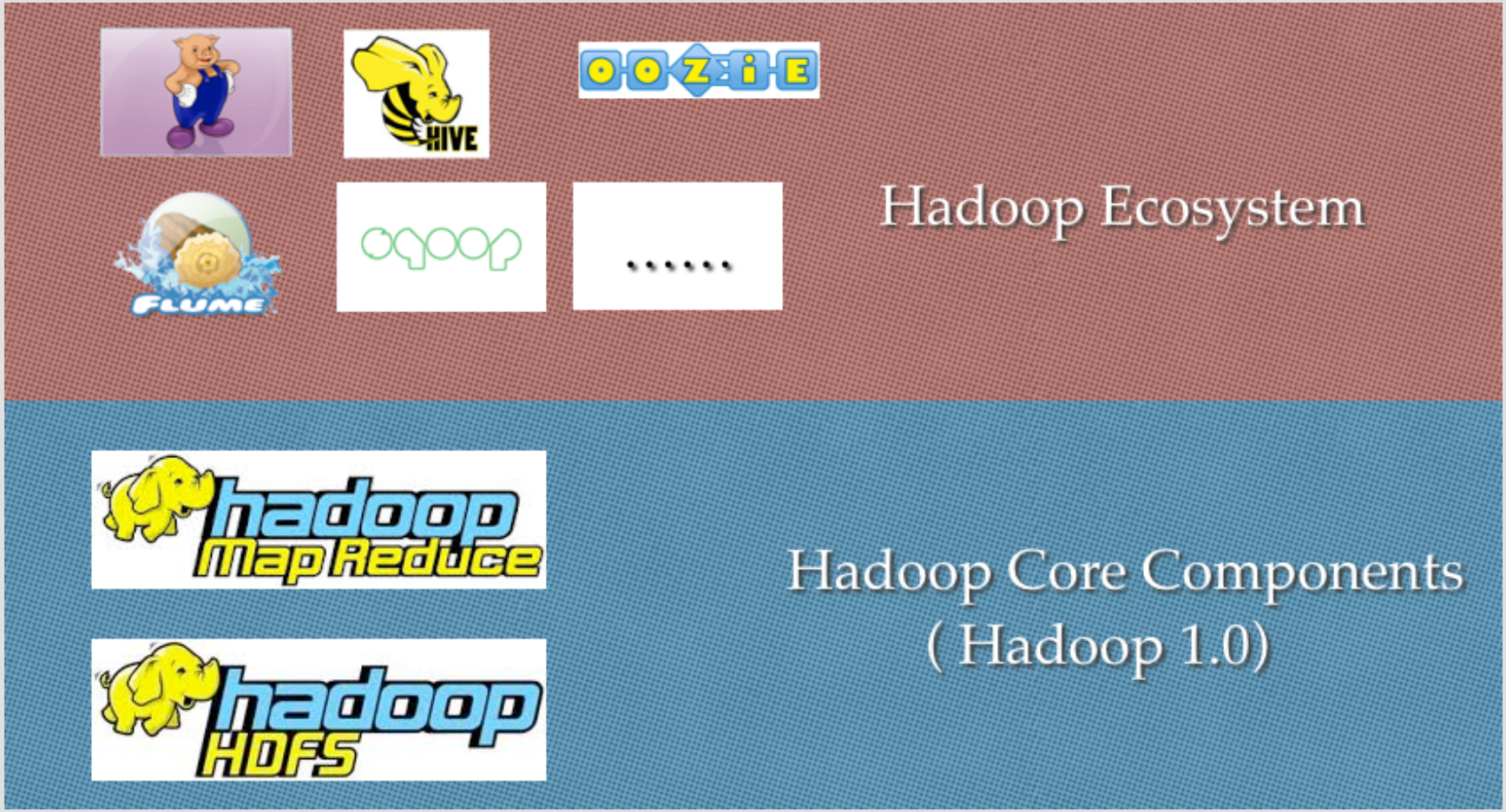
WHAT IS HADOOP?

- Application developed in high level Programming Language (Java,Python,Scala,Pig, Hive..etc)
- Very little communication between nodes
- Data is distributed and replicated in advance (HDFS)
- Data is processed on the stored location
- Hadoop scalable and fault tolerant

WHAT IS HADOOP?

- Scalable
  - Adding new nodes increases the system capability proportionally
- Fault Tolerance (If one of the nodes goes down)
  - No loss of data (replication)
  - No loss of tasks (failed task will be reassigned to another node)
  - Recovered nodes will rejoin to the cluster

WHAT IS HADOOP?

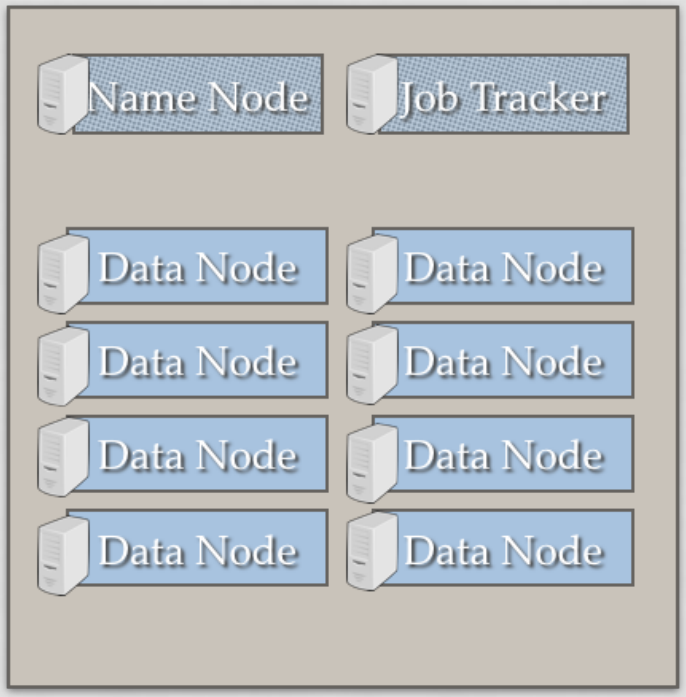


WHAT IS HADOOP?

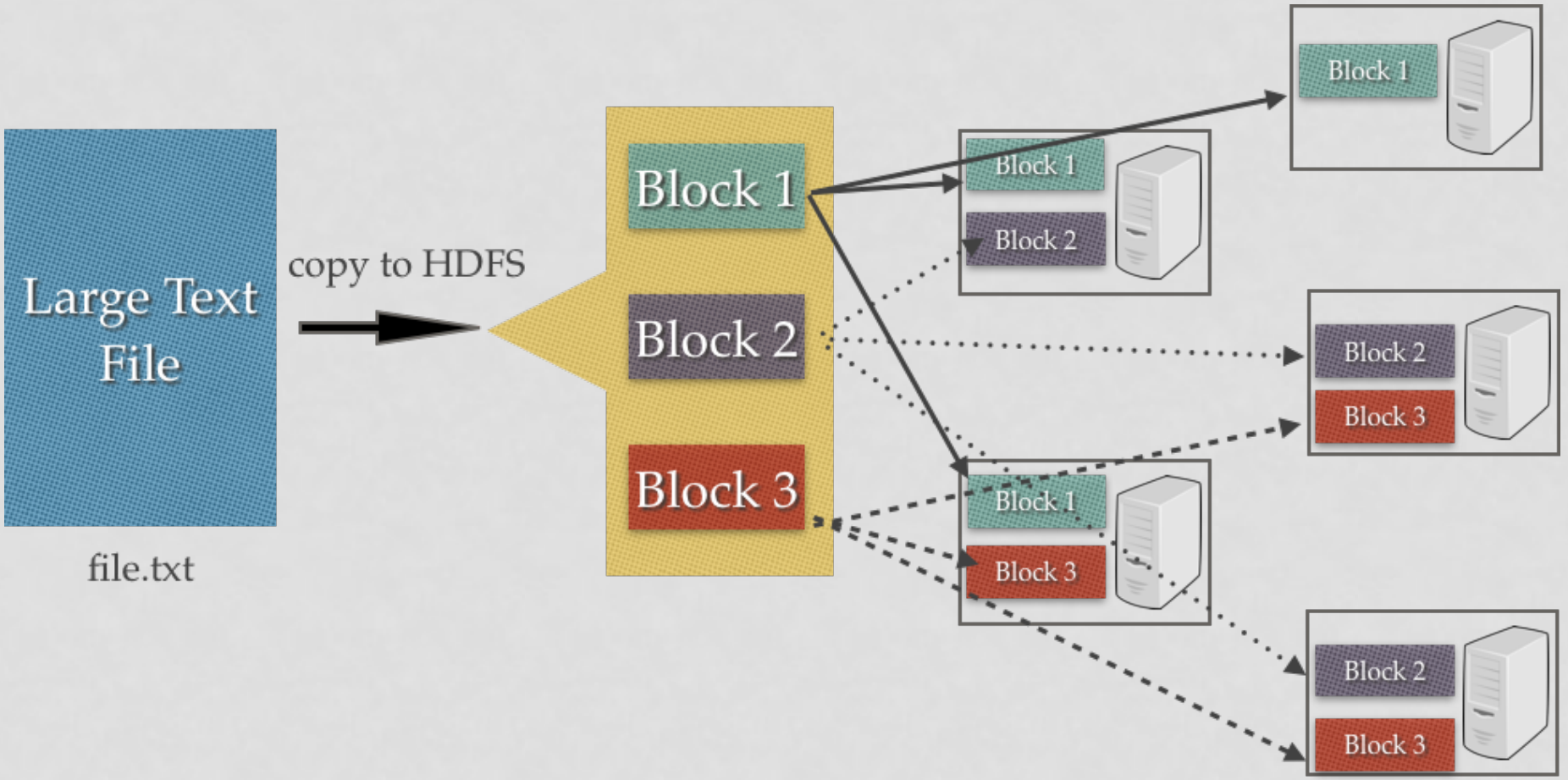
- HDFS (Hadoop Distributed File System)
  - Stores data on the cluster
  - A File system written in Java
  - Runs on top of native file system (like ext3,ext4,xfs)
  - Designed for dealing with massive amount of data
  - Keeps files Read-Only (for performance)
  - Performs best with large files (100MB or more)
  - Each block is replicated on 3 different nodes by default
  - Information about the files and locks (Metadata) is kept on the Name Node
- MapReduce
  - Processes data on the cluster
  - Manages Jobs

WHAT IS HADOOP?

- Hadoop Cluster : Group of computers working together store and process the big data
- Hadoop Cluster Nodes (Hadoop 1.0)
  - master nodes
    - cluster has two master nodes
      - Name Node (manages HDFS)
      - Job Tracker (manages MapReduce)
  - slave (worker) nodes
      - HDFS
      - MapReduce



WHAT IS HADOOP?



WHEN IS HADOOP?

- You have massive data (larger that GBs)
- Or operations that takes time on regular data
- And most importantly you have a parallelizable algorithm
- Or a better motivation
  - To get a good grade from this course ☺

WHERE IS HADOOP?

- You can install from scratch (Bad idea)
- You can use a distribution (Easy Install)
  - Cloudera
  - Hortonworks
  - MapR
- You can download quick-start-vm from (No install)
  - Cloudera
  - Hortonworks

WHERE IS HADOOP?

- Google Cloud
  - http://cloud.google.com
- Amazon EMR
  - https://aws.amazon.com/elasticmapreduce
- Microsoft Azure
  - https://azure.microsoft.com
- Ibm Bluemix
  - https://console.ng.bluemix.net/catalog
- Oracle Cloud
  - https://cloud.oracle.com/en_US/bigdata?tabID=1448073786401

HOW IS HADOOP?

- Demos
  - MapRaduce (Java)
  - Hive
  - Pig
  - Spark (Python)

MAP REDUCE

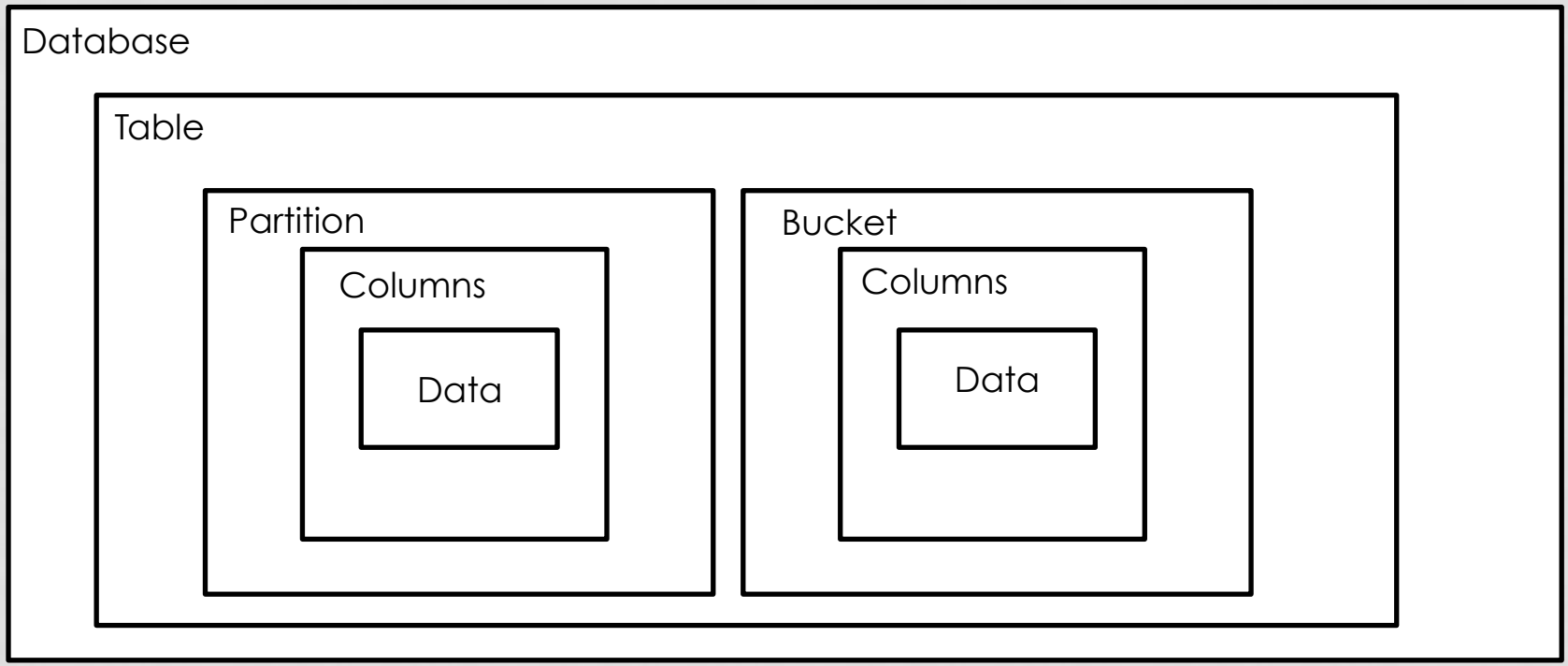
- Framework for processing data parallel on HDFS
- Written in Java
- Map -> Shuffle ->Reduce steps

HIVE

- http://hive.apache.org
- Data warehouse system for Hadoop
- ETL operations
- Easy data summarization and analysis of large volumes of data.
- Hive QL (based on SQL)
- Query execution in MapReduce
- Compiled in to MapReduce jobs (limitations)
- Runs with structured data (like csv)

HIVE
DATA UNITS

- Databases ,Tables , Partitions , Buckets (or Clusters)



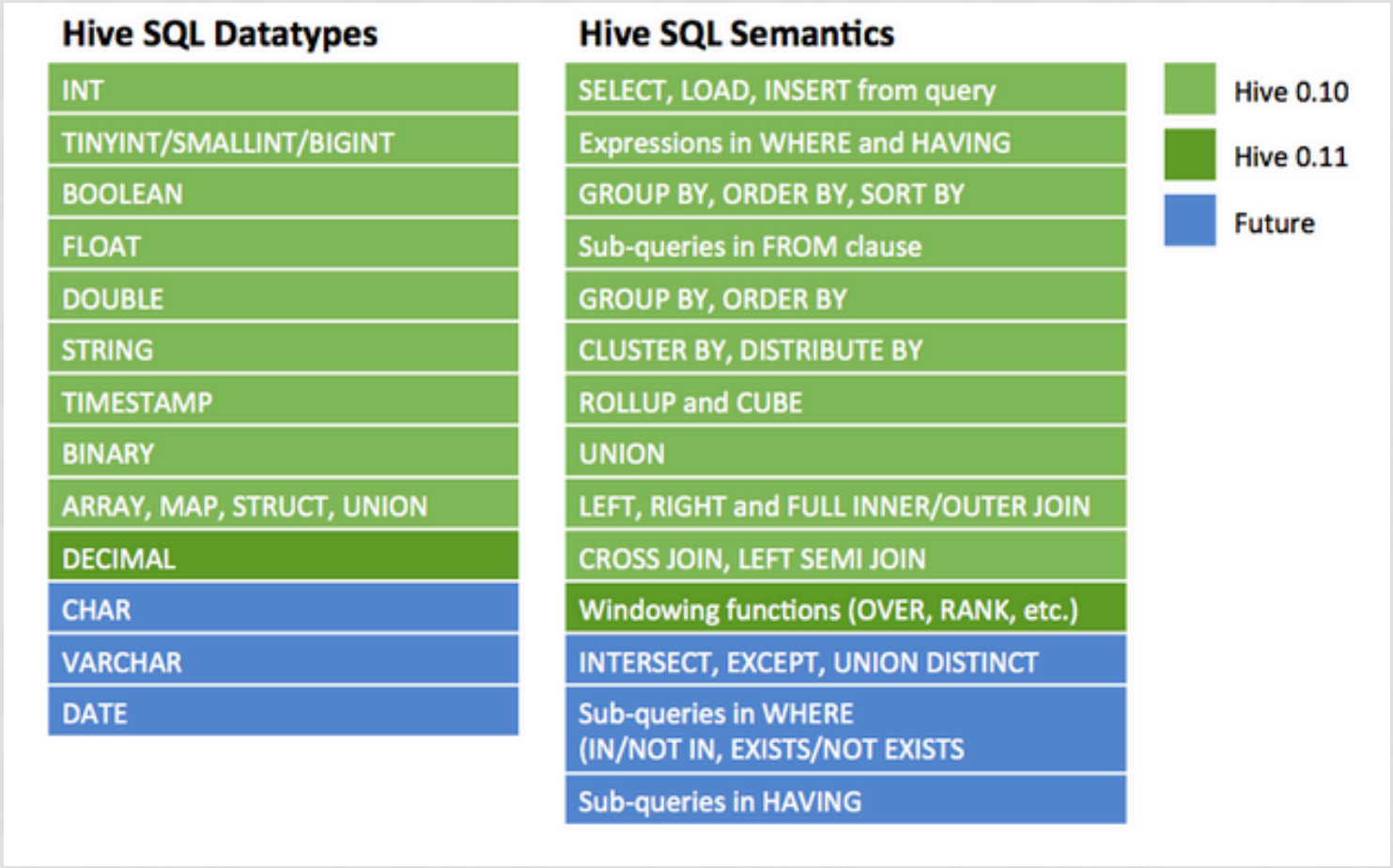
HIVE
PRIMITIVE TYPES

- Integers
  - TINYINT - 1 byte integer
  - SMALLINT - 2 byte integer
  - INT - 4 byte integer
  - BIGINT - 8 byte integer
- Boolean type
  - BOOLEAN - TRUE/FALSE
- Floating point numbers
  - FLOAT - single precision
  - DOUBLE - Double precision
- String type
  - STRING - sequence of characters in a specified character set

HIVE
HIVE QL

- Create tables, views and indexes
- Select ,where , group by, order by, joins
- No complex, correlated sub queries
- Not for small amounts of data
- No union, intersect, distinct queries

HIVE
LIKE SQL



*http://hortonworks.com/blog/hive-cheat-sheet-for-sql-users/

HIVE
LIKE SQL

Metadata

Function	MySQL	Hive
Selecting a database	`USE database;`	`USE database;`
Listing databases	`SHOW DATABASES;`	`SHOW DATABASES;`
Listing tables in a database	`SHOW TABLES;`	`SHOW TABLES;`
Describing the format of a table	`DESCRIBE table;`	`DESCRIBE (FORMATTED|EXTENDED) table;`
Creating a database	`CREATE DATABASE db_name;`	`CREATE DATABASE db_name;`
Dropping a database	`DROP DATABASE db_name;`	`DROP DATABASE db_name (CASCADE);`

*http://hortonworks.com/blog/hive-cheat-sheet-for-sql-users/

HIVE
LIKE SQL

Function	MySQL	Hive
Retrieving Information (General)	`SELECT from_columns FROM table WHERE conditions;`	`SELECT from_columns FROM table WHERE conditions;`
Retrieving All Values	`SELECT * FROM table;`	`SELECT * FROM table;`
Retrieving Some Values	`SELECT * FROM table WHERE rec_name = "value";`	`SELECT * FROM table WHERE rec_name = "value";`
Retrieving With Multiple Criteria	`SELECT * FROM TABLE WHERE rec1 = "value1" AND rec2 = "value2";`	`SELECT * FROM TABLE WHERE rec1 = "value1" AND rec2 = "value2";`
Retrieving Specific Columns	`SELECT column_name FROM table;`	`SELECT column_name FROM table;`
Retrieving Unique Output	`SELECT DISTINCT column_name FROM table;`	`SELECT DISTINCT column_name FROM table;`
Sorting	`SELECT col1, col2 FROM table ORDER BY col2;`	`SELECT col1, col2 FROM table ORDER BY col2;`
Sorting Reverse	`SELECT col1, col2 FROM table ORDER BY col2 DESC;`	`SELECT col1, col2 FROM table ORDER BY col2 DESC;`
Counting Rows	`SELECT COUNT(*) FROM table;`	`SELECT COUNT(*) FROM table;`
Grouping With Counting	`SELECT owner, COUNT(*) FROM table GROUP BY owner;`	`SELECT owner, COUNT(*) FROM table GROUP BY owner;`
Maximum Value	`SELECT MAX(col_name) AS label FROM table;`	`SELECT MAX(col_name) AS label FROM table;`
Selecting from multiple tables (Join same table using alias w/"AS")	`SELECT pet.name, comment FROM pet, event WHERE pet.name = event.name;`	`SELECT pet.name, comment FROM pet JOIN event ON (pet.name = event.name)`

*http://hortonworks.com/blog/hive-cheat-sheet-for-sql-users/

HIVE
RUN HIVE QUERIES

Command Line

Function	Hive
Run Query	`hive -e 'select a.col from tab1 a'`
Run Query Silent Mode	`hive -S -e 'select a.col from tab1 a'`
Set Hive Config Variables	`hive -e 'select a.col from tab1 a' -hiveconf hive.root.logger=DEBUG,console`
Use Initialization Script	`hive -i initialize.sql`
Run Non-Interactive Script	`hive -f script.sql`

*http://hortonworks.com/blog/hive-cheat-sheet-for-sql-users/

HIVE
SAMPLE DATASET

- 144 MB
- 22,000,000 ratings and 580,000 tag applications applied to 33,000 movies by 240,000 users. Last updated 1/2016
- The data are contained in four files,
  - movies.csv (movieId,title,genres)
  - ratings.csv (userId,movieId,rating,timestamp)
  - tags.csv (userId,movieId,tag,timestamp)
  - links.csv (movieId,imdbId,tmdbId)
    - movieId is an identifier for movies used by https://movielens.org. E.g., the movie Toy Story has the link https://movielens.org/movies/1.
    - imdbId is an identifier for movies used by http://www.imdb.com. E.g., the movie Toy Story has the link http://www.imdb.com/title/tt0114709/.
    - tmdbId is an identifier for movies used by https://www.themoviedb.org. E.g., the movie Toy Story has the link https://www.themoviedb.org/movie/862.

*http://grouplens.org/datasets/movielens/

HIVE
SAMPLE DATASET

- Upload data to your Google Cloud Bucket
- Copy data from you bucket to master instance
  - gsutil cp gs://<your-bucket>/yourfilename.csv .
- Create Table
  - hive> CREATE TABLE ratings (userid STRING, movieid STRING, rating DOUBLE, unixtime STRING) ROW FORMAT DELIMITED FIELDS TERMINATED BY ' ,' STORED AS TEXTFILE;
- Load data
  - hive> LOAD DATA LOCAL INPATH ' /usr/ahmet/ratings.csv' OVERWRITE INTO TABLE ratings;
- Query data
  - hive> SELECT COUNT(*) FROM ratings;

HIVE ACCESS

- Hive Shell
- Java, Python usind JDBC Driver
- .NET ODBC Driver
- Apache Thrift Client

PIG

- http://pig.apache.org
- Pig Engine parses, optimizes and runs scripts as MapReduce jobs
- Pig-Latin high level scripting language
- ETL operations
- No need for Java, Scala or Python knowledge
- Is able to store data at any point during a pipe line

PIG COMMANDS

- Load data
  - Movies = LOAD '/user/hadoop/movies.txt' USING PigStorage(',') as (id,name,year,rating,duration);
- Load data with types
  - Movies = LOAD '/user/hadoop/movies.txt' USING PigStorage(',') as (id:int,name:chararray,year:int,rating:float, duration:int);
- Check format of data
  - Describe Movies;
- Filter
  - filtered= FILTER Movies BY rating>3.5;

PIG COMMANDS

- Select part of data and assign to another variable
  - selected= **foreach** filtered **generate** year,rating,name;
  - Dump selected
- Store processed data to HDFS
  - STORE slectedINTO '/user/hadoop/processed.csv' USING PigStorage (',');

APACHE SPARK

- Word count example