

HIVE



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

What is HIVE?

- Data warehousing system for hadoop
- Run SQL like queries that get compiled and run as Map Reduce jobs.
- Displays the result back to the user
- Data is hadoop over though generally unstructured has some vague structure associated with it



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

Why Hadoop?

- To ease the use of hadoop file system and Map Reduce for non developers. User as scientist, analysts etc, who needs to know SQL syntax
- Writing SQL is harder than writing code



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

HIVE Features

- Create table, create view, create index – DDL
- Select, where clause, group by, order by and joins – DML
- Pluggable Input/Output format
- Pluggable:
 - User Defined Functions – UDF
 - User Defined Aggregate Functions –UDAf
 - User Defined Table Functions – UDT
- Pluggable Serializable Deserializable batches



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

What HIVI is not?

- It is not RDBMS
- Difficult to handle due to Inconsistency
- Correlated sub queries
- Even with small amount of data time taken to return the response can be comparable to RDBMS.



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

Connecting to Hadoop

- HDFS Shell
- JDBC Driver
- ODBC Driver
- Email Client



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163

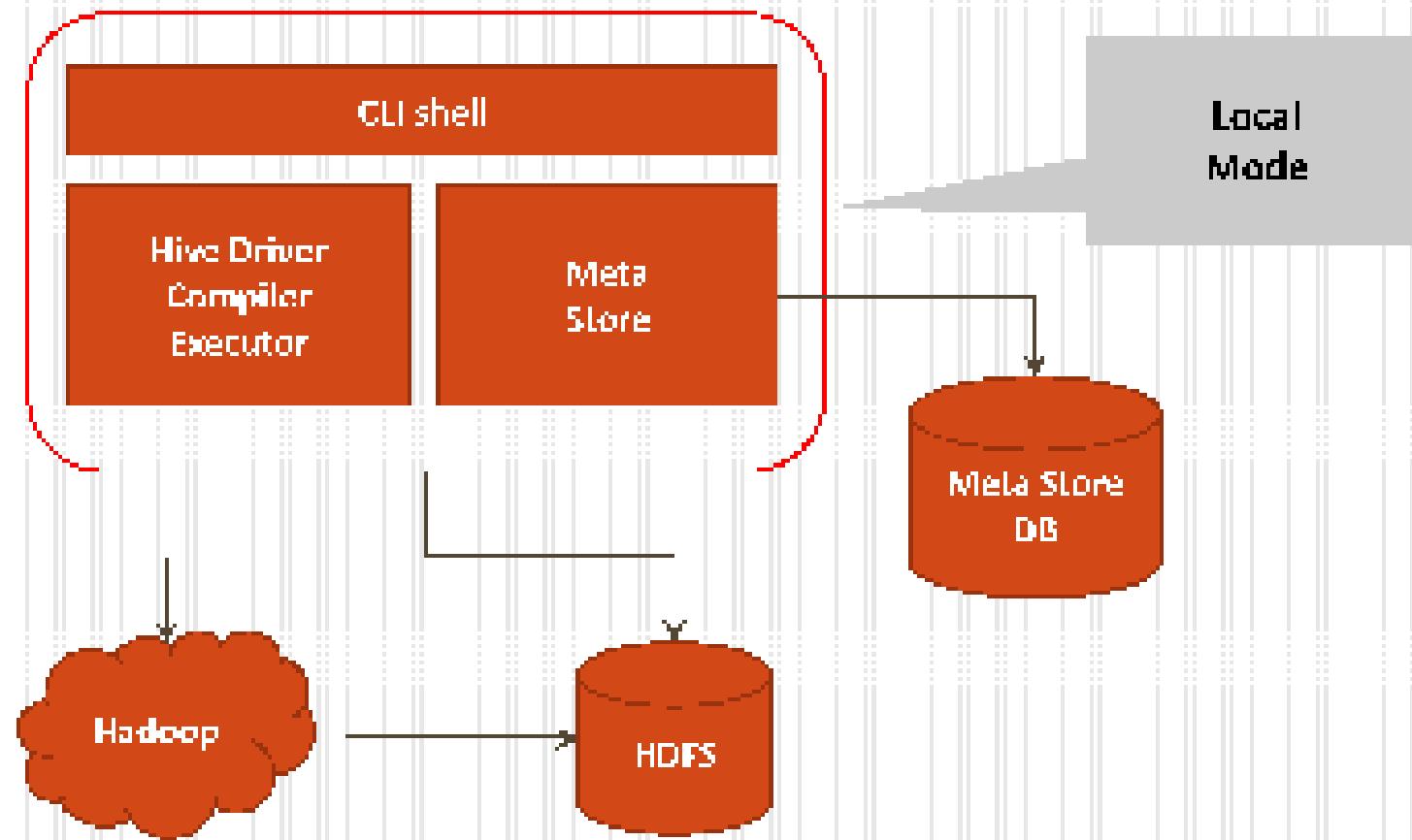


Skype ID : keylabstraining



info@keylabstraining.com

HIVE Architecture



HIVE MetaStore

- Persistent Schema, i.e. table definition (table name, columns, types)
- Location of table files
- Row format of table files
- Storage format - columns



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

How to create table in Hive ?

- CREATE TABLE t (column1 DATA TYPE,...)
 - ROW FORMAT IS DELIMITED
 - FILE FORMAT IS MINC HD BY char
 - STORED AS TEXTFILE PARSED..



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

Example create table

- Create table with all the employees (ie. ID, name, salary, Dept) in ROW FORMAT DELIMITED BY ',' STORED AS TEXTFILE.

```
CREATE TABLE emp (
    id INT,
    name STRING,
    salary DOUBLE,
    dept STRING
)
ROW FORMAT DELIMITED BY ',' 
STORED AS TEXTFILE;
```



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

Configuring hive on vm(hive-site.xml)

```
<?xml version="1.0"?>
<xslstylesheet type="text/xml" href="configuration.xml"?>

<configuration>

<!-- Hive Configuration can either be stored in this file or in the hadoop configuration files -->
<!-- That are applied by Mapred config variables. -->
<!-- Aside from Mapred config variables - this file is provided as a convenience so that Hive -->
<!-- users do not have to edit hadoop configuration files (that may be managed as a centralized -->
<!-- resource). -->

<!-- Hive Execution Parameters -->

<property>
  <name>javax.jdo.option.ConnectionURL</name>
  <value>jdbc:mysql://localhost:3306/hive?useUnicode=true&characterEncoding=UTF-8&autoReconnect=true</value>
  <description>JDBC connect string for a JDBC metastore</description>
</property>

<property>
  <name>javax.jdo.option.ConnectionClassName</name>
  <value>org.apache.derby.jdbc.EmbeddedDriver</value>
  <description>Driver class name for a JDBC metastore</description>
</property>

<property>
  <name>hive.root.logger.name</name>
  <value>/user/hive/lib/hive/lib/hive-hwi-0.7.0-cdh5.11.2-warc</value>
  <description>This is the HMR file with the jar content for Hive Web Interface</description>
</property>

</configuration>
-
```



USA : +1-908-366-7933



Skype ID : keylabstraining

Hyderabad : +91-9550645679
Bangalore : +91-8884584163



info@keylabstraining.com

HealthCheck

- After configuration, type command `show tables`. If you get output without any error, then it's correct and properly.
- The logs of hbase or debugging can be checked at `/tmp/$(username)/hbase`. If you did sudo then the logs will be at `/tmp/monty/hbase` otherwise `/tmp/$(username)/hbase`.



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

Troubleshoot hive

```
Hive> show databases;
FAILED: Error in metadata: java.sql.SQLException: Failed to start database '/home/cloudera/Desktop/metastore/metastore_db', see the next exception for details.
HadoopIOExceptions:
java.sql.SQLException: Failed to start database '/home/cloudera/Desktop/metastore/metastore_db', see the next exception for details.
FAILED: Execution Error, return code 1 from org.apache.hadoop.hive.jdbc.HQLParser
hive> quit;
root@cloudera-vm:/home/cloudera$ ls
cloudera derby.log imp.jar metastore prijarr schema.jar serc.jar
data Desktop my.jar part-c-00000 prijars.jar ssoo.jar
root@cloudera-vm:/home/cloudera$ cd Desktop/
root@cloudera-vm:/home/cloudera/Desktop$ ls
derby.log Learn_Hadoop.desktop metastore prijarr README.txt
root@cloudera-vm:/home/cloudera/Desktop$ cd metastore/
root@cloudera-vm:/home/cloudera/Desktop/metastore$ ls
metastore_db
root@cloudera-vm:/home/cloudera/Desktop/metastore$ cd metastore_db/
root@cloudera-vm:/home/cloudera/Desktop/metastore/metastore_db$ ls
dbms-lock-dir.log metastore.properties temp
root@cloudera-vm:/home/cloudera/Desktop/metastore/metastore_db$ rm -rf *.log
root@cloudera-vm:/home/cloudera/Desktop/metastore/metastore_db$ sudo hadoop
hive history file=/tmp/root/hive_job_log_root_201312111223_1902402719.txt
hive> show databases;
ok
default
Time taken: 3.247 seconds
hive> show tables;
ok
employee
Time taken: 0.061 seconds
```



USA : +1-908-366-7933
Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining
info@keylabstraining.com

Check table

```
hive> show tables;
OK
employee
Time taken: 0.561 seconds
hive> describe employee;
OK
id      int
name    string
salary  double
Time taken: 0.692 seconds
hive> [REDACTED]
```



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

External table

- If the user wants to change the default directory to some specific existing directory in HDFS, by using the command `jar` or `load`
- Create external table `t1as1` ...1 row for each committed file terminated by '\n' which stored at hdfs location '`location`'



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

Case study

- Create several table sorted by one or both an sorted output (done in MySQL).
- Display all the data using select statement.



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

Case study 1



Data types

- Non column data types can have any value. SQL column types as well as complex data types like Maps, Arrays, Struct



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

Syntax of collections

- Create table `employee (id int, name string, salary double, skills Array<String>)`
 - ROW FORMAT IS TINYINT
 - MAX LENGTH IS 100 BY DEFAULT
 - COLUMN SEPARATOR IS TERMINATED BY ','
 - STORED AS TEXTFILE



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

Delete table

- Dropping table deletes tables directory
- Dropping external table only deletes the metadata and not the directory



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

Writing output of query

- INSERT OVERWRITE OR REPLACE
- To write result in - DBS
- LOCAL to write on local FS



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

```
hive> msck repair table /princ select * from hivesubscribers limit 10;
Total MapReduce 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.mapred.reducer.bytes.per.reducer=number>
In order to limit the maximum number of reducers:
  set hive.mapred.reducers.max=number>
In order to set a concurrent number of reducers:
  set mapred.reducer.tasks=number>
Starting Job = job_201312130547_0003, Tracking URL = http://elasticsearch-vm:8030/jobsdetails?jobid=job_201312130547_0003
Kill Command = /usr/lib/hadoop/bin/hadoop job -Dmapred.job.tracker=elasticsearch-vm:8021 -kill job_201312130547_0003
2013-12-13 06:28:44,076 Stage-0 map = 0%,  reduce = 0%
2013-12-13 06:28:44,076 Stage-1 map = 0%,  reduce = 0%
2013-12-13 06:30:19,218 Stage-0 map = 0%,  reduce = 0%
2013-12-13 06:31:36,532 Stage-0 map = 79%,  reduce = 0%
2013-12-13 06:31:37,532 Stage-0 map = 100%,  reduce = 0%
2013-12-13 06:31:38,845 Stage-0 map = 100%,  reduce = 100%
Ended Job = job_201312130547_0003
Moving data to: /princ
10 Rows loaded to /princ
OK
Time taken: 296.347 seconds
hives> [REDACTED]
```



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

The progress of sql queries can also be checked at jobtracker page url:50030

Job Scheduling Information											
Job Identifier	Running Status	Priority	Run Date	Last Run Date	Last Run By	Completed Since Last Run	Completed Since Last Run %	Last Run Date	Last Run By	Completed Since Last Run	Completed Since Last Run %
ps_2011070001_0001	Running	Normal	2011-07-01 10:00:00	2011-07-01 10:00:00	System	0	0.0%	2011-07-01 10:00:00	System	0	0.0%

Scheduling Information:

Queue Name	Status	cheduling Information
default	running	PSA

New Queue, Priority Queue, Name:

Notes: You can run or stop by queue name or queue number and then start them.

Running Jobs:

[View](#)

Completed Jobs:

Job ID	Priority	User	Name	Step % Complete	Step Total	Steps Completed	Business % Complete	Business Total	Business Completed	Job Scheduling Information	Completion Date
ps_2011070001_0001	Normal	System	ps_2011070001_0001	100.0%	1	1	100.0%	1	1	PSA	2011-07-01 10:00:00
ps_2011070001_0002	Normal	test	ps_2011070001_0002	100.0%	1	1	100.0%	1	1	PSA	2011-07-01 10:00:00

Retired Jobs:

[View](#)

Local Logs:

[View](#)

Inventory Job Tracker history

Customer Distribution Inventory Administration 2010



Multiple DBs in Hive

- Default database is named default

```
root@cloudace-mac:/home/cloudace# sudo -i
hive history file=/tmp/sqoop/hive_job_log_root_201312130641_1819464056.txt
hive> show databases;
OK
default
Time taken: 21.004 seconds
hive> create database parttime;
OK
Time taken: 2.147 seconds
hive> use parttime;
OK
Time taken: 0.074 seconds
hive> show tables;
OK
Time taken: 0.068 seconds
hive> use default;
OK
Time taken: 0.032 seconds
hive> show tables;
OK
mysql
scottsubscribers
mkt
subd
subac
subaccrds
mktaccs
Time taken: 0.261 seconds
hive> [REDACTED]
```



USA : +1-908-366-7933



Skype ID : keylabstraining



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



info@keylabstraining.com

HIVEQL

- Interface similar to SQL - 92. However, some constructs are different
- Supports JOIN, sub queries etc.
- Does not support .. UPDATE OR INSERT



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

Select statement

- Select expr1, expr2 .. FROM tablename t
 - Where clause also supported



Order by

- Select expr, expr ... from tables no where condition
order by expr desc;

This will sort the data in descending order. The default ordering is ascending.



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

Sort by

- SORT BY EXPRESSION
 - Sorts the data before going to reduce

```
SELECT EXPRESSION FROM TABLE_NAME  
WHERE CONDITION  
SORT BY EXPRESSION
```



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

Comparison Sort by and order by

- **Sort by**
 - May use multiple reducers for this output
 - Only gives sorted ordering of rows with 1st reducer
 - May give partially ordered final results
- **Order by**
 - Uses single reducer to guarantee total ordering of output
 - LVI can be used to minimize sort time



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

Joining tables

- Hive supports:
 - Inner joins
 - Left outer join
 - Right outer joins
 - Full outer joins



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

Syntax

- `Select * from HR01 where table1.join_table2.on (condition)`



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

Join on 2 tables

Employee

EMP ID	Emp Name	Address
1	Rosa	..5
2	Fred	..5
3	Lisa	..6
4	Tom	..7

Employee Department

Emp ID	Department
1	IT
2	II
3	FO
4	Admin



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

Inner join

- `select * from employee join employee_department ON employee.empId=employee.department.emp_id;`

EMP ID	Emp Name	Address	Emp ID	Department
1	Ron-E	US	1	-
2	Fred	US	2	-
3	Lynn	IN	3	INFO
4	Froy	IN	4	ACTEUR



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

Left outer join

- Select emp_id, empName, department from employee @ Left outer join employee department on (emp_id=empId);

EMP ID	Emp Name	Department
1	Sue	IT
2	Fred	II
3	Tom	Engg
4	Frank	Admin
5	Chris	null



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

live functions

- `rand()`
- `floor()`
- `ceil()`
- `sin()`
- `cos()`
- `log()`
- `cosine()`
- `abs()`
- `sqrt()`
- `etc`



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

Use of functions

```
hadoop dfs -ls /user/hive/warehouse/retail_db/
Total MapReduce jobs = 1
Launching Job 1 out of 1.
Number of reduce tasks determined at compile time: 3
The number of reduce tasks can be changed by setting the property:
  set hive.exec.reducers.bytes.per.reducer=number
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=number
or
  set mapred.reduce.tasks=number
Starting Job = job_201312190249_0003, Tracking URL = http://cloud9-nm-50034/pid=14614/job_201312190249_0003
Kill Command = /usr/lib/hadoop/bin/hadoop job -Dmapred.job.tracker=nm-50034 -kill job_201312190249_0003
2013-12-19 14:03:28,026 Stage-1 map = 0%,  reduce = 0%
2013-12-19 14:03:28,477 Stage-1 map = 100%,  reduce = 0%
2013-12-19 14:03:28,700 Stage-1 map = 100%,  reduce = 100%
Final Job = job_201312190249_0003
OK
2013-12-19 14:03:28,700
Time taken: 25.505 seconds
hadoop [
```



USA : +1-908-366-7933



Skype ID : keylabstraining

Hyderabad : +91-9550645679
Bangalore : +91-8884584163



info@keylabstraining.com

Views

- CREATE VIEW view_name;

As Select ...

To remove view

Drop View view;



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

View

```
INFO-CREATE 4599 7/18, Job SINCE is from 8/2011  
DE  
Data copied 0.000 records  
Error occurs * from w/  
Data MapReduce job = 0  
Launching Job 0 out of 0  
Number of reduce tasks is set to 0 since there's no reduce operator  
Starting Job = job_00132155245_0004. Tracking URL = http://cloofera-mu00056/poweredby,http://job_00132155245_0004  
Kill Command = /usr/lib/hadoop/bin/killing job -Dmapred.job.tracker=192.168.1.113 job_00132155245_0004  
2013-07-18 14:08:58,247 Stage-0 map = 0%, reduce = 0%  
2013-07-18 14:08:58,999 Stage-1 map = 100%, reduce = 0%  
2013-07-18 14:08:59,028 Stage-1 map = 100%, reduce = 0%  
Ended Job = job_00132155245_0004  
DE  
Half of reducers  
00132000024008  
00132000024009  
00132000024010  
00132000024011  
00132000024012  
00132000024013  
00132000024014  
00132000024015  
00132000024016  
00132000024017  
00132000024018  
00132000024019  
00132000024020  
00132000024021  
00132000024022  
00132000024023  
00132000024024  
00132000024025  
00132000024026  
00132000024027
```



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

Partitioning

- Partitioning a data set means dividing and splitting the data into one or more partitions using values of columns.
- These partitions are stored in subdirectories of table directory



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

Use of partitioning

- Allows user to filter data on input path
- Example logs of amazon.com (one entry from UK) can be stored for each day in a directory based on dates.



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

Create partitioning table

- Create table `t(column_size)`

`Partitioned by (column_name strategy)`

`ROW BY RANGE`

`Insert row into partition table`

`Truncate table [partition table]`

`Drop table`

`Partitioned table`



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

```
spark-submit --master local[4] --driver-memory 1G --executor-memory 1G --executor-cores 4 --num-executors 4 --name "Hive" /home/hadoop/hadoop-2.7.1/bin/hadoop jar /usr/lib/spark/jars/*
```

```
Hive history file=/tmp/hadoop-hive/job_log_wrt_201312191440_00083879.log
hive> copy data useras '/user/hive/warehouse' into 'useras@subscriberepart partition (tabledate='2014-01-06')'
>Loading data to table default.subscriberepart partition (tabledate=2014-01-06)
OK
Time taken: 0.329 seconds
hivesql:spark
```

```
root@elasticsearch:/home/elasticsearch$ hadoop fs -lsr /user/hive/warehouse/
Found 9 items
drwxrwxrwx  - root supergroup          0 2013-12-13 06:42 /user/hive/warehouse/greetings
drwxrwxrwx  - root supergroup          0 2013-12-13 14:13 /user/hive/warehouse/retailstore
drwxrwxrwx  - root supergroup          0 2013-12-13 14:13 /user/hive/warehouse/retailstore/part
drwxrwxrwx  - root supergroup          0 2013-12-13 14:13 /user/hive/warehouse/retailstore/part/part
drwxrwxrwx  - root supergroup          0 2013-12-13 14:13 /user/hive/warehouse/retailstore/part/part/part
Found 1 item
drwxrwxrwx  - root supergroup          0 2013-12-13 14:13 /user/hive/warehouse/retailstore/part/part/part/part
```



USA : +1-908-366-7933



Skype ID : keylabstraining

Hyderabad : +91-9550645679
Bangalore : +91-8884584163



info@keylabstraining.com

Creating custom UDF's

- To implement custom UDF, need to extend org.apache.hadoop.hive.udf.exec.UDF class present in hive-exec-0.7.0-cdh4.0. This jar can be copied from /usr/lib/hive/lib of VM.
- The evaluate method will be implemented accordingly.



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

Custom UDF

```
1. import org.apache.hadoop.hive.ql.udf.UDF;
2.
3.
4.
5. public class PowerFunction extends UDF {
6.
7.     public Double evaluate(Double arg0) {
8.
9.         double bytes = arg0.doubleValue() / (1024 * 1024 * 1024);
10.        return new Double(bytes);
11.    }
12.
13. }
14.
```



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

How to run UDF

```
Added /home/alexander/mst.job to class path
Added properties: /home/alexander/mst.properties
Starting source dependency manager with 'maven-dependency'
[INFO] 
[INFO] Total time: 0.0005 seconds
[INFO] 
[INFO] Adding remote repository: https://maven.apache.org/:maven-repository-local-mirror
[INFO] Remote Repository: https://maven.apache.org/:maven-repository-local-mirror
[INFO] 
[INFO] Scanning Job 1 over all 1
[INFO] 
[INFO] Adding at phase: reduce to 0 since there's no reduce operations
[INFO] Starting job = job_201312121248_0010, tracking URL = https://maven.apache.org/:maven-repository-local-mirror/jobs/job_201312121248_0010
[INFO] Kill Command = /usr/lib/hadoop/bin/hadoop job -killredjob,stacksmolotore-mr0021 -kill job_201312121248_0010
[INFO] 2013-12-12 19:21:28,217 Stage-1 map = 0%, reduce = 0%
[INFO] 2013-12-12 19:21:29,400 Stage-1 map = 100%, reduce = 0%
[INFO] 2013-12-12 19:21:30,618 Stage-1 map = 100%, reduce = 100%
[INFO] 2013-12-12 19:21:31,248_0010
[INFO] 
[INFO] Total time: 0.0005 seconds
[INFO] 
[INFO] 123.9424499924949494
[INFO] 381.44138888323234
[INFO] 371.5554544842323555
[INFO] 388.344989349494972
[INFO] 388.4177764232323438
[INFO] 228.228896533551194
[INFO] 437.81493952832323438
[INFO] 381.323287233231818
[INFO] 274.989748199054
```

Serde

- Serde (Serializer/Deserializer) is used by Avro to control how bytes are read and written to storage files.
- When used as serializer (i.e insert) tables serde will convert Avro's internal representation of row of data into bytes written to output file.
- When used as deserializer (i.e querying the data), serde will de-serialize a row of data from bytes to objects.



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

Hive Serde

SerDe Name	Java Package	Description
LazySimpleSerDe	org.apache.hadoop.hive.serde2.lazy	The default serde. Delimited textual format.
LazyBinarySerDe	org.apache.hadoop.hive.serde2.lazy.binary	A more efficient version of LazySimpleSerde. Prints column with binary field types.
BinarySortableSerDe	org.apache.hadoop.hive.serde2.binarysortable	Optimized for sorting at expense of compactness.
ColumnarSerDe	org.apache.hadoop.hive.serde2.columnar	For column based storage with RCFile.
RegexSerDe	org.apache.hadoop.hive.common.serde2	For textual data where columns are specified by regular expression.



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

Storage format

- 2 common & identical value - storage in Line Feed Format or File Format
- Row separator is line feed, and fields in a row are separated
- Row separator is defined by Spool Initialization Record - SIR



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

Default storage format:

- Default storage format is ext 3 or ext 4 or ext 5
- It is a highly reliable file system for server
- Highly used by Sun Microsystems for their servers



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

Binary storage format RCFile and sequencefile

- Using `MapReduce` the output is generated by `Mapper` and `Reducer` which generate file (`key/value pair`)
- Sequence files are used to store it
- RCFile is a columnar HDFS file. Similar to sequence file but data is stored in column oriented form.



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

Logical table

	α_1	α_2	α_3
β_1	1	2	3
β_2	4	5	6
β_3	7	8	9
β_4	10	11	12

Row-oriented Input (Sequence File)

row1	row2	row3	row4
1	2	3	4

Column-oriented Input (K3 file)

row split 1			row split 2		
col1	col2	col3	col1	col2	col3
1	4	7	5	8	11



USA : +1-908-366-7933

Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

Performance Tuning



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

T0001.5

- To check how InnoDB works with queries & WHERE clause EXPLAIN command can be used. The steps of execution can be done by using `SHOW PROCESSLIST` command.
- What EXPLAIN the output will be.
- ABSTRACT SYNTAX TREE (TOK_QL_TREE)
- TOK FROM TOK_TABLE_LIST (TABLENAME enclosed) TOK_INSERT (TOK_DESTINAT OR TOK_CDIR) TOK_TYPE (FILE)
- TOK SELECT
- TOK_SELECTR
- TOK_FUNCTION call (TOK_TABLE OR COL number (1))()
- The stages/steps which produce object as inferred from this output



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

LIMIT/TUNING

- LIMIT clause specifies how many rows to return from the mapped records.
- To avoid scanning entire query, following property can be used:
 - Optimality
 - Granularity of limit optimize small partitions (partitioned values)
 - Co-partitioned table (partitioned by column) A smaller subset of data for simple LIMIT first will skip it's co-partitions
- NOTE: This will give correct results only when the query has map step. If the query also has reduce step the results can be incorrect. (Because the hive will not process all the data and will limit the records right at the start time of execution of job).



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

Join Optimization

- When joining 3 or more tables in MySQL ON clause, the same join by single MySQL job will be used.
- High - volumes In the last table in the query is the largest, it will be all other tables and it is available for performing joins. Therefore, the largest table should be last -est in the query.



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

PARALLEL EXECUTION

- Execute given queries in parallel in stages. By default it executes one stage at a time
- Execute queries in several stages. i.e. are not dependent on each other and can be executed in parallel along with other queries. For this, following property can be set

```
<property>
<name>hive.exec.parallel</name>
<value>true</value>
<description>Whether to execute job in parallel</description>
</property>
```



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

JVM REUSE

- When there are small number of tasks and lots of tasks associated with short execution time.
- Reuse is significant for larger number of tasks. If map/reduce tasks are linked by JVM, VM <map> has its own overhead of starting. This is because both tasks have same environment. To allow reuse of existing JVM's limit the size of property, <max>

```
<property>
<name>mapred.job.reuse.jvm.num.tasks</name>
<value>10</value>
<description>How many tasks to run per jvm. If set to -1, there is no limit.</description>
</property>
```



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

Limitations of HIVE



Keylabs Training
Key for IT Solutions



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com

Limitations of HIVE

- **Hive and its sub-languages** - It requires a single execution in multiple stages. Since it is iterative, it is slower but instead of the query, the multi-staging takes time.
- **HIVE Queries** - are structured, you have to clearly specify what needs to be performed in what order using sub-queries. This is a bit cumbersome if there are complex numbers of steps to be executed. For this, Pig can be used which is more procedural like - language where the sequence of steps to be performed can be written with just visual statements.
- Once data is inserted into table, data can't be updated.
- **Splitting data** - Data at various stages is merged and combined by SQL and in the end a single result is obtained. Hence splitting is not possible. Pig on the other hand can be used to save the result at different stages.



USA : +1-908-366-7933



Hyderabad : +91-9550645679
Bangalore : +91-8884584163



Skype ID : keylabstraining



info@keylabstraining.com