HIVE

type of tables
1. temporary table : Scope of this table is for current session only .Hive session closed this table will get dropped or deleted.
2. internal/manged table:
3.external table: will not fully bound to data .it will on t

By default table is internal table data and data are together

IF Drop this table - table/schema and will get dropped.

HDFS > /user/hive/warehouse/db.ddname/tablename

hive > drop table emp; <--> hdfs -->schema -->mysql

Partitioning and joins

Partitioning and joins

Please refer different file

```
hive> set hive.exec.dynamic.partition.mode=nonstrict;
hive> set hive.exec.dynamic.partition=true;
```

Joins
inner , Left

Inner
```
select e.ename,e.dno,d.dloc from employee e JOIN dept d ON e.dno=d.dno;
```

Left Outer
```
select e.name ,e.dno ,d.dloc from employee e LEFT OUTER JOIN dept d ON
e.dno=d.dno
```

Right Outer
```
select e.name,e.dno d.dloc form employee e RIGHT OUTER JOIN dept d on
e.dno = d.dno
```

Full Outer

```
select e.ename e.dno d.dloc from employee e FULL OUTER JOIN dept d ON
e.dno = d.dno
```

Bucketing :

In hive table Partitions are subdivided into buckets based on the hash function of a column in table.

1.It create a file.
2.There is always a one bucket column
3.Has to be in schema of table
4.Bucket numbers has to be defined.

5.Buketing can be used with partitions
6.Bucketing column depends on cardanility.If its less we select.
7.Bucket numbet will get decided after calculating hash of a column.

--Cardanilty -- Is defined as group of repeated elements it will high cardanilty

Data 10 GB   eid , ename,state ,country

Partitions : will be on state and country ---cardanility is high (more repeated elements)
Bucketing : will be on eid -------- cardanilty is less (less repeated elements)

Hash function MD5 algorithm to calculate the hash value of a bucketed column value.

Hash value of column value % No. of bucketes => Bucket Position

eid value:

```
value ,hash_valye,bucketing_number
5 => 23 => 23%3 = 2
10 => 22 => 22%3 => 1
like that .....
```

create table for Bucketing

use this command for every session
1.
```
set hive.enforce.bucketing=ture;
set hive.exec.dynamic.partition.mode=nonstrict;
set hive.exec.dynamic.partition=true;
```

2.
```
CREATE TABLE IF NOT EXISTS userbukpart (uid int,device string,browser
string,os string, osversion int,ip string,country string,city
string,street string,tt string,product string, day int)
PARTITIONED BY (month int,year int)
CLUSTERED BY (uid) INTO 3 BUCKETS
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
STORED AS TEXTFILE;
```

3.
```
INSERT INTO userbukpart PARTITION(year) select * from userlogext;
```
4. select * from userbuckpart;
5.hadoop fs -ls /user

```
/*
--- Exrernal table we should not need to load data beacuse table is
created on data which is existed  or internal table  location is always
in dirctory .
incase if you deleted the external table only that table and schema gets
deleted data will be there safely,

*/
```

UDF (user defined function)

Cube(3) => 27

Eclipse --> same as pig

jar file name will be hive-0.4.1.jar cp lib folder

right click on soruce and create new class name cube finished

this is code need to write.

```
import org.apache.hadoop.hive.ql.exec.UDF
public class cube extends UDF {

  public int evaluate(int num){
return num*num* num;

}
}
```

save the code .
right click on project name.
export as jar file
give the path to export on location

Move the jar file in hdfs
by using hadoop fs -put -f /umesh/jarfile /user/umesh

go to hive shell
ADD JAR "hdfs://localhost:9000/user/umesh/hiveudf.jar";

CREATE FUNCTION cube as 'cube' USING JAR
"hdfs://localhost:9000/user/umesh/hiveudf.jar";
--- here cube is class name

test functin;
select cube(3);
it will give output 27

to drop function
drop function function_name;

14/01/2019 10:05:37

Tom white (hadoop)
JDBC using Hive:

Driver Manger, class, forname,connection,prepeared statement ,Resultset

We do not create connection like   --connect //hdfs://localhost:mysql
...... like that

In production we can create config.properties which can contain userid ,password and url

config.properties

copy jar from /usr/lib/hadoop AND /usr/hive/lib

Mysql Dept(dno,name ,dloc)- 1,2,3

HDFS = /user/umesh/emp/emp.txt (eno,ename, esal, dno) 3,4,5

using JDBC/java code ... Create hive final table empdept (dno,avg(sal))
Consider only the depno which is availble in mysql

Hive with XML file:

hivexmlserde-1,0.5.3.jar

---- Serde = serilization and deserilization

In hive shell
```
hive > add jar
/home/umesh/NotesForBatch_Updated/OtherFiles&Jars/hivexmlserde-
1.0.5.3.jar;
```

XML file: book.xml

in hive shell

```
CREATE TABLE mybook(title string , author string , country string,
company string , price string, year int)
ROW FORMAT SERDE 'com.ibm.spss.hive.serde2.xml.XmlSerDe'
WITH SERDEPROPERTIES(
"column.xpath.title"="/BOOK/TITLE/text()",
"column.xpath.author"="/BOOK/AUTHOR/text()",
"column.xpath.country"="/BOOK/COUNTRY/text()",
"column.xpath.company"="/BOOK/COMPANY/text()",
"column.xpath.price"="/BOOK/PRICE/text()",
"column.xpath.year"="/BOOK/YEAR/text()"
)
STORED AS INPUTFORMAT 'com.ibm.spss.hive.serde2.xml.XmlInputFormat
OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.IgnoreKeyTextOutputFormat'
TBLPROPERTIES(
"xmlinput.start"="<BOOK>",
"xmlinput.end"="</BOOK>"
);
```

```
LOAD DATA LOCAL INPATH 'file path' into table mybook
```

```
load data local inpath
'/home/umesh/NotesForBatch_Updated/OtherFiles&Jars/book.xml' into table
mybook;
```

Through java

```
mybook.xml ---> put 5 book....
---> finalbook (partition in country where country = india)

------------------------------------------------------------------------
---------------------------------------------


SQOOP -- sql to hadoop and hadoop to sql
Sqoop is a tool designed  for import and export data between hadoop RDBMS
like mysql servers.

RDBMS(mysql)  <----- Sqoop import/export  <----- HDFS/hive

Trading : share Market:
nse, bse site .csv


Name of company share 1% grow....

trading ,investing, Gambling- X
----
1.import table_name from mysql to hdfs;

synatx: sqoop import/export --properties  --user  --pass --target --
export-dir


terminal > here '--' specified parmeter or properties

sqoop import --connect 'jdbc:mysql://localhost/umesh' --username root --P
--table maehsh --m 1

Default location for import is /user/umesh/umesh


hdfs dfs -cat /user/umesh/mahesh/part-m-00000


-
2. sqoop import --connect 'jdbc:mysql://localhost/mahesh' --username
root -password Admin@123 --table mahesh  --m 1 --target-dir /user/umesh

hdfs dfs -cat /user/umesh


3. Where Condition.
   sqoop import --connect 'jdbc:mysql://localhost/umesh' --username  root
--P --query "select * from mahesh where id>2 AND \$CONDITIONS"  --m 1 --
target-dir /user/prath
--output   hdfs dfs -cat  /user/prath/part-m-00000
/*3,shubangi
4,pramod
4,ganesh*/

4.import all table from DB:
sqoop import-all-tables --connect 'jdbc:mysql://localhost/umesh' --
username  root --P   --m 1

delete all dirctory
hdfs dfs -rm r /user/*
```

```
if want delete only file and keep the direcotries
hdfs dfs -rm -r /user/*.*


*/

5.list out all table in particular database
sqoop list-table --connect 'jdbc:mysql://localhost/umesh' --username root
--password Admin@123

6.Sqoop export:
export hdfs data to mysql table.
sqoop export --connect 'jdbc:mysql://localhost/umesh' --username root --
password Admin@123 --table emp --m 1 --export-dir /user/umesh/emp


Task...
Baking domain -> Mysql (customer) ---> transactions.log  --> server
/hdfs/ ---->hbase

7.work with sqoop jobs;
sqoop job --create empexport -- import --connect
'jdbc:mysql://localhost/umesh' --username root --password Admin@123 --
table emp --m 1 --export-dir /user/umesh/emp

--list of the job

sqoop job --list

--to delete a job

sqoop job --delete job_name

--to execute the job

sqoop job --exec job_name



1. create a table in mysql as emp(eid,ename, esal,dno) and
dept(dno,dlocation ,dname) using sqoop import both table in hdfs
in hive create external table on above data
2.join them on dno and insert into hive managed empstat
table(dno,dlocation, sum(esal))
3.Export this table data empstat  into mysql table
empstatistic(dno,dlocation, sum(esal)
4.drop above external table
5 Errors with solutions/efforts you have tried....

----solution------

-- to update particular value in table in mysql
UPDATE emp SET dno = 'math' WHERE eid = 1;


CREATE EXTERNAL TABLE IF NOT EXISTS emp(eid int,ename string,dno
string,esal int)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
```

```
STORED AS TEXTFILE
LOCATION '/user/umesh/emp';

--* you do not have load data for external table.. to be noted


CREATE EXTERNAL TABLE IF NOT EXISTS dept(dno string,dloc string, dnum
int)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
STORED AS TEXTFILE
LOCATION '/user/new';

--Joins of tables

select a.dno,a.esal,b.dloc from
emp a join dept b on a.dno = b.dno;


CREATE TABLE empstat (dno string,esal int,dloc string)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
STORED AS TEXTFILE;

--insert joins data into table empstat

INSERT INTO empstat select a.dno,a.esal,b.dloc from emp a join dept b on
a.dno = b.dno;

--Export empstat hive table in mysql empstatistics by using HDFS data .

sqoop export --connect 'jdbc:mysql://localhost/umesh' --username root --
password Admin@123 --table empstatistics --m 1 --export-dir
/user/hive/warehouse/umesh.db/empstat
------------------------------------------------------------------------
-------------------------------------------------
```

Hbase part 1
Hbase, cassandra,mongodb, dynamo etc   .. No-sql databases
<- use for stroing huge amount of data and access the random way.

Hash table --> Key,values pairs.....> fast read and write operations

What is Hbase:

Hbase is not for analysis it's for only write and read

Its a distributed column-oriented database built on top hdfs .Its an open source project by apache and horizontally scalble and vertically alligned.

Inherited form google's table designed to provide quick random access to huge amount of data.

Random real time Read/write access to data in hdfs.
HDFS: Storing large files does not support fast indiviual record lookup , provide sequential access of data

Vs Hbase:Built on to of hdfs ,provides fast lookups for large files ,uses hash tables(key,values pairs) for random access and store data in indexed hdfs files.

RDBMS: Schema is mandatory, small tables hard to scale , transactional , normalized data(its generally for avoid redundancy (duplication of data)) ACID

Vs Hbase: Schema is less, concept of column famlies ,wide table , horizontally scalable de-normalized. (Its genearally used for real time data like logs CAP theory (consistancy availblity,partitions)

Storage Mechanism:

Table is collection of rows
Row is a collection of column families
Colum family is a collection of columns
Colum is a collection of key values pairs.

Ex .Table

```
              Column Family1            Column Family2
RowID          col1 col2                 col1 col2
1
2
3
```

Ex Employee

```
               Personal              Professional
RowID          Name  Age             Job    Sal
1              umesh 23             data scientist 49K
```

Data Model:
Table -- Consist of multiple rows
Row -- consist of row_key and one or more colums with value associated with them
Column -- Consist of column family and column qualifier defined by ':' character.
Column Family -- Physically allocated the columns , and data is compressed or its row key are encoded.
Column Qualifier -- Its added to colums Family to provide the idex for a given columns family.
Cell-- Combination of row ,column family ,column qualifier ,value timestamp which represent values of version ,SCD(slowly chanaging Dimentions)

TimeStamp: Is written alongside with each values .Time when data was written by Hregionserver.

Hmaseter[managing Resources]
Regionserver[execution/writing
zookeeper -[cordination/cleanup]

Hmaster -- he is manager for habase who manages all resources
Hregion --- he is reponisble for query execution and writing the data
HQuorumPeer --- Maintain logs
Zookeeper -- he is cordinator and cleaup.

```
Create table
Syntax: create table_name 'CF1','CF2'
create 'candidate' 1,'personal','professional'

-- to insert data in table

Syntax: put
'table_name','row_key','Coluns_Famaliy:column_qualifier','value'
eg- put 'candidate',1,'personal:name','umesh'

--Display table
Syntax: scan 'table_name'
eg- scan 'candidate'

-- convert timestamp to date
import java.util.Date
Date.new(timestamp).toString()


--Get any particular cell
get 'emp', 1, 'personal:name'

--Delete particular cell
delete 'emp', 1, 'personal:name'

--Delete any row  (like below exmple 1st row)
delete 'emp',1

--Drop a table
drop 'emp'
We have Disable the table before deleting it

---To check wheathe table is Disable or enable
is_enable 'emp'
is_disable 'table_name'

-- to disable table
disable 'table_name'

--to disable table which start with 'em'
disable_all 'em*.*'  ---using regular exprssion
same for enable ,drop .
```