List of Statements

start-all.sh * how to start all nodes in hadoop *

jps * check the status *

hive * start Hive *

show databases; * lists all existing database*

create database if not exists HOSPITAL; * creates a database *

use HOSPITAL; * get into the database *

show tables; * lists all the tables in a database *

How to create internal /managed tables

create table patient(pid int,pfname string, age int,plname string,state string,reason string) row format delimited fields terminated by '/t' ;

LOAD DATA local INPATH '/home/hadoop/Documents/patient.txt' into table patient;

desc student;

create table app(pid int,did string,dname string,rating int,specialization string,hid string)row format delimited fields terminated by '\t';

How to insert values into a table

LOAD DATA local INPATH '/home/hadoop/Documents/app.txt' into table app ;

How to create External tables

create external table emp(eid int,ename string,rating float,department string,lname string,state string)row format delimited fields terminated by ',' stored as textfile;

show tables; * lists the tables in the database*
desc emp; * gives the structure of the table *

How to insert values into a table

LOAD DATA local INPATH '/home/hadoop/Documents/emp1.txt' into table emp;
OR
insert into emp values('eid','ename','rating','department','lname','state');

How display all values in a table
select * from emp;

How to drop a table
drop table emp;

HOW TO ADD COLUMN TO A EXISTING TABLE
alter table emp add columns (age int);

HOW TO DROP COLUMNS
alter table emp replace columns(sid int,sname string,grade float,department string,lname string,state string);

SAMPLE QUERIES

1. select * from patient;
2. select * from patient where age > 60 and reason<>'fever';
3. select did, dname, (rating + 1.0) AS raise from app;
4. select * from patient where reason='cold' or reason='fever';
5. Select dname from app where did in(1,2,3);
6. select max(rating) from app;
7. select min(age) from patient;
8. select max(rating) from app group by specialization; ent 4.8 derma 3.8 gyn 4.5
9. select avg(rating) from app;
10. select sum(rating) from app where specialization='cardiologist';

case statement

11. select dname,rating,case when rating<=2 then 'low' when rating >=3 and grade <=4 then 'average' when grade>=5 then 'excellent' else 'not valid' end as rating_range from app; john 1 not valid

12. select substr(dname,2,3) from app;
13. select concat(did,'_',dname) from app;

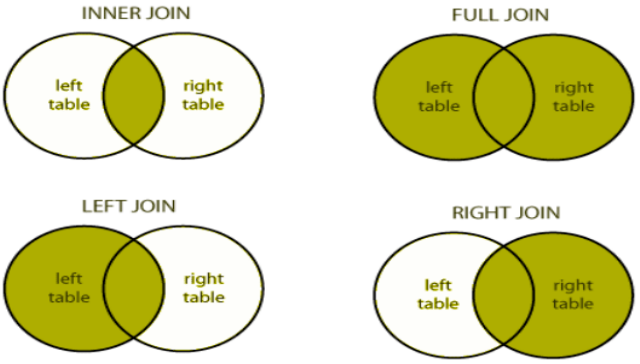
JOIN

14. select p.pid,p.pfname,a.dname from patient p join app a on (p.pid=a.pid);

15. select p.pid,p.pfname , a.dname from patient p left outer join app a on (p.pid=a.pid);

16. select p.pid,p.pfname , a.dname from patient p right outer join app a on (p.pid=a.pid);

17. select p.pid,p.pfname , a.dname from patient p full outer join app a on (p.pid=a.pid);



We will be working with two tables — customer and orders — that we imported in my sqoop import article, and we'll perform the following joins:

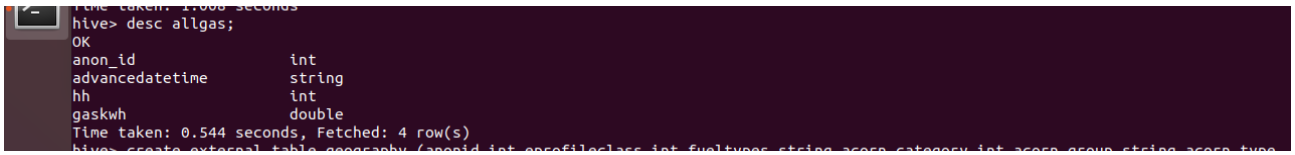
1. **INNER JOIN** – Select records that have matching values in both tables.
2. **LEFT JOIN** (LEFT OUTER JOIN) – Returns all the values from the left table, plus the matched values from the right table, or NULL in case of no matching join predicate
3. **RIGHT JOIN** (RIGHT OUTER JOIN) A RIGHT JOIN returns all the values from the right table, plus the matched values from the left table, or NULL in case of no matching join predicate
4. **FULL JOIN** (FULL OUTER JOIN) – Selects all records that match either left or right table records.

Screenshots

1. Show databases;

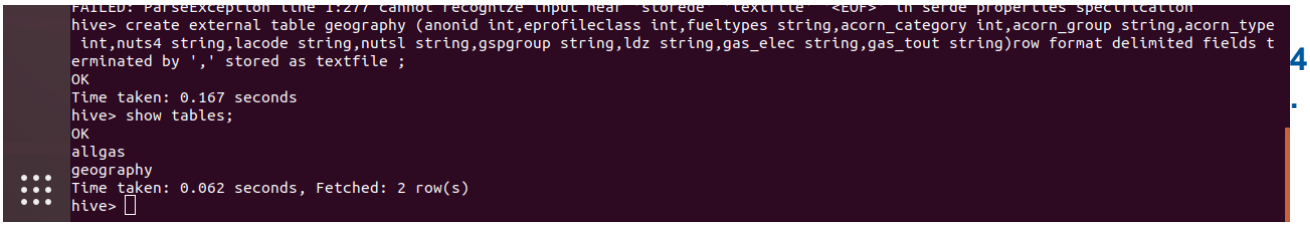
```
hive> show databases;
OK
default
firstdb
Time taken: 0.878 seconds, Fetched: 2 row(s)
hive> create external table allgas ( anon_id int,advancedatetime string,hh int ,gaskwh double)row format delimited fields terminated
by ',' stored as txtfile;
FAILED: SemanticException Unrecognized file format in STORED AS clause: 'TXTFILE'
hive> create external table allgas ( anon_id int,advancedatetime string,hh int ,gaskwh double)row format delimited fields terminated
by ',' stored as textfile;
OK
Time taken: 1.008 seconds
hive> desc allgas;
OK
anon_id                 int
advancedatetime         string
hh                      int
gaskwh                  double
Time taken: 0.544 seconds, Fetched: 4 row(s)
```

2. desc table_name;



```
hive> desc allgas;
OK
anon_id                 int
advancedatetime         string
hh                      int
gaskwh                  double
Time taken: 0.544 seconds, Fetched: 4 row(s)
```

3. create external table; (note without the keyword external it will be internal table)



```
hive> create external table geography (anonid int,eprofileclass int,fueltypes string,acorn_category int,acorn_group string,acorn_type
 int,nuts4 string,lacode string,nutsl string,gspgroup string,ldz string,gas_elec string,gas_tout string)row format delimited fields t
erminated by ',' stored as textfile ;
OK
Time taken: 0.167 seconds
hive> show tables;
OK
allgas
geography
Time taken: 0.062 seconds, Fetched: 2 row(s)
hive>
```

Show tables;

```
hive> show tables;
OK
allgas
geography
Time taken: 0.062 seconds, Fetched: 2 row(s)
hive>
```

5. Load values into table

```
hive> create table tanrecords(txno int,txndate string,custno int,amount double,category string,product string,city string,state strin
g,spendby string) row format delimited fields terminated by ',' stored as textfile;
OK
Time taken: 0.823 seconds
hive> show tables;
OK
tanrecords
Time taken: 0.035 seconds, Fetched: 1 row(s)
hive> select * from tanrecords;
OK
Time taken: 0.318 seconds
hive> load data local inpath '/home/hadoop/Documents/custtxn.txt' into table tanrecords;
Loading data to table trial.tanrecords
OK
Time taken: 2.039 seconds
hive> select * from tanrecords;
OK
NULL    NULL    NULL    NULL    NULL    NULL    NULL    NULL    NULL
1       12/05/2020      100     1234567.0       electronics     laptop  bangalore       karnataka       manager
2       31/02/2019      101     1234567.0       cloths  top     mangalore       karnataka       manager
3       02/05/2016      102     1234567.0       cloths  pant    bombay  maharastra      manager
4       12/05/1998      103     1234567.0       electronics     watch   mangalore       karnataka       manager
5       19/05/1994      104     1234567.0       electronics     tv      bangalore       karnataka       manager
6       24/04/2005      105     1234567.0       gold    jewlery bombay  maharastra      manager
7       12/06/2004      106     1234567.0       savings education       mysore  karnataka       manager
8       12/06/2015      107     1234567.0       trips   worldtour       bangalore       karnataka       manager
Time taken: 0.197 seconds, Fetched: 9 row(s)
hive>
```

OR
using Insert statement

```
Time taken: 0.25 seconds
hive> LOAD DATA local INPATH '/home/hadoop/Documents/app.txt' into table app ;
Loading data to table hospital.app
OK
Time taken: 0.646 seconds
hive> select * from app;
OK
1       d1      sam     2       Immunologists   H1
2       d2      mary    3       Cardiologists   H1
3       d3      john    4       Dermatologists  H2
4       d4      alex    5       Endocrinologists        H2
5       d5      rose    5       Family Physicians       H3
6       d6      richard 3       Nephrologists   H3
7       d7      melissa 4       Family Physicians       H3
8       d7      usha    5       Family Physicians       H3
9       d3      tom     3       Dermatologists  H3
10      d4      kyane   4       Immunologists   H2
Time taken: 0.353 seconds, Fetched: 10 row(s)
hive> insert into app values('11','d8','george','3','dermatologist','H4);
FAILED: ParseException line 1:67 character '<EOF>' not supported here
hive> insert into app values('11','d8','george','3','dermatologist','H4');
Query ID = hadoop_20200711182803_ca3cc257-955b-4489-857f-6688e5fb13c9
Total jobs = 3
Launching Job 1 out of 3
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_1594385239703_0005, Tracking URL = http://bigdata-OptiPlex-360:8088/proxy/application_1594385239703_0005/
Kill Command = /usr/local/hadoop/bin/mapred job  -kill job_1594385239703_0005
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2020-07-11 18:29:40,123 Stage-1 map = 0%,  reduce = 0%
2020-07-11 18:30:40,535 Stage-1 map = 0%,  reduce = 0%
2020-07-11 18:31:11,590 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 4.09 sec
2020-07-11 18:32:03,644 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 6.94 sec
MapReduce Total cumulative CPU time: 6 seconds 940 msec
Ended Job = job_1594385239703_0005
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 directory hdfs://localhost:9000/user/hive/warehouse/hospital.db/app/.hive-staging_hive_2020-07-11_18-28-03_416_2225429
802119696188-1/-ext-10000
Loading data to table hospital.app
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 6.94 sec   HDFS Read: 20207 HDFS Write: 400 SUCCESS
Total MapReduce CPU Time Spent: 6 seconds 940 msec
OK
Time taken: 246.452 seconds
hive>
```

6. Query sample for Count aggregate function

```
hive> select count(category) from tanrecords;
Query ID = hadoop_20200630142936_9ab577d8-d6e7-41a2-906b-766624c5472f
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_1593505514430_0001, Tracking URL = http://bigdata-OptiPlex-360:8088/proxy/application_1593505514430_0001/
Kill Command = /usr/local/hadoop/bin/mapred job  -kill job_1593505514430_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2020-06-30 14:30:23,497 Stage-1 map = 0%,  reduce = 0%
```

7. Query sample for sum aggregate function

```
hive> select sum(amount) from tanrecords group by category;
Query ID = hadoop_20200630143252_c51c83dc-4bbc-4496-aca1-c00b85b90338
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 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_1593505514430_0002, Tracking URL = http://bigdata-OptiPlex-360:8088/proxy/application_1593505514430_0002/
Kill Command = /usr/local/hadoop/bin/mapred job  -kill job_1593505514430_0002
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2020-06-30 14:33:05,586 Stage-1 map = 0%,  reduce = 0%
2020-06-30 14:33:22,334 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.34 sec
2020-06-30 14:33:43,005 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 4.95 sec
MapReduce Total cumulative CPU time: 4 seconds 950 msec
Ended Job = job_1593505514430_0002
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 4.95 sec   HDFS Read: 15075 HDFS Write: 212 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 950 msec
OK
NULL
2469134.0
3703701.0
1234567.0
1234567.0
1234567.0
Time taken: 52.055 seconds, Fetched: 6 row(s)
hive>
```

8.

Queries using and & or

```
hive> select * from tanrecords where custno='102';
OK
3       02/05/2016      102     1234567.0       cloths  pant    bombay  maharastra      manager
Time taken: 0.602 seconds, Fetched: 1 row(s)
hive> select * from tanrecords where custno='102' or custno='103';
OK
3       02/05/2016      102     1234567.0       cloths  pant    bombay  maharastra      manager
4       12/05/1998      103     1234567.0       electronics     watch   mangalore       karnataka       manager
Time taken: 0.214 seconds, Fetched: 2 row(s)
hive> select * from tanrecords where custno='102' and txno='3';
OK
3       02/05/2016      102     1234567.0       cloths  pant    bombay  maharastra      manager
Time taken: 0.253 seconds, Fetched: 1 row(s)
hive>
```

9. usage of limit keyword

```
hive> select * from tanrecords limit 2;
OK
NULL    NULL    NULL    NULL    NULL    NULL    NULL    NULL    NULL
1       12/05/2020      100     1234567.0       electronics     laptop  bangalore       karnataka       manager
Time taken: 0.165 seconds, Fetched: 2 row(s)
hive> select * from tanrecords limit 3;
OK
NULL    NULL    NULL    NULL    NULL    NULL    NULL    NULL    NULL
1       12/05/2020      100     1234567.0       electronics     laptop  bangalore       karnataka       manager
2       31/02/2019      101     1234567.0       cloths  top     mangalore       karnataka       manager
Time taken: 0.189 seconds, Fetched: 3 row(s)
hive>
```

10.JOIN query

```
Activities  Terminal                Fri 13:37
                                    hadoop@bigdata-OptiPlex-360: ~
File  Edit  View  Search  Terminal  Help
OK
Time taken: 0.588 seconds
hive> select * from stu_course;
OK
1       c1      sql     4       dms     f1
2       c1      sql     4       dms     f1
3       c2      os      4       co      f2
4       c2      os      4       co      f2
5       c3      java    4       oops    f3
6       c3      java    4       oops    f3
7       c3      java    4       oops    f3
8       c3      java    4       oops    f3
9       c3      java    4       oops    f3
10      c4      oops    4       c       f2
Time taken: 0.172 seconds, Fetched: 10 row(s)
hive> select st.sid,st.fname,sc.cname from s st join stu_course sc on (st.sid=sc.sid);
Query ID = hadoop_20200710133606_e95a6a85-4ed7-429b-b5cf-66afd6926148
Total jobs = 1

Execution completed successfully
MapredLocal task succeeded
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1594363444337_0004, Tracking URL = http://bigdata-OptiPlex-360:8088/proxy/application_1594363444337_
0004/
Kill Command = /usr/local/hadoop/bin/mapred job  -kill job_1594363444337_0004
Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 0
2020-07-10 13:36:40,197 Stage-3 map = 0%,  reduce = 0%
```

Difference between Internal & External tables :

External Tables -

External table stores files on the HDFS server but tables are not linked to the source file completely.

If you delete an external table the file still remains on the HDFS server.

As an example if you create an **external table** called **"table_test"** in HIVE using HIVE-QL and link the table to file **"file"**, *then deleting "table_test" from HIVE will not delete "file" from HDFS*.

External table files are accessible to anyone who has access to HDFS file structure and therefore security needs to be managed at the HDFS file/folder level.

Meta data is maintained on the master node, and deleting an external table from HIVE only deletes the metadata not the data/file.

For Internal Tables-

Stored in a directory based on settings in hive.metastore.warehouse.dir, *by default internal tables are stored* in the following directory *"/user/hive/warehouse"* you can change it by updating the location in the config file .
Deleting the table deletes the metadata and data from master-node and HDFS respectively.
Internal table file security is controlled solely via HIVE. Security needs to be managed within HIVE, probably at the schema level (depends on organization).

Hive may have internal or external tables, this is a choice that affects how data is loaded, controlled, and managed.

Use EXTERNAL tables when:

The **data is also used outside of Hive**. For example, the data files are read and processed by an existing program that doesn't lock the files.
Data needs to remain in the underlying location even after a DROP TABLE. This can apply if you are pointing multiple schema (tables or views) at a single data set or if you are iterating through various possible schema.
Hive should not own data and control settings, directories, etc., you may have another program or process that will do those things.
You are not creating table based on existing table (AS SELECT).

Use INTERNAL tables when:

The **data is temporary**.
You want **Hive to completely manage the life-cycle of the table and data**