Good to know things in Hive

...

IMPORTANT

Copyright Infringement and Illegal Content Sharing Notice

Enable No drop feature

If we enable no drop on a table then the table can’t be dropped.

alter table employee enable no_drop;

drop table employee;

```
hive> alter table employee enable no_drop;
OK
Time taken: 0.302 seconds
hive> drop table employee;
FAILED: Execution Error, return code 1 from org.apache.hadoop.hive.ql.exec.DDLTask.
 Table employee is protected from being dropped
hive>
```

Disable No drop feature

alter table employee disable no_drop;

drop table employee;

```
hive> alter table employee disable no_drop;
OK
Time taken: 0.107 seconds
hive> drop table employee;
OK
Time taken: 0.237 seconds
hive>
```

Enabling no drop on a particular partition

Note: We can also enable no drop feature on a particular partition of a partitioned table in hive.

alter table <tablename> partition (deptname='HR') enable no_drop;

Enable Offline feature in hive

If we enable offline feature on a table then we won't be able to query the table.

alter table orders enable offline;
select * from orders;

```
hive> select * from orders;
OK
111111  phone   1111    3       1200.0
111112  camera  1111    1       5200.0
111113  broom   1111    1       10.0
111114  broom   2222    2       20.0
111115  t-shirt 4444    2       66.0
Time taken: 0.099 seconds, Fetched: 5 row(s)
hive> alter table orders enable offline;
OK
Time taken: 0.165 seconds
hive> select * from orders;
FAILED: SemanticException [Error 10113]: Query against an offline table or partitio
n Table orders
hive>
```

disable Offline feature in hive

We can disable the feature also.

alter table orders disable offline;

select * from orders;

```
hive> alter table orders disable offline;
OK
Time taken: 0.199 seconds
hive> select * from orders;
OK
111111  phone   1111    3       1200.0
111112  camera  1111    1       5200.0
111113  broom   1111    1       10.0
111114  broom   2222    2       20.0
111115  t-shirt 4444    2       66.0
Time taken: 0.08 seconds, Fetched: 5 row(s)
hive>
```

Skipping headers when loading data

Let’s say when loading data we want to skip few the first few rows then we can mention this in tblproperties.

Let us first see how the data looks like.

```
[cloudera@quickstart Downloads]$ cat skip_dataset.csv
Name,score
name1,score1
name2,score2
John,1500
Albert,1500
Mark,1000
Frank,1150
Loopa,1100
Lui,1300
John,1300
John,900
Lesa,1500
Lesa,900
Pars,800
leo,700
leo,1500
lock,650
Bhut,800
Lio,500
```

Skipping the header lines

create table skip_test(name string,score int) row format delimited fields terminated by ',' lines terminated by '\n' stored as textfile tblproperties("skip.header.line.count"="3");

load data local inpath '/home/cloudera/Downloads/skip_dataset.csv' into table skip_test;

```
hive> create table skip_test(name string,score int) row format delimited fields ter
minated by ',' lines terminated by '\n' stored as textfile tblproperties("skip.head
er.line.count"="3");
OK
Time taken: 0.274 seconds
hive> load data local inpath '/home/cloudera/Downloads/skip_dataset.csv' into table
 skip_test;
Loading data to table trendytech.skip_test
Table trendytech.skip_test stats: [numFiles=1, totalSize=190]
OK
Time taken: 0.656 seconds
hive>
```

Skipping headers

Let us try to see the data in table now. We can see that first 3 rows are skipped.

```
hive> select * from skip_test;
OK
John    1500
Albert  1500
Mark    1000
Frank   1150
Loopa   1100
Lui     1300
John    1300
John    900
Lesa    1500
Lesa    900
Pars    800
leo     700
leo     1500
lock    650
Bhut    800
Lio     500
Time taken: 0.138 seconds, Fetched: 16 row(s)
hive>
```

Making table immutable

If we set tblproperties("immutable"="true")

Then this will allow to load data in table only for first time.

That means you won't be able to append the data in this table.

however you will be able to overwrite the data.

drop vs truncate vs purge

1. Drop:

In case of managed table drop will drop both data and schema.

However in case of external table drop will drop just the schema.

2. Truncate:

Truncate will delete all the data. Schema will still be there

3. Purge:

tblproperties("auto.purge"="true")

if set to true the data will be permanently deleted and won't go to trash.

Treating empty strings as Null

If the file has empty spaces for string field then in table they show as empty.

What if we want to show them in our tables as null's

We can use the below table property

```
tblproperties("serialization.null.format"="");
```

See the data as it looks in file

```
hive> select * from sparse_test;
OK
John    1500
Albert  NULL
        1000
Frank   1150
        1100
        1300
John    1300
        900
Lesa    1500
Lesa    NULL
Pars    800
leo     700
leo     1500
lock    650
Bhut    800
Lio     500
```

Treating empty strings as Null

Let us create a normal table and load the data.

```
create table sparse_test(name string,score int) row format
delimited fields terminated by ',' lines terminated by '\n' stored as
textfile;

load data local inpath
'/home/cloudera/Downloads/sparse_dataset.csv' into table
sparse_test;

select * from sparse_test;
```

Screenshots of previous commands

```
hive> create table sparse_test(name string,score int) row format delimited fields t
erminated by ',' lines terminated by '\n' stored as textfile;
OK
Time taken: 0.089 seconds
hive> load data local inpath '/home/cloudera/Downloads/sparse_dataset.csv' into tab
le sparse_test;
Loading data to table trendytech.sparse_test
Table trendytech.sparse_test stats: [numFiles=1, totalSize=130]
OK
Time taken: 0.428 seconds
hive>
```

```
hive> select * from sparse_test;
OK
John    1500
Albert  NULL
        1000
Frank   1150
        1100
        1300
John    1300
        900
Lesa    1500
Lesa    NULL
Pars    800
leo     700
leo     1500
lock    650
Bhut    800
Lio     500
```

Treating empty strings as Null

Now, recreate the table with the table property.

```
drop table sparse_test;

create table sparse_test(name string,score int) row format
delimited fields terminated by ',' lines terminated by '\n' stored as
textfile tblproperties("serialization.null.format"="");

load data local inpath
'/home/cloudera/Downloads/sparse_dataset.csv' into table
sparse_test;

select * from sparse_test;
```

Screenshots of previous commands

```
hive> drop table sparse_test;
OK
Time taken: 0.293 seconds
hive> create table sparse_test(name string,score int) row format delimited fields t
erminated by ',' lines terminated by '\n' stored as textfile tblproperties("seriali
zation.null.format"="");
OK
Time taken: 0.212 seconds
hive> load data local inpath '/home/cloudera/Downloads/sparse_dataset.csv' into tab
le sparse_test;
Loading data to table trendytech.sparse_test
Table trendytech.sparse_test stats: [numFiles=1, totalSize=130]
OK
Time taken: 0.453 seconds
hive>
```

```
hive> select * from sparse_test;
OK
John    1500
Albert  NULL
NULL    1000
Frank   1150
NULL    1100
NULL    1300
John    1300
NULL    900
Lesa    1500
Lesa    NULL
Pars    800
leo     700
leo     1500
lock    650
Bhut    800
Lio     500
Time taken: 0.177 seconds, Fetched: 16 row(s)
hive>
```

Run hdfs commands from hive

We need to use the dfs command

dfs -ls /user/cloudera;

```
hive> dfs -ls /user/cloudera;
Found 10 items
drwxr-xr-x   - cloudera cloudera          0 2020-04-16 16:02 /user/cloudera/_sqoop
drwxr-xr-x   - cloudera cloudera          0 2020-04-22 14:26 /user/cloudera/customers
drwxr-xr-x   - cloudera cloudera          0 2020-04-29 16:47 /user/cloudera/data
-rw-------   1 cloudera cloudera        507 2020-04-14 17:44 /user/cloudera/mysql.password.jceks
drwxr-xr-x   - cloudera cloudera          0 2020-04-22 14:25 /user/cloudera/orders
drwxr-xr-x   - cloudera cloudera          0 2020-04-09 15:03 /user/cloudera/ordersboundval
drwxr-xr-x   - cloudera cloudera          0 2020-04-11 14:59 /user/cloudera/outputfolder
-rw-r--r--   1 cloudera cloudera       1355 2020-04-25 13:01 /user/cloudera/udf_example.jar
drwxr-xr-x   - cloudera cloudera          0 2020-04-09 16:14 /user/cloudera/verboseresult
drwxr-xr-x   - cloudera cloudera          0 2020-04-09 15:26 /user/cloudera/whereclauseresult
hive>
```

Run linux commands from hive

We need to use the ! symbol before the command

!ls -ltr /home/cloudera/Desktop;

```
hive> !ls -ltr /home/cloudera/Desktop;
total 9484
-rwxrwxr-x 1 cloudera cloudera     237 Oct 23  2017 Parcels.desktop
-rwxrwxr-x 1 cloudera cloudera     238 Oct 23  2017 Kerberos.desktop
-rwxrwxr-x 1 cloudera cloudera     259 Oct 23  2017 Express.desktop
-rwxrwxr-x 1 cloudera cloudera     284 Oct 23  2017 Enterprise.desktop
-rwxrwxr-x 1 cloudera cloudera     281 Oct 23  2017 Eclipse.desktop
-rw-r--r-- 1 cloudera cloudera 4829457 Mar 14 09:27 card_transactions (copy).csv~
drwxrwxr-x 2 cloudera cloudera    4096 Apr 11 05:58 mapreduce-required-jars
-rw-r--r-- 1 cloudera cloudera 4825594 Apr 14 15:54 card_transactions_new.csv~
-rw-rw-r-- 1 cloudera cloudera    1355 Apr 25 12:43 udf_example.jar
drwxrwxr-x 2 cloudera cloudera    4096 May  2 02:25 inputfolder
drwxrwxr-x 2 cloudera cloudera    4096 May  2 02:26 outputfolder8
drwxrwxr-x 2 cloudera cloudera    4096 May  2 02:31 outputfolder9
drwxrwxr-x 2 cloudera cloudera    4096 May  2 02:31 outputfolder10
drwxrwxr-x 2 cloudera cloudera    4096 May  2 02:36 outputfolder11
hive>
```

Setting hivevar

We can set the value of hive variables and use the value dynamically in our query.

select * from orders limit 5;

```
hive> select * from orders limit 5;
OK
111111  phone   1111    3       1200.0
111112  camera  1111    1       5200.0
111113  broom   1111    1       10.0
111114  broom   2222    2       20.0
111115  t-shirt 4444    2       66.0
Time taken: 0.19 seconds, Fetched: 5 row(s)
hive> describe orders;
OK
id                      bigint
product id              string
customer id             bigint
quantity                int
amount                  double
Time taken: 0.205 seconds, Fetched: 5 row(s)
hive>
```

We can see that third column is customer id.

Setting hivevar

Let us now set a hivevar & then use it in our query.

set hivevar:favourite_customer=1111;

select * from orders where customer_id=${favourite_customer};

```
hive> set hivevar:favourite_customer=1111;
hive> select * from orders where customer_id=${favourite_customer};
OK
111111  phone   1111    3       1200.0
111112  camera  1111    1       5200.0
111113  broom   1111    1       10.0
Time taken: 0.165 seconds, Fetched: 3 row(s)
hive>
```

Printing headers along with data

By default headers are not show as print.header property is set to false by default.

set hive.cli.print.header;

select * from orders limit 5;

```
hive> set hive.cli.print.header;
hive.cli.print.header=false
hive> select * from orders limit 5;
OK
111111  phone   1111    3       1200.0
111112  camera  1111    1       5200.0
111113  broom   1111    1       10.0
111114  broom   2222    2       20.0
111115  t-shirt 4444    2       66.0
Time taken: 0.145 seconds, Fetched: 5 row(s)
hive>
```

Printing headers along with data

Now set the property to true and then see the results.

set hive.cli.print.header=true;

select * from orders limit 5;

```
hive> set hive.cli.print.header=true;
hive> select * from orders limit 5;
OK
orders.id       orders.product_id       orders.customer_id      orders.quantity ord
ers.amount
111111  phone   1111    3       1200.0
111112  camera  1111    1       5200.0
111113  broom   1111    1       10.0
111114  broom   2222    2       20.0
111115  t-shirt 4444    2       66.0
Time taken: 0.193 seconds, Fetched: 5 row(s)
hive>
```

Cartesian product

This is like a cross join.

If there are 2 tables.

Table 1 with 100 rows

Table 2 with 200 rows

Then resulting cartesian join will give 100*200 = 20000 rows.

```
select * from table1,table2
```

We have learnt few good to know things in Hive.

Happy Learning!!!

Follow US

Trainer	Mr. Sumit Mittal
Phone	9108179578
Email	trendytech.sumit@gmail.com
Website	https://trendytech.in/courses/big-data-online-training/
LinkedIn	https://www.linkedin.com/in/bigdatabysumit/
Twitter	@BigdataBySumit
Instagram	bigdatabysumit
Facebook	https://www.facebook.com/trendytech.in/
Youtube	https://www.youtube.com/channel/UCbTggJVf0NDTfWX-C_gUGSg