Hive Partitioning – Cloudera:

Partitioning:

Partition will the split the data into manageable parts and which in turn gives better performance. Partition will be based on column value and data splits may not be of same size per partition

Two types of Partitioning:

Static partitioning

Manually mention the partition name and we need to aware of the incoming data. It is much faster than dynamic partitioning and we ourself loading the data into respective partitions.

Sample Scenario:

Step 1 :

```
create table orders_static_partition(

id string,

customer_id string,

product_id string,

quantity int,

amount double,

zipcode char(5)

)

partitioned by (state char(2))

row format delimited fields terminated by ',';
```

step2:

```
load data local inpath '/home/cloudera/projects/orders_CA.csv into table orders_static_partition partition (state="CA");
```

```
hive> load data local inpath  '/home/cloudera/projects/order_ca.csv' into table testing.orders_static_partition partition (st
ate="CA");
Loading data to table testing.orders_static_partition partition (state=CA)
Partition testing.orders_static_partition{state=CA} stats: [numFiles=1, numRows=0, totalSize=108, rawDataSize=0]
OK
Time taken: 5.361 seconds
hive>
```

Step 3:

Display partition details in hive terminal:

show partitions orders_static_partition;

```
hive> show partitions testing.orders_static_partition;
OK
state=CA
Time taken: 1.741 seconds, Fetched: 1 row(s)
hive>
```

Cloudera Live : Welco... | cloudera@quickstart:...

Step 4:

Query the results using partition

select id,customer_id,amount from testing.orders_static_partition where state = "CA";

```
hive> select id,customer_id,amount from testing.orders_static_partition where state = "CA";
OK
o1      c1      1.11
o2      c2      2.22
o3      c3      3.33
o4      c4      4.44
Time taken: 4.967 seconds, Fetched: 4 row(s)
hive>
```

Cloudera Live : Welco... | cloudera@quickstart:...

HDFS location static partitioning data files:

```
[cloudera@quickstart ~]$ hadoop fs -ls /user/hive/warehouse/testing.db/orders_static_partition/
Found 1 items
drwxrwxrwx   - cloudera supergroup          0 2022-06-25 06:17 /user/hive/warehouse/testing.db/orders_static_partition/state=
CA
[cloudera@quickstart ~]$
```

Dynamic partitioning

Data will get inserted dynamically in the respective partitions without having explicitly creating partitions in hive. Partitions will be created during runtime

Enable the two dynamic properties in Hive to perform this partition:

SET hive.exec.dynamic.partition=true;

SET hive.exec.dynamic.partition.mode=nonstrict;

```
[cloudera@quickstart ~]$ hive

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.properties
WARNING: Hive CLI is deprecated and migration to Beeline is recommended.
hive>
    > SET hive.exec.dynamic.partition=true;
hive> SET hive.exec.dynamic.partition.mode=nonstrict;
hive> create table testing.orders(
    > id string,
    > customer_id string,
    > product_id string,
    > quantity int,
    > amount double,
    > zipcode char(5),
    > state char(2)
    > )
    > row format delimited fields terminated by ',';
OK
```

Step 1:

Create a internal table with No partition with the name “orders” under “testing” database

create table testing.orders(

id string,

customer_id string,

product_id string,

quantity int,

amount double,

zipcode char(5),

state char(2)

)

row format delimited fields terminated by ',';

Step 2:

Load the data from Local file system to non-partitioned table

load data local inpath '/home/cloudera/projects/orders_CA_with_state.csv into table testing.orders;

load data local inpath '/home/cloudera/projects/orders_CT_with_state.csv into table testing.orders;

load data local inpath '/home/cloudera/projects/orders_NY_with_state.csv into table testing.orders;

```
hive> load data local inpath '/home/cloudera/projects/orders_CA_with_state.csv' into table testing.orders;
Loading data to table testing.orders
Table testing.orders stats: [numFiles=1, totalSize=120]
OK
Time taken: 2.39 seconds
hive> load data local inpath '/home/cloudera/projects/orders_NY_with_state.csv' into table testing.orders;
Loading data to table testing.orders
Table testing.orders stats: [numFiles=2, totalSize=272]
OK
Time taken: 0.712 seconds
hive> load data local inpath '/home/cloudera/projects/orders_CT_with_state.csv' into table testing.orders;
Loading data to table testing.orders
Table testing.orders stats: [numFiles=3, totalSize=416]
OK
Time taken: 0.66 seconds
hive>
```

```
hive> select id,state from testing.orders;
OK
o1      CA
o2      CA
o3      CA
o4      CA
o10     CT
o20     CT
o30     CT
o40     CT
o100    NY
o200    NY
o300    NY
o400    NY
Time taken: 1.172 seconds, Fetched: 12 row(s)
hive>
```

Cloudera Live : Welco... | cloudera@quickstart:~

Step 3:

Create a dynamic partitioned table in Hive:

```
create table testing.orders_dynamic(
id string,
customer_id string,
product_id string,
quantity int,
amount double,
zipcode char(5)
)
partitioned by (state char(2))
row format delimited fields terminated by ',';
```

Load non partitioned table to Dynamic partitioned table

```
insert into table testing.orders_dynamic partition(state) select * from testing.orders;
```

```
hive> insert into table testing.orders_dynamic partition(state) select * from testing.orders;
Query ID = cloudera_20220625064444_25010207-8403-4905-9022-a0dff41bfb38
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1656161082814_0001, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1656161082814_0001/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1656161082814_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2022-06-25 06:44:59,977 Stage-1 map = 0%,  reduce = 0%
2022-06-25 06:45:18,083 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.69 sec
MapReduce Total cumulative CPU time: 1 seconds 690 msec
Ended Job = job_1656161082814_0001
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: hdfs://quickstart.cloudera:8020/user/hive/warehouse/testing.db/orders_dynamic/.hive-staging_hive_2022-06-25_06-44-12_922_641784741972575381 4-1/-ext-10000
Loading data to table testing.orders_dynamic partition (state=null)
         Time taken for load dynamic partitions : 1326
        Loading partition {state=CA}
        Loading partition {state=CT}
        Loading partition {state=NY}
         Time taken for adding to write entity : 30
Partition testing.orders_dynamic{state=CA} stats: [numFiles=1, numRows=4, totalSize=100, rawDataSize=96]
Partition testing.orders_dynamic{state=CT} stats: [numFiles=1, numRows=4, totalSize=116, rawDataSize=112]
Partition testing.orders_dynamic{state=NY} stats: [numFiles=1, numRows=4, totalSize=124, rawDataSize=120]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 1.69 sec   HDFS Read: 5747 HDFS Write: 529 SUCCESS
Total MapReduce CPU Time Spent: 1 seconds 690 msec
OK
Time taken: 69.407 seconds
```

```
FAILED: SemanticException [Error 10001]: Table not found
hive> show partitions testing.orders_dynamic;
OK
state=CA
state=CT
state=NY
Time taken: 0.413 seconds, Fetched: 3 row(s)
hive>
```

Cloudera Live : Welco... | cloudera@quickstart:~

Final Output files for dynamic Partitioning:

```
[cloudera@quickstart ~]$ hadoop fs -ls /user/hive/warehouse/testing.db/orders_dynamic/
Found 3 items
drwxrwxrwx   - cloudera supergroup          0 2022-06-25 06:45 /user/hive/warehouse/testing.db/orders_dynamic/state=CA
drwxrwxrwx   - cloudera supergroup          0 2022-06-25 06:45 /user/hive/warehouse/testing.db/orders_dynamic/state=CT
drwxrwxrwx   - cloudera supergroup          0 2022-06-25 06:45 /user/hive/warehouse/testing.db/orders_dynamic/state=NY
[cloudera@quickstart ~]$
```

Cloudera Live : Welco... | cloudera@quickstart:~