Hive File Formats-Cloudera

Creation of Parquet file format in Hive:

create external table testing.productinfo_test(id int,

name string,

cost double,

category string

) stored as parquet;

```
[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> create external table testing.productinfo_data(
    > id int,
    > name string,
    > cost double,
    > category string
    > ) stored as parquet;
```

Load data to Parquet file table:

insert into productinfo_data select * from product_info;

```
hive>
    > insert into productinfo_tests select * from product_info;
FAILED: SemanticException org.apache.hadoop.hive.ql.metadata.InvalidTableException: Table not found productinfo_tests
hive> insert into productinfo_tests select * from testing.product_info;
FAILED: SemanticException org.apache.hadoop.hive.ql.metadata.InvalidTableException: Table not found productinfo_tests
hive> insert into testing.productinfo_tests select * from testing.product_info;
FAILED: SemanticException org.apache.hadoop.hive.ql.metadata.InvalidTableException: Table not found productinfo_tests
hive> insert into testing.productinfo_test select * from testing.product_info;
Query ID = cloudera_20220708105252_e39d629b-f3ab-443e-9cc6-ff708b270450
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_1657301056190_0001, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1657301056190_0001/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1657301056190_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2022-07-08 10:55:54,457 Stage-1 map = 0%,  reduce = 0%
2022-07-08 10:56:44,473 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 3.15 sec
MapReduce Total cumulative CPU time: 3 seconds 150 msec
Ended Job = job_1657301056190_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/productinfo_test/.hive-staging_hive_2022-07-08
_10-52-44_247_5669286475737339012-1/-ext-10000
Loading data to table testing.productinfo_test
Table testing.productinfo_test stats: [numFiles=1, numRows=10, totalSize=852, rawDataSize=40]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 3.15 sec   HDFS Read: 4617 HDFS Write: 932 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 150 msec
OK
Time taken: 248.028 seconds
hive>
```

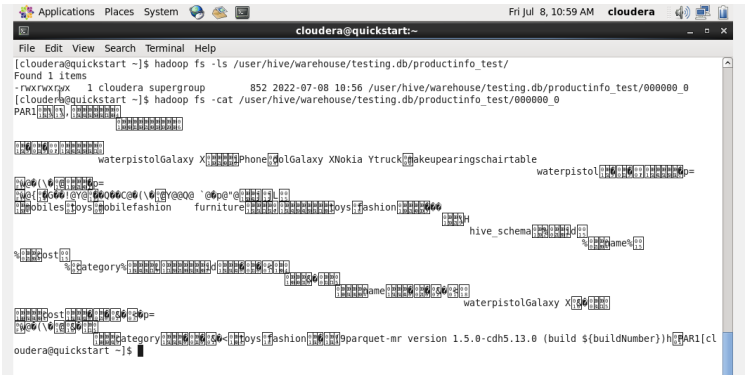
Cloudera Live : Welco... | cloudera@quickstart:~

<u>To see the Parquet file data in HDFS location:</u>

List & Read the Files:

hadoop fs -ls /user/hive/warehouse/testing.db/productinfo_test/

hadoop fs -cat /user/hive/warehouse/testing.db/productinfo_test/000000_0



<u>Commands to get the Metadata & Data stored in Parquet file:</u>

parquet-tools meta 000000_0;

parquet-tools cat 000000_0;

```
[cloudera@quickstart ~]$ parquet-tools meta 000000_0;
creator:     parquet-mr version 1.5.0-cdh5.13.0 (build ${buildNumber})

file schema: hive_schema
--------------------------------------------------------------------------------
id:          OPTIONAL INT32 R:0 D:1
name:        OPTIONAL BINARY O:UTF8 R:0 D:1
cost:        OPTIONAL DOUBLE R:0 D:1
category:    OPTIONAL BINARY O:UTF8 R:0 D:1

row group 1: RC:10 TS:480
--------------------------------------------------------------------------------
id:           INT32 UNCOMPRESSED DO:0 FPO:4 SZ:79/79/1.00 VC:10 ENC:RLE,BIT_PACKED,PLAIN
name:         BINARY UNCOMPRESSED DO:0 FPO:83 SZ:156/156/1.00 VC:10 ENC:RLE,BIT_PACKED,PLAIN
cost:         DOUBLE UNCOMPRESSED DO:0 FPO:239 SZ:129/129/1.00 VC:10 ENC:RLE,BIT_PACKED,PLAIN
category:     BINARY UNCOMPRESSED DO:0 FPO:368 SZ:116/116/1.00 VC:10 ENC:RLE,PLAIN_DICTIONARY,BIT_PACKED
```

```
[cloudera@quickstart ~]$ parquet-tools cat 000000_0;
id = 1
name = iPhone
cost = 379.99
category = mobiles

id = 2
name = doll
cost = 8.99
category = toys

id = 3
name = Galaxy X
cost = 100.0
category = mobile

id = 5
name = Nokia Y
cost = 39.99
category = mobile

id = 6
name = truck
cost = 7.99
category = toys

id = 7
name = makeup
cost = 100.0
category = fashion
```

Hive Serde

To get the generated output hive file as Json Format . SerDe means Serializer and Deserializer. Hive uses SerDe and FileFormat to read and write table rows. Main use of SerDe interface is for IO operations. A SerDe allows hive to read the data from the table and write it back to the HDFS in any custom format

Download the below Jar from the link below and keep in local disk:

- www.congiu.net/hive-json-serde/1.3.7/cdh5/json-serde-1.3.7-jar-with-dependencies.jar
- Add jar /home/cloudera/Downloads/json-serde-1.3.7-jar-with-dependencies.jar;

Table command:

CREATE TABLE orders_json(id bigint, product_id string, customer_id bigint, quantity int, amount double) ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe';

Now insert the data in this table from orders table:

insert overwrite table orders_json select * from orders;

ORC file format Creation:

create table testing.customers_data

(

id bigint,

name string,

address string) stored as orc;

Note: Follow the same process as per Parquet file format

MSCK Repair command in HIVE:

The MSCK REPAIR TABLE command was designed to manually add partitions that are added to or removed from the file system, but are not present in the Hive metastore. This action renders the metastore inconsistent with the file system. You repair the discrepancy manually to synchronize the metastore with the file system, HDFS for example

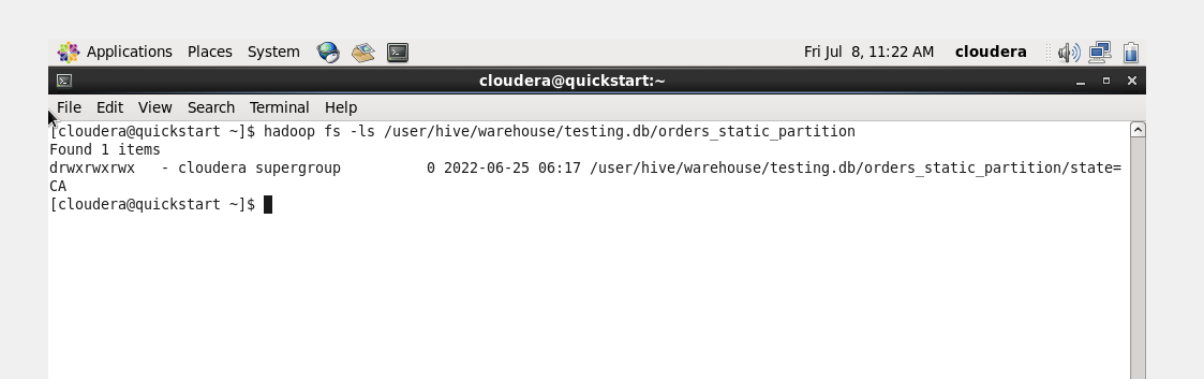
Create a table as below DDL:

Create a table with partitioned by “State=CA” and go the hdfs location.

```
CREATE TABLE `testing.orders_static_partition`(
  `id` string,
  `customer_id` string,
  `product_id` string,
  `quantity` int,
  `amount` double,
  `zipcode` char(5))
PARTITIONED BY (
  `state` char(2));
```

HDFS Location:

/user/hive/warehouse/testing.db/orders_static_partition'



Insert New partitions manually in HDFS location:

Inserted three partitions (State : NT,UN,CK)

```
[cloudera@quickstart ~]$ hadoop fs -mkdir /user/hive/warehouse/testing.db/orders_static_partition/state=NT;
[cloudera@quickstart ~]$ hadoop fs -mkdir /user/hive/warehouse/testing.db/orders_static_partition/state=UN;
[cloudera@quickstart ~]$ hadoop fs -mkdir /user/hive/warehouse/testing.db/orders_static_partition/state=CK;
[cloudera@quickstart ~]$ hadoop fs -ls /user/hive/warehouse/testing.db/orders_static_partition
Found 4 items
drwxrwxrwx   - cloudera supergroup          0 2022-06-25 06:17 /user/hive/warehouse/testing.db/orders_static_partition/state=
CA
drwxr-xr-x   - cloudera supergroup          0 2022-07-08 11:24 /user/hive/warehouse/testing.db/orders_static_partition/state=
CK
drwxr-xr-x   - cloudera supergroup          0 2022-07-08 11:24 /user/hive/warehouse/testing.db/orders_static_partition/state=
NT
drwxr-xr-x   - cloudera supergroup          0 2022-07-08 11:24 /user/hive/warehouse/testing.db/orders_static_partition/state=
UN
[cloudera@quickstart ~]$
```

Even after inserting the partitions it won't show if we run show partitions command since the hive metadata was not missing

```
hive> show partitions testing.orders_static_partition;
OK
state=CA
Time taken: 2.015 seconds, Fetched: 1 row(s)
hive>
```

Run MSCK repair command to bring all the new partitions added in HDFS location:

msck repair table testing.orders_static_partition;

```
hive> msck repair table testing.orders_static_partition;
OK
Partitions not in metastore:    orders_static_partition:state=CK        orders_static_partition:state=NT        orders_static
_partition:state=UN
Repair: Added partition to metastore testing.orders_static_partition:state=CK
Repair: Added partition to metastore testing.orders_static_partition:state=NT
Repair: Added partition to metastore testing.orders_static_partition:state=UN
Time taken: 3.102 seconds, Fetched: 4 row(s)
hive>
```

Post MSCK repair command we can see all the new inserted partitions below:

show partitions orders_static_partition;

```
hive> show partitions testing.orders_static_partition;
OK
state=CA
state=CK
state=NT
state=UN
Time taken: 0.333 seconds, Fetched: 4 row(s)
hive>
```

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