Tech Tip Hive Big SQL

Tips for populating Big SQL and Hive Hadoop tables with DATE types

Nailah Bissoon
Published on December 16, 2016 / *Updated on September 11, 2017*
1

When creating external Hive tables defined with DATE columns, ensure that the values in the data files on HDFS correspond to DATE values and not a mix of DATE and TIMESTAMP values. The same is true for when creating Hive tables and using the Hive INSERT or INSERT...SELECT commands to add data to tables. When Hive expects a DATE type, but instead finds a TIMESTAMP type in the data file, then a NULL value is inserted to the table. NULL values can have a negative impact on query performance especially for queries performed against partitioned tables where the partitioning keys are NULL values. This is because Hive will put all NULL values into one partition.

Big SQL implicitly casts values during external table creation and insert which avoids unwanted NULL values being stored in the tables. Therefore if tables are created using Big SQL and populated using Big SQL then there can be a mix of DATE and TIMESTAMP values in the data files. However, in Big SQL, the DATE data type is mapped and stored as a Hive TIMESTAMP type by default. This means that the actual date fields for tables created with the DATE type in Big SQL will consist of date and time portions in Hive, but in Big SQL only the date portion is returned by Big SQL. This also means that if a table is created in Big SQL but populated using Hive INSERT or INSERT...SELECT then the entries in the data files should be TIMESTAMP values instead of DATE types. There could be a lot of confusion around this therefore there are a few examples below to illustrate this behavior better.

Note also that there is a new DATE STORED AS DATE type in which some optimizations were added in Big SQL 4.2 and later releases so that the DATE type is mapped to a DATE type in Hive. If this is the case then if the table is created from Big SQL and populated using Hive INSERT/INSERT...SELECT then a DATE type is expected in the input data file.

Due to the linux kernel performance issues with TIMESTAMPs generated using Map Reduce type applications such as Hive, mapping a DATE to a DATE type in Hive could perform better than mapping a DATE type to a TIMESTAMP in Hive but it also depends on the frequency of times the DATE values are referenced as well. For Big SQL INSERT...SELECT since Map Reduce is not used, DATE stored as TIMESTAMP could perform better than DATE stored as DATE types. There are some examples in this blog to guide you through the decision of whether the DATE type should be stored as a DATE in Hive or as a TIMESTAMP in Hive.

The next section will take you through a few examples of creating Big SQL and Hive tables with DATE types.

Create Big SQL Hadoop table with DATE type

Consider the example table and contents of one of the files of this external table as follows:

```
$hadoop fs -mkdir '/user/hadoop/dates/tab1'
$hadoop fs -put dttab.txt '/user/hadoop/dates/tab1
$hadoop fs -ls '/user/hadoop/dates/tab1'
Found 1 items
-rw-r--r--   3 nbissoon hdfs         87 2016-12-16 13:41 /user/hadoop/dates/tab1/dttab.txt

$hadoop fs -cat /user/hadoop/dates/tab1/f1.dat
1,1997-12-15 11:32:21
2,2011-12-01 00:00:00.000
3,2014-09-10 00:00:00.000
4,2008-04-21

jsqsh>CREATE EXTERNAL HADOOP TABLE dttab1
```

```
jsqsh> select * from dttab1;
+----+------------+
| C1 | C2         |
+----+------------+
|  1 | 1997-12-15 |
|  2 | 2011-12-01 |
|  3 | 2014-09-10 |
|  4 | 2008-04-21 |
+----+------------+

hive> describe dttab1;
c1                      int
c2                      timestamp               /*@type=date*/

hive> select * from dttab;
1       1997-12-15 11:32:21
2       2011-12-01 00:00:00
3       2014-09-10 00:00:00
4       NULL
```

Note that all the rows are successfully inserted into the table in Big SQL and the time portion of the TIMESTAMP entries are simply omitted. However, in Hive, because the DATE is stored as a TIMESTAMP, and the last entry is not a valid TIMESTAMP then it would be added as a NULL value in the table.

Also note that the values returned from Big SQL are slightly different than Hive in that there is one less NULL value in Big SQL. As a side note, if the table is created in Big SQL but a TIMESTAMP type is used instead of a DATE type then Big SQL and Hive will store both the date and the time portions of the entries. If there is no time portion in the input file i.e. a DATE is in the input file, then Big SQL will return the time portion as 00:00:00.000 but in Hive this value will be returned as a NULL.

```
$hadoop fs -mkdir '/user/hadoop/dates/tab2'
$hadoop fs -put dttab.txt '/user/hadoop/dates/tab2
$hadoop fs -ls '/user/hadoop/dates/tab2'
Found 1 items
-rw-r--r--   3 nbissoon hdfs         87 2016-12-16 13:41 /user/hadoop/dates/tab2/dttab.txt

$hadoop fs -cat /user/hadoop/dates/tab2/f1.dat
1,1997-12-15 11:32:21
2,2011-12-01 00:00:00.000
3,2014-09-10 00:00:00.000
4,2008-04-21

jsqsh> CREATE HADOOP TABLE dttab2
   ( c1 int, c2 timestamp )
   ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
   LOCATION '/user/hadoop/dates/tab2';

hive> describe dttab2;
c1                      int
c2                      timestamp

jsqsh> select * from dttab2;
+----+-------------------------+
| C1 | C2                      |
+----+-------------------------+
|  1 | 1997-12-15 11:32:21.000 |
|  2 | 2011-12-01 00:00:00.000 |
|  3 | 2014-09-10 00:00:00.000 |
|  4 | 2008-04-21 00:00:00.000 |
+----+-------------------------+

hive> select * from dttab2;
1       1997-12-15 11:32:21
2       2011-12-01 00:00:00
3       2014-09-10 00:00:00
4       NULL
```

Create Big SQL Hadoop table with DATE STORED AS DATE type

In the majority of cases, if a DATE is expected in the table, the data file usually has a DATE type. Therefore, in Big SQL 4.2 and later releases, one can alternatively store the DATE type as a DATE in Hive by using the DATE STORED AS DATE clause during table creation. When this type is used, if the input entries are not actual DATE entries then Big SQL and Hive will store these entries as NULL values.

```
$hadoop fs -mkdir '/user/hadoop/dates/tab3'
$hadoop fs -put dttab.txt '/user/hadoop/dates/tab3'
$hadoop fs -ls '/user/hadoop/dates/tab3'
Found 1 items
```

```
1,1997-12-15 11:32:21
2,2011-12-01 00:00:00.000
3,2014-09-10 00:00:00.000
4,2008-04-21

jsqsh> CREATE HADOOP TABLE dttab3
   ( c1 int, c2 date stored as date )
   ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
   LOCATION '/user/hadoop/dates/tab3';

hive> describe dttab3;
OK
c1                      int
c2                      date

jsqsh> select * from dttab3;
+----+------------+
| C1 | C2         |
+----+------------+
|  1 | [NULL]     |
|  2 | [NULL]     |
|  3 | [NULL]     |
|  4 | 2008-04-21 |
+----+------------+

hive> select * from dttab3;
1       NULL
2       NULL
3       NULL
4       2008-04-21
```

Note that when the table is created with the DATE STORED AS DATE type the entries in the table match the Hive entries but also note that there are more NULL values. Therefore when choosing to use the DATE STORED AS DATE type in Big SQL ensure that the input file consist of actual valid DATE entries instead of a mix of DATE and TIMESTAMP entries.

Create Hive table with DATE type

If the table is created in Hive with a DATE type then this will be mapped to a DATE type in Big SQL. Note below that the HCAT_SYNC_OBJECTS stored procedure needs to be called to sync the Big SQL and the Hive catalog when the table is created in Hive. More information on syncing of the Big SQL and Hive catalogs is also available for further reading.

```
$hadoop fs -mkdir '/user/hadoop/dates/tabh1'
$hadoop fs -put dttab.txt '/user/hadoop/dates/tabh1'
$hadoop fs -cat /user/hadoop/dates/tabh1/dttab.txt
1,1997-12-15 11:32:21
2,2011-12-01 00:00:00.000
3,2014-09-10 00:00:00.000
4,2008-04-21

hive> CREATE TABLE dttabh1
   (c1 int, c2 date )
   ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
   LOCATION '/user/hadoop/dates/tabh1';

hive> describe dttabh1;
c1                      int
c2                      date

hive> select * from dttabh1 ;
1       NULL
2       NULL
3       NULL
4       2008-04-21

jsqsh> call syshadoop.hcat_sync_objects('nbissoon','dttabh1','a','REPLACE','CONTINUE');

  Result set 1
  --------------

  OBJSCHEMA OBJNAME OBJATTRIB TYPE STATUS DETAILS
  --------- ------- --------- ---- ------ -------
  NBISSOON  DTTABH1 -         T    OK     -

  1 record(s) selected.

  Return Status = 0
jsqsh> select * from dttabh1;
+----+------------+
| C1 | C2         |
+----+------------+
|  1 | [NULL]     |
|  2 | [NULL]     |
|  3 | [NULL]     |
|  4 | 2008-04-21 |
```

Note that both Big SQL and Hive is now reporting the same output but there ar
mostly consists of TIMESTAMP values.

Create Big SQL Hadoop table with DATE ty ve INSERT

If a table is created in Big SQL with a DATE type but Hive Insert or INSERT...SE e input data file or table being selected from must consist of TIMESTAMP values ive. This is because when the table is created in Big SQL DATE types are mapped t hat when INSERT is used there is 1 file for each row of the table. This is not optima e optimized for large files, but INSERT statements are used for demonstration p

```
jsqsh> CREATE HADOOP TABLE dttab4
   ( c1 int, c2 date )
   ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
   LOCATION '/user/hadoop/dates/tab4'
0 rows affected (total: 1.9s)

hive> describe dttab4;
c1                      int
c2                      timestamp               /*@type=date*/

hive> insert into dttab4 values (1, '1997-12-15 11:32:21');
hive> insert into dttab4 values(2,'2011-12-01 00:00:00.000');
hive> insert into dttab4 values(3,'2014-09-10 00:00:00.000');
hive> insert into dttab4 values(4,'2008-04-21');

hive> select * from dttab4;
1       1997-12-15 11:32:21
2       2011-12-01 00:00:00
3       2014-09-10 00:00:00
4       NULL

jsqsh> select * from dttab4 order by c1;
+----+------------+
| C1 | C2         |
+----+------------+
|  1 | 1997-12-15 |
|  2 | 2011-12-01 |
|  3 | 2014-09-10 |
|  4 | [NULL]     |
+----+------------+

$hadoop fs -ls '/user/hadoop/dates/tab4'
Found 4 items
-rwxr-xr-x   3 nbissoon hdfs         22 2016-12-16 13:08 /user/hadoop/dates/tab4/000000_0
-rwxr-xr-x   3 nbissoon hdfs         22 2016-12-16 13:09 /user/hadoop/dates/tab4/000000_0_copy_1
-rwxr-xr-x   3 nbissoon hdfs         22 2016-12-16 13:09 /user/hadoop/dates/tab4/000000_0_copy_2
-rwxr-xr-x   3 nbissoon hdfs          5 2016-12-16 13:10 /user/hadoop/dates/tab4/000000_0_copy_3

$hadoop fs -cat '/user/hadoop/dates/tab4/000000_0'
1,1997-12-15 11:32:21
$hadoop fs -cat '/user/hadoop/dates/tab4/000000_0_copy_1'
2,2011-12-01 00:00:00
$hadoop fs -cat '/user/hadoop/dates/tab4/000000_0_copy_2'
3,2014-09-10 00:00:00
$hadoop fs -cat '/user/hadoop/dates/tab4/000000_0_copy_3'
4,\N
```

However, if the table is created in Big SQL with a DATE STORED AS DATE type and Hive Insert or INSERT...SELECT is used to populate the table then the input data file must contain a DATE type otherwise NULL values will be added by Hive.

```
jsqsh> CREATE HADOOP TABLE dttab5
   ( c1 int, c2 date stored as date )
   ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
   LOCATION '/user/hadoop/dates/tab5'

hive> describe dttab5;
c1                      int
c2                      date

hive> insert into dttab5 values (1, '1997-12-15 11:32:21');
hive> insert into dttab5 values(2,'2011-12-01 00:00:00.000');
hive> insert into dttab5 values(3,'2014-09-10 00:00:00.000');
hive> insert into dttab5 values(4,'2008-04-21');

jsqsh> select * from dttab5 order by c1;
+----+------------+
| C1 | C2         |
+----+------------+
```

```
|  4 | 2008-04-21 |
+----+------------+

hive> select * from dttab5;
OK
1       NULL
2       NULL
3       NULL
4       2008-04-21

$hadoop fs -ls '/user/hadoop/dates/tab5'
Found 4 items
-rwxr-xr-x   3 nbissoon hdfs          5 2016-12-16 13:25 /user/
-rwxr-xr-x   3 nbissoon hdfs          5 2016-12-16 13:26 /user/
-rwxr-xr-x   3 nbissoon hdfs          5 2016-12-16 13:26 /user/
-rwxr-xr-x   3 nbissoon hdfs         13 2016-12-16 13:26 /user/

$hadoop fs -cat '/user/hadoop/dates/tab5/000000_0'
1,\N
$hadoop fs -cat '/user/hadoop/dates/tab5/000000_0_copy_1'
2,\N
$hadoop fs -cat '/user/hadoop/dates/tab5/000000_0_copy_2'
3,\N
$hadoop fs -cat '/user/hadoop/dates/tab5/000000_0_copy_3'
4,2008-04-21
```


Create Big SQL Hadoop Partitioned table with DATE types populated using Big SQL INSERT...SELECT

The table created in the examples above can be used to populate partitioned tables using Big SQL INSERT...SELECT. Partitioned tables are recommended for performance advantages.

```
jsqsh> CREATE HADOOP TABLE dttab6 ( c1 int )
partitioned by (c2 date)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
LOCATION '/user/hadoop/dates/tab6'
0 rows affected (total: 1.9s)

jsqsh> select * from dttab1;
+----+------------+
| C1 | C2         |
+----+------------+
|  1 | 1997-12-15 |
|  2 | 2011-12-01 |
|  3 | 2014-09-10 |
|  4 | 2008-04-21 |
+----+------------+

hive> select * from dttab1;
1       1997-12-15 11:32:21
2       2011-12-01 00:00:00
3       2014-09-10 00:00:00
4       NULL

jsqsh> insert into dttab6 select * from dttab1;

jsqsh> select * from dttab6;
+----+------------+
| C1 | C2         |
+----+------------+
|  2 | 2011-12-01 |
|  1 | 1997-12-15 |
|  3 | 2014-09-10 |
|  4 | 2008-04-21 |

hive> select * from dttab6;
1       1997-12-15 00:00:00
4       2008-04-21 00:00:00
2       2011-12-01 00:00:00
3       2014-09-10 00:00:00
```

Note that even though the last entry 2008-04-21 is not in the Hive dttab1 table, because the INSERT is driven by Big SQL and the origin table consisted of this field, the destination table will have the 2008-04-21 00:00:00 entry in Hive.

Looking at HDFS, note below that there are 4 directories for each of the partitions and they resemble a TIMESTAMP (the colons in the time portion is represented as %3A on HDFS). This is expected because the DATE type in Big SQL is being mapped to a TIMESTAMP in Hive.

```
$hadoop fs -ls '/user/hadoop/dates/tab6'
```

```
drwxrwxrwx   - nbissoon hdfs          0 2016-11-22 15:56 /user/
00%3A00%3A00
drwxrwxrwx   - nbissoon hdfs          0 2016-11-22 15:56 /user/
00%3A00%3A00
drwxrwxrwx   - nbissoon hdfs          0 2016-11-22 15:56 /user/
00%3A00%3A00
```

The next section will show how the NULLs could persist even when INSERT..SE ive.

Create Big SQL Hadoop Partitioned table w ted using Hive INSERT...SELECT

If Hive is used to populate the partitioned tables using INSERT...SELECT then a the table in which it is selecting from and insert the rows into the new table.

```
jsqsh> CREATE HADOOP TABLE dttab7 ( c1 int ) partitioned by (c2 date) ROW FORMAT DELIMITED FIELDS
TERMINATED BY ',' LOCATION '/user/hadoop/dates/tab7'
0 rows affected (total: 1.9s)

hive> select * from dttab1;
1       1997-12-15 11:32:21
2       2011-12-01 00:00:00
3       2014-09-10 00:00:00
4       NULL

hive> set hive.exec.dynamic.partition.mode=nonstrict;
hive> insert into dttab7 partition(c2) select * from dttab1;

hive> select * from dttab7;
1       1997-12-15 11:32:21
2       2011-12-01 00:00:00
3       2014-09-10 00:00:00
4       NULL

jsqsh> select * from dttab7;
+----+------------+
| C1 | C2         |
+----+------------+
|  3 | 2014-09-10 |
|  1 | 1997-12-15 |
|  2 | 2011-12-01 |
|  4 | [NULL]     |
+----+------------+
```

Since the origin table dttab1 did not have the 2008-04-21 date, the dttab7 table also does not have this entry and it is stored as NULL.

If we compare the two tables you will notice that the table populated through Big SQL will have less NULL values. Recall one was populated using Big SQL Insert...Select and the other was populated using Hive Insert...Select.

In HDFS, NULL values are stored in the __HIVE_DEFAULT_PARTITION___ directory. If there are a lot of NULL values in partitioned tables this can cause performance problems because all the NULL values will go to one partition.

```
$hadoop fs -ls /user/hadoop/dates/tab6
Found 4 items
drwxr-xr-x   - nbissoon hdfs          0 2016-11-23 10:53 /user/hadoop/dates/tab6/c2=1997-12-15
11%3A32%3A21
drwxr-xr-x   - nbissoon hdfs          0 2016-11-23 10:53 /user/hadoop/dates/tab6/c2=2011-12-01
00%3A00%3A00
drwxr-xr-x   - nbissoon hdfs          0 2016-11-23 10:53 /user/hadoop/dates/tab6/c2=2014-09-10
00%3A00%3A00
drwxr-xr-x   - nbissoon hdfs          0 2016-11-23 10:53
/user/hadoop/dates/tab6/c2=__HIVE_DEFAULT_PARTITION__
```

Conclusion

Always make sure if table creation or INSERT/INSERT...SELECT statements are driven from Hive that the original data files from which the table is created or selected from do not have a mix of TIMESTAMP and DATE values as DATE values are expected. If a DATE value is not found a NULL value is added. Big SQL external table creation and INSERT/INSERT...SELECT will implicitly cast the values to the expected format so there is much less chance of NULL values when there is a mix of DATE

type in Big SQL to Hive Date Type. If a table is created in Big SQL but Hive INSE he table then TIMESTAMP values should be in the input file instead of DATE value DATE STORED AS DATE type.

Thanks to Kathy Mcknight for reviewing this blog

TAGS **HIVE, BIG SQL, INSERT, INSERT...SELECT, DATE, TIMESTAMP, NULL, HIVE_DEFAULT_PARTITIO**

Nailah Bissoon

Nailah Bissoon is a technical leader in the Toronto Canada IBM Lab. She has over 12 years experie management systems. She is the currently the Big SQL Development Performance Architect in the responsible for managing development items that reinforce the robustness of Big SQL from the pe since Jan 2015.

1 comment on"Tips for populating Big SQL and Hive Hadoop tables with DATE types"

Big SQL Supported Data Types in v4 and v5 - biva • August 07, 2017

[...] By default, Big SQL stores the DATE type as a TIMESTAMP in Hive. In Big SQL 4.2, support was added to store the DATE as a DATE type in Hive by using the CREATE HADOOP TABLE ...DATE STORED AS DATE clause.You can not partition a Hadoop table using a TIMESTAMP type but you can partition a Hadoop table using a DATE type. When importing Hive DATE types (via HCAT_SYNC_OBJECTS) or when the DATE STORED AS DATE clause is used, DATE types in Hive are represented as DATE types in Big SQL. In this case the Java I/O interface is used when accessing and writing data to these tables. The usage of the ORC file format is encouraged when DATE types are used in this manner. More information on DATE types in Big SQL and Hive and be found in tips for populating tables with date types in Big SQL and Hive. [...]

Reply

Join The Discussion

Your email address will not be published. Required fields are marked *

Enter your comments...

Name *

Email *

Website

Post Comment