Peer-Graded Assignment: Data Management
Course: Managing Big Data in Clusters and Cloud Storage
Name: Suhaimi William Chan
Date: April 17, 2021

(Include your name and today's date above.)

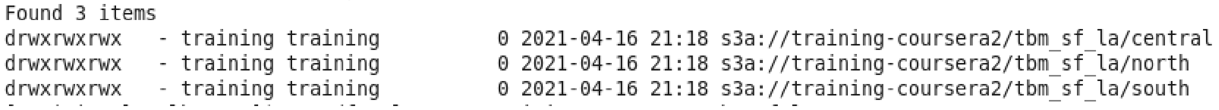
Assignment

Create a table named **tbm_sf_la** in the database named **dig** to store the data from three tunnel boring machines (TBMs), which is currently stored in S3 in three separate subdirectories under a directory named **tbm_sf_la** in the bucket named **training-coursera2**. In this document, describe the steps taken to complete this task.

Solution

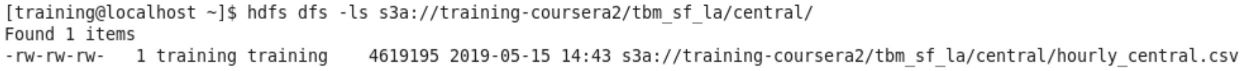
I performed the following steps to complete this task:

1. Check the content of the aws s3 file directory using the following command line:
2. $ hdfs dfs –ls s3a://training-coursera2/tbm-sf-la/



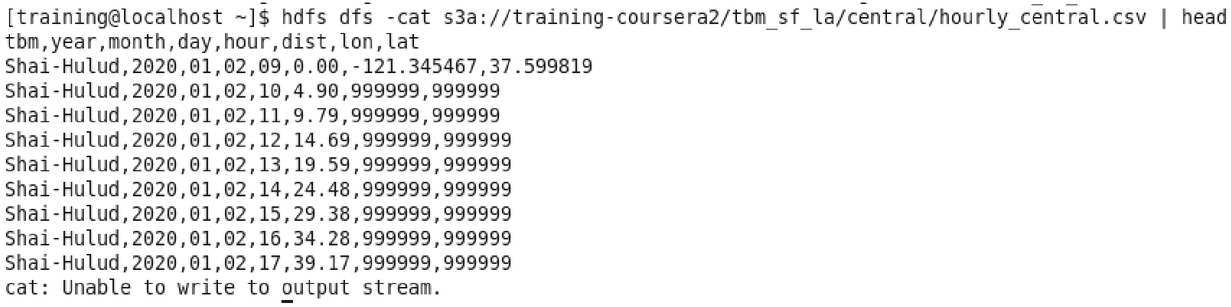
```
Found 3 items
drwxrwxrwx   - training training          0 2021-04-16 21:18 s3a://training-coursera2/tbm_sf_la/central
drwxrwxrwx   - training training          0 2021-04-16 21:18 s3a://training-coursera2/tbm_sf_la/north
drwxrwxrwx   - training training          0 2021-04-16 21:18 s3a://training-coursera2/tbm_sf_la/south
```

3. Check the content of central directory using the following command line:
   $ hdfs dfs –ls s3a://training-coursera2/tbm_sf_la/central/



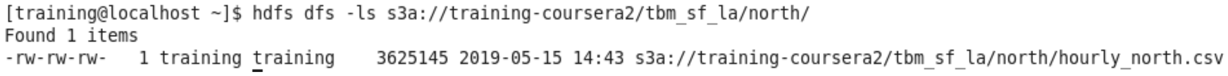
```
[training@localhost ~]$ hdfs dfs -ls s3a://training-coursera2/tbm_sf_la/central/
Found 1 items
-rw-rw-rw-   1 training training    4619195 2019-05-15 14:43 s3a://training-coursera2/tbm_sf_la/central/hourly_central.csv
```

4. Check the sample content of hourly_central.csv file using the following command line:
   $ hdfs dfs –cat s3a://training-coursera2/tbm_sf_la/central/hourly_central.csv | head



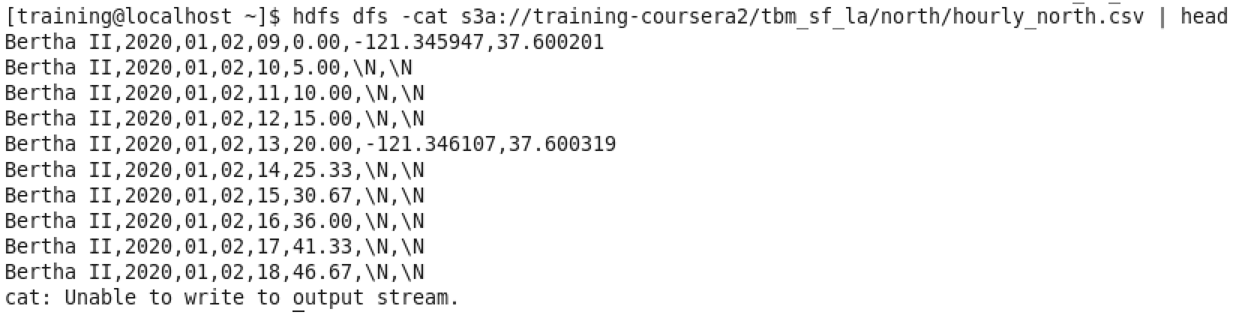
```
[training@localhost ~]$ hdfs dfs -cat s3a://training-coursera2/tbm_sf_la/central/hourly_central.csv | head
tbm,year,month,day,hour,dist,lon,lat
Shai-Hulud,2020,01,02,09,0.00,-121.345467,37.599819
Shai-Hulud,2020,01,02,10,4.90,999999,999999
Shai-Hulud,2020,01,02,11,9.79,999999,999999
Shai-Hulud,2020,01,02,12,14.69,999999,999999
Shai-Hulud,2020,01,02,13,19.59,999999,999999
Shai-Hulud,2020,01,02,14,24.48,999999,999999
Shai-Hulud,2020,01,02,15,29.38,999999,999999
Shai-Hulud,2020,01,02,16,34.28,999999,999999
Shai-Hulud,2020,01,02,17,39.17,999999,999999
cat: Unable to write to output stream.
```

5. Check the content of north directory using the following command line:
   $ hdfs dfs –ls s3a://training-coursera2/tbm_sf_la/north/



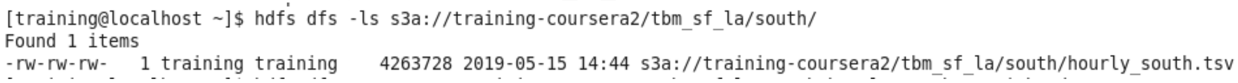
```
[training@localhost ~]$ hdfs dfs -ls s3a://training-coursera2/tbm_sf_la/north/
Found 1 items
-rw-rw-rw-   1 training training    3625145 2019-05-15 14:43 s3a://training-coursera2/tbm_sf_la/north/hourly_north.csv
```

6. Check the sample content of hourly_north.csv file using the following command line:
   $ hdfs dfs –cat s3a://training-coursera2/tbm_sf_la/north/hourly_north.csv | head



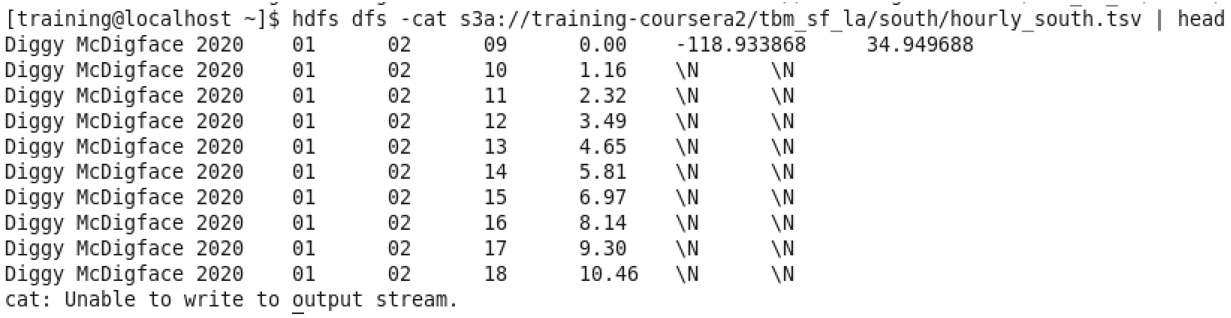
```
[training@localhost ~]$ hdfs dfs -cat s3a://training-coursera2/tbm_sf_la/north/hourly_north.csv | head
Bertha II,2020,01,02,09,0.00,-121.345947,37.600201
Bertha II,2020,01,02,10,5.00,\N,\N
Bertha II,2020,01,02,11,10.00,\N,\N
Bertha II,2020,01,02,12,15.00,\N,\N
Bertha II,2020,01,02,13,20.00,-121.346107,37.600319
Bertha II,2020,01,02,14,25.33,\N,\N
Bertha II,2020,01,02,15,30.67,\N,\N
Bertha II,2020,01,02,16,36.00,\N,\N
Bertha II,2020,01,02,17,41.33,\N,\N
Bertha II,2020,01,02,18,46.67,\N,\N
cat: Unable to write to output stream.
```

7. Check the content of south directory using the following command line:
   $ hdfs dfs –ls s3a://training-coursera2/tbm_sf_la/south/



```
[training@localhost ~]$ hdfs dfs -ls s3a://training-coursera2/tbm_sf_la/south/
Found 1 items
-rw-rw-rw-   1 training training    4263728 2019-05-15 14:44 s3a://training-coursera2/tbm_sf_la/south/hourly_south.tsv
```

8. Check the sample content of hourly_south.tsv file using the following command line:
   $ hdfs dfs –cat s3a://training-coursera2/tbm_sf_la/south/hourly_south.tsv | head



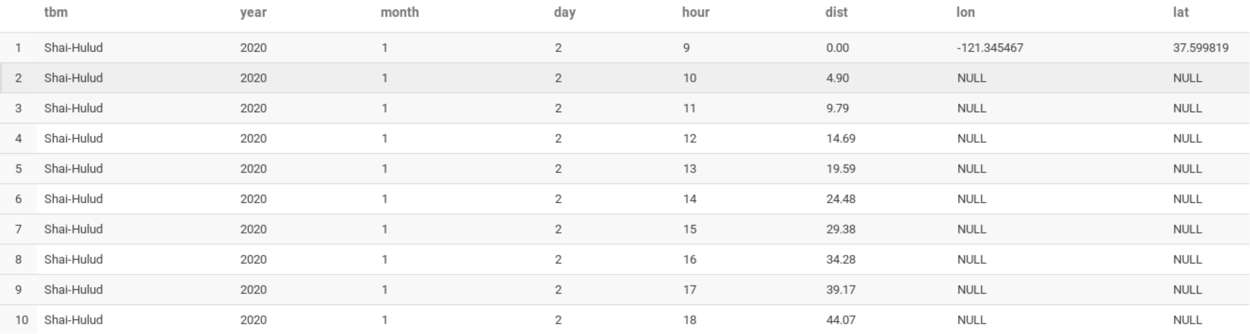
```
[training@localhost ~]$ hdfs dfs -cat s3a://training-coursera2/tbm_sf_la/south/hourly_south.tsv | head
Diggy McDigface 2020    01      02      09      0.00    -118.933868     34.949688
Diggy McDigface 2020    01      02      10      1.16    \N      \N
Diggy McDigface 2020    01      02      11      2.32    \N      \N
Diggy McDigface 2020    01      02      12      3.49    \N      \N
Diggy McDigface 2020    01      02      13      4.65    \N      \N
Diggy McDigface 2020    01      02      14      5.81    \N      \N
Diggy McDigface 2020    01      02      15      6.97    \N      \N
Diggy McDigface 2020    01      02      16      8.14    \N      \N
Diggy McDigface 2020    01      02      17      9.30    \N      \N
Diggy McDigface 2020    01      02      18      10.46   \N      \N
cat: Unable to write to output stream.
```

9. Create an external table dig.hourly_central for hourly_central.csv file using the header name from the file. Convert 999999 values to NULL. Here is the SQL command in Impala:

```
CREATE EXTERNAL TABLE dig.hourly_central (
   tbm STRING,
   year SMALLINT,
   month TINYINT,
   day TINYINT,
   hour TINYINT,
   dist DECIMAL(8,2),
   lon DECIMAL(12,6),
   lat DECIMAL(12,6)
   )
   ROW FORMAT DELIMITED
   FIELDS TERMINATED BY ','
   STORED AS TEXTFILE
   LOCATION 's3a://training-coursera2/tbm_sf_la/central/'
   TBLPROPERTIES('skip.header.line.count'='1','serialization.null.format'='999999');
```

10. Check a few samples of the data in newly created external table dig.hourly_central using the following SQL command in Impala:

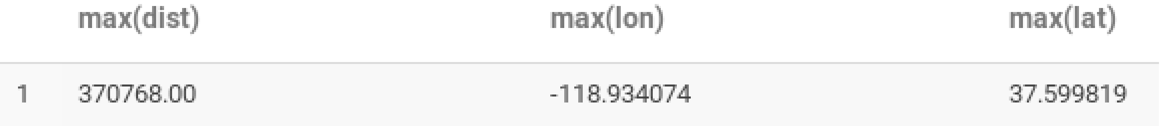
```
SELECT * FROM dig.hourly_central LIMIT 10;
```



	tbm	year	month	day	hour	dist	lon	lat
1	Shai-Hulud	2020	1	2	9	0.00	-121.345467	37.599819
2	Shai-Hulud	2020	1	2	10	4.90	NULL	NULL
3	Shai-Hulud	2020	1	2	11	9.79	NULL	NULL
4	Shai-Hulud	2020	1	2	12	14.69	NULL	NULL
5	Shai-Hulud	2020	1	2	13	19.59	NULL	NULL
6	Shai-Hulud	2020	1	2	14	24.48	NULL	NULL
7	Shai-Hulud	2020	1	2	15	29.38	NULL	NULL
8	Shai-Hulud	2020	1	2	16	34.28	NULL	NULL
9	Shai-Hulud	2020	1	2	17	39.17	NULL	NULL
10	Shai-Hulud	2020	1	2	18	44.07	NULL	NULL

11. Check the maximum values for field dist, lon and lat in dig.hourly_central using the following SQL command in Impala:

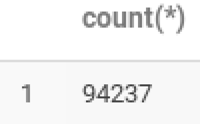
```
SELECT MAX(dist), MAX(lon), MAX(lat)
   FROM dig.hourly_central;
```



	max(dist)	max(lon)	max(lat)
1	370768.00	-118.934074	37.599819

12. Check how many records in dig.hourly_central using the following SQL command in Impala:

```
SELECT COUNT(*) FROM dig.hourly_central;
```



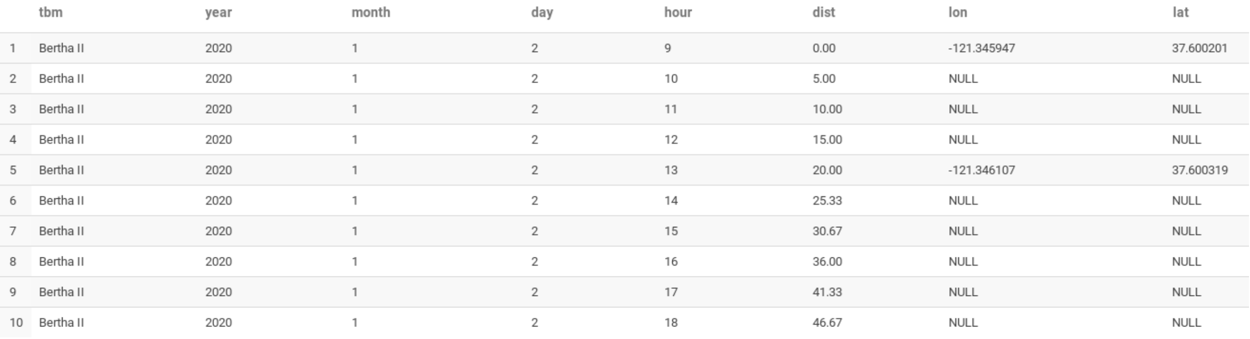
	count(*)
1	94237

13. Create an external table dig.hourly_north for hourly_north.csv file using the following SQL command in Impala:

```
CREATE EXTERNAL TABLE dig.hourly_north (
   tbm STRING,
   year SMALLINT,
   month TINYINT,
   day TINYINT,
   hour TINYINT,
   dist DECIMAL(8,2),
   lon DECIMAL(12,6),
   lat DECIMAL(12,6)
   )
   ROW FORMAT DELIMITED
   FIELDS TERMINATED BY ','
   STORED AS TEXTFILE
   LOCATION 's3a://training-coursera2/tbm_sf_la/north/';
```

14. Check a few samples of the data in newly created external table dig.hourly_north using the following SQL command in Impala:

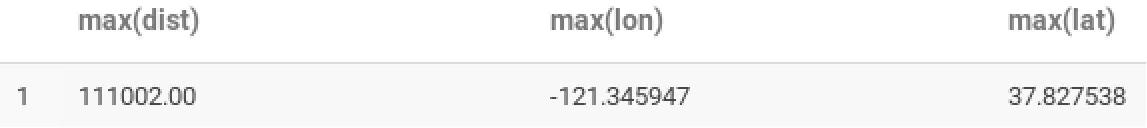
```
SELECT * FROM dig.hourly_north LIMIT 10;
```



	tbm	year	month	day	hour	dist	lon	lat
1	Bertha II	2020	1	2	9	0.00	-121.345947	37.600201
2	Bertha II	2020	1	2	10	5.00	NULL	NULL
3	Bertha II	2020	1	2	11	10.00	NULL	NULL
4	Bertha II	2020	1	2	12	15.00	NULL	NULL
5	Bertha II	2020	1	2	13	20.00	-121.346107	37.600319
6	Bertha II	2020	1	2	14	25.33	NULL	NULL
7	Bertha II	2020	1	2	15	30.67	NULL	NULL
8	Bertha II	2020	1	2	16	36.00	NULL	NULL
9	Bertha II	2020	1	2	17	41.33	NULL	NULL
10	Bertha II	2020	1	2	18	46.67	NULL	NULL

15. Check the maximum values for field dist, lon and lat in dig.hourly_north using the following SQL command in Impala:

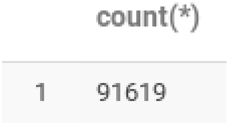
```
SELECT MAX(dist), MAX(lon), MAX(lat)
   FROM dig.hourly_north;
```



	max(dist)	max(lon)	max(lat)
1	111002.00	-121.345947	37.827538

16. Check how many records in dig.hourly_north using the following SQL command in Impala:

```
SELECT COUNT(*) FROM dig.hourly_north;
```



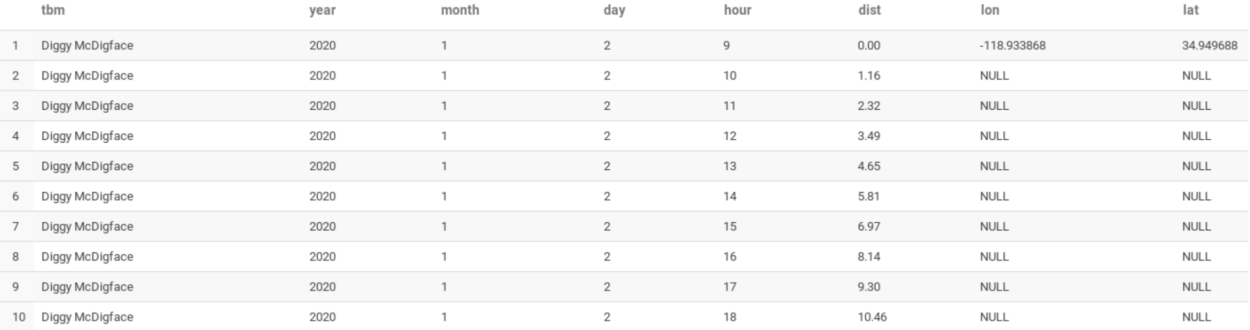
	count(*)
1	91619

17. Create an external table dig.hourly_south for hourly_south.tsv file using the following SQL command in Impala:

```
CREATE EXTERNAL TABLE dig.hourly_south (
   tbm STRING,
   year SMALLINT,
   month TINYINT,
   day TINYINT,
   hour TINYINT,
   dist DECIMAL(8,2),
   lon DECIMAL(12,6),
   lat DECIMAL(12,6)
   )
   ROW FORMAT DELIMITED
   FIELDS TERMINATED BY '\t'
   STORED AS TEXTFILE
   LOCATION 's3a://training-coursera2/tbm_sf_la/south/';
```

18. Check a few samples of the data in newly created external table dig.hourly_south using the following SQL command in Impala:

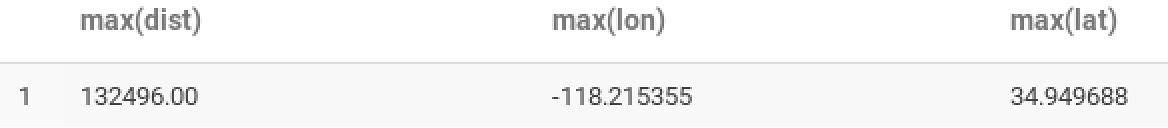
```
SELECT * FROM dig.hourly_south LIMIT 10;
```



	tbm	year	month	day	hour	dist	lon	lat
1	Diggy McDigface	2020	1	2	9	0.00	-118.933868	34.949688
2	Diggy McDigface	2020	1	2	10	1.16	NULL	NULL
3	Diggy McDigface	2020	1	2	11	2.32	NULL	NULL
4	Diggy McDigface	2020	1	2	12	3.49	NULL	NULL
5	Diggy McDigface	2020	1	2	13	4.65	NULL	NULL
6	Diggy McDigface	2020	1	2	14	5.81	NULL	NULL
7	Diggy McDigface	2020	1	2	15	6.97	NULL	NULL
8	Diggy McDigface	2020	1	2	16	8.14	NULL	NULL
9	Diggy McDigface	2020	1	2	17	9.30	NULL	NULL
10	Diggy McDigface	2020	1	2	18	10.46	NULL	NULL

19. Check the maximum values for field dist, lon and lat in dig.hourly_south using the following SQL command in Impala:

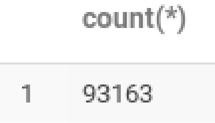
```
SELECT MAX(dist), MAX(lon), MAX(lat)
  FROM dig.hourly_south;
```



	max(dist)	max(lon)	max(lat)
1	132496.00	-118.215355	34.949688

20. Check how many records in dig.hourly_south using the following SQL command in Impala:

```
SELECT COUNT(*) FROM dig.hourly_south;
```



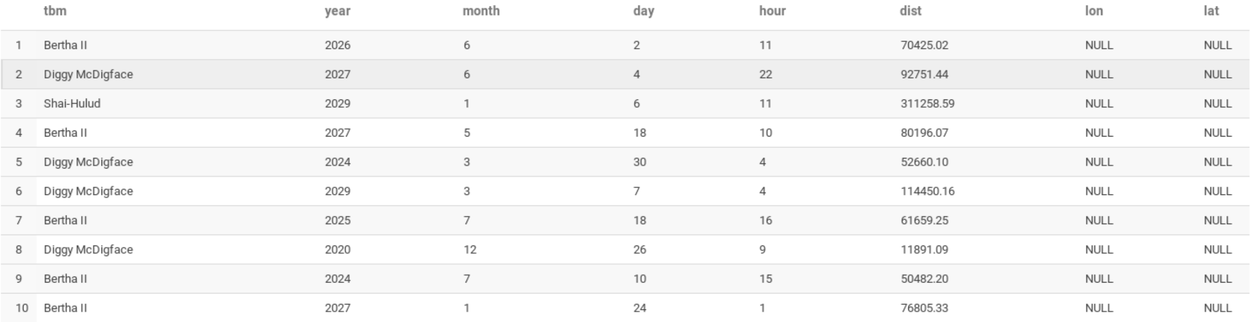
	count(*)
1	93163

21. Create a dig.tbm_sf_la view that combines all those three newly created external tables above, that way our view will always be up-to-date with any new data added to the source files in s3a://training-coursera2/tbm_sf_la/
Here is the SQL command in Impala that created the view:

```
CREATE VIEW dig.tbm_sf_la AS
  SELECT * FROM dig.hourly_central
  UNION
  SELECT * FROM dig.hourly_north
  UNION
  SELECT * FROM dig.hourly_south;
```

22. Check a few samples of the data in newly created view dig.tbm_sf_la using the following SQL command in Impala:

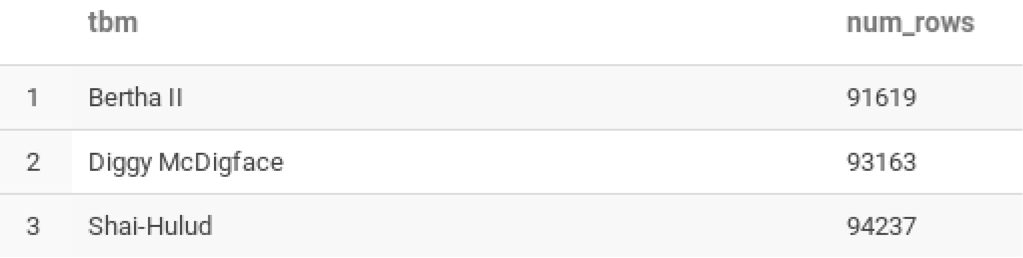
```
SELECT * FROM dig.tbm_sf_la LIMIT 10;
```



	tbm	year	month	day	hour	dist	lon	lat
1	Bertha II	2026	6	2	11	70425.02	NULL	NULL
2	Diggy McDigface	2027	6	4	22	92751.44	NULL	NULL
3	Shai-Hulud	2029	1	6	11	311258.59	NULL	NULL
4	Bertha II	2027	5	18	10	80196.07	NULL	NULL
5	Diggy McDigface	2024	3	30	4	52660.10	NULL	NULL
6	Diggy McDigface	2029	3	7	4	114450.16	NULL	NULL
7	Bertha II	2025	7	18	16	61659.25	NULL	NULL
8	Diggy McDigface	2020	12	26	9	11891.09	NULL	NULL
9	Bertha II	2024	7	10	15	50482.20	NULL	NULL
10	Bertha II	2027	1	24	1	76805.33	NULL	NULL

23. Check how many records for each tbm in view dig.tbm_sf_la, so we can verify that all the amount data in the view are matching with the amount of data of each external table using the following SQL command in Impala:

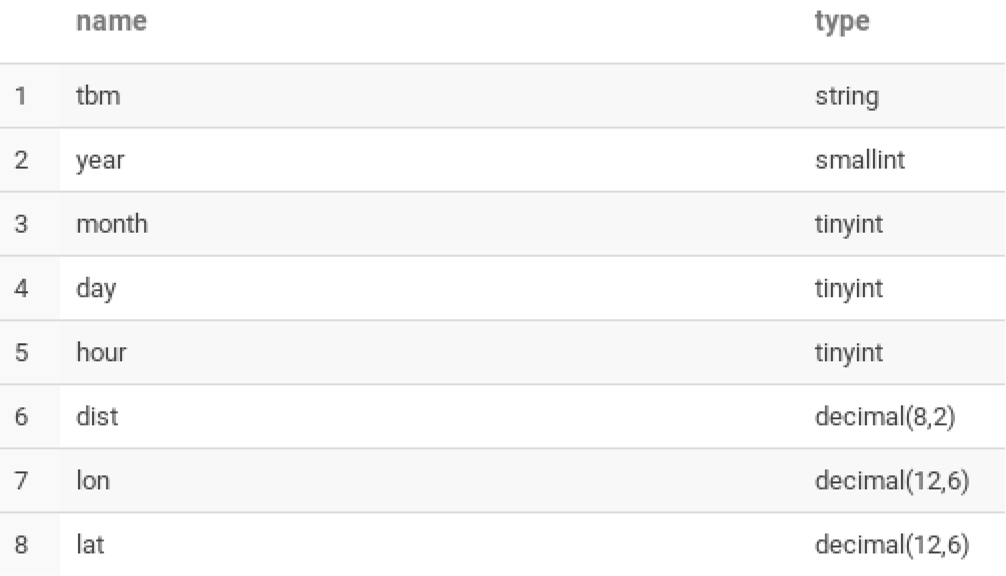
```
SELECT tbm, COUNT(*) AS num_rows
   FROM dig.tbm_sf_la
   GROUP BY tbm
   ORDER BY tbm;
```



	tbm	num_rows
1	Bertha II	91619
2	Diggy McDigface	93163
3	Shai-Hulud	94237

24. Check the metadata of view dig.tbm_sf_la using the following SQL command in Impala:

```
DESCRIBE dig.tbm_sf_la;
```



	name	type
1	tbm	string
2	year	smallint
3	month	tinyint
4	day	tinyint
5	hour	tinyint
6	dist	decimal(8,2)
7	lon	decimal(12,6)
8	lat	decimal(12,6)

(Describe all the steps you performed. Include the commands or SQL statements you ran.)

Result

After performing the steps described above, I ran the following queries and they produced the following result sets:

SELECT tbm, COUNT(*) AS num_rows FROM dig.tbm_sf_la GROUP BY tbm ORDER BY tbm;

Tbm	**num_rows**
Bertha II	91,619
Diggy McDigface	93,163
Shai-Hulud	94,237

DESCRIBE dig.tbm_sf_la;

Name	**Type**
tbm	string
year	smallint
month	tinyint
day	tinyint
hour	tinyint
dest	decimal(8,2)
Lon	decimal(12,6)
lat	decimal(12,6)

(Fill in the above tables.)

Notes

I could have made it more efficient as follow:

1. I could have created a partition table by tbm field for better query performance if I created tables, instead of external tables
2. I could have stored the data in parquet format to compress the data to save hdfs space and faster query performance instead of text files
3. I could have standardized the raw text files by pre-processing the raw files to the same format, but I may need to do the same thing over and over for any new data raw files. So the best way is to standardize the data format from the source by communicating it with the people who created the raw files

(In this section, describe ways that you could further optimize the table. You may also describe other methods you considered or attempted.)