

Data Engineer TEST

For this exercise I have used **Hortonworks Ambari Sandbox** and focused mainly on Data Engineering skills. The CSV files provided (weather.20160201.csv and weather.20160301.csv) were both saved locally first, on the Ambari machine.

Within the HIVE view I have uploaded the .csv files and created 2 separate tables, one for each .csv file provided. I have provided screenshots below.

-Upload .csv files from local to **HIVE**

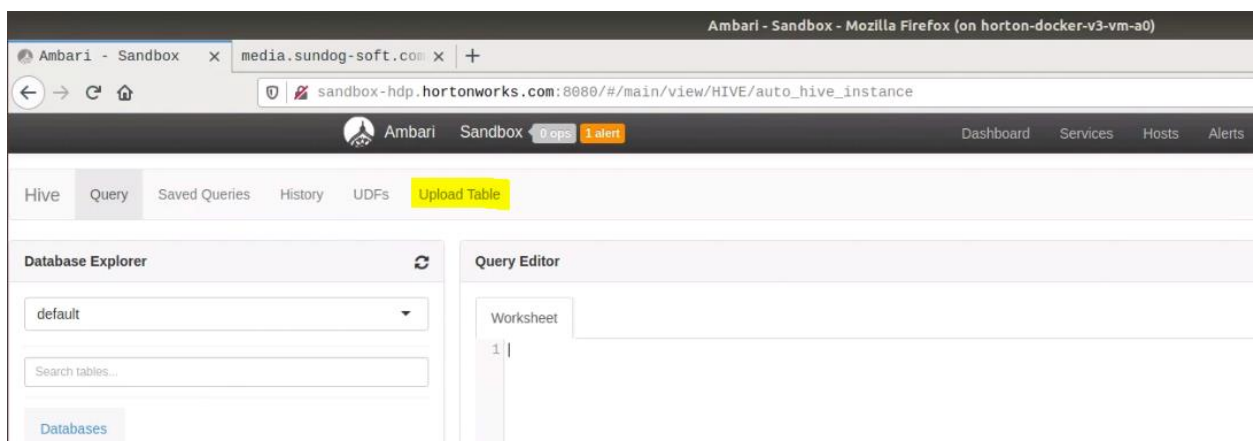


Fig.1

Upload from Local and **Select CSV** as **File Type** and click **Browse**. Once you have located the .CSV file open it and then in the next step make sure **Parquet** is the option selected in the **Stored as** field. Then you will have to re-name the columns and select the appropriate data type

Make sure **Is first row header?** is ticked.

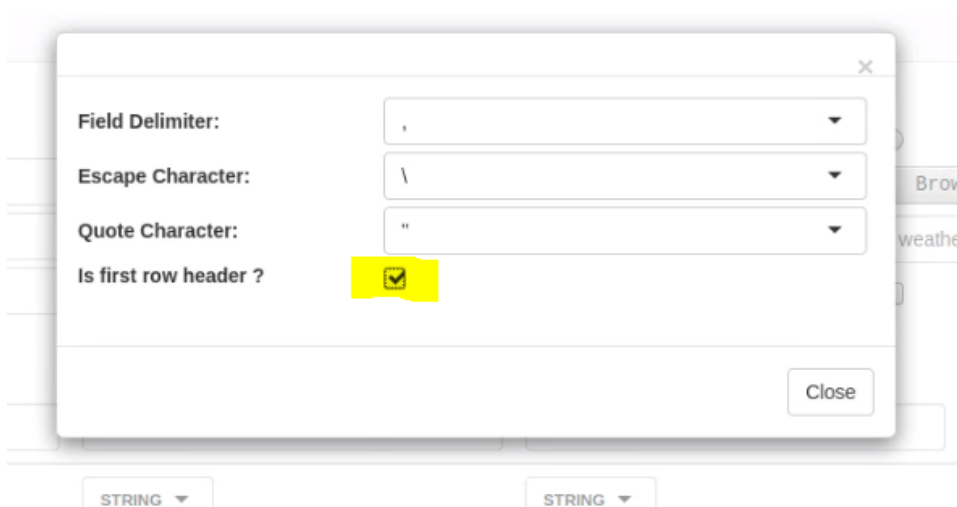


Fig. 2

The Data type fields are as follows for all tables:



forecastsitecode	STRING
observationtime	STRING
observationdate	STRING
winddirection	STRING
windspeed	STRING
windgust	STRING
visibility	STRING
screentemperature	DECIMAL(5,2)
pressure	STRING
significantweather...	STRING
sitename	STRING
latitude	STRING
longitude	STRING
region	STRING
country	STRING

Fig. 3

Then click on Upload Table and you will see the Upload Progress as below.

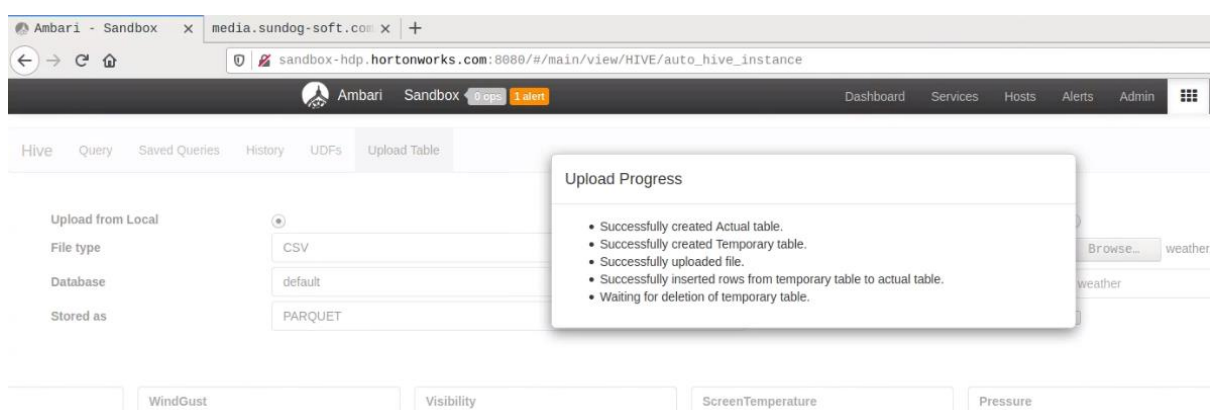


Fig. 4

I have uploaded both .csv files creating 2 tables in HIVE called “weather” and “weather1”

Then I have inserted all tables into a single table called weather2, keeping all properties making sure is still saved as PARQUET. (please see the next screenshot)

CREATE TABLE weather2 LIKE weather;

INSERT INTO weather2 SELECT * FROM weather;

INSERT INTO weather2 SELECT * FROM weather1;

The screenshot shows the Hive web interface with the 'TABLES' tab selected. The database is 'default'. Under 'TABLES | 4', the 'weather2' table is highlighted. The 'TABLE > WEATHER2' view shows the 'STORAGE INFORMATION' tab. The table's storage details are as follows:

PROPERTY	VALUE
Information	
SerDe Library	org.apache.hadoop.hive ql.io.parquet.serde.ParquetHiveSerDe
Input Format	org.apache.hadoop.hive ql.io.parquet.MapredParquetInputFormat
Output Format	org.apache.hadoop.hive ql.io.parquet.MapredParquetOutputFormat
Compressed	No
Number of Buckets	-1
Bucket Columns	
Sort Columns	
Parameters	{ "serialization.format": "1" }

Fig. 5

The screenshot shows the Hive web interface with the 'TABLES' tab selected. The database is 'default'. Under 'TABLES | 4', the 'weather2' table is highlighted. The 'TABLE > WEATHER2' view shows the 'DDL' tab. The table's DDL is as follows:

```
3  `observationtime` string,
4  `observationdate` string,
5  `winddirection` string,
6  `windspeed` string,
7  `windgust` string,
8  `visibility` string,
9  `screentemperature` decimal(5,2),
10 `pressure` string,
11 `significantweathercode` string,
12 `sitename` string,
13 `latitude` string,
14 `longitude` string, |
15 `region` string,
16 `country` string)
17 ROW FORMAT SERDE
18 'org.apache.hadoop.hive ql.io.parquet.serde.ParquetHiveSerDe'
19 STORED AS INPUTFORMAT
20 'org.apache.hadoop.hive ql.io.parquet.MapredParquetInputFormat'
21 OUTPUTFORMAT
22 'org.apache.hadoop.hive ql.io.parquet.MapredParquetOutputFormat'
23 LOCATION
24 'hdfs://sandbox-hdp.hortonworks.com:8020/apps/hive/warehouse/weather2'
25 TBLPROPERTIES (
26 'COLUMN_STATS_ACCURATE'='{\"BASIC_STATS\": \"true\"}',
```

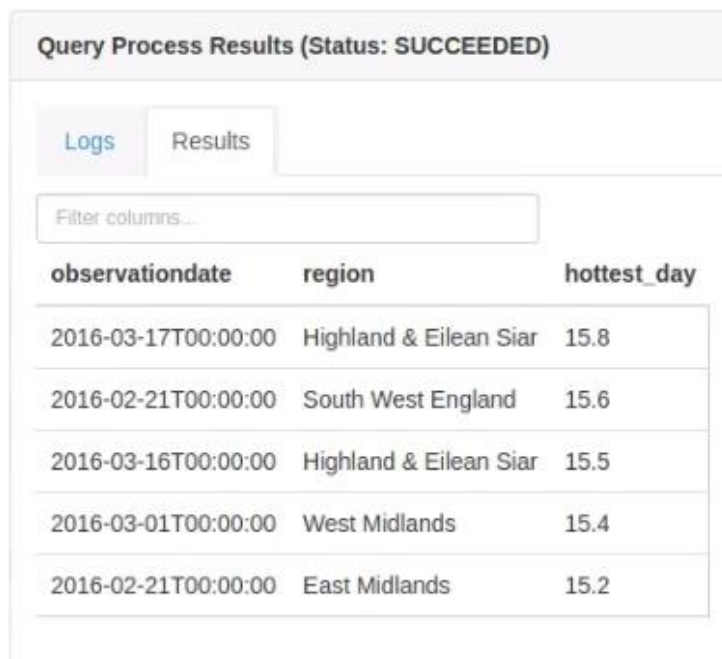
Fig. 6

DESCRIBE FORMATTED weather2; #checking all properties

SELECT (*) FROM weather2; #checking all entries (to see if both tables are present)

To answer all questions in the initial test I have run the following in HIVE Query:

```
SELECT observationdate, region  
      ,MAX(screentemperature) AS Hottest_Day  
FROM weather2  
GROUP BY observationdate, region  
ORDER BY Hottest_Day DESC  
LIMIT 5;
```



The screenshot shows the 'Query Process Results (Status: SUCCEEDED)' interface. It has two tabs: 'Logs' and 'Results'. Below the tabs is a 'Filter columns...' input field. The results are displayed in a table with three columns: 'observationdate', 'region', and 'hottest_day'. The table contains five rows of data, sorted by 'hottest_day' in descending order.

observationdate	region	hottest_day
2016-03-17T00:00:00	Highland & Eilean Siar	15.8
2016-02-21T00:00:00	South West England	15.6
2016-03-16T00:00:00	Highland & Eilean Siar	15.5
2016-03-01T00:00:00	West Midlands	15.4
2016-02-21T00:00:00	East Midlands	15.2

Fig.7

Looking at the previous queries and figures (mainly Fig. 7) all questions have been answered:

- Which date was the hottest day?
- What was the temperature on that day?
- In which region was the hottest day?