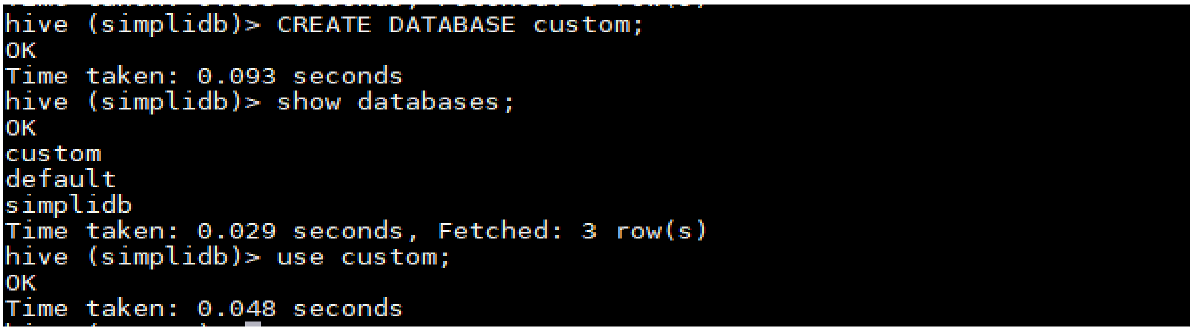
Assignment Submission- Session 8

Task 1

Create a database named 'custom'.
Create a table named temperature_data inside custom having below fields:
1. date (mm-dd-yyyy) format
2. zip code
3. temperature
The table will be loaded from comma-delimited file.
Load the dataset.txt (which is ',' delimited) in the table.

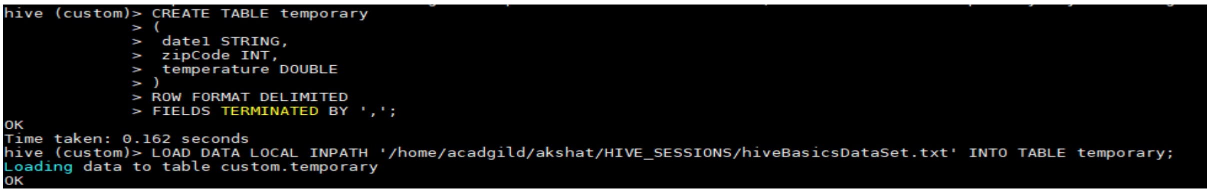
Solution:

Created database, as shown in screenshot.



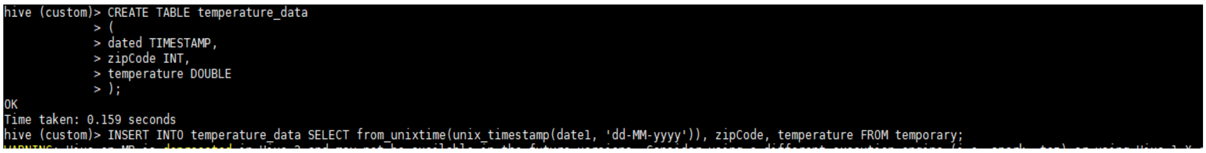
```
hive (simplidb)> CREATE DATABASE custom;
OK
Time taken: 0.093 seconds
hive (simplidb)> show databases;
OK
custom
default
simplidb
Time taken: 0.029 seconds, Fetched: 3 row(s)
hive (simplidb)> use custom;
OK
Time taken: 0.048 seconds
```

To store column 1 data into table as "timestamp" storing data into temporary table first.



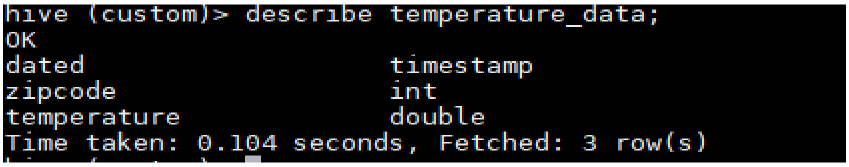
```
hive (custom)> CREATE TABLE temporary
             > (
             >  date1 STRING,
             >  zipCode INT,
             >  temperature DOUBLE
             > )
             > ROW FORMAT DELIMITED
             > FIELDS TERMINATED BY ',';
OK
Time taken: 0.162 seconds
hive (custom)> LOAD DATA LOCAL INPATH '/home/acadgild/akshat/HIVE_SESSIONS/hiveBasicsDataSet.txt' INTO TABLE temporary;
Loading data to table custom.temporary
OK
```

Creating main table and inserting data into it.



```
hive (custom)> CREATE TABLE temperature_data
             > (
             > dated TIMESTAMP,
             > zipCode INT,
             > temperature DOUBLE
             > );
OK
Time taken: 0.159 seconds
hive (custom)> INSERT INTO temperature_data SELECT from_unixtime(unix_timestamp(date1, 'dd-MM-yyyy')), zipCode, temperature FROM temporary;
```

Checking by describe table:



```
hive (custom)> describe temperature_data;
OK
dated                   timestamp
zipcode                 int
temperature             double
Time taken: 0.104 seconds, Fetched: 3 row(s)
```

Hive queries :

```
CREATE TABLE temporary
(
 date1 STRING,
 zipCode INT,
 temperature DOUBLE
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ',';
```

****

```
LOAD DATA LOCAL INPATH '/home/acadgild/akshat/HIVE_SESSIONS/hiveBasicsDataSet.txt' INTO
TABLE temporary;
```

****

```
CREATE TABLE temperature_data
(
dated TIMESTAMP,
zipCode INT,
temperature DOUBLE
);
```

****

```
INSERT INTO temperature_data SELECT from_unixtime(unix_timestamp(date1, 'dd-MM-yyyy')),
zipCode, temperature FROM temporary;
```

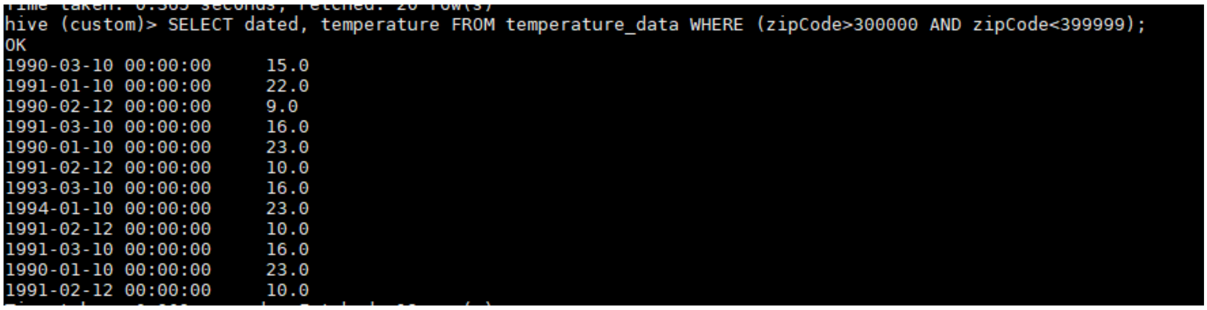
****

Task 2:

- Fetch date and temperature from temperature_data where zip code is greater than 300000 and less than 399999.

Hive query:

```
SELECT dated, temperature FROM temperature_data WHERE (zipCode>300000 AND
zipCode<399999);
```

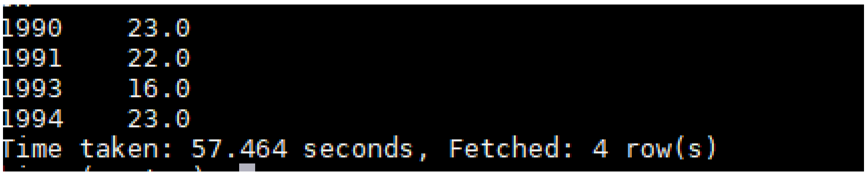


```
Time taken: 0.303 seconds, Fetched: 20 row(s)
hive (custom)> SELECT dated, temperature FROM temperature_data WHERE (zipCode>300000 AND zipCode<399999);
OK
1990-03-10 00:00:00     15.0
1991-01-10 00:00:00     22.0
1990-02-12 00:00:00     9.0
1991-03-10 00:00:00     16.0
1990-01-10 00:00:00     23.0
1991-02-12 00:00:00     10.0
1993-03-10 00:00:00     16.0
1994-01-10 00:00:00     23.0
1991-02-12 00:00:00     10.0
1991-03-10 00:00:00     16.0
1990-01-10 00:00:00     23.0
1991-02-12 00:00:00     10.0
```

● Calculate maximum temperature corresponding to every year from temperature_data table.

Hive query:

SELECT year(dated), max(temperature) FROM temperature_data GROUP BY year(dated);



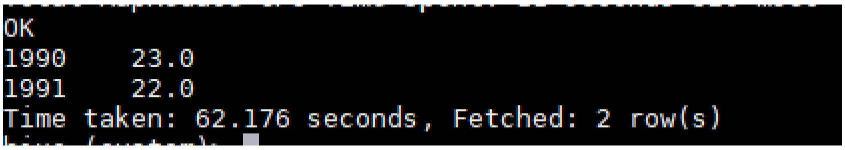
```
1990    23.0
1991    22.0
1993    16.0
1994    23.0
Time taken: 57.464 seconds, Fetched: 4 row(s)
```

● Calculate maximum temperature from temperature_data table corresponding to those years which have at least 2 entries in the table.

Done it with atleast 3 entries, as the dataset had 2 entries for each year.

Hive query:

SELECT year(dated), max(temperature) FROM temperature_data GROUP BY year(dated) HAVING count(year(dated)) >=3;

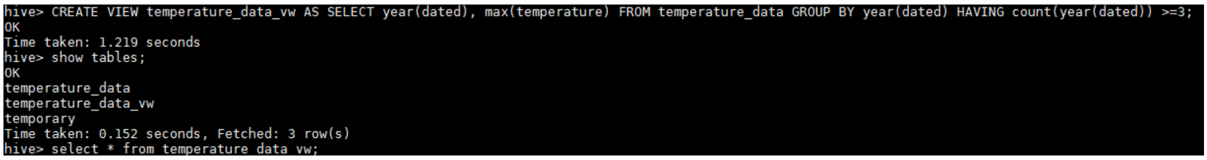


```
OK
1990    23.0
1991    22.0
Time taken: 62.176 seconds, Fetched: 2 row(s)
```

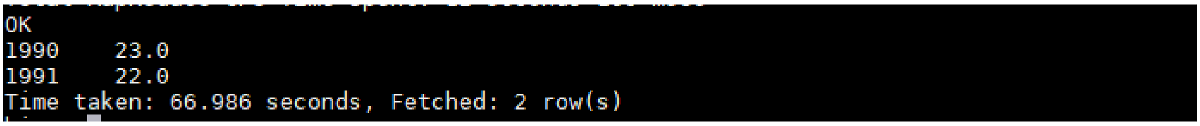
● Create a view on the top of last query, name it temperature_data_vw.

Hive query:

CREATE VIEW temperature_data_vw AS SELECT year(dated), max(temperature) FROM temperature_data GROUP BY year(dated) HAVING count(year(dated)) >=3;



```
hive> CREATE VIEW temperature_data_vw AS SELECT year(dated), max(temperature) FROM temperature_data GROUP BY year(dated) HAVING count(year(dated)) >=3;
OK
Time taken: 1.219 seconds
hive> show tables;
OK
temperature_data
temperature_data_vw
temporary
Time taken: 0.152 seconds, Fetched: 3 row(s)
hive> select * from temperature_data_vw;
```

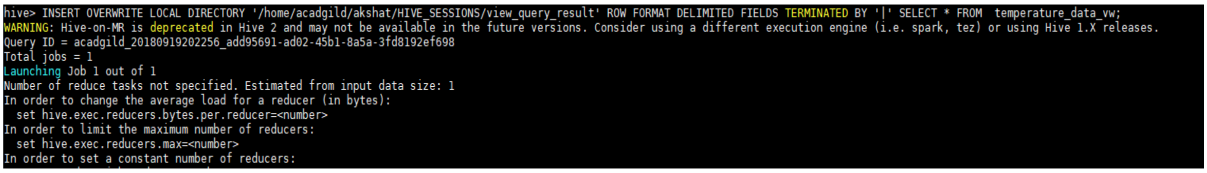


```
OK
1990    23.0
1991    22.0
Time taken: 66.986 seconds, Fetched: 2 row(s)
```

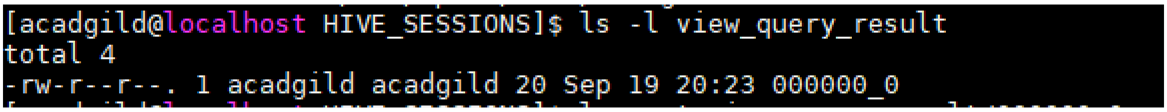
● Export contents from temperature_data_vw to a file in local file system, such that each file is '|' delimited.

Hive query:

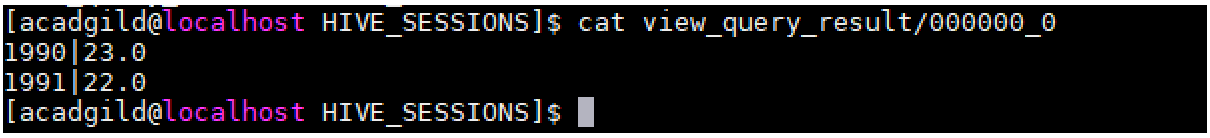
INSERT OVERWRITE LOCAL DIRECTORY ‘/home/acadgild/akshat/HIVE_SESSIONS/view_query_result’ ROW FORMAT DELIMITED FIELDS TERMINATED BY ‘|’ SELECT * FROM temperature_data_vw;



```
hive> INSERT OVERWRITE LOCAL DIRECTORY '/home/acadgild/akshat/HIVE_SESSIONS/view_query_result' ROW FORMAT DELIMITED FIELDS TERMINATED BY '|' SELECT * FROM  temperature_data_vw;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = acadgild_20180919202256_add95691-ad02-45b1-8a5a-3fd8192ef698
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
```



```
[acadgild@localhost HIVE_SESSIONS]$ ls -l view_query_result
total 4
-rw-r--r--. 1 acadgild acadgild 20 Sep 19 20:23 000000_0
```



```
[acadgild@localhost HIVE_SESSIONS]$ cat view_query_result/000000_0
1990|23.0
1991|22.0
[acadgild@localhost HIVE_SESSIONS]$ 
```