

Hive Exercise 2 - Solutions

Create a new database called exercise

```
create database exercise;

hive> show databases;
OK
default
emp
sales
Time taken: 1.204 seconds, Fetched: 3 row(s)
hive> create database exercise;
OK
Time taken: 2.48 seconds
hive> show databases;
OK
default
emp
exercise
sales
Time taken: 0.054 seconds, Fetched: 4 row(s)
hive>
```

In this newly created database, create an empty table named housing_price

First few lines of the csv

	status	bed	bath	acre_lot	city	state	zip_code	house_size	prev_sold_date	price
0	for_sale	3.0	2.0	0.12	Adjuntas	Puerto Rico	601.0	920.0	NaN	105000.0
1	for_sale	4.0	2.0	0.08	Adjuntas	Puerto Rico	601.0	1527.0	NaN	80000.0
2	for_sale	2.0	1.0	0.15	Juana Díaz	Puerto Rico	795.0	748.0	NaN	67000.0
3	for_sale	4.0	2.0	0.10	Ponce	Puerto Rico	731.0	1800.0	NaN	145000.0
4	for_sale	6.0	2.0	0.05	Mayaguez	Puerto Rico	680.0	NaN	NaN	65000.0
5	for_sale	4.0	3.0	0.46	San Sebastian	Puerto Rico	612.0	2520.0	NaN	179000.0
6	for_sale	3.0	1.0	0.20	Ciales	Puerto Rico	639.0	2040.0	NaN	50000.0
7	for_sale	3.0	2.0	0.08	Ponce	Puerto Rico	731.0	1050.0	NaN	71600.0
8	for_sale	2.0	1.0	0.09	Ponce	Puerto Rico	730.0	1092.0	NaN	100000.0
9	for_sale	5.0	3.0	7.46	Las Marias	Puerto Rico	670.0	5403.0	NaN	300000.0

```
CREATE TABLE IF NOT EXISTS housing_price (
  status STRING,
  bed FLOAT,
  bath FLOAT,
  acre_lot FLOAT,
  city STRING,
```

```
state STRING,  
zip_code FLOAT,  
house_size FLOAT,  
prev_sold_date STRING,  
price FLOAT  
)  
ROW FORMAT DELIMITED  
FIELDS TERMINATED BY ','  
STORED AS TEXTFILE  
TBLPROPERTIES ("skip.header.line.count"="1");
```

Load data from the local CSV file into the Hive Table

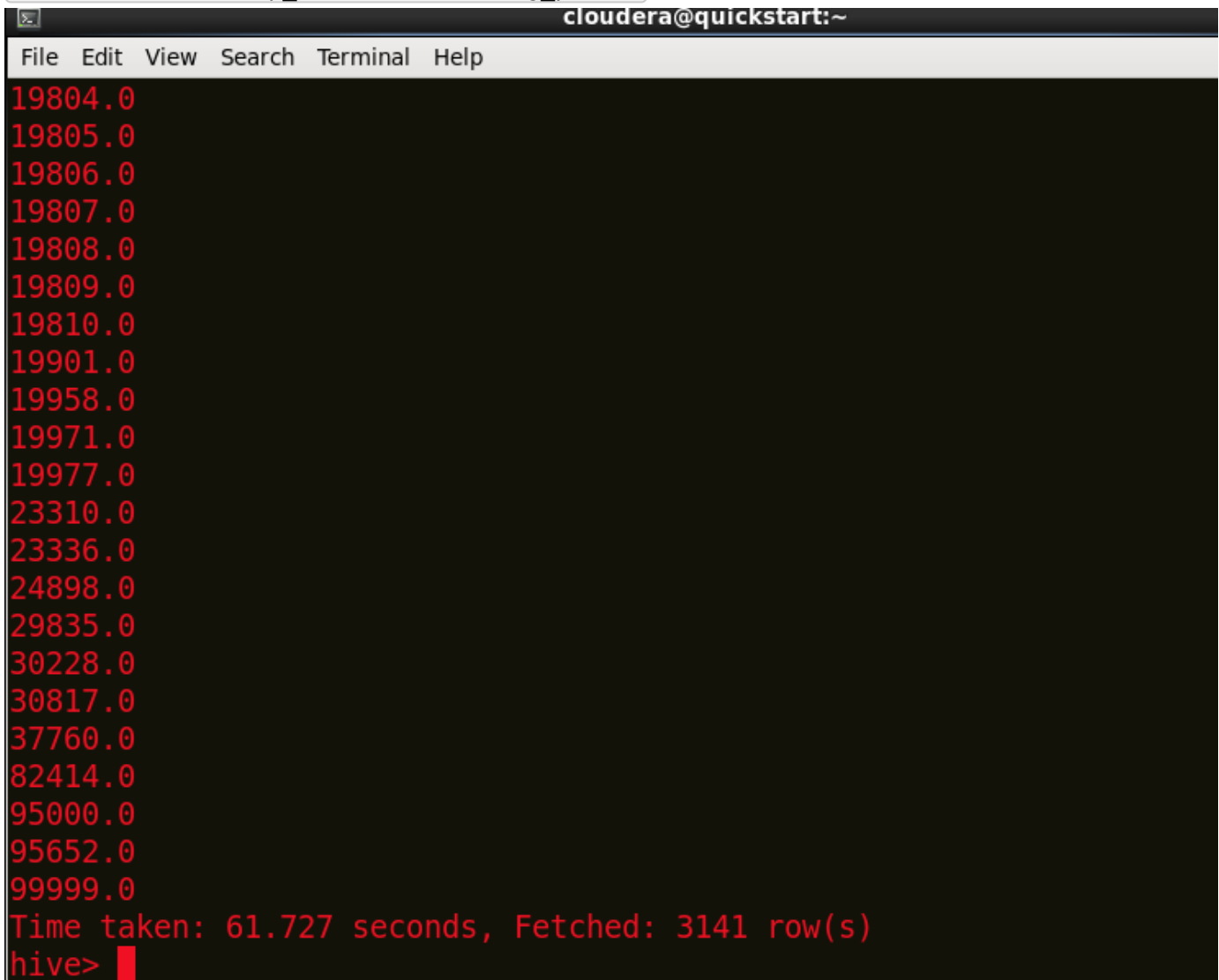
```
LOAD DATA LOCAL INPATH '/home/cloudera/Desktop/sf/realtor-data.csv' OVERWRITE INTO  
TABLE housing_price;
```

Display the first 5 rows of the table

```
SELECT * FROM housing_price LIMIT 5;
```

Get distinct zip codes from the dataset

```
SELECT DISTINCT zip_code FROM housing_price;
```



A terminal window titled "cloudera@quickstart:~" with a menu bar (File, Edit, View, Search, Terminal, Help). The output of the SQL query is displayed in red text on a black background. It lists 20 distinct zip codes followed by a summary line and a prompt.

```
19804.0  
19805.0  
19806.0  
19807.0  
19808.0  
19809.0  
19810.0  
19901.0  
19958.0  
19971.0  
19977.0  
23310.0  
23336.0  
24898.0  
29835.0  
30228.0  
30817.0  
37760.0  
82414.0  
95000.0  
95652.0  
99999.0  
Time taken: 61.727 seconds, Fetched: 3141 row(s)  
hive> █
```

List the total number of records in the table

```
SELECT COUNT(*) AS total_records FROM housing_price;
```

```

Starting Job = job_1695154310689_0002, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1695154310689_0002/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1695154310689_0002
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2023-09-19 13:43:15,289 Stage-1 map = 0%, reduce = 0%
2023-09-19 13:43:31,534 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 5.63 sec
2023-09-19 13:43:46,598 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 6.61 sec
MapReduce Total cumulative CPU time: 6 seconds 610 msec
Ended Job = job_1695154310689_0002
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 6.61 sec HDFS Read: 7 SUCCESS
Total MapReduce CPU Time Spent: 6 seconds 610 msec
OK
total_records
904966
Time taken: 50.43 seconds, Fetched: 1 row(s)

```

Get the Total number of beds

```
SELECT COUNT(bed) AS bed_count FROM housing_price;
```

```

Applications  Places  System
cloudera@quickstart:~
File Edit View Search Terminal Help
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1695154310689_0004, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1695154310689_0004/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1695154310689_0004
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2023-09-19 13:48:09,470 Stage-1 map = 0%, reduce = 0%
2023-09-19 13:48:27,806 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 5.63 sec
2023-09-19 13:48:43,879 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 8.3 sec
MapReduce Total cumulative CPU time: 8 seconds 300 msec
Ended Job = job_1695154310689_0004
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 8.3 sec HDFS Read: 62029280 HDFS Write: 7 SUCCESS
Total MapReduce CPU Time Spent: 8 seconds 300 msec
OK
bed_count
775126
Time taken: 54.662 seconds, Fetched: 1 row(s)
hive>

```

Get number of properties where state = "Massachusetts"

```
SELECT COUNT(*) AS massachusetts_count FROM housing_price WHERE state =  
'Massachusetts';
```

```
2023-09-19 13:50:13,713 Stage-1 map = 0%, reduce = 0%  
2023-09-19 13:50:30,721 Stage-1 map = 100%, reduce = 0%, Cumulative CPU  
C  
2023-09-19 13:50:47,955 Stage-1 map = 100%, reduce = 100%, Cumulative CPU  
sec  
MapReduce Total cumulative CPU time: 8 seconds 370 msec  
Ended Job = job_1695154310689_0005  
MapReduce Jobs Launched:  
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 8.37 sec HDFS  
3 HDFS Write: 7 SUCCESS  
Total MapReduce CPU Time Spent: 8 seconds 370 msec  
OK  
massachusetts_count  
175248  
Time taken: 53.571 seconds, Fetched: 1 row(s)  
hive>
```

Create a new table based on the previous table where city "San Juan"

```
CREATE TABLE IF NOT EXISTS san_juan_housing_price AS SELECT * FROM housing_price  
WHERE city = 'San Juan';
```

Write a query such that if there is an entry called Warwick print "My City" else "Not My City"

```
SELECT  
    CASE  
        WHEN COUNT(*) > 0 THEN 'My City'  
        ELSE 'Not My City'  
    END AS result  
FROM housing_price  
WHERE city = 'Warwick';
```