Advanced Hive Assignment 1 :

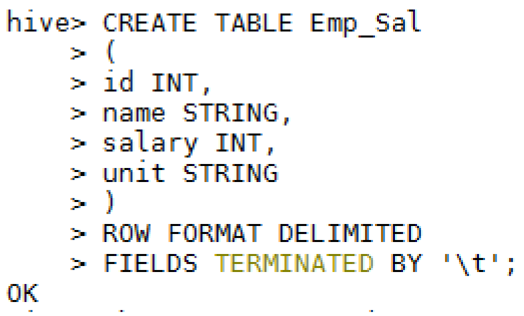
This Data set is about Employee Salary. This assignment uses Emp_Sal.txt file. First we have to create the table and load the data for Emp_Sal.

Creating the table and loading the data

1) First, we have to create a table named Emp_Sal, with the fields corresponding to the data in the Emp_Sal.txt data file. The command used is as below.

```
CREATE TABLE Emp_Sal
(
id INT,
name STRING,
salary INT,
unit STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t';
```

Screenshot of Mobaxterm for creating the table Emp_Sal:



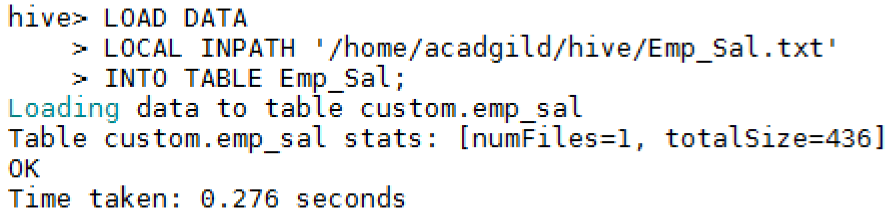
```
hive> CREATE TABLE Emp_Sal
    > (
    > id INT,
    > name STRING,
    > salary INT,
    > unit STRING
    > )
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY '\t';
OK
```

2) Next we have to load the contents of Emp_Sal.txt. I stored the file in the path ‘/home/acadgild/hive/Emp_Sal.txt'

   So we are loading the contents of the Emp_Sal.txt into the table Emp_Sal using the below commands.

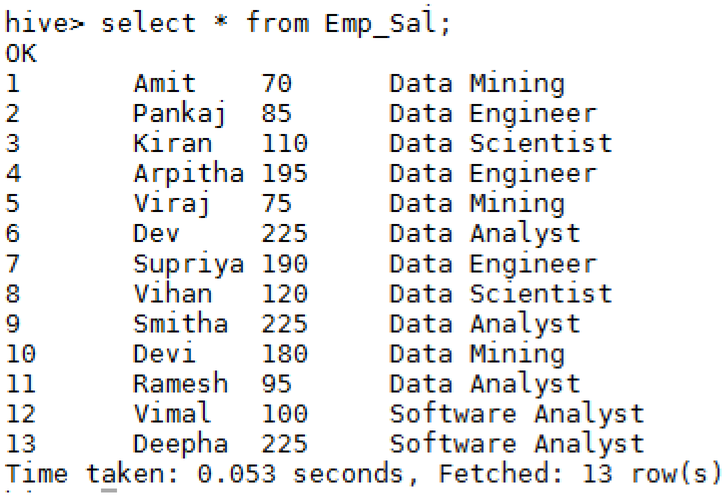
```
LOAD DATA
LOCAL INPATH '/home/acadgild/hive/Emp_Sal.txt'
INTO TABLE Emp_Sal;
```

Screenshot of Mobaxterm for loading Emp_Sal.txt into Emp_Sal:



```
hive> LOAD DATA
    > LOCAL INPATH '/home/acadgild/hive/Emp_Sal.txt'
    > INTO TABLE Emp_Sal;
Loading data to table custom.emp_sal
Table custom.emp_sal stats: [numFiles=1, totalSize=436]
OK
Time taken: 0.276 seconds
```

Screenshot of Mobaxterm for viewing the contents of Emp_Sal:



```
hive> select * from Emp_Sal;
OK
1       Amit    70      Data Mining
2       Pankaj  85      Data Engineer
3       Kiran   110     Data Scientist
4       Arpitha 195     Data Engineer
5       Viraj   75      Data Mining
6       Dev     225     Data Analyst
7       Supriya 190     Data Engineer
8       Vihan   120     Data Scientist
9       Smitha  225     Data Analyst
10      Devi    180     Data Mining
11      Ramesh  95      Data Analyst
12      Vimal   100     Software Analyst
13      Deepha  225     Software Analyst
Time taken: 0.053 seconds, Fetched: 13 row(s)
```

Using this table we are going to provide solution for all the queries in this assignment.

Q1) Get a list of employees who receive a salary less than 100, compared to their immediate employee with higher salary in the same unit.

The requirement is we need to find the list of employees who get 100 less than their immediate employee of higher salary within the same unit.

1) First we need to partition the employee by unit, we also need to arrange the employees within the unit in terms of ascending order of salary. This will give us the immediate employee within same unit
2) In Hadoop Lead and Lag are the Hive analytic functions used to compare different rows of a table by specifying an offset from the current row. We can use these functions to analyze change and variation in the data.
3) Using Lead we can find the Lead of salary for each employee, grouping by the unit and arranging the employee items in terms of salary in ascending order.

<u>Query To Find lead_salary</u>

SELECT id, name, salary, unit, LEAD(salary) OVER (PARTITION BY unit ORDER BY salary) AS lead_salary FROM Emp_Sal;

Output :

id , name, salary, lead_salary

```
11    Ramesh  95     Data Analyst    225
9     Smitha  225    Data Analyst    225
6     Dev     225    Data Analyst    NULL
2     Pankaj  85     Data Engineer   190
7     Supriya 190    Data Engineer   195
4     Arpitha 195    Data Engineer   NULL
1     Amit    70     Data Mining     75
5     Viraj   75     Data Mining     180
10    Devi    180    Data Mining     NULL
3     Kiran   110    Data Scientist  120
8     Vihan   120    Data Scientist  NULL
12    Vimal   100    Software Analyst       225
13    Deepha  225    Software Analyst       NULL
```

4) Now we need to find the list of employees who draw a salary less than 100 compared to their lead employee's salary. So we are writing an outerquery which takes LeadSalary value from the innerquery and filters and displays the employees with Leadsalary to their salary difference of over 100.

Query: To Find the list of employee who draw 100 less than their lead's salary

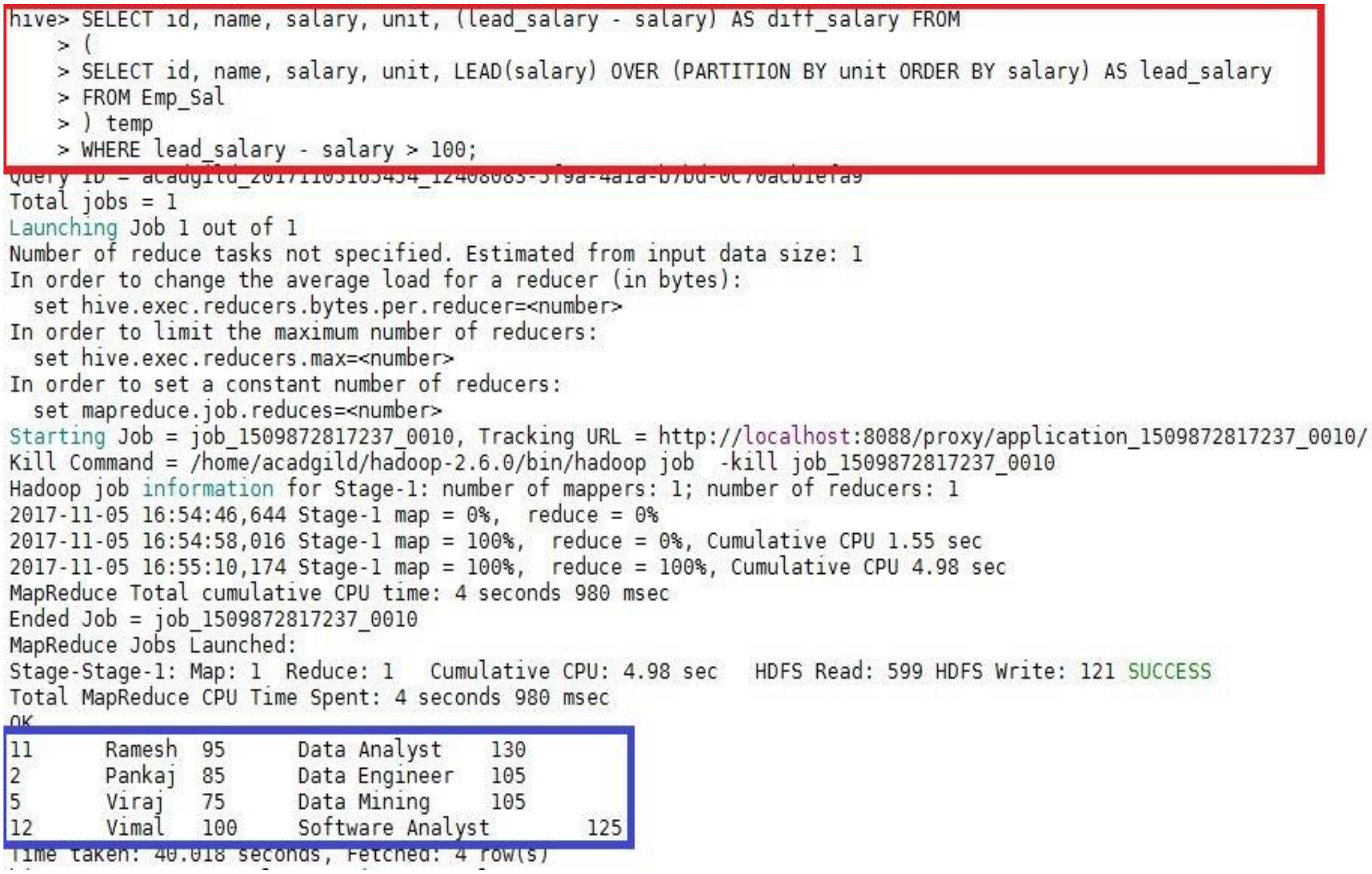
```
SELECT id, name, salary, unit, (lead_salary - salary) AS diff_salary FROM
(
SELECT id, name, salary, unit, LEAD(salary) OVER (PARTITION BY unit ORDER BY salary) AS
lead_salary
FROM Emp_Sal
) temp
WHERE lead_salary - salary > 100;
```

Output : :

id , name, salary, lead_salary

```
11    Ramesh  95     Data Analyst    130
2     Pankaj  85     Data Engineer   105
5     Viraj   75     Data Mining     105
12    Vimal   100    Software Analyst       125
```

Screenshot of Mobaxterm for the query and output



```
hive> SELECT id, name, salary, unit, (lead_salary - salary) AS diff_salary FROM
    > (
    > SELECT id, name, salary, unit, LEAD(salary) OVER (PARTITION BY unit ORDER BY salary) AS lead_salary
    > FROM Emp_Sal
    > ) temp
    > WHERE lead_salary - salary > 100;
Query ID = acadgild_20171105165454_12408083-5f9a-4a1a-b7bd-0c70acb1efa9
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:
  set mapreduce.job.reduces=<number>
Starting Job = job_1509872817237_0010, Tracking URL = http://localhost:8088/proxy/application_1509872817237_0010/
Kill Command = /home/acadgild/hadoop-2.6.0/bin/hadoop job  -kill job_1509872817237_0010
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2017-11-05 16:54:46,644 Stage-1 map = 0%,  reduce = 0%
2017-11-05 16:54:58,016 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.55 sec
2017-11-05 16:55:10,174 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 4.98 sec
MapReduce Total cumulative CPU time: 4 seconds 980 msec
Ended Job = job_1509872817237_0010
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 4.98 sec   HDFS Read: 599 HDFS Write: 121 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 980 msec
OK
11      Ramesh  95      Data Analyst    130
2       Pankaj  85      Data Engineer   105
5       Viraj   75      Data Mining     105
12      Vimal   100     Software Analyst        125
Time taken: 40.018 seconds, Fetched: 4 row(s)
```

Q2) List of all employees who draw higher salary than the average salary of that department..

The requirement is we need to find the list of employees who draw a higher salary than their unit(departments) average salary.

1) First we need to partition the employee by unit and finding the average salary drawn in that particular unit.
2) Using the below query we are querying and finding the avg. salary of the department and we are displaying the employee details along with the avg salary for each employee.

<u>Query: To find the avg salary of each unit, along with employee details</u>

```
SELECT id, name, salary, unit, avg(salary) OVER (PARTITION BY unit) AS avg_salary
FROM Emp_Sal;
```

<u>Output :</u>

<u>id , name, salary, avg_salary</u>

```
6      Dev     225     Data Analyst    181.66666666666666
11      Ramesh  95      Data Analyst    181.66666666666666
9      Smitha  225     Data Analyst    181.66666666666666
```

```
7      Supriya 190     Data Engineer   156.66666666666666
2      Pankaj  85      Data Engineer   156.66666666666666
4      Arpitha 195     Data Engineer   156.66666666666666
1      Amit    70      Data Mining     108.33333333333333
10      Devi    180     Data Mining     108.33333333333333
5      Viraj   75      Data Mining     108.33333333333333
8      Vihan   120     Data Scientist  115.0
3      Kiran   110     Data Scientist  115.0
12      Vimal   100     Software Analyst       162.5
13      Deepha  225     Software Analyst       162.5
```

3) Now we need to find the list of employees who draw a higher salary than their departments average salary. So we are writing an outerquery which takes, avg_salary from the innerquery and filters and displays the employees who draw a salary more than average salary.

Query: To Find the list of employee who draw a salary more than their units avg. salary

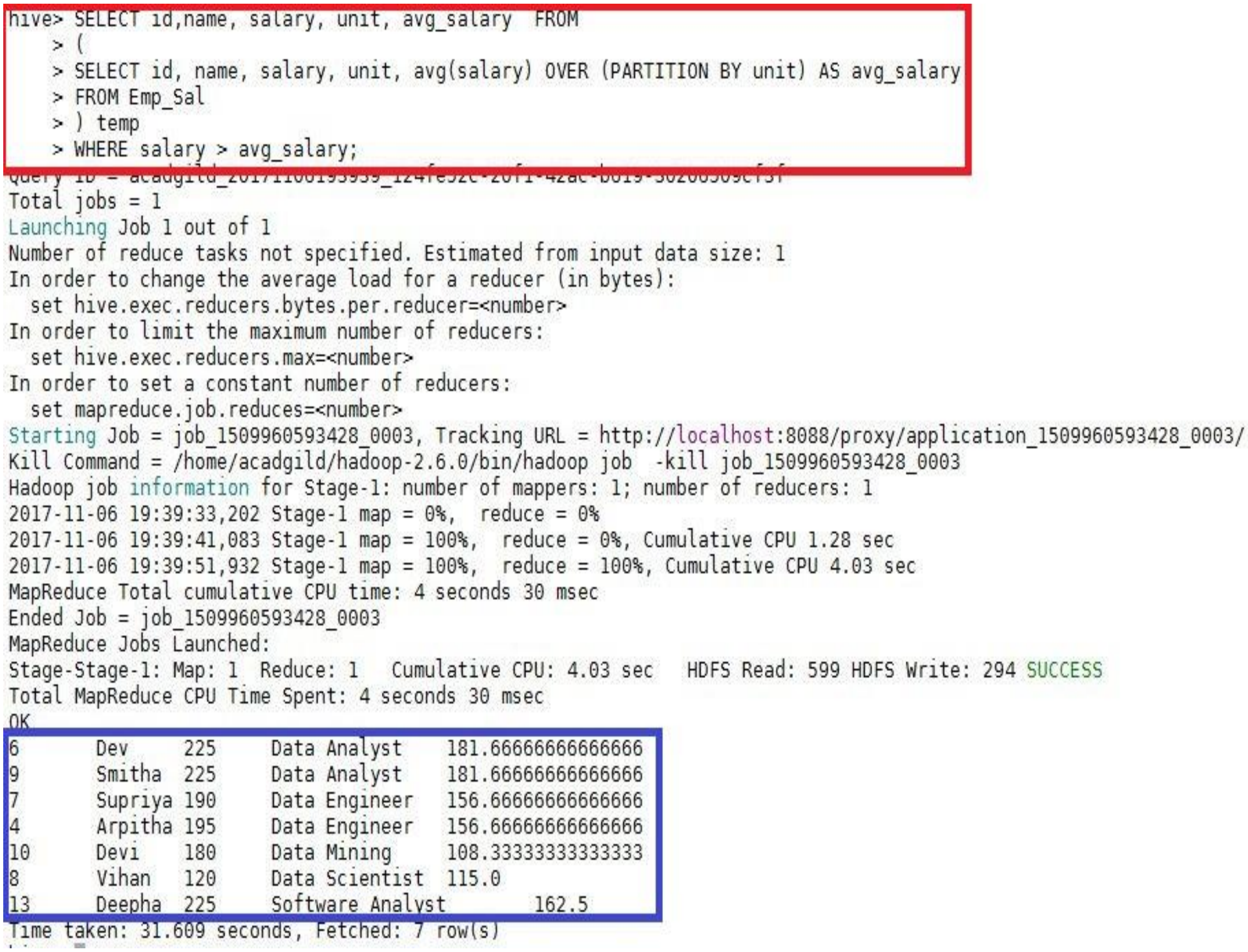
```
SELECT id,name, salary, unit, avg_salary  FROM
(
SELECT id, name, salary, unit, avg(salary) OVER (PARTITION BY unit) AS avg_salary
FROM Emp_Sal
) temp
WHERE salary > avg_salary;
```

Output : :

id , name, salary, avg_salary

```
Dev     225     Data Analyst    181.66666666666666
Smitha  225     Data Analyst    181.66666666666666
Supriya 190     Data Engineer   156.66666666666666
Arpitha 195     Data Engineer   156.66666666666666
Devi    180     Data Mining     108.33333333333333
Vihan   120     Data Scientist  115.0
Deepha  225     Software Analyst       162.5
```

<u>Screenshot of Mobaxterm for the query and output</u>



```
hive> SELECT id,name, salary, unit, avg_salary  FROM
    > (
    > SELECT id, name, salary, unit, avg(salary) OVER (PARTITION BY unit) AS avg_salary
    > FROM Emp_Sal
    > ) temp
    > WHERE salary > avg_salary;
Query ID = acadgild_20171106193939_124fe52c-20f1-42ac-b019-30206509cf3f
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:
  set mapreduce.job.reduces=<number>
Starting Job = job_1509960593428_0003, Tracking URL = http://localhost:8088/proxy/application_1509960593428_0003/
Kill Command = /home/acadgild/hadoop-2.6.0/bin/hadoop job  -kill job_1509960593428_0003
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2017-11-06 19:39:33,202 Stage-1 map = 0%,  reduce = 0%
2017-11-06 19:39:41,083 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.28 sec
2017-11-06 19:39:51,932 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 4.03 sec
MapReduce Total cumulative CPU time: 4 seconds 30 msec
Ended Job = job_1509960593428_0003
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 4.03 sec   HDFS Read: 599 HDFS Write: 294 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 30 msec
OK
6       Dev     225     Data Analyst    181.66666666666666
9       Smitha  225     Data Analyst    181.66666666666666
7       Supriya 190     Data Engineer   156.66666666666666
4       Arpitha 195     Data Engineer   156.66666666666666
10      Devi    180     Data Mining     108.33333333333333
8       Vihan   120     Data Scientist  115.0
13      Deepha  225     Software Analyst        162.5
Time taken: 31.609 seconds, Fetched: 7 row(s)
```