Project Report

Submitted On : 09/08/2018

Topic:

"IBM Dataset Analysis"

Submitted By:

Ankita

Sonali Priya

Sofia Jamil

Zaid Alam

ACKNOWLEDGEMENT

We express our sincere gratitude to our guide,

Mrs. Prity Dwivedi, NIVT Skills Training Instructor, for guiding and correcting various documents of ours with attention and care. She has taken pain to go through the project and make necessary corrections as and when needed.

We are also indebted to her, for her unconditional help and inspiration. Last but not the least, we would like to thank her for giving us an opportunity to take part in this project and helping us out in her own way whenever needed.

Thank You.

Mrs. Prity Dwivedi

NIVT Skills Training Instructor

INDEX

ABSTRACT

Big data is the collection and analysis of large set of data which holds many intelligence and raw information based on user data, Sensor data, Medical and Enterprise data. The Hadoop platform is used to Store, Manage, and Distribute Big data across several server nodes.

Hive is a data warehouse infrastructure tool to process structured data in Hadoop. It resides on top of Hadoop to summarize Big Data, and makes querying and analyzing easy.

Initially Hive was developed by Facebook, later the Apache Software Foundation took it up and developed it further as an open source under the name Apache Hive. It is used by different companies. For example, Amazon uses it in Amazon Elastic MapReduce.

Features of Hive

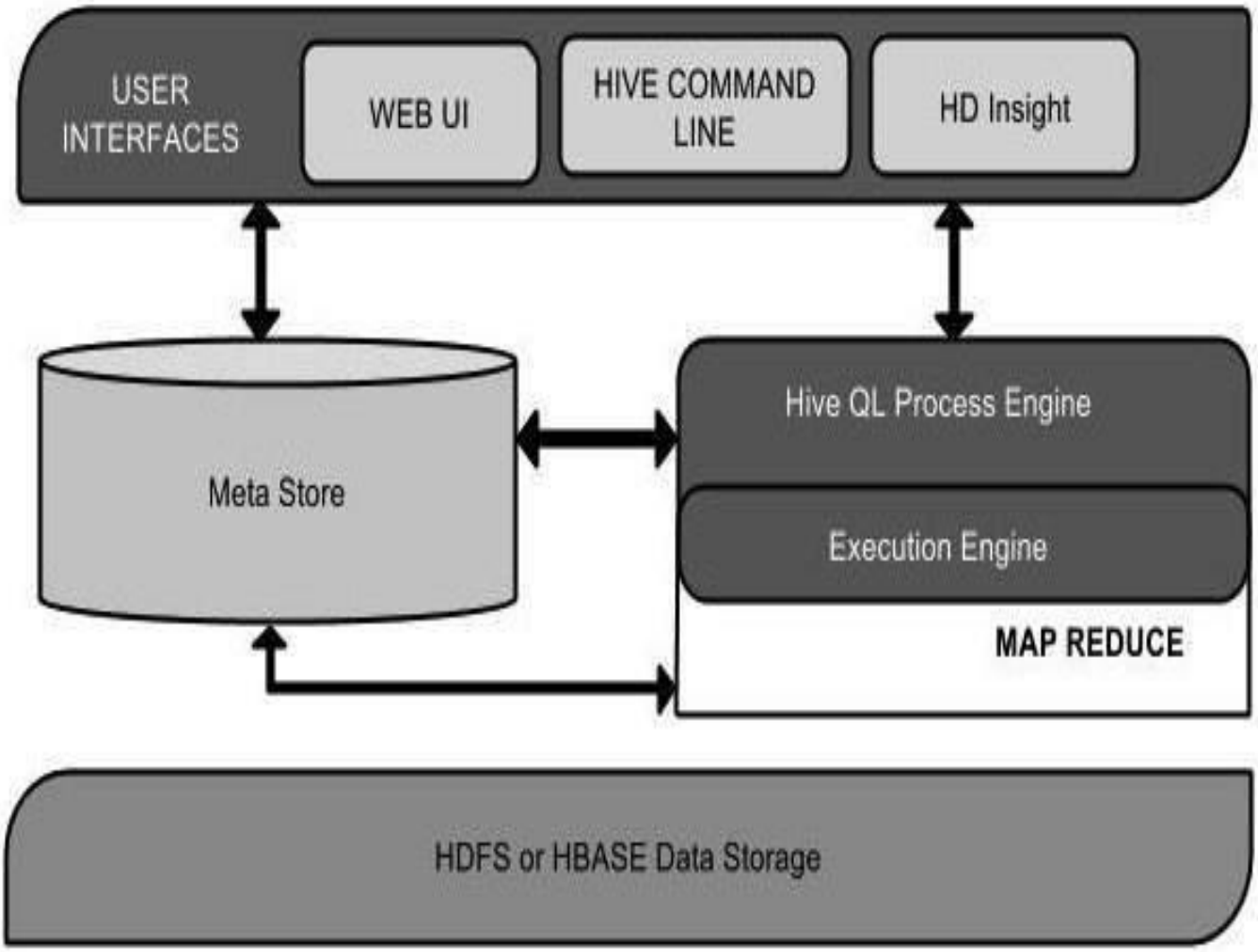
- It stores schema in a database and processed data into HDFS.
- It is designed for OLAP.(Online Analytical Processing)
- It provides SQL type language for querying called HiveQL or HQL.
- It is familiar, fast, scalable, and extensible.

Hive is not

- A relational database
- A design for OnLine Transaction Processing (OLTP)
- A language for real-time queries and row-level updates

Architecture of Hive :

The following component diagram depicts the architecture of Hive:

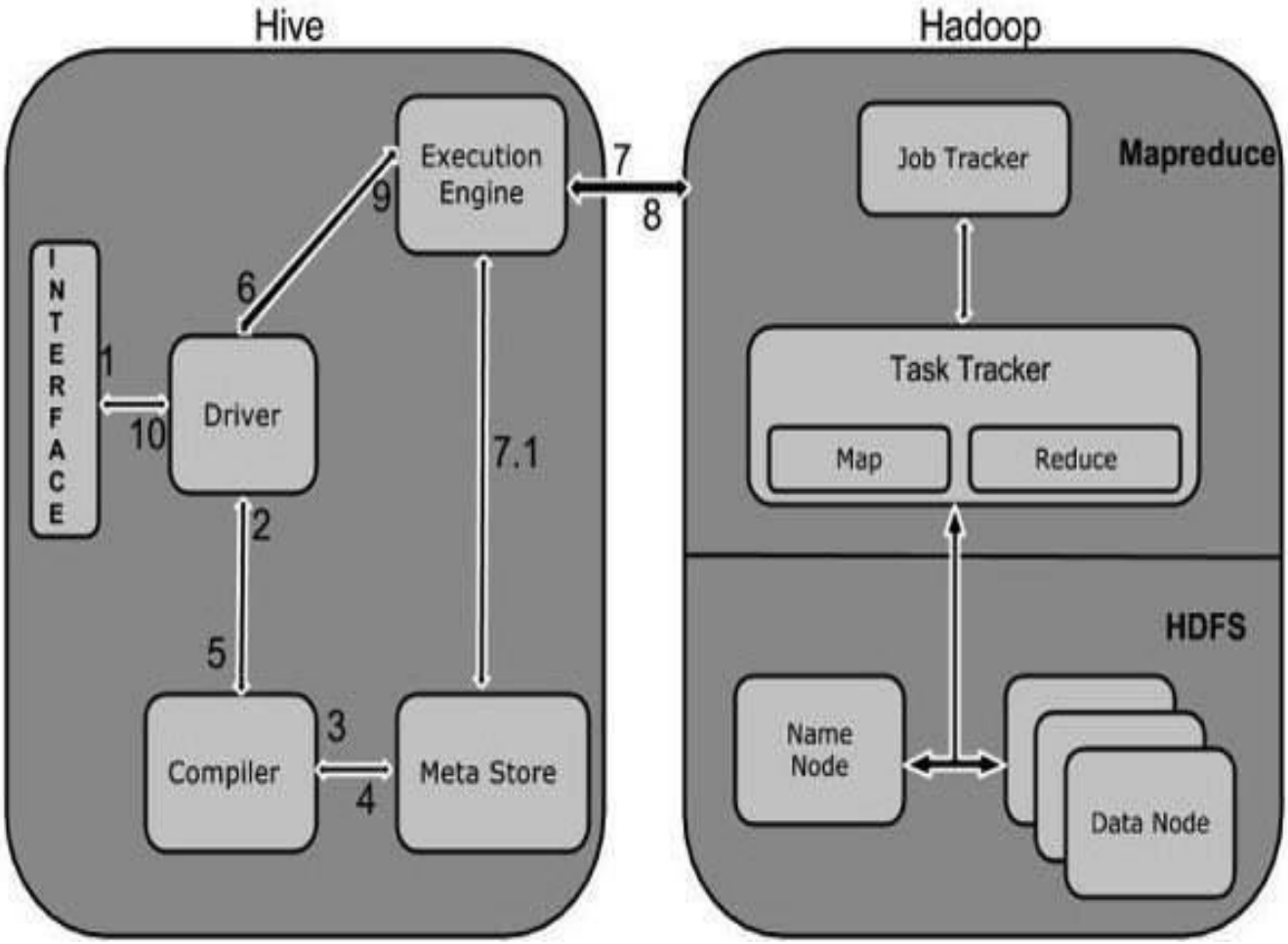


This component diagram contains different units. The following table describes each unit:

Unit Name	Operation
User Interface	Hive is a data warehouse infrastructure software that can create interaction between user and HDFS. The user interfaces that Hive supports are Hive Web UI, Hive command line, and Hive HD Insight (In Windows server).
Meta Store	Hive chooses respective database servers to store the schema or Metadata of tables, databases, columns in a table, their data types, and HDFS mapping.
HiveQL Process Engine	HiveQL is similar to SQL for querying on schema info on the Metastore. It is one of the replacements of traditional approach for MapReduce program. Instead of writing MapReduce program in Java, we can write a query for MapReduce job and process it.
Execution Engine	The conjunction part of HiveQL process Engine and MapReduce is Hive Execution Engine. Execution engine processes the query and generates results as same as MapReduce results. It uses the flavor of MapReduce.
HDFS or HBASE	Hadoop distributed file system or HBASE are the data storage techniques to store data into file system.

Working of Hive :

The following diagram depicts the workflow between Hive and Hadoop:



The following table defines how Hive interacts with Hadoop framework:

Step No.	Operation
1	**Execute Query** The Hive interface such as Command Line or Web UI sends query to Driver (any database driver such as JDBC, ODBC, etc.) to execute.
2	**Get Plan** The driver takes the help of query compiler that parses the query to check the syntax and query plan or the requirement of query.
3	**Get Metadata** The compiler sends metadata request to Metastore (any database).
4	**Send Metadata** Metastore sends metadata as a response to the compiler.
5	**Send Plan** The compiler checks the requirement and resends the plan to the driver. Up to here, the parsing and compiling of a query is complete.
6	**Execute Plan** The driver sends the execute plan to the execution engine.
7	**Execute Job** Internally, the process of execution job is a MapReduce job.

	The execution engine sends the job to JobTracker, which is in Name node and it assigns this job to TaskTracker, which is in Data node. Here, the query executes MapReduce job.
7.1	**Metadata Ops** Meanwhile in execution, the execution engine can execute metadata operations with Metastore.
8	**Fetch Result** The execution engine receives the results from Data nodes.
9	**Send Results** The execution engine sends those resultant values to the driver.
10	**Send Results** The driver sends the results to Hive Interfaces.

The IBM data that has been analyzed in this project consists of the following details as ',' delimited values.

1. Age
2. Attrition
3. Business Travel
4. Daily Rate
5. Department
6. Distance From Home
7. Education
8. Education Field
9. Employee Count
10. Employee Number
11. Hourly Rate
12. Job Role
13. Marital Status
14. Monthly Income
15. Monthly Rate.
16. Number Companies Worked
17. Over Time
18. Percent Salary Hike
19. Performance Rating
20. Standard Hours

21. Total Working Years

22. Training Times Last Year

23. Work Life Balance

24. Years At Company

25. Years In Current Role

26. Years Since Last Promotion

27. Years With Current Manager

The following details are analyzed out of the dataset:

1. The employee number and department of the employee who do Overtime

2. The last five employees based on last promotion received.

3. List of all employees who's income is more than average income of all employees of same department.

4. Employee details whose monthly income is above 5000.

INTRODUCTION

The IBM dataset is a csv file which contains details of 99 employees working in IBM. It contains details like employee's age, department, job status, monthly income, details about promotion and few other details. We analysed this data with the help of Hive Query Language (HQL).

- For finding the employee number and their department who do overtime, we have analysed the overtime column and the one with "Yes" were counted.

- For finding the last 5 employee number and their department who received the last promotion, we analysed the column "Years Since Last Promotion" and find the result by running the HQL in descending order.

- For finding the employees whose monthly income is greater than the average income of all the employees present in the same department, firstly we group by the employees on department and then calculated the average income and then we compared every employee monthly income with this reduced data.

- For finding the employee details like their employee number, department, role, gender and monthly income, whose monthly income is greater than the 5000, we run a query where we select all the employees whose monthly income is greater than 5000.

PROBLEM SET AND CODES

1. Find out the employee number and dept of employee who does overtime?

   - SELECT EmployeeNumber , Department FROM Ibmanalysis

     WHERE OverTime='Yes';

2. Find out last 5 employees based on last promotion received?

   - SELECT EmployeeNumber , Department, YearsSinceLastPromotion FROM Ibmanalysis ORDER BY YearsSinceLastPromotion DESC LIMIT 5;

3. Find out the list of employee whose income is more than the average income of all the employee's present in the same department?

   - SELECT i.EmployeeNumber, i.MonthlyIncome, i.Department

     FROM Ibmanalysis as i INNER JOIN (select

     Department,avg(MonthlyIncome) as sal from Ibmanalysis group by

     department) as t ON (i.Department=t.Department) WHERE

     (i.MonthlyIncome>t.sal);

4. Find out the all employees details whose monthly income above 5000.

   - SELECT EmployeeNumber , Department, Gender, JobRole,

     MonthlyIncome FROM Ibmanalysis WHERE MonthlyIncome>5000;

OUTPUT SCREENSHOT

- Database creation in Hive

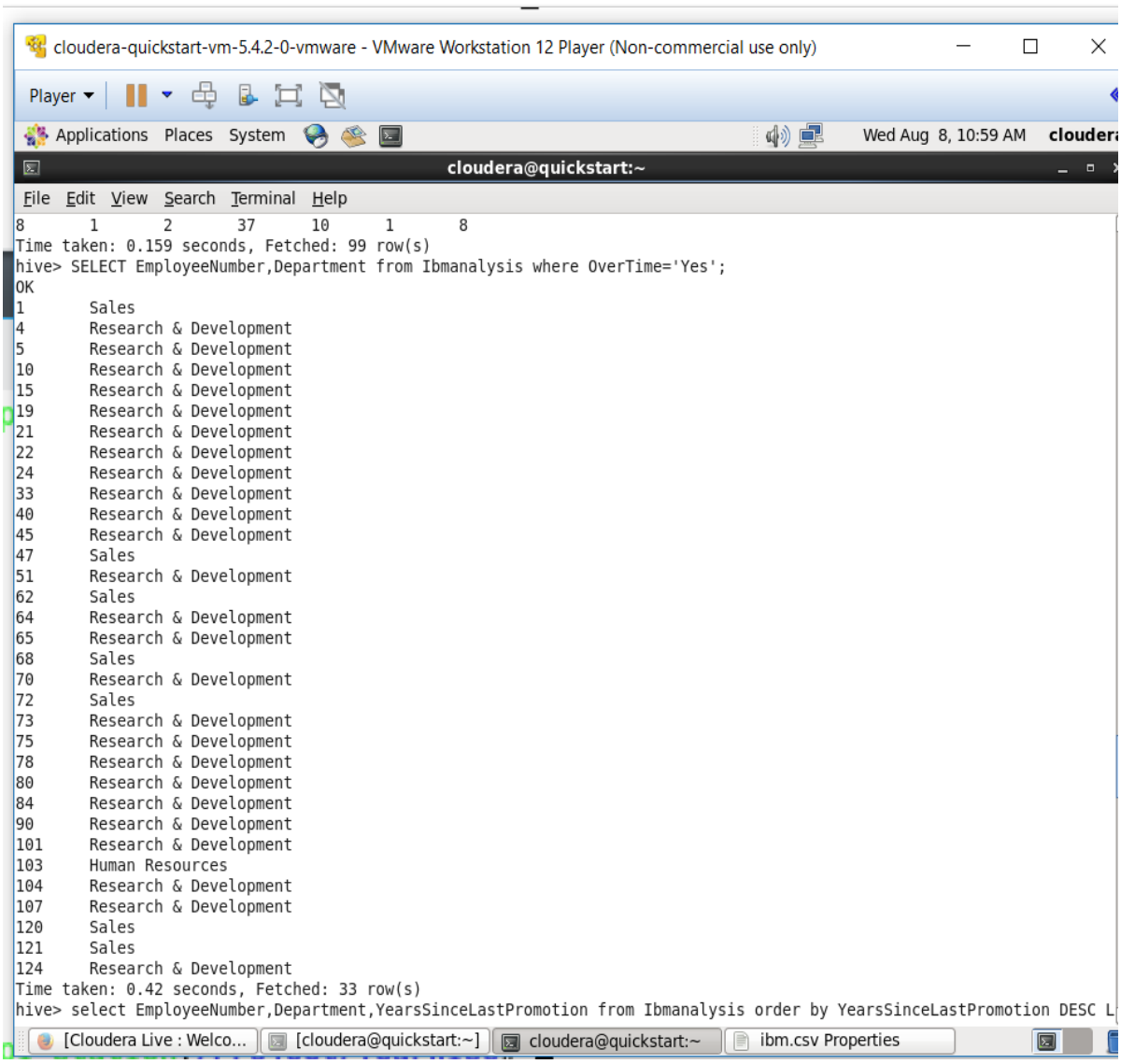
```
[cloudera@quickstart ~]$ hive

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.p
roperties
WARNING: Hive CLI is deprecated and migration to Beeline is recommended.
hive> create database project1;
OK
Time taken: 0.631 seconds
hive> show databases;
OK
default
project
project1
Time taken: 0.452 seconds, Fetched: 3 row(s)
hive>
    > use project1;
OK
Time taken: 0.042 seconds
hive> create table Ibmanalysis(
    > Age int,Attrition string,BusinessTravel string,DailyRate int,
    > Department string,DistanceFromHome int,Education int,
    > EducationField string,EmployeeCount int,EmployeeNumber int,
    > EnvironmentSatisfaction int,Gender string,HourlyRate int,
    > JobInvolvement int,JobLevel int,JobRole string,
    > JobSatisfaction int,MartialStatus string,MonthlyIncome number,
    > MonthlyRate number,NumCompaniesWorked int,
    > Over18 string,OverTime string,PercentSalaryHike int,
    > PerformanceRating int,RelationshipSatisfaction int,
    > StandardHours int,StockOptionLevel int,TotalWorkingYears int,
    > TrainingTimesLastYear int,WorkLifeBalance int,
    > YearsAtCompany int,YearsInCurrentRole int,
    > YearsSinceLastPromotion int,YearsWithCurrManager int)
    > row format delimited fields terminated by "," tblproperties ("skip.header.
```

- Loading IBM csv file from local storage to HDFS

```
132699
140858
152811
150154
```

```
OK
Time taken: 0.257 seconds
hive> load data local inpath '/home/cloudera/Desktop/ibm.csv' into table Ibmanalysis;
Loading data to table project1.ibmanalysis
Table project1.ibmanalysis stats: [numFiles=1, totalSize=15804]
OK
Time taken: 1.262 seconds
```

1. The employee number and department of the employee who do Overtime.



2. The last five employees based on last promotion received.

```
Time taken: 0.42 seconds, Fetched: 33 row(s)
hive> select EmployeeNumber,Department,YearsSinceLastPromotion from Ibmanalysis order by YearsSinceLastPromotion DESC
Query ID = cloudera_20180808103737_56971fa7-803f-4bcb-ae2e-28b4bee56370
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 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_1533723269683_0001, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1533723269683_
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1533723269683_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-08-08 10:37:46,282 Stage-1 map = 0%,  reduce = 0%
2018-08-08 10:37:58,891 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.64 sec
2018-08-08 10:38:13,509 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 3.45 sec
MapReduce Total cumulative CPU time: 3 seconds 450 msec
Ended Job = job_1533723269683_0001
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 3.45 sec   HDFS Read: 26179 HDFS Write: 147 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 450 msec
OK
58      Research & Development  15
80      Research & Development  13
83      Research & Development  12
119     Research & Development  11
117     Research & Development  10
Time taken: 57.154 seconds, Fetched: 5 row(s)
```

3. List of all employees who's income is more than average income of all employees of same department.

```
File  Edit  View  Search  Terminal  Help
hive> select i.EmployeeNumber,i.MonthlyIncome,i.Department from Ibmanalysis as i INNER JOIN (select Department,avg(MonthlyIncome) as sal from Ibmanalysis group by Department)t ON (i.Department=t.Department) WHERE (i.MonthlyIncome>t.sal);

Query ID = cloudera_20180808205757_315a57a7-7108-4273-bfec-6fecea6109ec
Total jobs = 2
Launching Job 1 out of 2
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_1533723269683_0004, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1533723269683_0004/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1533723269683_0004
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2018-08-08 20:57:22,375 Stage-1 map = 0%,  reduce = 0%
2018-08-08 20:57:50,950 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 12.54 sec
2018-08-08 20:58:04,128 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 14.95 sec
MapReduce Total cumulative CPU time: 14 seconds 950 msec
Ended Job = job_1533723269683_0004
Execution log at: /tmp/cloudera/cloudera_20180808205757_315a57a7-7108-4273-bfec-6fecea6109ec.log
2018-08-08 08:58:14     Starting to launch local task to process map join;      maximum memory = 1013645312
2018-08-08 08:58:17     Dump the side-table for tag: 0 with group count: 3 into file: file:/tmp/cloudera/b72956b9-4b1c-43d3-bb5f-6d43a91006e5/hive_2018-08-08_20-57-01_239_3673907340327666096-1/-local-10004/HashTable-Stage-4/MapJoin-mapfi
le10--.hashtable
2018-08-08 08:58:17     Uploaded 1 File to: file:/tmp/cloudera/b72956b9-4b1c-43d3-bb5f-6d43a91006e5/hive_2018-08-08_20-57-01_239_3673907340327666096-1/-local-10004/HashTable-Stage-4/MapJoin-mapfile10--.hashtable (1241 bytes)
2018-08-08 08:58:17     End of local task; Time Taken: 2.861 sec.
Execution completed successfully
MapredLocal task succeeded
Launching Job 2 out of 2
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1533723269683_0005, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1533723269683_0005/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1533723269683_0005
Hadoop job information for Stage-4: number of mappers: 1; number of reducers: 0
2018-08-08 20:58:30,540 Stage-4 map = 0%,  reduce = 0%
2018-08-08 20:58:41,597 Stage-4 map = 100%,  reduce = 0%, Cumulative CPU 2.29 sec
MapReduce Total cumulative CPU time: 2 seconds 290 msec
Ended Job = job_1533723269683_0005
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 14.95 sec   HDFS Read: 26688 HDFS Write: 216 SUCCESS
Stage-Stage-4: Map: 1   Cumulative CPU: 2.29 sec   HDFS Read: 8950 HDFS Write: 896 SUCCESS
Total MapReduce CPU Time Spent: 17 seconds 240 msec
OK
```

- OUTPUT

```
cloudera-quickstart-vm-5.4.2-0-vmware - VMware Workstation 12 Player (Non-commercial use only)
Player
Applications  Places  System
cloudera@quicksta
File  Edit  View  Search  Terminal  Help
MapredLocal task succeeded
Launching Job 2 out of 2
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1533723269683_0005, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1533723269683_0005/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1533723269683_0005
Hadoop job information for Stage-4: number of mappers: 1; number of reducers: 0
2018-08-08 20:58:30,540 Stage-4 map = 0%,  reduce = 0%
2018-08-08 20:58:41,597 Stage-4 map = 100%,  reduce = 0%, Cumulative CPU 2.29 sec
MapReduce Total cumulative CPU time: 2 seconds 290 msec
Ended Job = job_1533723269683_0005
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 14.95 sec   HDFS Read: 26688 HDFS Write: 216 SUCCESS
Stage-Stage-4: Map: 1   Cumulative CPU: 2.29 sec   HDFS Read: 8950 HDFS Write: 896 SUCCESS
Total MapReduce CPU Time Spent: 17 seconds 240 msec
OK
12      9526    Research & Development
20      9980    Research & Development
28      11994   Research & Development
32      19094   Research & Development
36      10248   Research & Development
40      6465    Research & Development
58      19545   Research & Development
70      9884    Research & Development
73      13458   Research & Development
76      5915    Research & Development
77      5993    Research & Development
78      6162    Research & Development
80      18740   Research & Development
83      10096   Research & Development
84      14756   Research & Development
85      6499    Research & Development
86      9724    Research & Development
96      6220    Research & Development
101     13245   Research & Development
102     13664   Research & Development
112     7260    Research & Development
119     13503   Research & Development
124     10673   Research & Development
126     13549   Research & Development
23      15427   Sales
35      6825    Sales
38      18947   Sales
56      8726    Sales
74      9069    Sales
81      7637    Sales
106     10239   Sales
118     9619    Sales
131     13872   Sales
Time taken: 101.538 seconds, Fetched: 33 row(s)
hive>
[Cloudera Live : Welco...]  [cloudera@quickstart:~]  cloudera@quickstart:~  ibm.csv Properties
```

4. Employee details whose monthly income is above 5000.

```
cloudera@qui
File  Edit  View  Search  Terminal  Help
hive> SELECT EmployeeNumber,Age,Department,Gender,JobRole,MonthlyIncome from Ibmanalysis where MonthlyIncome>5000;
OK
1       41      Sales   Female  Sales Executive 5993
2       49      Research & Development  Male    Research Scientist      5130
12      38      Research & Development  Male    Manufacturing Director  9526
13      36      Research & Development  Male    Healthcare Representative       5237
20      29      Research & Development  Female  Manufacturing Director  9980
23      53      Sales   Female  Manager 15427
28      34      Research & Development  Female  Research Director       11994
32      53      Research & Development  Female  Manager 19094
35      42      Sales   Male    Sales Executive 6825
36      44      Research & Development  Female  Healthcare Representative       10248
38      46      Sales   Female  Manager 18947
40      44      Research & Development  Male    Healthcare Representative       6465
52      33      Sales   Female  Sales Executive 5376
56      27      Sales   Male    Sales Executive 8726
58      41      Research & Development  Female  Research Director       19545
62      46      Sales   Male    Sales Executive 5772
64      48      Research & Development  Male    Laboratory Technician   5381
68      44      Sales   Female  Sales Executive 5454
70      35      Research & Development  Male    Healthcare Representative       9884
73      33      Research & Development  Female  Research Director       13458
74      35      Sales   Male    Sales Executive 9069
76      31      Research & Development  Male    Laboratory Technician   5915
77      37      Research & Development  Male    Manufacturing Director  5993
78      32      Research & Development  Male    Manufacturing Director  6162
80      50      Research & Development  Female  Research Director       18740
81      59      Sales   Female  Sales Executive 7637
83      36      Research & Development  Female  Healthcare Representative       10096
84      55      Research & Development  Female  Manager 14756
85      36      Research & Development  Male    Manufacturing Director  6499
86      45      Research & Development  Male    Research Scientist      9724
91      59      Sales   Female  Sales Executive 5473
96      32      Research & Development  Male    Research Scientist      6220
101     45      Research & Development  Male    Research Director       13245
102     37      Research & Development  Male    Research Director       13664
103     46      Human Resources Male    Human Resources 5021
104     30      Research & Development  Male    Laboratory Technician   5126
106     55      Sales   Male    Sales Executive 10239
[Cloudera Live : Welco...  [cloudera@quickstart:~]  cloudera@quickstart:~  ibm.csv Properties
```

ANALYSIS AND CONCLUSION

- Out of given number of employees only 33 does overtime and most of the employees are from Research And Development Department.

- The last five employee who got promotion were from Research And Development Department.

- 24 employees from Research and Development Department whereas 9 employees from Sales Department get more than average monthly income of other employee of their respective department.

- 38 employees have salary more than 5000.

From the results of the Dataset we can conclude that the employees in Research And Development are more hardworking and that's why they get more paid and promotion rate is also better than other department employees.