

### Assignment - Data Transform Activity (20%)

- 1) a) Create a new table for your analysis called “employee\_sales”.  
b) Load the table “employee” into this table.  
c) Select these columns: Attrition, Department, JobSatisfaction & MonthlyIncome.

The screenshot shows the Ambari Hive View 2.0 interface. On the left is a sidebar with services like HDFS, YARN, MapReduce2, Tez, Hive, Oozie, ZooKeeper, Ambari Metrics, and WebHCat. The main area has tabs for QUERY, JOBS, TABLES, SAVED QUERIES, UDFs, and SETTINGS. A green banner at the top says "Query has been submitted. (details)". The QUERY tab is active, showing a worksheet titled "Worksheet2". The database is set to "default". The query editor contains the following SQL:

```
1 CREATE TABLE employee_sales
2 (
3     Attrition string,
4     Department string,
5     JobSatisfaction int,
6     MonthlyIncome int
7 );
8
```

Below the editor are buttons for Execute, Save As, Insert UDF, and Visual Explain. To the right is a sidebar for "default" showing tables: employee (selected), hivesampletable, and employee\_sales (highlighted). At the bottom are navigation icons.

This screenshot shows the same Ambari Hive View 2.0 interface. The QUERY tab is active, showing a worksheet titled "Worksheet2". The database is set to "default". The query editor contains the following SQL:

```
1 INSERT OVERWRITE TABLE employee_sales
2 SELECT Attrition, Department, JobSatisfaction, MonthlyIncome
3 FROM employee;
4
```

Below the editor are buttons for Execute, Save As, Insert UDF, and Visual Explain. To the right is a sidebar for "default" showing tables: employee, employee\_sales (selected), and hivesampletable. At the bottom are navigation icons.

- 2) Round the data found in the “MonthlyIncome” column to the nearest \$1000.

The screenshot shows two panels of the Ambari Hive View 2.0 interface. The top panel displays a query window with the following code:

```
1 INSERT OVERWRITE TABLE employee_sales
2 SELECT Attrition, Department, JobSatisfaction, ROUND(MonthlyIncome,-3)
3 FROM employee_sales;
4
```

A green banner at the top right indicates "Query has been submitted. (details)". The bottom panel shows the results of the executed query:

```
1 SELECT * FROM employee_sales LIMIT 10;
```

The results table contains the following data:

	employee_sales.attrition	employee_sales.department	employee_sales.jobssatisfaction	employee_sales.monthlyincome
1	Yes	Sales	4	6000
2	No	Research & Development	2	5000
3	Yes	Research & Development	3	2000
4	No	Research & Development	3	3000
5	No	Research & Development	2	3000
6	No	Research & Development	4	3000
7	No	Research & Development	1	3000
8	No	Research & Development	3	3000
9	No	Research & Development	3	10000
10	No	Research & Development	3	5000

3) Filter the data to only look at those items in the “Sales Department”.

The screenshot shows two stacked windows from the Ambari Hive View 2.0 interface.

**Top Window:** A "Worksheet2" tab is active. The query editor contains the following SQL:

```
1 INSERT OVERWRITE TABLE employee_sales
2 SELECT *
3 FROM employee_sales
4 WHERE Department LIKE "%Sales%";
```

A green status bar at the top right says "Query has been submitted. (details)". The sidebar on the left lists services like HDFS, YARN, MapReduce2, Tez, Hive, Oozie, ZooKeeper, and Ambari Metrics. The right sidebar shows the "default" database with three tables: employee, employee\_sales (which is highlighted), and hivesamplabletable.

**Bottom Window:** A "RESULTS" tab is active, displaying the output of the previous query:

employee_sales.attrition	employee_sales.department	employee_sales.jobsatisfaction	employee_sales.monthlyincome
Yes	Sales	4	6000
No	Sales	4	15000
Yes	Sales	1	3000
No	Sales	2	7000
No	Sales	1	19000

The sidebar on the left includes Cluster Admin, Stack and Versions, Service Accounts, and Service Auto Start.

- 4) Order the data by “JobSatisfaction” from highest to lowest.

The screenshot shows two views of the Ambari Hive View 2.0 interface. The top view displays a query window with the following code:

```
1 INSERT OVERWRITE TABLE employee_sales
2 SELECT *
3 FROM employee_sales
4 ORDER BY JobSatisfaction DESC;
```

A green status bar at the top right indicates "Query has been submitted. (details)". The bottom view shows the results of the executed query:

```
1 SELECT * FROM employee_sales LIMIT 10;
```

The results table displays the following data:

employee_sales.attrition	employee_sales.department	employee_sales.jobsatisfaction	employee_sales.monthlyincome
No	Sales	4	5000
No	Sales	4	9000
No	Sales	4	18000
No	Sales	4	10000
No	Sales	4	7000
No	Sales	4	8000
Yes	Sales	4	1000
No	Sales	4	5000