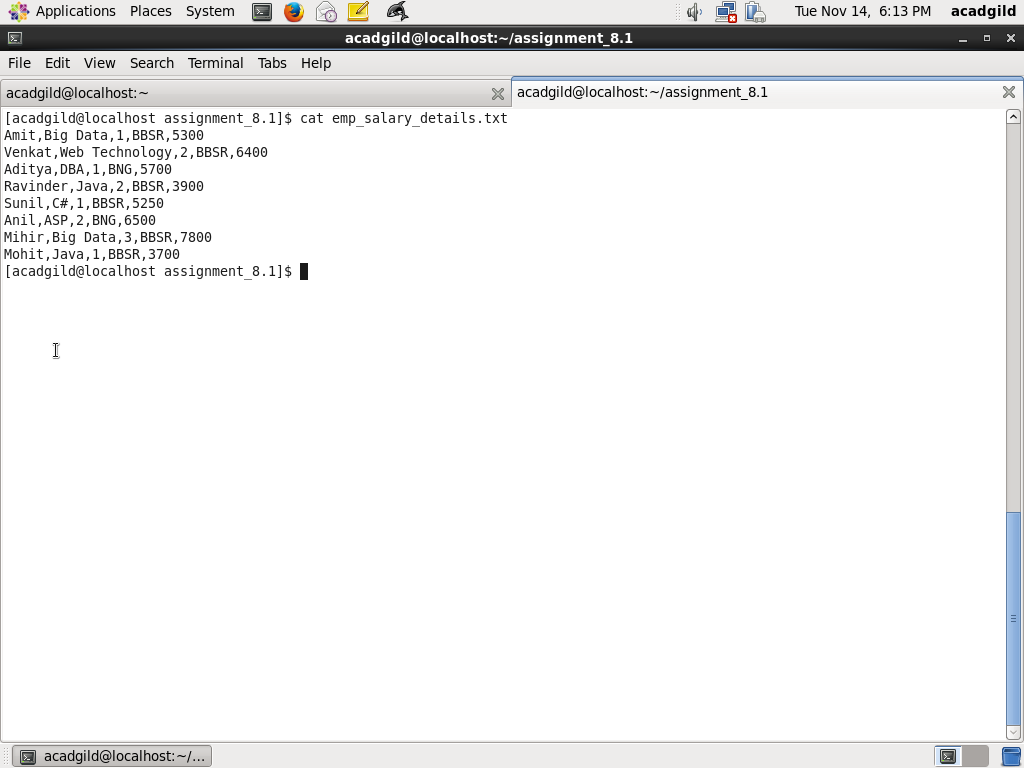
Assignment 8.1 ( Advanced Hive)

Step1: Create a dataset emp\_salary\_details.txt which has employee and his salary

The screenshot is as below:



Step2:

|  |
| --- |
| Use the following HIVE SQL command to create Hive table employee |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| CREATE TABLE IF NOT EXISTS employee |
|  |

|  |
| --- |
| ( |
|  |

|  |
| --- |
| name string, |
|  |

|  |
| --- |
| skill string, |
|  |

|  |
| --- |
| department\_id int, |
|  |

|  |
| --- |
| location string, |
|  |

|  |
| --- |
| salary float |
|  |

|  |
| --- |
| ) |
|  |

|  |
| --- |
| ROW FORMAT DELIMITED FIELDS TERMINATED BY ','; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
| Step3: |
|  |

|  |
| --- |
| Use the following Hive SQL Command to load employee salary details form emp\_salary\_details INOTO employee |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| LOAD DATA LOCAL INPATH '/home/acadgild/assignment\_8.1/emp\_salary\_details.txt' |
|  |

|  |
| --- |
| OVERWRITE INTO TABLE employee; |
| Screenshots for Step1 and Step2 is as below: |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| Hive SQL Query for Task1: Find all the employees whose salary is more than |
|  |

|  |
| --- |
| the department average salary  Use the following Hive SQL query where employee table with alias a is joined with  Alias b which has avg salary calculated for each department using avg function. While joining criterias used are same department\_id and alias a has salary greater than average salary |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| SELECT a.name, a.department\_id, a.salary |
|  |

|  |
| --- |
| FROM employee a JOIN |
|  |

|  |
| --- |
| (SELECT department\_id,avg(salary) avg\_sal |
|  |

|  |
| --- |
| FROM employee GROUP BY department\_id) b |
|  |

|  |
| --- |
| WHERE a.department\_id=b.department\_id and a.salary > b.avg\_sal; |
| The screenshot of query is as below: |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| The output after executing query is as below:    Task2: Hive SQL Query for Find all the employees whose salary is less than |
|  |

|  |
| --- |
| his immediate colleague in unit by 100 |
| First create a view ranked\_employee\_salary\_view which has all the employees ranked from smaller to higher salary for each  Department. Rank() OVER is PARTITIONED by department\_id |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| CREATE VIEW ranked\_employee\_salary\_view AS |
|  |

|  |
| --- |
| SELECT name, department\_id, salary, rank() OVER |
|  |

|  |
| --- |
| (PARTITION by department\_id order by salary) rnk |
|  |

|  |
| --- |
| FROM employee |
| Next join the ranked\_employee\_salary with itself by using aliases a and b where a and b has same department\_id and b has rank 1 more than a and salary of a is less than from b by least 100 |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| SELECT a.name, a.department\_id, a.salary |
|  |

|  |
| --- |
| FROM ranked\_employee\_salary\_view a JOIN |
|  |

|  |
| --- |
| ranked\_employee\_salary\_view b |
|  |

|  |
| --- |
| WHERE |
|  |

|  |
| --- |
| a.department\_id=b.department\_id |
|  |

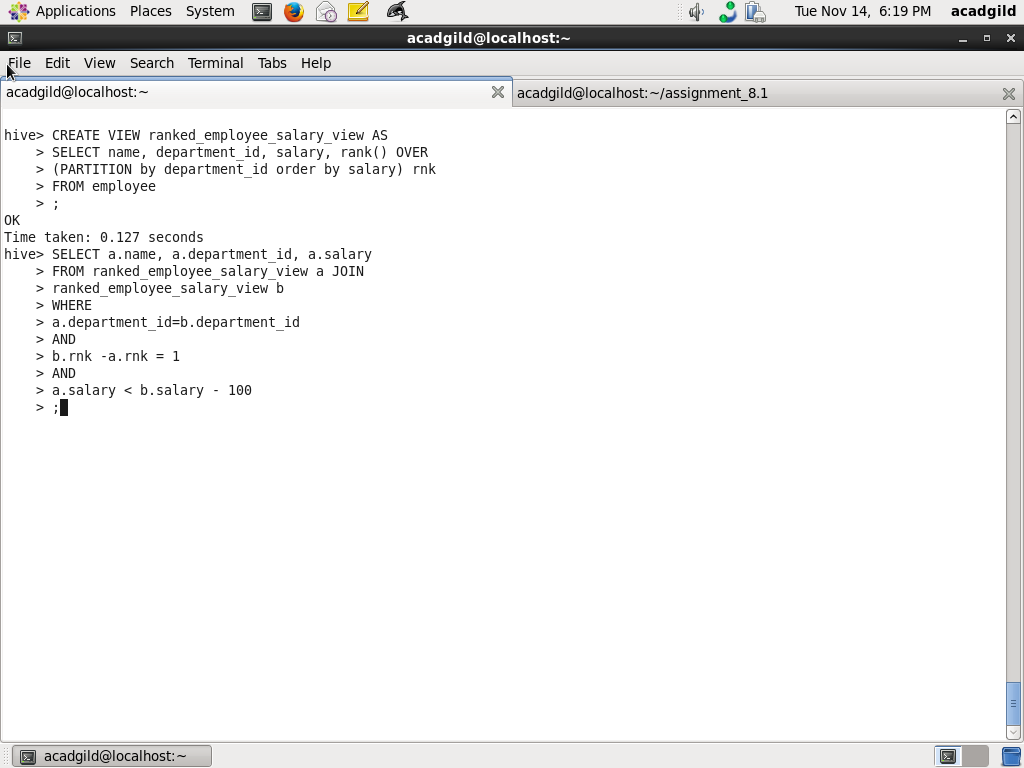
|  |
| --- |
| AND |
|  |

|  |
| --- |
| b.rnk -a.rnk = 1 |
|  |

|  |
| --- |
| AND |
|  |

a.salary < b.salary - 100

Screenshot of query is as below:



Output is as below:

