CSEE5590 Big Data Programming

In Class Programming –7 Report
(Jongkook Son)

Project Overview:

Independent Column based No SQL Tool – Cassandra

Requirements/Task(s):

Create KEYSPACE and a table.

Import the employees' data from the given dataset and apply the following quires

1. List the empID,ename,jobtitle,and hiredate of employee from the employee table.

2. List the name,salary of the employees who are clerks.

3. List the name,job,salary of every employee joined on ‘february 18,2000’,

4. List name and annual salary of all the employees.

5. Display employees’ names, salary and manager values of those employees whose

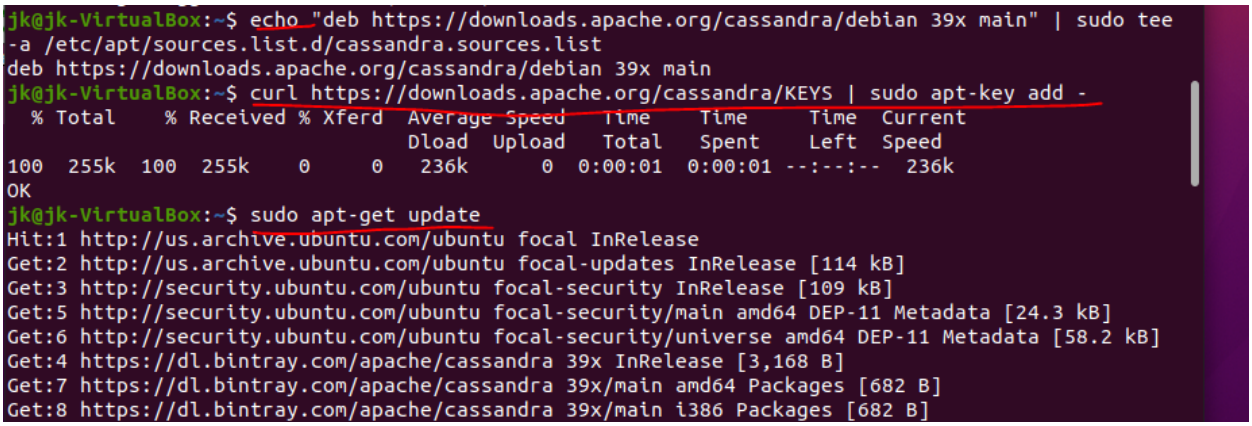
salary is 45000 from EMP table using SELECT statement.

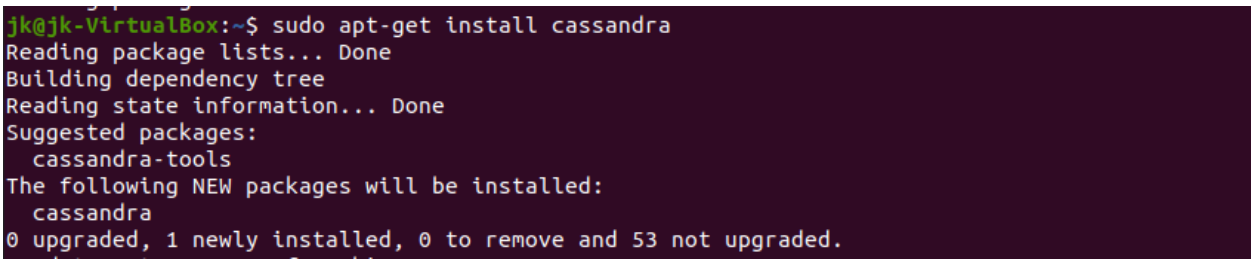
What I learned in ICP:

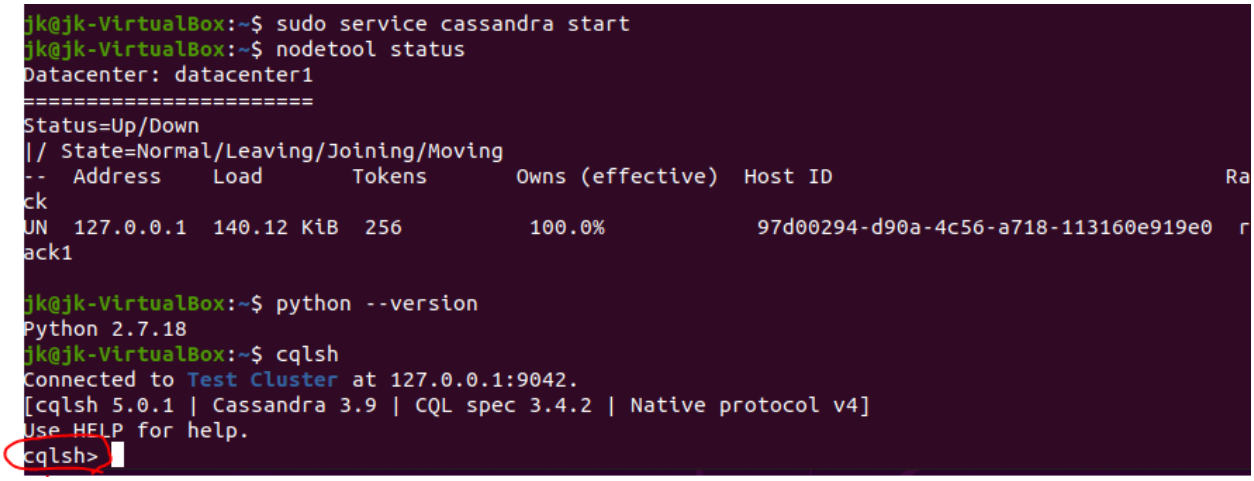
I could have learned how to download Cassandra on my virtual maschine and installing the dependencies like java and python. Cassandra is an open-source NoSQL database management system. Cassandra uses tables and indexes to store data. Its strength are scalability and availably for the end user. Cassandra can operate strongly on multiple data centres which are distributed over the World, along with replication which gives the Cassandra the ability to operate on low latency servers. Many companies like netfilx and reddit use Cassandra for those good points.

INSTALLATION OF CASSANDRA AND START

(I installed it on the ubuntu virtual maschine.)







$ echo "deb http://www.apache.org/dist/cassandra/debian 39x main" | sudo tee -a /etc/apt/sources.list.d/cassandra.sources.list

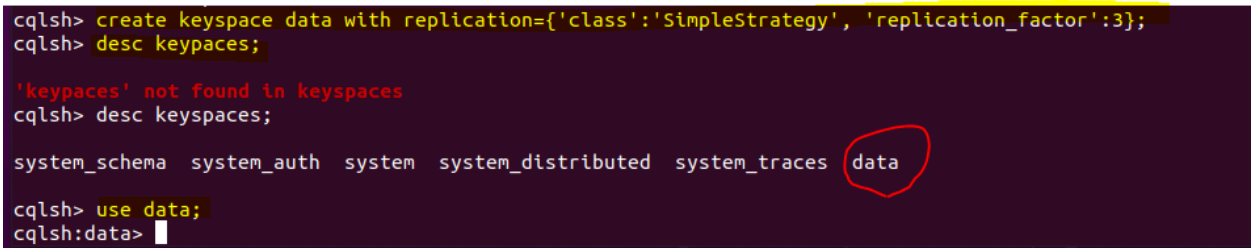
$ curl https://www.apache.org/dist/cassandra/KEYS | sudo apt-key add -

$ sudo apt-get update

$ sudo apt-get install Cassandra

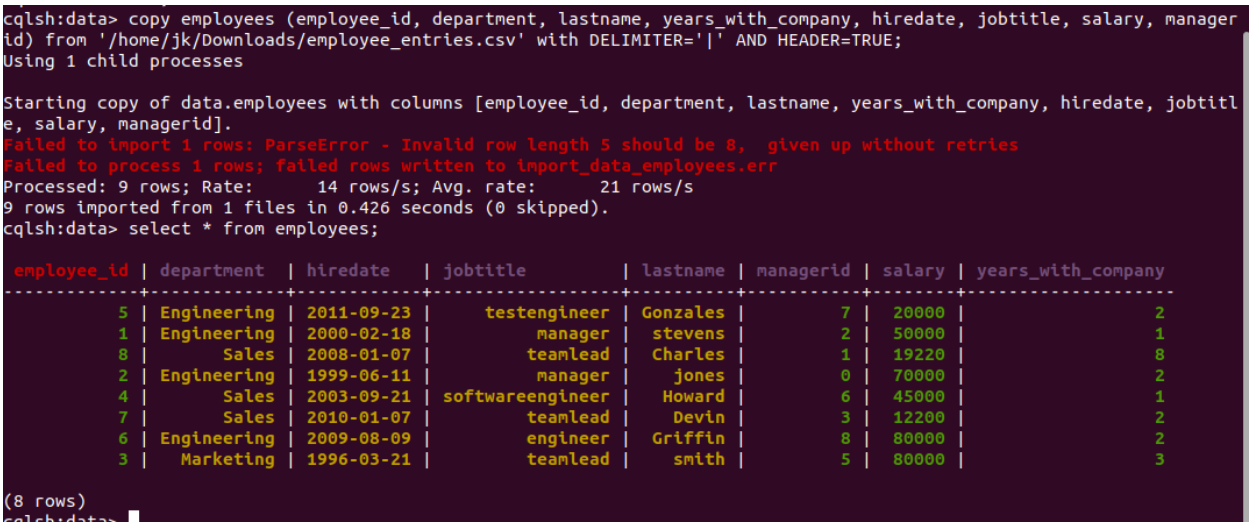
$ sudo service cassandra start

CREATE KEY SPACE



$ create keyspace data with replication={'class':'SimpleStrategy', 'replication_factor':3};

CREATE TABLE AND IMPORT



$ use data;

$ create table data.employees (employee_id int PRIMARY KEY, department text, lastname text, year_with_company int, hiredate text, jobtitle text, salary int, managerid int);

$ copy employees (employee_id, department, lastname, year_with_company, hiredate, jobtitle, salary, managerid) from '/home/jk/Downloads/employee_entries.csv' with DELIMITER='|' AND HEADER=TRUE;

$ select * from employees;

QUERY1 (List the employee_id, lastname, jobtitle, and hiredate of employee from the employee table)

```
cqlsh:data> select employee_id, lastname, jobtitle, hiredate from employees;

 employee_id | lastname | jobtitle         | hiredate
-------------+----------+------------------+------------
           5 | Gonzales |     testengineer | 2011-09-23
           1 |  stevens |          manager | 2000-02-18
           8 |  Charles |         teamlead | 2008-01-07
           2 |    jones |          manager | 1999-06-11
           4 |   Howard | softwareengineer | 2003-09-21
           7 |    Devin |         teamlead | 2010-01-07
           6 |  Griffin |         engineer | 2009-08-09
           3 |    smith |         teamlead | 1996-03-21

(8 rows)
```

$ select employee_id, lastname, jobtitle, hiredate from employees;

QUERY2 (List the lastname, salary, of the employees who are clerks.)

```
(0 rows)
cqlsh:data> SELECT lastname, salary from employees where jobtitle = 'clerk' ALLOW FILTERING;

 lastname | salary
----------+--------

(0 rows)
cqlsh:data> SELECT lastname, salary from employees where jobtitle = 'teamlead' ALLOW FILTERING;

 lastname | salary
----------+--------
  Charles |  19220
    Devin |  12200
    smith |  80000

(3 rows)
```

$ select lastname, salary from employees where jobtitle = 'clerk' ALLOW FILTERING;

QUERY3 (List the lastname, job, salary, of every employee joined on 'February 18,2000'.)

```
cqlsh:data> SELECT lastname, jobtitle, salary from employees where hiredate='2000-02-18' ALLOW FILTERING;

 lastname | jobtitle | salary
----------+----------+--------
  stevens |  manager |  50000
```

$ select lastname, jobtitle, salary from employees where hiredate='2000-02-18' ALLOW FILTERING;

QUERY4 (List lastname, and annual salary of all the employees.)

```
cqlsh:data> SELECT lastname, salary from employees;

 lastname | salary
----------+--------
 Gonzales |  20000
  stevens |  50000
  Charles |  19220
    jones |  70000
   Howard |  45000
    Devin |  12200
  Griffin |  80000
    smith |  80000
```

$ select lastname, salary from employees;

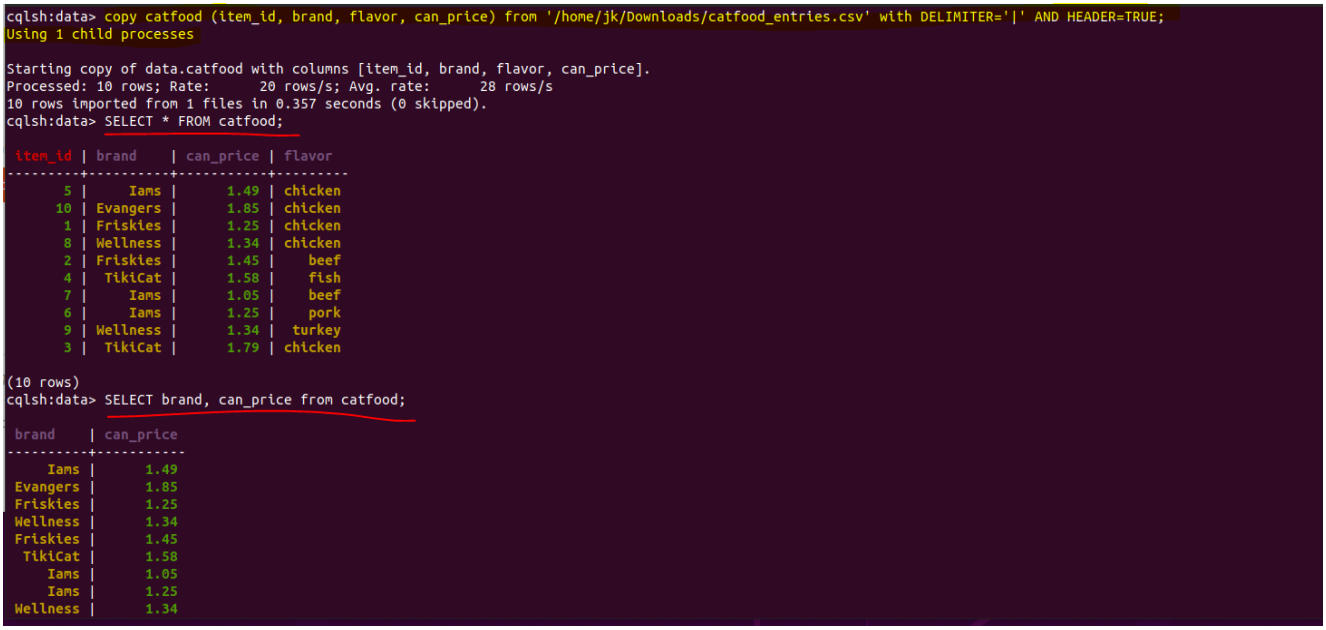
QUERY5 (Display employees' lastname, salary and manager values of those employees whose salary is 45000 from employees table using select statement.)

```
cqlsh:data> SELECT lastname, salary, managerid from employees where salary=45000 ALLOW FILTERING;

 lastname | salary | managerid
----------+--------+-----------
   Howard |  45000 |         6

(1 rows)
```

$ select lastname, salary, managerid from employees where salary=45000 ALLOW FILTERING;

BONUS

=>I used catfood_entries dataset for import it and apply some commands like above



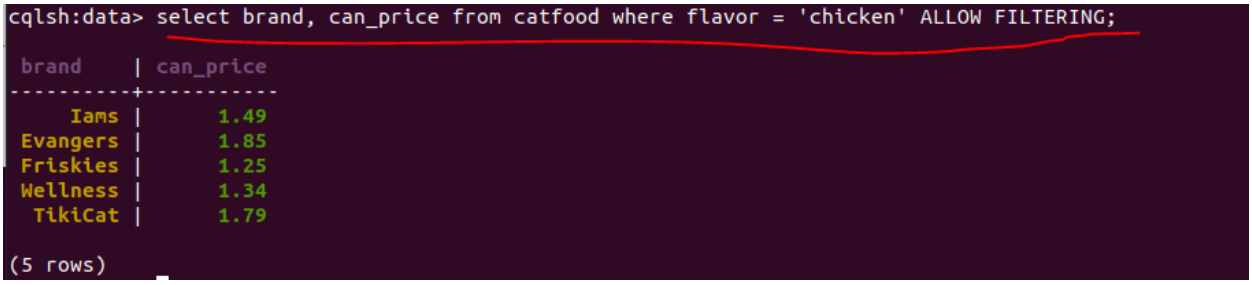
```
cqlsh:data> copy catfood (item_id, brand, flavor, can_price) from '/home/jk/Downloads/catfood_entries.csv' with DELIMITER='|' AND HEADER=TRUE;
Using 1 child processes

Starting copy of data.catfood with columns [item_id, brand, flavor, can_price].
Processed: 10 rows; Rate:      20 rows/s; Avg. rate:      28 rows/s
10 rows imported from 1 files in 0.357 seconds (0 skipped).
cqlsh:data> SELECT * FROM catfood;

 item_id | brand    | can_price | flavor
---------+----------+-----------+---------
       5 |     Iams |      1.49 | chicken
      10 | Evangers |      1.85 | chicken
       1 | Friskies |      1.25 | chicken
       8 | Wellness |      1.34 | chicken
       2 | Friskies |      1.45 |    beef
       4 |  TikiCat |      1.58 |    fish
       7 |     Iams |      1.05 |    beef
       6 |     Iams |      1.25 |    pork
       9 | Wellness |      1.34 |  turkey
       3 |  TikiCat |      1.79 | chicken

(10 rows)
cqlsh:data> SELECT brand, can_price from catfood;

 brand    | can_price
----------+-----------
     Iams |      1.49
 Evangers |      1.85
 Friskies |      1.25
 Wellness |      1.34
 Friskies |      1.45
  TikiCat |      1.58
     Iams |      1.05
     Iams |      1.25
 Wellness |      1.34
```

$ create table catfood (item_id int PRIMARY KEY, brand text, flavor text, can_price float);

$ copy catfood (item_id, brand, flavor, can_price) from '/home/jk/Downloads/catfood_entries.csv' with DELIMITER='|' AND HEADER=TRUE;

$SELECT * FROM catfood;

$SELECT brand, can_price from catfood;



```
cqlsh:data> select brand, can_price from catfood where flavor = 'chicken' ALLOW FILTERING;

 brand    | can_price
----------+-----------
     Iams |      1.49
 Evangers |      1.85
 Friskies |      1.25
 Wellness |      1.34
  TikiCat |      1.79

(5 rows)
```

$ select brand, can_price from catfood where flavor = ‘chicken’ ALLOW FILTERING;

=>Display the brand, can_price where the flavor’s value is chicken

```
cqlsh:data> select item_id, brand, flavor from catfood where can_price<1.5 ALLOW FILTERING;

 item_id | brand    | flavor
---------+----------+---------
       5 |     Iams | chicken
       1 | Friskies | chicken
       8 | Wellness | chicken
       2 | Friskies |    beef
       7 |     Iams |    beef
       6 |     Iams |    pork
       9 | Wellness |  turkey

(7 rows)
```

$ select item_id, brand, flavor from catfood where can_price<1.5 ALLOW FILTERING;

=>Display item id and brand where can price is below 1.5

```
cqlsh:data> SELECT brand, flavor, can_price from catfood where can_price>1.5 AND flavor='chicken' ALLOW FILTERING;

 brand    | flavor  | can_price
----------+---------+-----------
 Evangers | chicken |      1.85
  TikiCat | chicken |      1.79
```

$ select brand, flavor, can_price from catfood where can_price>1.5 AND flavor =”chicken” ALLOW FILTERING;

=>Display flavor and can price where can price is over 1.5 and flavor value is chicken

```
cqlsh:data> SELECT brand, flavor from catfood where brand='Iams' ALLOW FILTERING;

 brand | flavor
-------+---------
  Iams | chicken
  Iams |    beef
  Iams |    pork

(3 rows)
```

$ select brand, flavor from catfood where brand=”Iams” ALLOW FILTERING;

=>Display brand and flavor where brand value is Iams