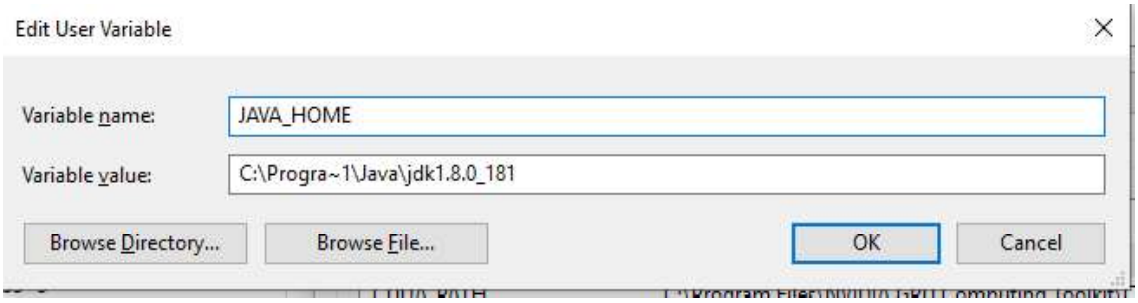
Steps for Cassandra installation:

1. Cassandra need JDK to run. First need to install JDK on the PC.

- Go To Oracle (SignUp required) or Filehippo to download JDK 1.8 from which you find suitable.
- Run the Install as it is.
- Configure JDK path As:
- Copy the JDK where is installed it's bin directory. Mine is 64-bit found inside "C:\Program Files\Java\jdk1.8.0_181\bin" and place on Environment Variable as new Env_name : "JAVA_HOME" later we use for cassandra.



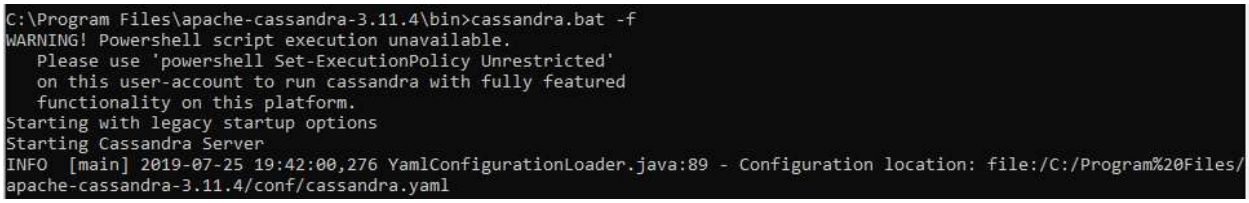
2. Go to Apache Cassandra Download Page. And Download the latest version. The latest version at that time is cassandra-3.11.4

- Unzip it and place all files inside sub folder into ……. "C:\Program Files\apache-cassandra-3.11.4"
- Run "cassandra.bat -f" with CMD Run as Administrator.

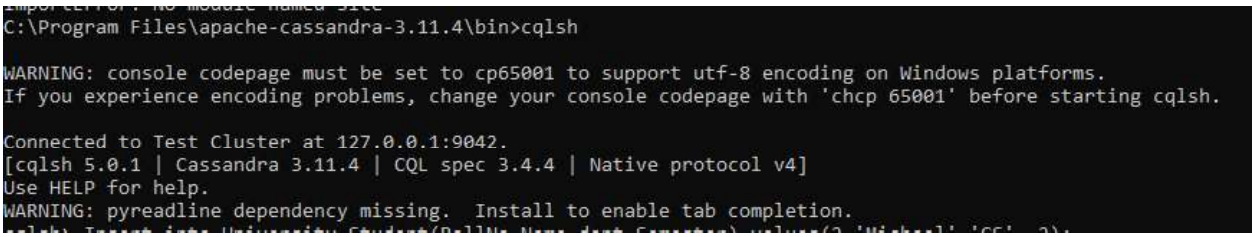
3. Need Python2.7 to run Cassandra Query shell cqlsh .

- Download Python2.7 latest version and extract inside the "C:\Program Files\apache-cassandra-3.11.4\bin" during installing.

4. Finally run the Cassandra Server as "cassandra.bat -f" with CMD Run as Administrator from the bin directory. Following screen should see on successful start.



- Don't close it, keep running.
- Then, open other CMD go over the bin directory of cassandra installed. To run cqlsh by type **cqlsh .** Following output should get.



Creating Keyspace, Table and Inserting Values:

```
cqlsh> CREATE KEYSPACE students WITH REPLICATION={
   ... 'class':'SimpleStrategy',
   ... 'replication_factor':1};
```

```
cqlsh> use students;
cqlsh:students> CREATE TABLE stu(
           ... roll_no int PRIMARY KEY,
           ... name text,
           ... join_date timestamp,
           ... marks double );
cqlsh:students> INSERT INTO stu(roll_no,name,join_date,marks)
           ... values(1,'Ajay','2018-03-09',85.4);
cqlsh:students> INSERT INTO stu(roll_no,name,join_date,marks)
           ... values(2,'Anil','2018-06-07',65.8);
cqlsh:students> INSERT INTO stu(roll_no,name,join_date,marks)
           ... values(3,'Suman','2019-10-08',78.76);
```

Displaying the details of Student Table:

```
cqlsh:students> select * from stu;

 roll_no | join_date                       | marks | name
---------+---------------------------------+-------+-------
       1 | 2018-03-08 18:30:00.000000+0000 |  85.4 |  Ajay
       2 | 2018-06-06 18:30:00.000000+0000 |  65.8 |  Anil
       3 | 2019-10-07 18:30:00.000000+0000 | 78.76 | Suman
```

Creating a CSV file: cass.csv

```
4,Anusha,2019-08-10,76.8
5,Varun,2018-07-12,89.00
6,Sheethal,2017-02-04,80.55
7,Nisha,2018-06-11,75.68
8,Roshni,2019-08-01,90.5
```

Importing data from cass.csv:

```
cqlsh:students> COPY stu(roll_no,name,join_date,marks) FROM 'C:\Users\hp\Desktop\cass.csv';
Using 3 child processes

Starting copy of students.stu with columns [roll_no, name, join_date, marks].
```

Data after importing from cass.csv:

```
cqlsh:students> select * from stu;

 roll_no | join_date                                | marks | name
---------+------------------------------------------+-------+----------
       5 | 2018-07-12 05:30:00.000000+0000 |    89 |    Varun
       1 | 2018-03-08 18:30:00.000000+0000 |  85.4 |     Ajay
       8 | 2019-08-01 05:30:00.000000+0000 |  90.5 |   Roshni
       2 | 2018-06-06 18:30:00.000000+0000 |  65.8 |     Anil
       4 | 2019-08-10 05:30:00.000000+0000 |  76.8 |   Anusha
       7 | 2018-06-11 05:30:00.000000+0000 | 75.68 |    Nisha
       6 | 2017-02-04 05:30:00.000000+0000 | 80.55 | Sheethal
       3 | 2019-10-07 18:30:00.000000+0000 | 78.76 |    Suman
```

Defining Collection Set:

```
cqlsh:students> ALTER TABLE stu ADD hobbies set<text>;
```

```
cqlsh:students> update stu set hobbies=hobbies+{'Chess,Music,Dance'}
           ... where roll_no=6;
cqlsh:students> update stu set hobbies=hobbies+{'Chess,Dance'}
           ... where roll_no=7;
cqlsh:students> update stu set hobbies=hobbies+{'Cricket,Dance'}
           ... where roll_no=8;
cqlsh:students> select * from stu;

 roll_no | hobbies                | join_date                                | marks | name
---------+------------------------+------------------------------------------+-------+----------
       5 |       {'Cricket,Jog'} | 2018-07-12 05:30:00.000000+0000 |    89 |    Varun
       1 |       {'Chess,Dance'} | 2018-03-08 18:30:00.000000+0000 |  85.4 |     Ajay
       8 |     {'Cricket,Dance'} | 2019-08-01 05:30:00.000000+0000 |  90.5 |   Roshni
       2 |        {'Dance,Sing'} | 2018-06-06 18:30:00.000000+0000 |  65.8 |     Anil
       4 |      {'Cricket,Golf'} | 2019-08-10 05:30:00.000000+0000 |  76.8 |   Anusha
       7 |       {'Chess,Dance'} | 2018-06-11 05:30:00.000000+0000 | 75.68 |    Nisha
       6 | {'Chess,Music,Dance'} | 2017-02-04 05:30:00.000000+0000 | 80.55 | Sheethal
       3 |    {'Dance,Football'} | 2019-10-07 18:30:00.000000+0000 | 78.76 |    Suman
```

Exporting this data to a CSV file:

```
cqlsh:students> COPY stu(roll_no,name,join_date,marks,hobbies) to 'C:\Users\hp\Desktop\cexp.csv';
Using 3 child processes

Starting copy of students.stu with columns [roll_no, name, join_date, marks, hobbies].
Processed: 8 rows; Rate:      18 rows/s; Avg. rate:       1 rows/s
8 rows exported to 1 files in 6.832 seconds.
```

Exported Data :

```
1,Ajay,2018-03-08 18:30:00.000+0000,85.4,"{'Chess,Dance'}"
4,Anusha,2019-08-10 05:30:00.000+0000,76.8,"{'Cricket,Golf'}"
7,Nisha,2018-06-11 05:30:00.000+0000,75.68,"{'Chess,Dance'}"
5,Varun,2018-07-12 05:30:00.000+0000,89,"{'Cricket,Jog'}"
6,Sheethal,2017-02-04 05:30:00.000+0000,80.55,"{'Chess,Music,Dance'}"
8,Roshni,2019-08-01 05:30:00.000+0000,90.5,"{'Cricket,Dance'}"
2,Anil,2018-06-06 18:30:00.000+0000,65.8,"{'Dance,Sing'}"
3,Suman,2019-10-07 18:30:00.000+0000,78.76,"{'Dance,Football'}"
```

Defining Collections Map:

```
cqlsh:students> alter table stu add todo map<int,text>;
```

```
cqlsh:students> update stu set todo={1:'Cassandra'}
           ... where roll_no=2;
cqlsh:students> update stu set todo={1:'MongoDB'}
           ... where roll_no=3;
cqlsh:students> update stu set todo={1:'MongoDB',2:'RDBMS'}
           ... where roll_no=4;
cqlsh:students> update stu set todo={1:'MongoDB',2:'Membase'}
           ... where roll_no=5;
cqlsh:students> update stu set todo={1:'Cassandra',2:'HyperTable'}
           ... where roll_no=6;
cqlsh:students> update stu set todo={1:'Cassandra',2:'MongoDB'}
           ... where roll_no=7;
cqlsh:students> update stu set todo={1:'Hypertable'}
           ... where roll_no=8;
cqlsh:students>
cqlsh:students> select * from stu;

 roll_no | hobbies              | join_date                       | marks | name     | todo
---------+----------------------+---------------------------------+-------+----------+-----------------------------------
       5 |       {'Cricket,Jog'} | 2018-07-12 05:30:00.000000+0000 |    89 |    Varun |      {1: 'MongoDB', 2: 'Membase'}
       1 |       {'Chess,Dance'} | 2018-03-08 18:30:00.000000+0000 |  85.4 |     Ajay |    {1: 'Cassandra', 2: 'MongoDB'}
       8 |     {'Cricket,Dance'} | 2019-08-01 05:30:00.000000+0000 |  90.5 |   Roshni |                {1: 'Hypertable'}
       2 |        {'Dance,Sing'} | 2018-06-06 18:30:00.000000+0000 |  65.8 |     Anil |                 {1: 'Cassandra'}
       4 |      {'Cricket,Golf'} | 2019-08-10 05:30:00.000000+0000 |  95.6 |   Anusha |        {1: 'MongoDB', 2: 'RDBMS'}
       7 |       {'Chess,Dance'} | 2018-06-11 05:30:00.000000+0000 | 75.68 |    Nisha |    {1: 'Cassandra', 2: 'MongoDB'}
       6 | {'Chess,Music,Dance'} | 2017-02-04 05:30:00.000000+0000 | 80.55 | Sheethal | {1: 'Cassandra', 2: 'HyperTable'}
       3 |    {'Dance,Football'} | 2019-10-07 18:30:00.000000+0000 | 78.76 |    Suman |                   {1: 'MongoDB'}
```

Queries:

Retrieving the details of those students whose marks are greater than 90:

```
cqlsh:students> select roll_no,name,marks from stu where marks>90.00
           ... ALLOW FILTERING;

 roll_no | name   | marks
---------+--------+-------
       8 | Roshni |  90.5
       4 | Anusha |  95.6
```

Retrieving details of those students who have opted for the course of Cassandra:

```
cqlsh:students> select * from stu where todo[1]='Cassandra'
          ... ALLOW FILTERING;

 roll_no | hobbies              | join_date                       | marks | name     | todo
---------+----------------------+---------------------------------+-------+----------+---------------------------------
       1 |       {'Chess,Dance'} | 2018-03-08 18:30:00.000000+0000 |  85.4 |     Ajay |    {1: 'Cassandra', 2: 'MongoDB'}
       2 |        {'Dance,Sing'} | 2018-06-06 18:30:00.000000+0000 |  65.8 |     Anil |                {1: 'Cassandra'}
       7 |       {'Chess,Dance'} | 2018-06-11 05:30:00.000000+0000 | 75.68 |    Nisha |    {1: 'Cassandra', 2: 'MongoDB'}
       6 | {'Chess,Music,Dance'} | 2017-02-04 05:30:00.000000+0000 | 80.55 | Sheethal | {1: 'Cassandra', 2: 'HyperTable'}
```

Counting the total marks of all students:

```
cqlsh:students> select sum(marks) as tot_marks from stu;

 tot_marks
-----------
    661.29
```

To demonstrate usage of IN operator:

```
cqlsh:students> select roll_no,name,marks from stu where roll_no in(2,4,6);

 roll_no | name     | marks
---------+----------+-------
       2 |     Anil |  65.8
       4 |   Anusha |  95.6
       6 | Sheethal | 80.55
```

Updating the marks of particular students:

```
cqlsh:students> update stu set marks=95.6 where roll_no=4;
cqlsh:students> select * from stu where roll_no=4;

 roll_no | hobbies          | join_date                       | marks | name
---------+------------------+---------------------------------+-------+--------
       4 | {'Cricket,Golf'} | 2019-08-10 05:30:00.000000+0000 |  95.6 | Anusha
```