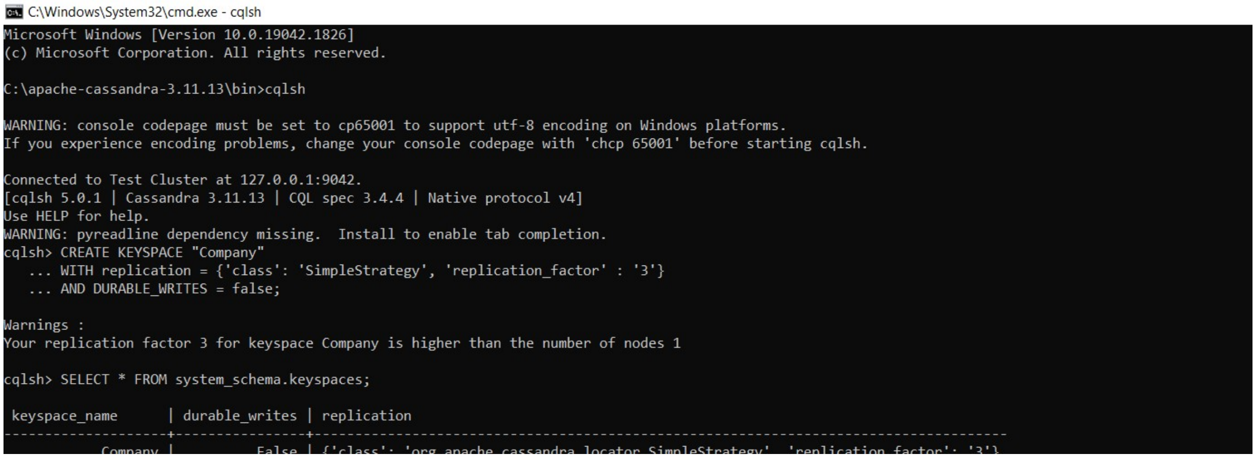
NO SQL LAB – 4

NAME : SREENIDHI GANACHARI

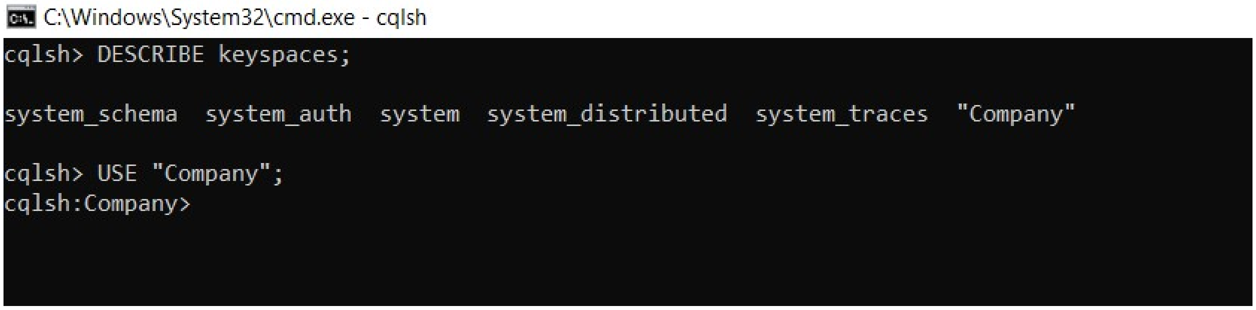
REGISTRATION NUMBER : 19BCE7230

Write the CassandraDBqueries for the following

1.Create a keyspace called ‘Company’ with properties and write a query to select keyspace as ‘Company’.

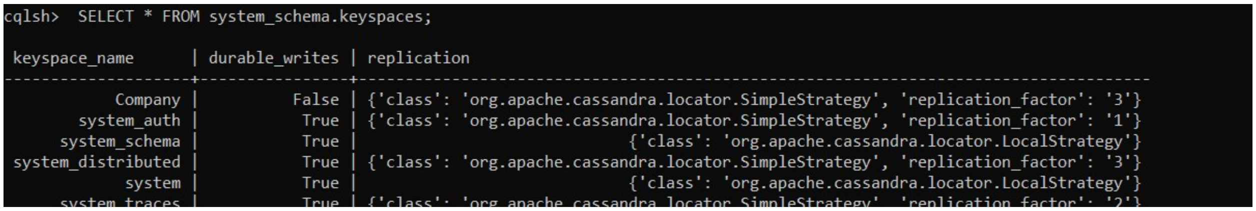


```
Microsoft Windows [Version 10.0.19042.1826]
(c) Microsoft Corporation. All rights reserved.

C:\apache-cassandra-3.11.13\bin>cqlsh

WARNING: console codepage must be set to cp65001 to support utf-8 encoding on Windows platforms.
If you experience encoding problems, change your console codepage with 'chcp 65001' before starting cqlsh.

Connected to Test Cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.11.13 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
WARNING: pyreadline dependency missing.  Install to enable tab completion.
cqlsh> CREATE KEYSPACE "Company"
   ... WITH replication = {'class': 'SimpleStrategy', 'replication_factor' : '3'}
   ... AND DURABLE_WRITES = false;

Warnings :
Your replication factor 3 for keyspace Company is higher than the number of nodes 1

cqlsh> SELECT * FROM system_schema.keyspaces;

 keyspace_name      | durable_writes | replication
--------------------+----------------+-------------------------------------------------------------------------------------
            Company |          False | {'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '3'}
```



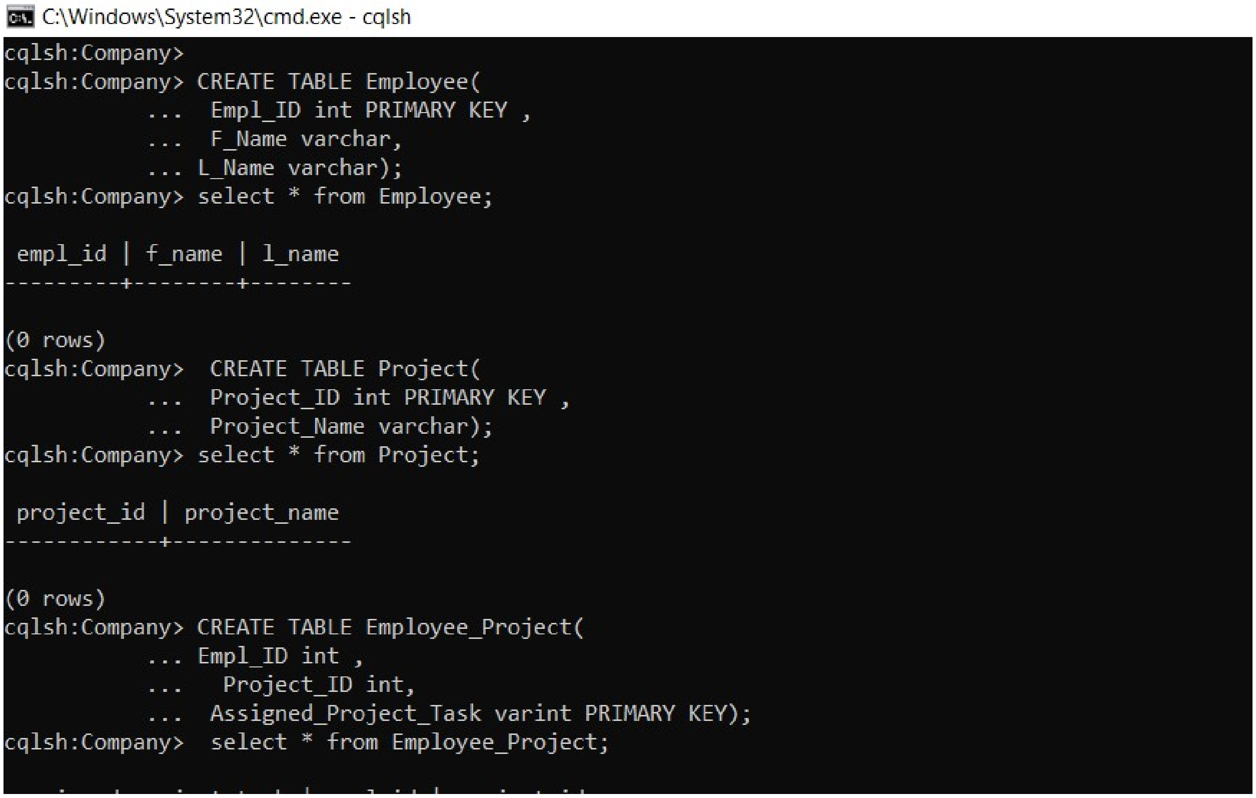
```
cqlsh> DESCRIBE keyspaces;

system_schema  system_auth  system  system_distributed  system_traces  "Company"

cqlsh> USE "Company";
cqlsh:Company>
```

2.Write a CQL query to verify and display keyspace.



```
cqlsh>  SELECT * FROM system_schema.keyspaces;

 keyspace_name      | durable_writes | replication
--------------------+----------------+-------------------------------------------------------------------------------------
            Company |          False | {'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '3'}
        system_auth |           True | {'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '1'}
      system_schema |           True |                             {'class': 'org.apache.cassandra.locator.LocalStrategy'}
 system_distributed |           True | {'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '3'}
             system |           True |                             {'class': 'org.apache.cassandra.locator.LocalStrategy'}
      system_traces |           True | {'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '2'}
```

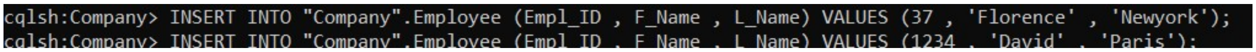
3.Create all the above tables.



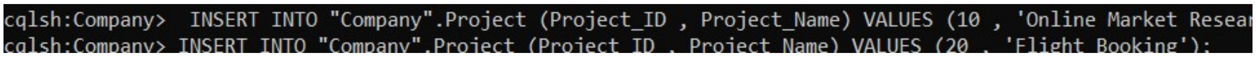
C:\Windows\System32\cmd.exe - cqlsh

```
cqlsh:Company>
cqlsh:Company> CREATE TABLE Employee(
           ...  Empl_ID int PRIMARY KEY ,
           ...  F_Name varchar,
           ... L_Name varchar);
cqlsh:Company> select * from Employee;

 empl_id | f_name | l_name
---------+--------+--------

(0 rows)
cqlsh:Company>  CREATE TABLE Project(
           ...  Project_ID int PRIMARY KEY ,
           ...  Project_Name varchar);
cqlsh:Company> select * from Project;

 project_id | project_name
------------+-------------

(0 rows)
cqlsh:Company> CREATE TABLE Employee_Project(
           ... Empl_ID int ,
           ...   Project_ID int,
           ...  Assigned_Project_Task varint PRIMARY KEY);
cqlsh:Company>  select * from Employee_Project;
```

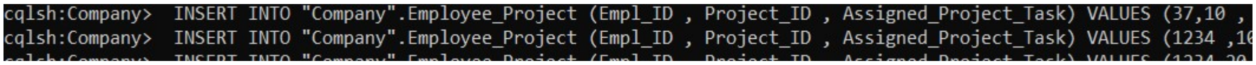
4.Implement all the referential integrity constraints.



```
cqlsh:Company> INSERT INTO "Company".Employee (Empl_ID , F_Name , L_Name) VALUES (37 , 'Florence' , 'Newyork');
cqlsh:Company> INSERT INTO "Company".Employee (Empl_ID , F_Name , L_Name) VALUES (1234 , 'David' , 'Paris');
```

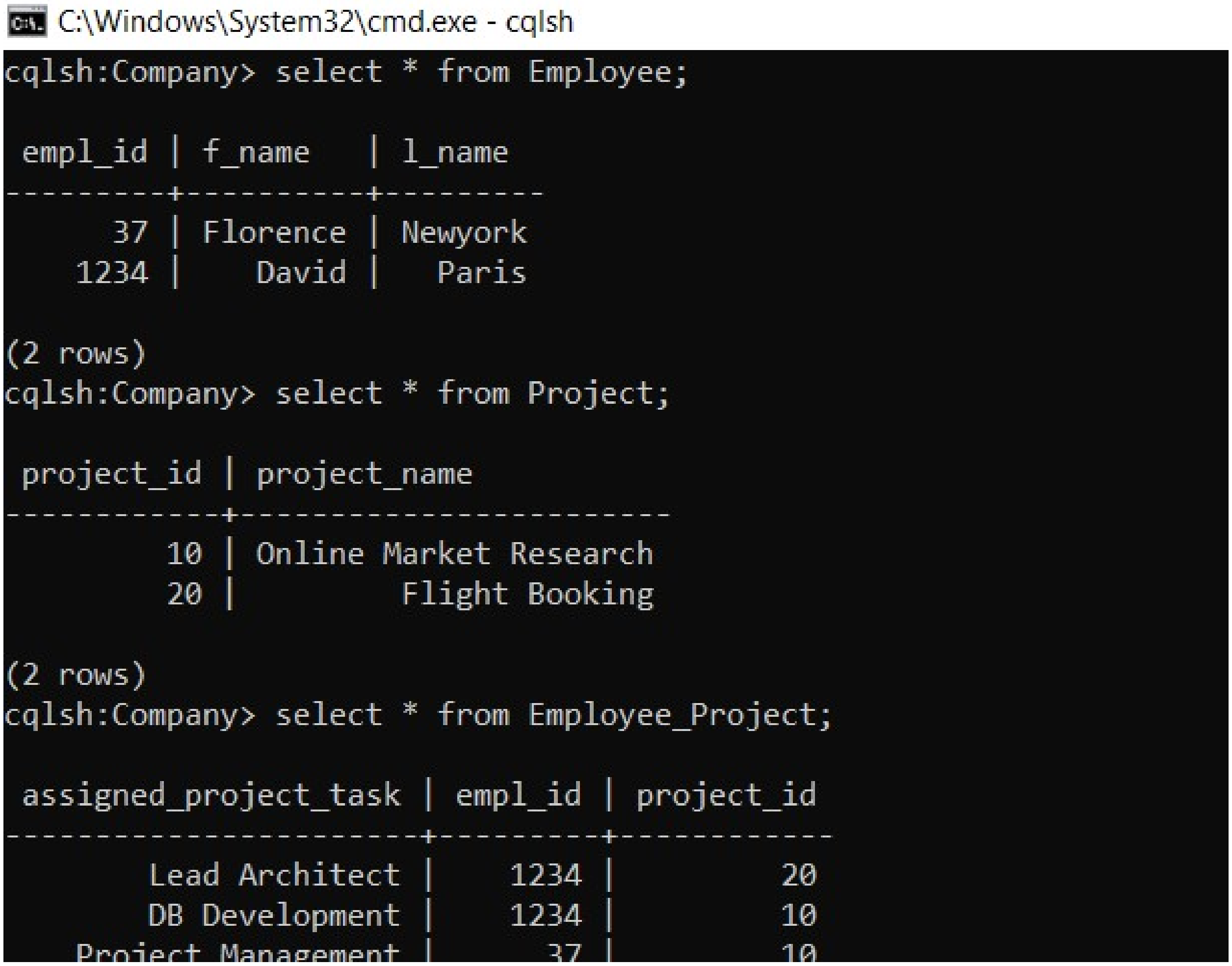


```
cqlsh:Company>  INSERT INTO "Company".Project (Project_ID , Project_Name) VALUES (10 , 'Online Market Resear
cqlsh:Company> INSERT INTO "Company".Project (Project_ID , Project_Name) VALUES (20 , 'Flight Booking');
```



```
cqlsh:Company>  INSERT INTO "Company".Employee_Project (Empl_ID , Project_ID , Assigned_Project_Task) VALUES (37,10 ,
cqlsh:Company>  INSERT INTO "Company".Employee_Project (Empl_ID , Project_ID , Assigned_Project_Task) VALUES (1234 ,10
cqlsh:Company>  INSERT INTO "Company".Employee_Project (Empl_ID , Project_ID , Assigned_Project_Task) VALUES (1234 ,20
```

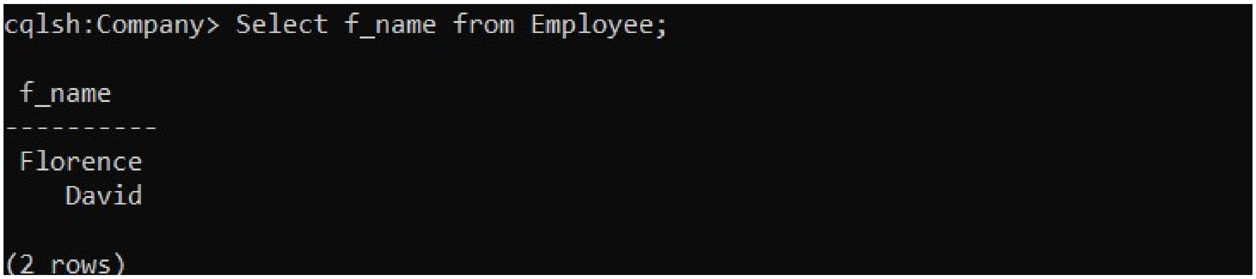
5.Insert all the values using CQL to the respective tables



C:\Windows\System32\cmd.exe - cqlsh

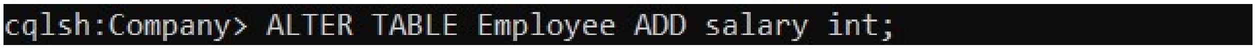
```
cqlsh:Company> select * from Employee;

 empl_id | f_name   | l_name
---------+----------+---------
      37 | Florence | Newyork
    1234 |    David |   Paris

(2 rows)
cqlsh:Company> select * from Project;

 project_id | project_name
------------+------------------------
         10 | Online Market Research
         20 |         Flight Booking

(2 rows)
cqlsh:Company> select * from Employee_Project;

 assigned_project_task | empl_id | project_id
-----------------------+---------+------------
        Lead Architect |    1234 |         20
        DB Development |    1234 |         10
    Project Management |      37 |         10
```

6.Write a CQL query to display all the employee names.

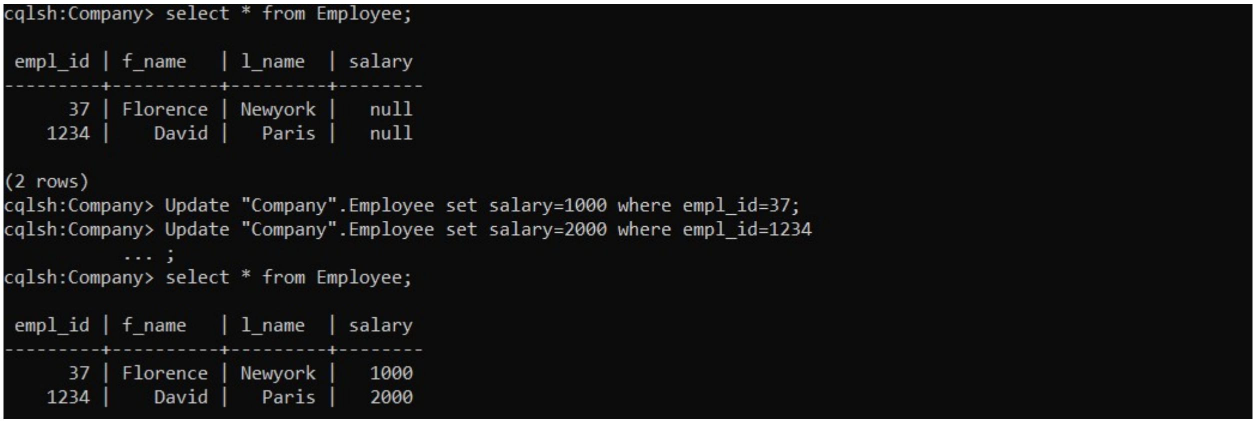


```
cqlsh:Company> Select f_name from Employee;

 f_name
----------
 Florence
    David

(2 rows)
```

7.Alter table with salary column and insert values.

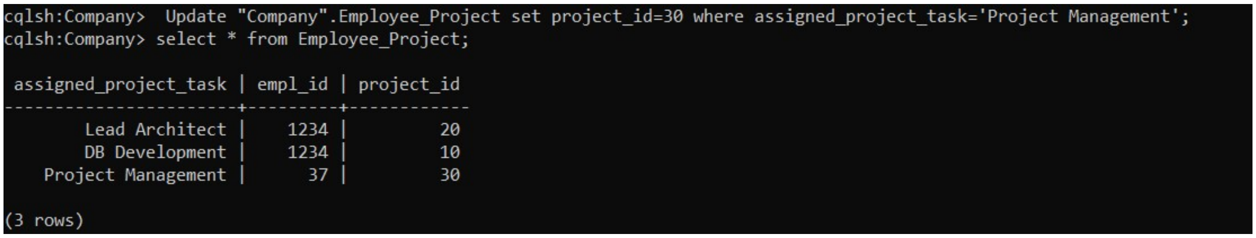


```
cqlsh:Company> ALTER TABLE Employee ADD salary int;
```



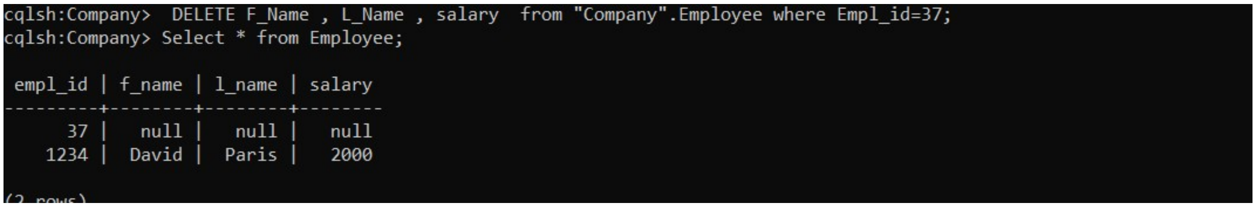
```
cqlsh:Company> select * from Employee;

 empl_id | f_name   | l_name  | salary
---------+----------+---------+--------
      37 | Florence | Newyork |   null
    1234 |    David |   Paris |   null

(2 rows)
cqlsh:Company> Update "Company".Employee set salary=1000 where empl_id=37;
cqlsh:Company> Update "Company".Employee set salary=2000 where empl_id=1234
           ... ;
cqlsh:Company> select * from Employee;

 empl_id | f_name   | l_name  | salary
---------+----------+---------+--------
      37 | Florence | Newyork |   1000
    1234 |    David |   Paris |   2000
```

8.Write a CQL query to update the project id where empl_id = 37.



```
cqlsh:Company>  Update "Company".Employee_Project set project_id=30 where assigned_project_task='Project Management';
cqlsh:Company> select * from Employee_Project;

 assigned_project_task | empl_id | project_id
-----------------------+---------+------------
        Lead Architect |    1234 |         20
        DB Development |    1234 |         10
    Project Management |      37 |         30

(3 rows)
```

9.Write a CQL query to delete the data where empl_id=37.

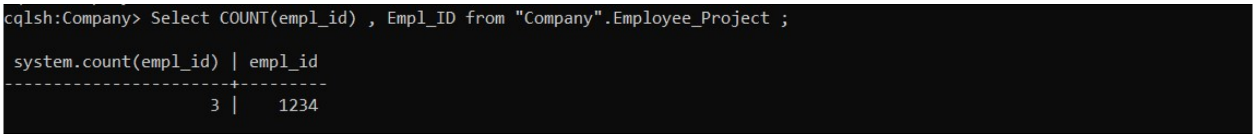


```
cqlsh:Company>  DELETE F_Name , L_Name , salary  from "Company".Employee where Empl_id=37;
cqlsh:Company> Select * from Employee;

 empl_id | f_name | l_name | salary
---------+--------+--------+--------
      37 |   null |   null |   null
    1234 |  David |  Paris |   2000

(2 rows)
```

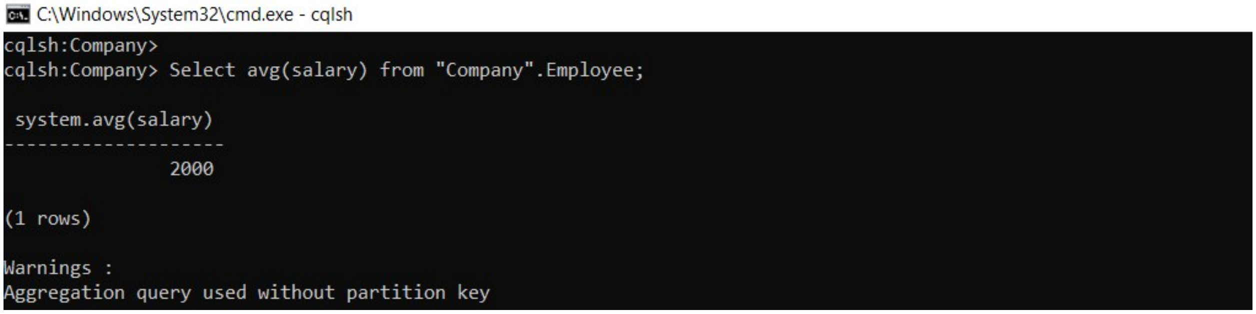
10.CQL Querying:

▪Find the employee which has more than two projects.

▪Find the average salary of all employees.



```
cqlsh:Company> Select COUNT(empl_id) , Empl_ID from "Company".Employee_Project ;

 system.count(empl_id) | empl_id
-----------------------+---------
                     3 |    1234
```



```
cqlsh:Company>
cqlsh:Company> Select avg(salary) from "Company".Employee;

 system.avg(salary)
--------------------
               2000

(1 rows)

Warnings :
Aggregation query used without partition key
```