LAB – 5

CASSANDRA QUERIES

Consider the following tables.

Table 1: Employee

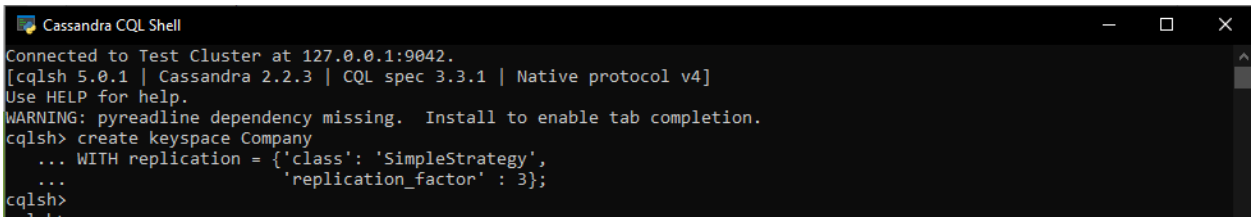
Empl_ID (number(3))	F_Name (varchar(10))	L_Name (varchar(10))
37	Florence	Newyork
1234	David	Paris

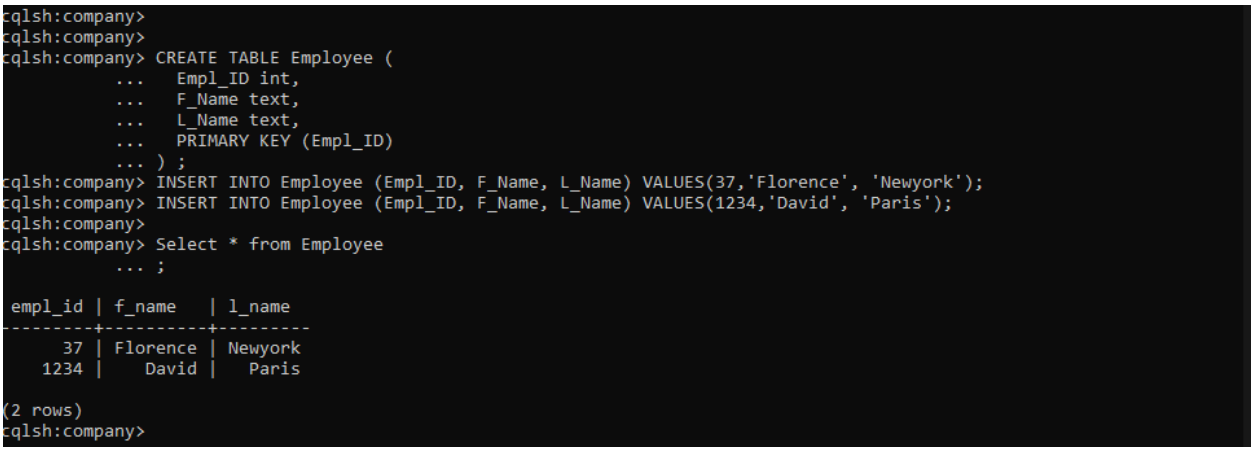
Table 2: Project

Project_ID (number(2))	Project_Name (varchar(20))
10	Online Market Research
20	Flight_Booking

Table 3: Employee_Project

Empl_ID (number(3))	Project_ID (number(2))	Assigned_Project_Task (varchar(20))
37	10	Project Management
1234	10	DB Development
1234	20	Lead Architect





```
cqlsh:company>
cqlsh:company>
cqlsh:company> CREATE TABLE Project (
          ...   Project_ID int,
          ...   Project_Name text,
          ...   PRIMARY KEY (Project_ID)
          ... ) ;
cqlsh:company>
cqlsh:company> INSERT INTO Project (Project_ID, Project_Name) VALUES (10, 'Online Market Research');
cqlsh:company> INSERT INTO Project (Project_ID, Project_Name) VALUES (20, 'Flight Booking');
cqlsh:company>
cqlsh:company> select * from Project;

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

(2 rows)
cqlsh:company>
```

```
cqlsh:company>
cqlsh:company> CREATE TABLE Employee_Project (
          ...   Empl_ID int,
          ...   Project_ID int,
          ...   Assigned_Project_Task text,
          ...   PRIMARY KEY (Empl_ID, Project_ID)
          ... ) ;
cqlsh:company>
cqlsh:company>
cqlsh:company> INSERT INTO Employee_Project (Empl_ID, Project_ID, Assigned_Project_Task) VALUES(37,10, 'Project Manageme
nt');
cqlsh:company> INSERT INTO Employee_Project (Empl_ID, Project_ID, Assigned_Project_Task) VALUES(1234,10, 'DB Development
');
cqlsh:company> INSERT INTO Employee_Project (Empl_ID, Project_ID, Assigned_Project_Task) VALUES(1234,20, 'Lead Architect
');
cqlsh:company>
cqlsh:company> select * from Employee_Project;

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

(3 rows)
cqlsh:company>
```

Answer the following questions

1. Create a keyspace called '*Company*' with properties *and write* a query to select keypsace as '*Company*'.
2. Write a CQL query to verify and display keyspace.
3. Create all the above tables.
4. Implement all the referential integrity constraints.
5. Insert all the values using CQL to the respective tables
6. Write a CQL query to display all the employee names.
7. Alter table with salary column and insert values.
8. Write a CQL query to update the project id where empl_id = 37.
9. Write a CQL query to delete the data where empl_id=1234.
10. CQL Querying:
    - Find the employee which has more than two projects.

- Find the average salary of all employees.

```
cqlsh:company> ALTER TABLE employee ADD salary int;
cqlsh:company> select * employee;
SyntaxException: <ErrorMessage code=2000 [Syntax error in CQL query] message="line 1:9 missing K_FROM at 'employee' (sel
ect * [employee];)">
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 employee SET salary=80000 WHERE Empl_ID = 37;
cqlsh:company> UPDATE employee SET salary=120000 WHERE Empl_ID = 1234;
cqlsh:company> select * from employee;

 empl_id | f_name   | l_name  | salary
---------+----------+---------+--------
      37 | Florence | Newyork |  80000
    1234 |    David |   Paris | 120000

(2 rows)
cqlsh:company>
```

```
cqlsh:company>
cqlsh:company> UPDATE project SET project_name='Cassandra DB' WHERE Project_ID = 10;
cqlsh:company> DELETE FROM Company.employee WHERE Empl_ID=37;
cqlsh:company> select * from project;

 project_id | project_name
------------+----------------
         10 |   Cassandra DB
         20 | Flight Booking

(2 rows)
cqlsh:company> select * from employee;

 empl_id | f_name | l_name | salary
---------+--------+--------+--------
    1234 |  David |  Paris | 120000

(1 rows)
cqlsh:company>
```

```
cqlsh:company>
cqlsh:company> SELECT AVG(salary) AS Average FROM Company.employee;

 average
---------
  120000

(1 rows)
```

CODE

```
CREATE TABLE Project (
  Project_ID int,
  Project_Name text,
  PRIMARY KEY (Project_ID)
) ;

INSERT INTO Project (Project_ID, Project_Name) VALUES (10, 'Online Market Research');
INSERT INTO Project (Project_ID, Project_Name) VALUES (20, 'Flight Booking');

CREATE TABLE Employee_Project (
  Empl_ID int,
  Project_ID int,
  Assigned_Project_Task text,
```

```
 PRIMARY KEY (Empl_ID, Project_ID)

) ;


INSERT INTO Employee_Project (Empl_ID, Project_ID, Assigned_Project_Task) VALUES(37,10, 'Project Management');



INSERT INTO Employee_Project (Empl_ID, Project_ID, Assigned_Project_Task) VALUES(1234,10, 'DB Development');



INSERT INTO Employee_Project (Empl_ID, Project_ID, Assigned_Project_Task) VALUES(1234,20, 'Lead Architect');

ALTER TABLE employee ADD salary int;

UPDATE employee SET salary=80000 WHERE Empl_ID = 37;

UPDATE employee SET salary=120000 WHERE Empl_ID = 1234;

UPDATE project SET project_name='Cassandra DB' WHERE Project_ID = 10;

DELETE FROM Company.employee WHERE Empl_ID=37;

SELECT AVG(salary) AS Average FROM Company.employee;
```