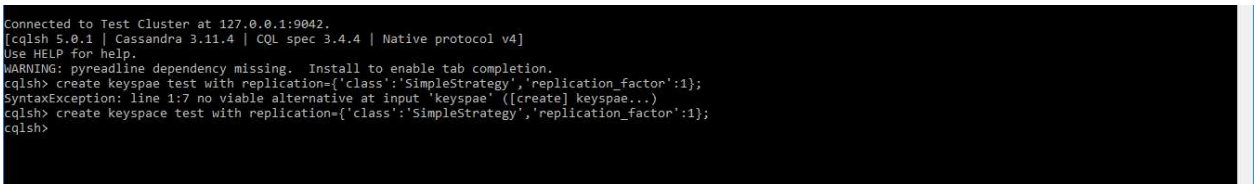
ICP-7

Name : Anusha Jasti

Class ID : 10

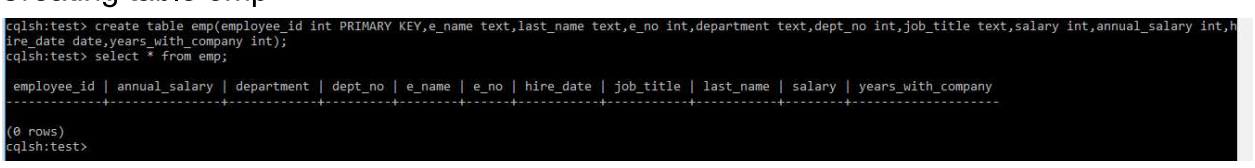
Cassandra

Create a Keyspace test



Use the keyspace test to perform queries

Creating table emp



Inserting values into the table emp

```
cqlsh:test> INSERT INTO emp(employee_id,e_name,last_name,e_no,department,dept_no,job_title,salary,annual_salary,hire_date,years_with_company) VALUES (1,'Anusha','Jast
i',303,'Engineering',30,'clerk',10000,120000,'1999-01-31',20);
cqlsh:test> select * from emp;

 employee_id | annual_salary | department  | dept_no | e_name | e_no | hire_date  | job_title | last_name | salary | years_with_company
-------------+---------------+-------------+---------+--------+------+------------+-----------+-----------+--------+--------------------
           1 |        120000 | Engineering |      30 | Anusha |  303 | 1999-01-31 |     clerk |     Jasti |  10000 |                 20

(1 rows)
cqlsh:test> INSERT INTO emp(employee_id,e_name,last_name,e_no,department,dept_no,job_title,salary,annual_salary,hire_date,years_with_company) VALUES (2,'Apoorva','Gee
ta',506,'Engineering',28,'clerk',1000,12000,'2018-12-17',1);
cqlsh:test> INSERT INTO emp(employee_id,e_name,last_name,e_no,department,dept_no,job_title,salary,annual_salary,hire_date,years_with_company) VALUES (3,'Bindu','Hima'
,405,'Marketing',38,'assistant',500,60000,'2018-12-16',1);
cqlsh:test> INSERT INTO emp(employee_id,e_name,last_name,e_no,department,dept_no,job_title,salary,annual_salary,hire_date,years_with_company) VALUES (4,'James','Howar
d',987,'Sales',34,'clerk',200,2400,'2017-06-24',2);
cqlsh:test> INSERT INTO emp(employee_id,e_name,last_name,e_no,department,dept_no,job_title,salary,annual_salary,hire_date,years_with_company) VALUES (5,'Jacob','Gonza
le',256,'Engineering',23,'clerk',300,3600,'2015-11-23',4);
```

The output of the table

```
cqlsh:test> select * from emp;

 employee_id | annual_salary | department  | dept_no | e_name | e_no | hire_date  | job_title | last_name | salary | years_with_company
-------------+---------------+-------------+---------+--------+------+------------+-----------+-----------+--------+--------------------
           5 |          3600 | Engineering |      23 |  Jacob |  256 | 2015-11-23 |     clerk |   Gonzale |    300 |                  4
           1 |        120000 | Engineering |      30 | Anusha |  303 | 1999-01-31 |     clerk |     Jasti |  10000 |                 20
           2 |         12000 | Engineering |      28 | Apoorva |  506 | 2018-12-17 |     clerk |     Geeta |   1000 |                  1
           4 |          2400 |       Sales |      34 |  James |  987 | 2017-06-24 |     clerk |    Howard |    200 |                  2
           3 |         60000 |   Marketing |      38 |  Bindu |  405 | 2018-12-16 | assistant |      Hima |    500 |                  1

(5 rows)
cqlsh:test>
```

The following are the queries performed on the table emp.

1. Query to display the e_no,e_name,job_title,hire_date.

```
cqlsh:test> select e_no,e_name,job_title,hire_date from emp;

 e_no | e_name  | job_title | hire_date
------+---------+-----------+------------
  256 |   Jacob |     clerk | 2015-11-23
  303 |  Anusha |     clerk | 1999-01-31
  506 | Apoorva |     clerk | 2018-12-17
  987 |   James |     clerk | 2017-06-24
  405 |   Bindu | assistant | 2018-12-16

(5 rows)
cqlsh:test>
```

2. Query to display name and salary of employee who are clerks.

```
cqlsh:test> select e_name,salary from emp WHERE job_title='clerk' ALLOW FILTERING;

 e_name  | salary
---------+--------
   Jacob |    300
  Anusha |  10000
 Apoorva |   1000
   James |    200

(4 rows)
cqlsh:test>
```

3. Query to display name,job,salary of employees who joined on ‘Jan 31 1999’.

```
cqlsh:test> select e_name,job_title,salary FROM emp WHERE hiredate='1999-01-31' ALLOW FILTERING;
InvalidRequest: Error from server: code=2200 [Invalid query] message="Undefined column name hiredate"
cqlsh:test> select e_name,job_title,salary FROM emp WHERE hire_date='1999-01-31' ALLOW FILTERING;

 e_name | job_title | salary
--------+-----------+--------
 Anusha |     clerk |  10000

(1 rows)
cqlsh:test>
```

4. Query to display name and annual salary of all employees.

```
cqlsh:test> select e_name,annual_salary FROM emp;

 e_name  | annual_salary
---------+---------------
   Jacob |          3600
  Anusha |        120000
 Apoorva |         12000
   James |          2400
   Bindu |         60000

(5 rows)
cqlsh:test>
```

5. Query to display name,dept no for the departments having dept no greater than 20.

```
cqlsh:test> select department,dept_no FROM emp WHERE dept_no>=20 ALLOW FILTERING;

 department  | dept_no
-------------+---------
 Engineering |      23
 Engineering |      30
 Engineering |      28
       Sales |      34
   Marketing |      38

(5 rows)
cqlsh:test>
```

Bonus :

I have choosen the dataset books for the bonus part.

- Creating the table books and importing data from csv file.

```
cqlsh:test> CREATE TABLE books (book_id int,category text,title text,number_pages int,publisher text,pub_date timestamp,author text,PRIMARY KEY(publisher, pub_date,bo
ok_id));
cqlsh:test> COPY books (book_id, category, title, author, publisher, pub_date, number_pages) FROM 'C:\Users\kite\Downloads\Data\books\book_entries.csv' WITH HEADER=Tr
ue AND DELIMITER='|';
Using 7 child processes

Starting copy of test.books with columns [book_id, category, title, author, publisher, pub_date, number_pages].
PPProcess ImportProcess-44:    5 rows/s; Avg. rate:       5 rows/s
PTProcess ImportProcess-41:
rocess ImportProcess-46:
rocess ImportProcess-43:
Process ImportProcess-47:
Trocess ImportProcess-45:
```

```
cqlsh:test> select * from books;

 publisher | pub_date                        | book_id | author         | category        | number_pages | title
-----------+---------------------------------+---------+----------------+-----------------+--------------+---------------------
       TOR | 1985-02-14 18:00:00.000000+0000 |       3 | Isaac Asimover | science fiction |          550 |          Foundation
       TOR | 1999-04-21 18:00:00.000000+0000 |       4 |   David Weaver |         fantasy |          600 |        On Your Honor
   Penguin | 2013-08-11 18:00:00.000000+0000 |       1 | Dorothy Sawyer |         mystery |          500 |      Murder By Alibi
   Penguin | 2014-09-29 18:00:00.000000+0000 |       2 |      Tom Jones |         mystery |          500 | Murder at the School

(4 rows)
```

- Performing few queries on the table books.

1. Query to display author,book id whose category is fantasy.

```
cqlsh:test> select author,book_id FROM books WHERE category='fantasy' ALLOW FILTERING;

 author       | book_id
--------------+---------
 David Weaver |       4

(1 rows)
cqlsh:test>
```

2. Query to display the maximum number of pages in all the books.

```
cqlsh:test> select max(number_pages) FROM books;

 system.max(number_pages)
--------------------------
                      600

(1 rows)

Warnings :
Aggregation query used without partition key
```

3. Query to display author,book id,publisher whose number of pages=500.

```
cqlsh:test> select author,book_id,publisher FROM books WHERE number_pages = 500 ALLOW FILTERING;

 author         | book_id | publisher
----------------+---------+-----------
 Dorothy Sawyer |       1 |   Penguin
      Tom Jones |       2 |   Penguin

(2 rows)
cqlsh:test>
```

4. Query to display min number of pages in all the books.

```
cqlsh:test> select min(number_pages) FROM books;

 system.min(number_pages)
--------------------------
                      500

(1 rows)

Warnings :
Aggregation query used without partition key

cqlsh:test>
```

5. Query to display author,category whose publisher is Penguin.

```
cqlsh:test> select author,category FROM books WHERE publisher='Penguin' ALLOW FILTERING;

 author         | category
----------------+----------
 Dorothy Sawyer |  mystery
      Tom Jones |  mystery

(2 rows)
cqlsh:test>
```