Task No: 6 Date:	Implement CURD operations on Casandra **Tools:** Casandra, Docker desktop	CO3

AIM:

To install and configure the apache Cassandra in windows operating system, and to perform the CURD operations.

PROCEDURE:

1. Install Docker desktop, Java 8 or latest version, and Python
2. Visit the Apache Cassandra website at https://cassandra.apache.org/download/ to download the latest stable release of Cassandra.
3. Start the Docker engine and type the following commands in the command prompt
4. Check the Docker Vesion

   ```
   docker --version
   ```

5. Creating a Cluster of two Cassandra nodes. Name the nodes Cassandra-1 and Cassandra-2

   ```
   docker run --name cassandra-1 -d cassandra:latest
   ```

6. Run the second Container, Cassandra-2, and link it to Cassandra-1.

   ```
   docker run --name cassandra-2 -d --link cassandra-1:cassandra cassandra:latest
   ```

7. Check the status of our node using the nodetool utility.

   ```
   docker exec -it cassandra-1 nodetool status
   docker exec -it cassandra-2 nodetool status
   ```

8. Use a Command Line Interface (CQLSH) to interact with the Cluster to create databases and tables using bash command

   ```
   docker exec -it cassandra-1 bash -c 'cqlsh'
   ```

9. Create the keyspace and tables, then execute the CURD operations – create, update, read, delete.
10. Execute various no-sql queries and find the results.
11. Stop the cassandra server.

    ```
    docker stop cassandra-1
    ```

    ```
    docker stop cassandra-2
    ```

Syntax:

Create Keyspace:

A keyspace is like RDBMS database which contains column families, indexes, user defined types, data center awareness, strategy used in keyspace, replication factor, etc.

```
CREATE KEYSPACE <identifier> WITH <properties>
```

Or

```
Create keyspace KeyspaceName with replicaton={'class':strategy name,

'replication_factor': No of replications on different nodes}
```

Alter Keyspace:

```
ALTER KEYSPACE <identifier> WITH <properties>
```

Or

```
ALTER KEYSPACE "KeySpace Name"

WITH replication = {'class': 'Strategy name', 'replication_factor' : 'No.Of  replicas'};
```

Drop Keyspace:

```
DROP  keyspace KeyspaceName ;
```

Create Table Operation:

Syntax of Create Operation-

```
CREATE (TABLE | COLUMNFAMILY) <tablename>

('<column-definition>' , '<column-definition>')

(WITH <option> AND <option>)
```

CRUD Operatons:

CREATE KEYSPACE test_cassandra WITH replication = {'class':'SimpleStrategy' , 'replication_factor' : 1};

Use test_cassandra ;

```
CREATE TABLE student(

   student_id int PRIMARY KEY,

   student_name text,
   student_city text,
   student_fees varint,
   student_phone varint
   );
```

Create Data:

```
INSERT INTO student (student_id, student_fees, student_name)
VALUES(1,5000, 'Ajeet');
INSERT INTO student (student_id, student_fees, student_name)
VALUES(2,3000, 'Kanchan');
INSERT INTO student (student_id, student_fees, student_name)
VALUES(3, 2000, 'Shivani');
```

READ Data

```
SELECT  * FROM student;
SELECT * FROM student WHERE student_id=2;
```

Update Data:

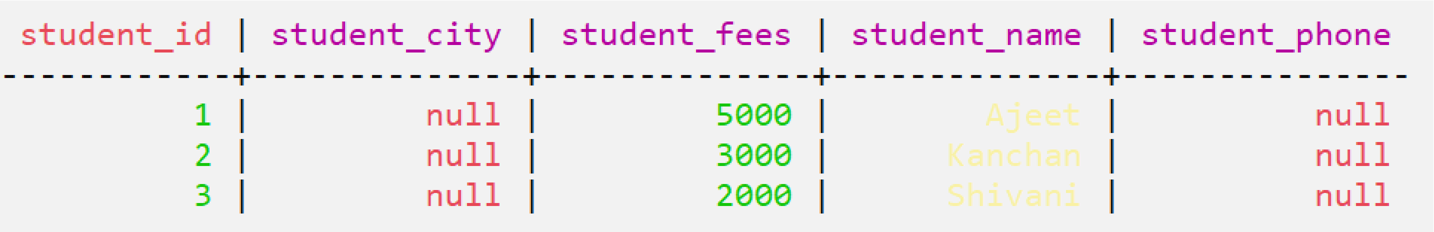
```
UPDATE student SET student_fees=10000,student_name='Rahul'
WHERE student_id=2;
```

DELETE Data:

```
DELETE student_fees FROM student WHERE student_id=3;
```

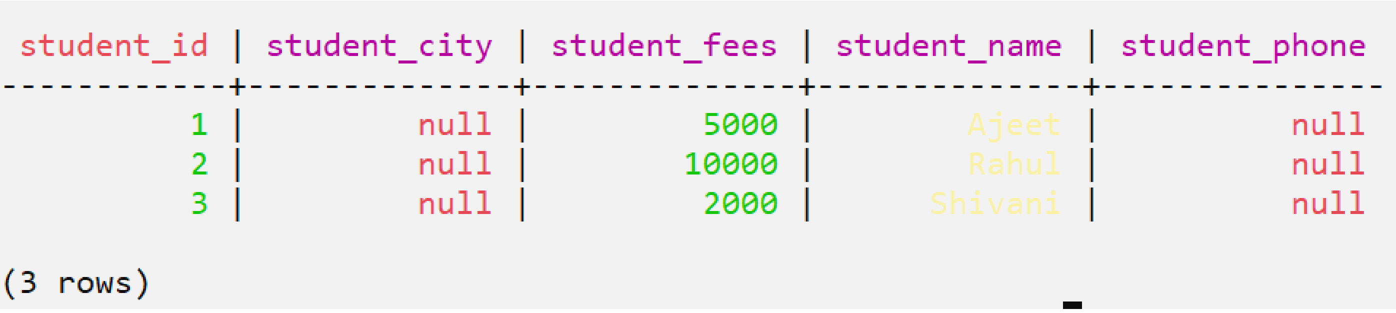
Output:

Create, Insert and Select(Read) Query Output:



```
 student_id | student_city | student_fees | student_name | student_phone
------------+--------------+--------------+--------------+---------------
          1 |         null |         5000 |        Ajeet |          null
          2 |         null |         3000 |      Kanchan |          null
          3 |         null |         2000 |      Shivani |          null
```

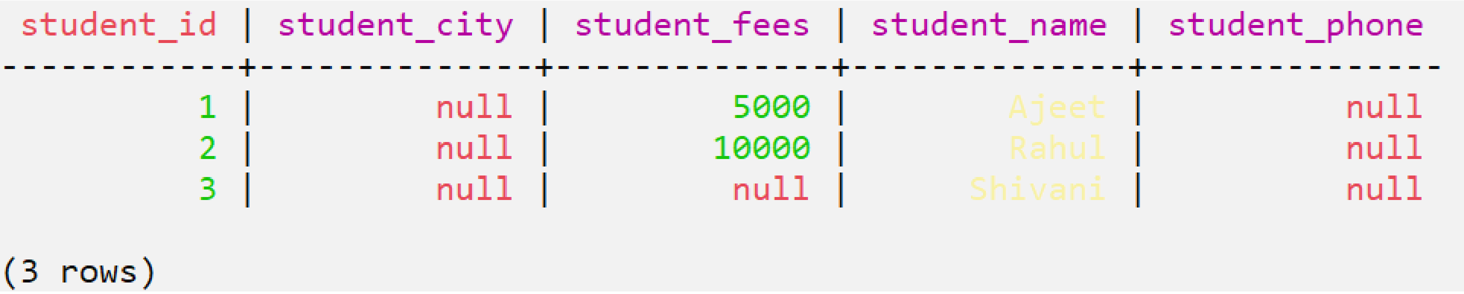
Update Query Output



```
 student_id | student_city | student_fees | student_name | student_phone
------------+--------------+--------------+--------------+---------------
          1 |         null |         5000 |        Ajeet |          null
          2 |         null |        10000 |        Rahul |          null
          3 |         null |         2000 |      Shivani |          null

(3 rows)
```

Delete Query Output:

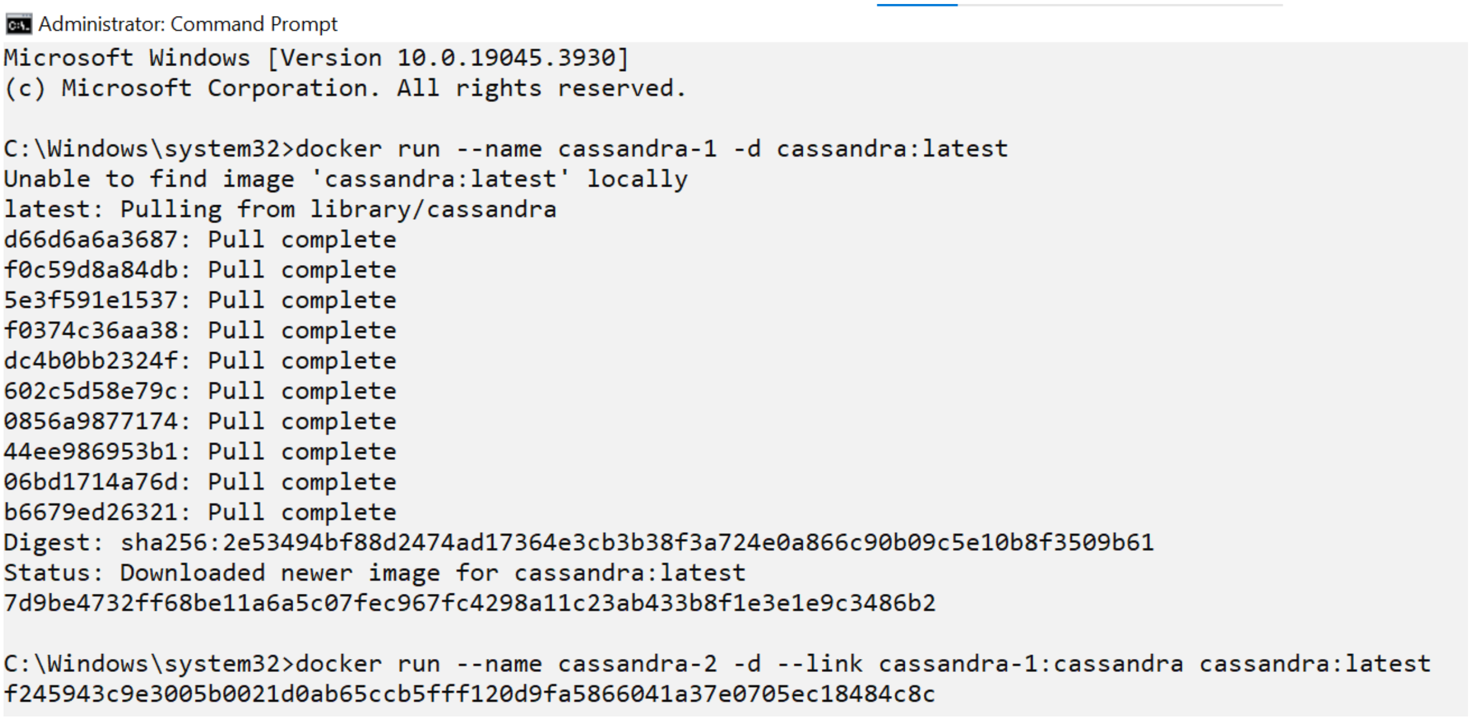


```
 student_id | student_city | student_fees | student_name | student_phone
------------+--------------+--------------+--------------+---------------
          1 |         null |         5000 |        Ajeet |          null
          2 |         null |        10000 |        Rahul |          null
          3 |         null |         null |      Shivani |          null

(3 rows)
```

Result:

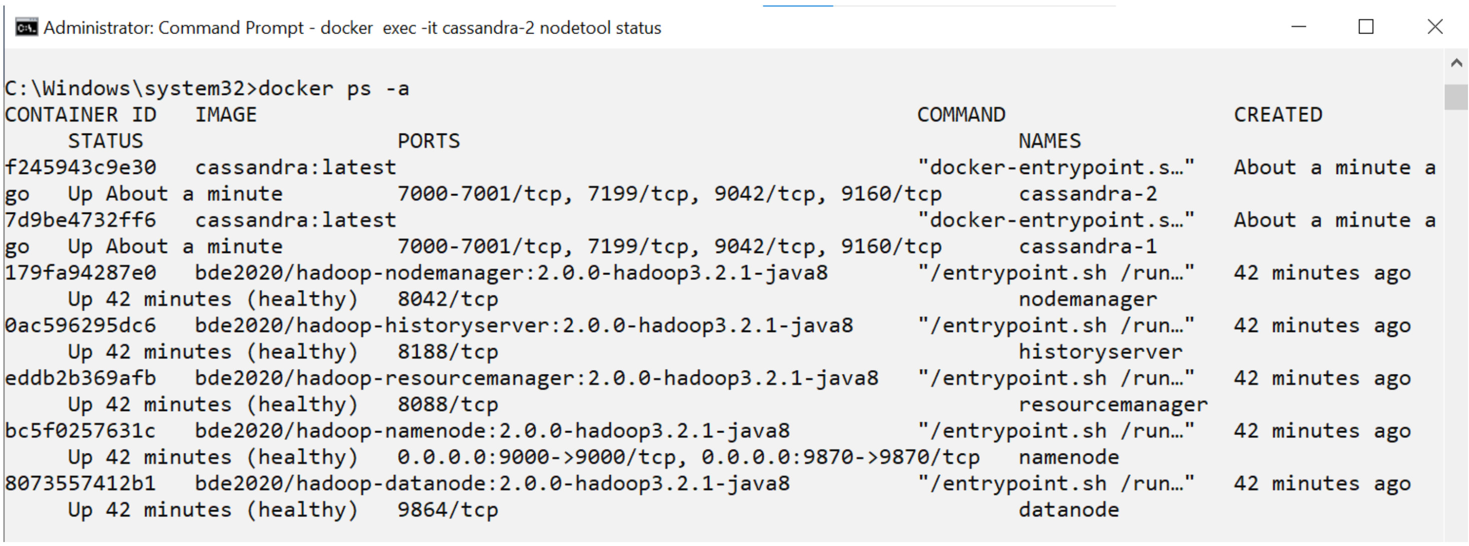
Thus the installation and configuration of Cassandra and its CURD operation are executed successfully.



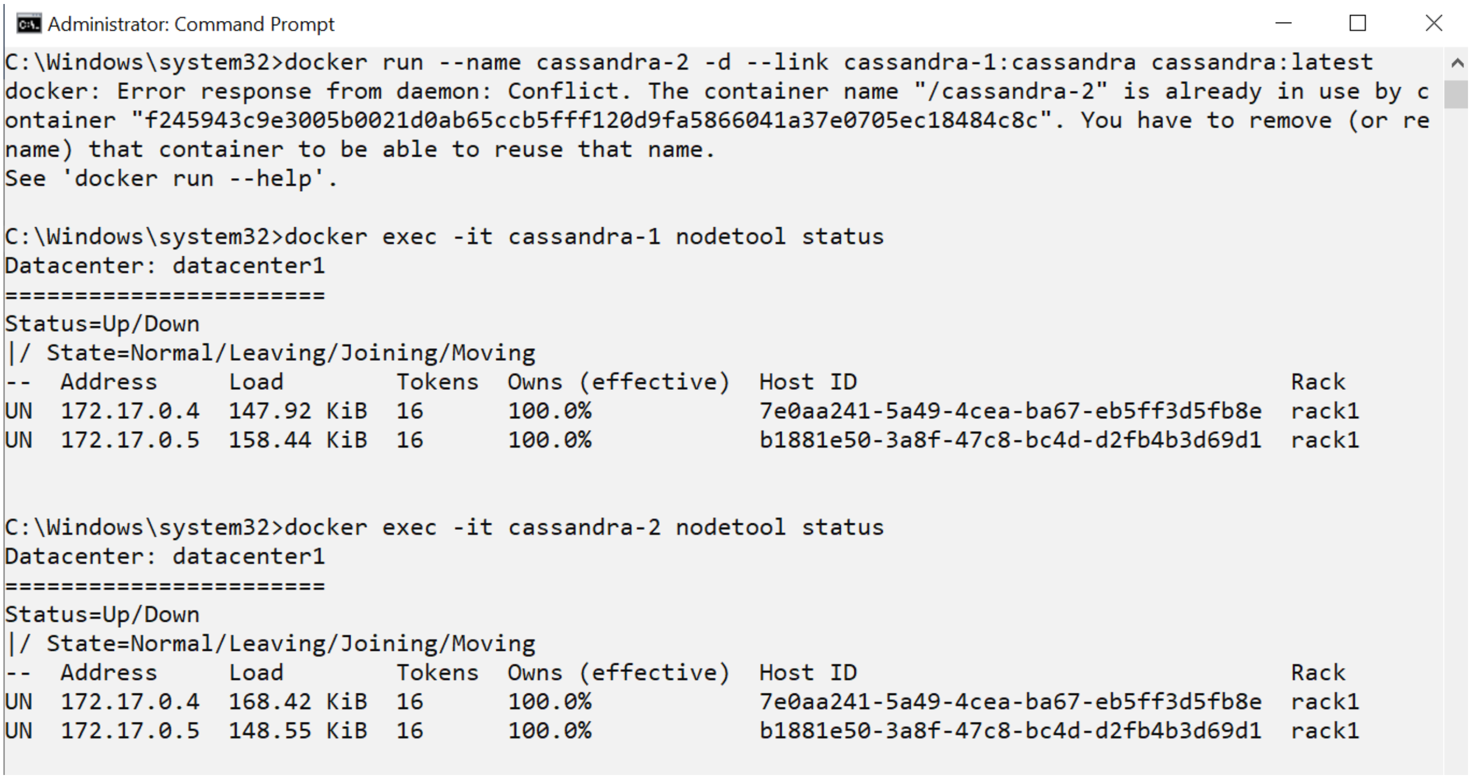
```
Microsoft Windows [Version 10.0.19045.3930]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>docker run --name cassandra-1 -d cassandra:latest
Unable to find image 'cassandra:latest' locally
latest: Pulling from library/cassandra
d66d6a6a3687: Pull complete
f0c59d8a84db: Pull complete
5e3f591e1537: Pull complete
f0374c36aa38: Pull complete
dc4b0bb2324f: Pull complete
602c5d58e79c: Pull complete
0856a9877174: Pull complete
44ee986953b1: Pull complete
06bd1714a76d: Pull complete
b6679ed26321: Pull complete
Digest: sha256:2e53494bf88d2474ad17364e3cb3b38f3a724e0a866c90b09c5e10b8f3509b61
Status: Downloaded newer image for cassandra:latest
7d9be4732ff68be11a6a5c07fec967fc4298a11c23ab433b8f1e3e1e9c3486b2

C:\Windows\system32>docker run --name cassandra-2 -d --link cassandra-1:cassandra cassandra:latest
f245943c9e3005b0021d0ab65ccb5fff120d9fa5866041a37e0705ec18484c8c
```


Administrator: Command Prompt - docker exec -it cassandra-2 nodetool status

```
C:\Windows\system32>docker ps -a
CONTAINER ID   IMAGE                                                    COMMAND                  CREATED
     STATUS                    PORTS                                            NAMES
f245943c9e30   cassandra:latest                                         "docker-entrypoint.s…"   About a minute a
go     Up About a minute         7000-7001/tcp, 7199/tcp, 9042/tcp, 9160/tcp      cassandra-2
7d9be4732ff6   cassandra:latest                                         "docker-entrypoint.s…"   About a minute a
go     Up About a minute         7000-7001/tcp, 7199/tcp, 9042/tcp, 9160/tcp      cassandra-1
179fa94287e0   bde2020/hadoop-nodemanager:2.0.0-hadoop3.2.1-java8       "/entrypoint.sh /run…"   42 minutes ago
     Up 42 minutes (healthy)   8042/tcp                                         nodemanager
0ac596295dc6   bde2020/hadoop-historyserver:2.0.0-hadoop3.2.1-java8     "/entrypoint.sh /run…"   42 minutes ago
     Up 42 minutes (healthy)   8188/tcp                                         historyserver
eddb2b369afb   bde2020/hadoop-resourcemanager:2.0.0-hadoop3.2.1-java8   "/entrypoint.sh /run…"   42 minutes ago
     Up 42 minutes (healthy)   8088/tcp                                         resourcemanager
bc5f0257631c   bde2020/hadoop-namenode:2.0.0-hadoop3.2.1-java8          "/entrypoint.sh /run…"   42 minutes ago
     Up 42 minutes (healthy)   0.0.0.0:9000->9000/tcp, 0.0.0.0:9870->9870/tcp   namenode
8073557412b1   bde2020/hadoop-datanode:2.0.0-hadoop3.2.1-java8          "/entrypoint.sh /run…"   42 minutes ago
     Up 42 minutes (healthy)   9864/tcp                                         datanode
```

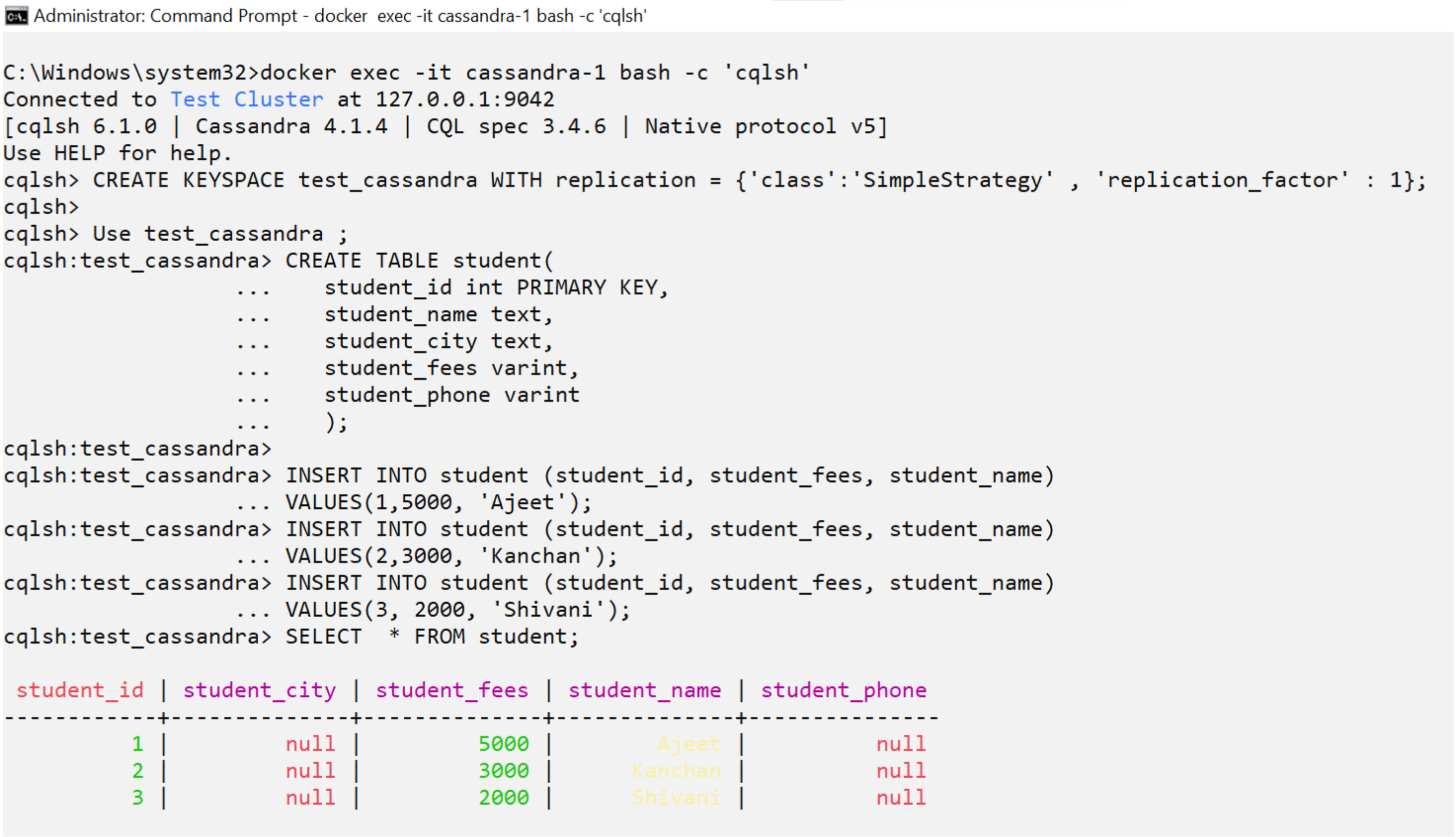


```
C:\Windows\system32>docker run --name cassandra-2 -d --link cassandra-1:cassandra cassandra:latest
docker: Error response from daemon: Conflict. The container name "/cassandra-2" is already in use by c
ontainer "f245943c9e3005b0021d0ab65ccb5fff120d9fa5866041a37e0705ec18484c8c". You have to remove (or re
name) that container to be able to reuse that name.
See 'docker run --help'.

C:\Windows\system32>docker exec -it cassandra-1 nodetool status
Datacenter: datacenter1
=======================
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
--  Address     Load        Tokens  Owns (effective)  Host ID                               Rack
UN  172.17.0.4  147.92 KiB  16      100.0%            7e0aa241-5a49-4cea-ba67-eb5ff3d5fb8e  rack1
UN  172.17.0.5  158.44 KiB  16      100.0%            b1881e50-3a8f-47c8-bc4d-d2fb4b3d69d1  rack1


C:\Windows\system32>docker exec -it cassandra-2 nodetool status
Datacenter: datacenter1
=======================
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
--  Address     Load        Tokens  Owns (effective)  Host ID                               Rack
UN  172.17.0.4  168.42 KiB  16      100.0%            7e0aa241-5a49-4cea-ba67-eb5ff3d5fb8e  rack1
UN  172.17.0.5  148.55 KiB  16      100.0%            b1881e50-3a8f-47c8-bc4d-d2fb4b3d69d1  rack1
```

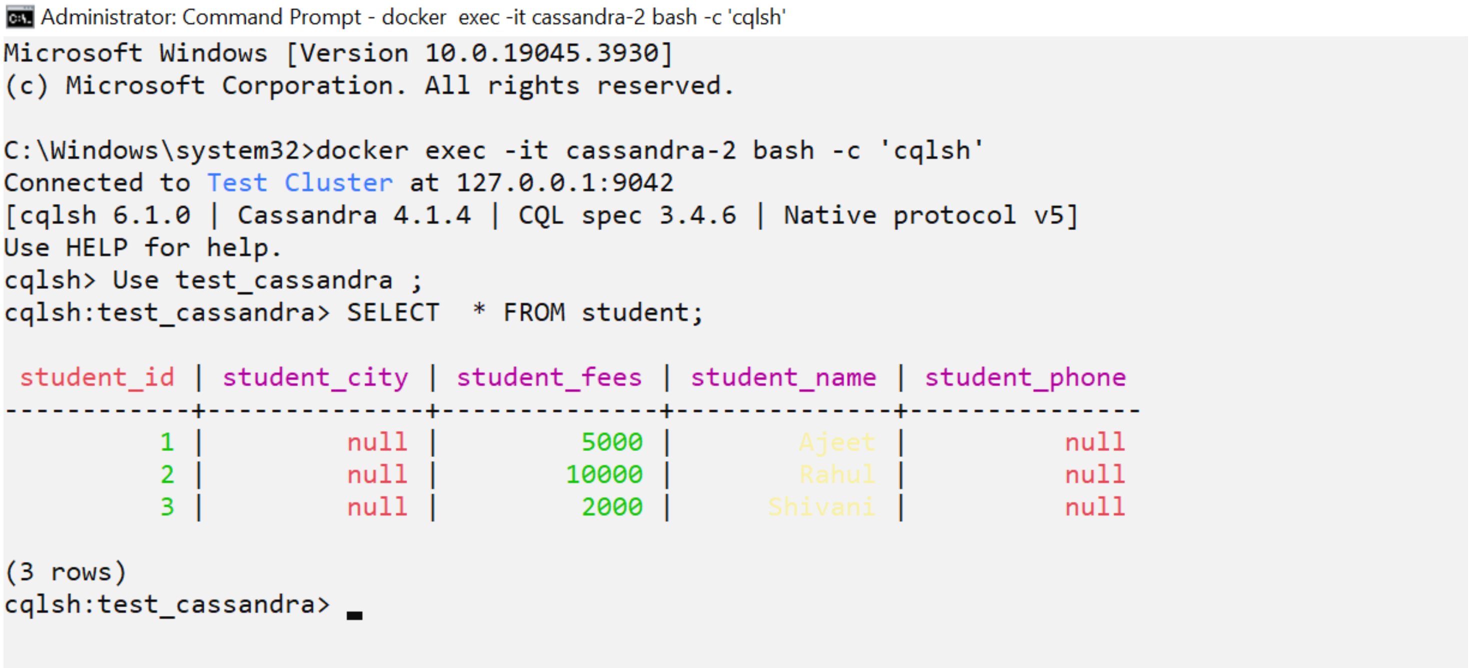
Node Cassandra-1:


Administrator: Command Prompt - docker exec -it cassandra-1 bash -c 'cqlsh'

```
C:\Windows\system32>docker exec -it cassandra-1 bash -c 'cqlsh'
Connected to Test Cluster at 127.0.0.1:9042
[cqlsh 6.1.0 | Cassandra 4.1.4 | CQL spec 3.4.6 | Native protocol v5]
Use HELP for help.
cqlsh> CREATE KEYSPACE test_cassandra WITH replication = {'class':'SimpleStrategy' , 'replication_factor' : 1};
cqlsh>
cqlsh> Use test_cassandra ;
cqlsh:test_cassandra> CREATE TABLE student(
                  ...    student_id int PRIMARY KEY,
                  ...    student_name text,
                  ...    student_city text,
                  ...    student_fees varint,
                  ...    student_phone varint
                  ...    );
cqlsh:test_cassandra>
cqlsh:test_cassandra> INSERT INTO student (student_id, student_fees, student_name)
                  ... VALUES(1,5000, 'Ajeet');
cqlsh:test_cassandra> INSERT INTO student (student_id, student_fees, student_name)
                  ... VALUES(2,3000, 'Kanchan');
cqlsh:test_cassandra> INSERT INTO student (student_id, student_fees, student_name)
                  ... VALUES(3, 2000, 'Shivani');
cqlsh:test_cassandra> SELECT  * FROM student;

 student_id | student_city | student_fees | student_name | student_phone
------------+--------------+--------------+--------------+---------------
          1 |         null |         5000 |        Ajeet |          null
          2 |         null |         3000 |      Kanchan |          null
          3 |         null |         2000 |      Shivani |          null
```

Node Cassandra-2:



Administrator: Command Prompt - docker exec -it cassandra-2 bash -c 'cqlsh'

```
Microsoft Windows [Version 10.0.19045.3930]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>docker exec -it cassandra-2 bash -c 'cqlsh'
Connected to Test Cluster at 127.0.0.1:9042
[cqlsh 6.1.0 | Cassandra 4.1.4 | CQL spec 3.4.6 | Native protocol v5]
Use HELP for help.
cqlsh> Use test_cassandra ;
cqlsh:test_cassandra> SELECT  * FROM student;

 student_id | student_city | student_fees | student_name | student_phone
------------+--------------+--------------+--------------+---------------
          1 |         null |         5000 |        Ajeet |          null
          2 |         null |        10000 |        Rahul |          null
          3 |         null |         2000 |      Shivani |          null

(3 rows)
cqlsh:test_cassandra> 
```