CQL Basics Operations

```
cqlsh
```

Keyspace

Create a keyspace

- similar to Database in SQL

```
CREATE KEYSPACE admatic WITH replication = {'class':'SimpleStrategy', 'r
eplication_factor' : 3};

CREATE KEYSPACE admatic WITH replication = {'class' : 'NetworkTopologySt
rategy', 'dc1' : 2, 'dc2' : 2};
```

Using the keyspace that was created

```
use admatic;
```

Lising all keyspaces

```
DESCRIBE keyspaces;
```

Alter a keyspace

```
ALTER KEYSPACE admatic WITH replication = {'class':'NetworkTopologyStrat
egy'};
```

Drop keyspaces

```
DROP KEYSPACE admatic;
```

CRUD Operations

1. Creating a table

Syntax

```
CREATE (TABLE | COLUMNFAMILY) <tablename>
('<column-definition>' , '<column-definition>')
(WITH <option> AND <option>)
```

- `COLUMNFAMILY` syntax is supported as an alias for `TABLE` (for historical reasons)

Examples

```
CREATE TABLE emp(
emp_id int PRIMARY KEY,
emp_name text,
emp_city text,
emp_sal varint,
emp_phone varint
);
```

```
CREATE TABLE monkeySpecies (
    species text PRIMARY KEY,
    common_name text,
    population varint,
    average_size int
) WITH comment='Important biological records'
   AND read_repair_chance = 1.0;
```

```
CREATE COLUMNFAMILY timeline (
    userid uuid,
    posted_month int,
    posted_time uuid,
    body text,
    posted_by text,
    PRIMARY KEY (userid, posted_month, posted_time)
) WITH compaction = { 'class' : 'LeveledCompactionStrategy' };
```

Output

```
select * from emp;

emp_id | emp_city | emp_name | emp_phone | emp_sal
```

```
--------+----------+----------+-----------+---------

(0 rows)
```

2. Capture query outputs to a text file

Syntax

```
CAPTURE '<Output_File_Path>'
```

Example

```
CAPTURE '/home/hadoop/Cassandra_capture.txt'

Now capturing query output to '/home/hadoop/Cassandra_capture.txt'.
```

```
select * from emp;
```

Output

```
cat Cassandra_capture.txt

 emp_id | emp_city | emp_name | emp_phone | emp_sal
--------+----------+----------+-----------+---------

(0 rows)
```

3. Display the table

```
select * from emp;
```

4. Inserting data into table

Example

```
INSERT INTO emp (emp_id, emp_name, emp_city, emp_phone, emp_sal) VALUES(
1,'ram', 'Hyderabad', 9848022338, 50000);
INSERT INTO emp (emp_id, emp_name, emp_city, emp_phone, emp_sal) VALUES(
2,'robin', 'Hyderabad', 9848022339, 40000);
INSERT INTO emp (emp_id, emp_name, emp_city, emp_phone, emp_sal) VALUES(
3,'rahman', 'Chennai', 9848022330, 45000);
```

Output

```
SELECT * FROM emp;

emp_id |  emp_city | emp_name |  emp_phone | emp_sal
--------+-----------+----------+------------+---------
      1 | Hyderabad |      ram | 9848022338 | 50000
      2 | Hyderabad |    robin | 9848022339 | 40000
      3 |   Chennai |   rahman | 9848022330 | 45000

(3 rows)
```

5. Update Data

Example

```
UPDATE emp SET emp_city='Delhi',emp_sal=50000 WHERE emp_id=2;
```

Output

```
select * from emp;

emp_id |  emp_city | emp_name |  emp_phone | emp_sal
--------+-----------+----------+------------+---------
      1 | Hyderabad |      ram | 9848022338 | 50000
      2 |     Delhi |    robin | 9848022339 | 50000
      3 |   Chennai |   rahman | 9848022330 | 45000

(3 rows)
```

6. Deleting a value from row

Example

```
DELETE emp_sal FROM emp WHERE emp_id=3;
```

Output

```
select * from emp;

emp_id |  emp_city | emp_name |  emp_phone | emp_sal
--------+-----------+----------+------------+---------
```

```
      1 | Hyderabad |      ram | 9848022338 | 50000
      2 |     Delhi |    robin | 9848022339 | 50000
      3 |   Chennai |   rahman | 9848022330 | null
(3 rows)
```

7. Deleting an entire row

Example

```
DELETE FROM emp WHERE emp_id=3;
```

Output

```
select * from emp;

emp_id |  emp_city | emp_name |  emp_phone | emp_sal
--------+-----------+----------+------------+---------
      1 | Hyderabad |      ram | 9848022338 | 50000
      2 |     Delhi |    robin | 9848022339 | 50000

(2 rows)
```