1. STUDENT DATABASE

```
bmsce@bmsce-HP-Elite-Tower-600-G9-Desktop-PC:~$ cqlsh
Connected to Test Cluster at 127.0.0.1:9042
[cqlsh 6.1.0 | Cassandra 4.1.8 | CQL spec 3.4.6 | Native protocol v5]
Use HELP for help.
cqlsh> DROP KEYSPACE IF EXISTS College;
cqlsh> CREATE KEYSPACE College
   ... WITH replication = {'class':'SimpleStrategy','replication_factor':1};
cqlsh> USE College;
cqlsh:college> CREATE TABLE Student_Info (
          ...     USN text PRIMARY KEY,
          ...     Name text,
          ...     Branch text,
          ...     Semester int,
          ...     CGPA decimal
          ... );
```

```
cqlsh:college> INSERT INTO Student_Info (USN, Name, Branch, Semester, CGPA)
          ... VALUES ('1','Amit','CSE',5,8.5);
cqlsh:college>
cqlsh:college> INSERT INTO Student_Info (USN, Name, Branch, Semester, CGPA)
          ... VALUES ('2','Priya','ECE',4,8.8);
cqlsh:college>
cqlsh:college> INSERT INTO Student_Info (USN, Name, Branch, Semester, CGPA)
          ... VALUES ('3','Rahul','ME',6,7.9);
cqlsh:college>
cqlsh:college> UPDATE Student_Info
          ... SET CGPA = 9.0
          ... WHERE USN = '1';
cqlsh:college>
cqlsh:college> SELECT * FROM Student_Info
          ... ;

 usn | branch | cgpa | name  | semester
-----+--------+------+-------+----------
   3 |     ME |  7.9 | Rahul |        6
   2 |    ECE |  8.8 | Priya |        4
   1 |    CSE |  9.0 |  Amit |        5

(3 rows)
cqlsh:college>
```

2. BOOK MANAGEMENT SYSTEM

```
bmsce@bmsce-HP-Elite-Tower-600-G9-Desktop-PC:~$ cqlsh
Connected to Test Cluster at 127.0.0.1:9042
[cqlsh 6.1.0 | Cassandra 4.1.8 | CQL spec 3.4.6 | Native protocol v5]
Use HELP for help.
cqlsh> DROP KEYSPACE IF EXISTS Library;
cqlsh> CREATE KEYSPACE Library
   ... WITH replication = {'class':'SimpleStrategy','replication_factor':1};
cqlsh> USE Library;
cqlsh:library>
cqlsh:library> CREATE TABLE Book_Info (
           ...     Book_Id text PRIMARY KEY,
           ...     Title text,
           ...     Author text,
           ...     Price decimal,
           ...     Category text
           ... );
cqlsh:library> BEGIN BATCH
           ... INSERT INTO Book_Info (Book_Id, Title, Author, Price, Category)
           ... VALUES ('B1','DBMS','Korth',550,'Education');
           ...
           ... INSERT INTO Book_Info (Book_Id, Title, Author, Price, Category)
           ... VALUES ('B2','OS','Galvin',620,'Education');
           ...
           ... INSERT INTO Book_Info (Book_Id, Title, Author, Price, Category)
           ... VALUES ('B3','Alchemist','Paulo Coelho',300,'Fiction');
           ... APPLY BATCH;
cqlsh:library> UPDATE Book_Info
           ... SET Price = 350
           ... WHERE Book_Id = 'B3';
cqlsh:library> DELETE FROM Book_Info
           ... WHERE Book_Id = 'B2';
cqlsh:library>
cqlsh:library> SELECT * FROM Book_Info;

 book_id | author       | category  | price | title
---------+--------------+-----------+-------+-----------
      B1 |        Korth | Education |   550 |      DBMS
      B3 | Paulo Coelho |   Fiction |   350 | Alchemist

(2 rows)
cqlsh:library>
```

3. CUSTOMER ORDERS

```
bmsce@bmsce-HP-Elite-Tower-600-G9-Desktop-PC:~$ cqlsh
Connected to Test Cluster at 127.0.0.1:9042
[cqlsh 6.1.0 | Cassandra 4.1.8 | CQL spec 3.4.6 | Native protocol v5]
Use HELP for help.
cqlsh> DROP KEYSPACE IF EXISTS Ecommerce;
cqlsh> CREATE KEYSPACE Ecommerce
   ... WITH replication = {'class':'SimpleStrategy','replication_factor':1};
cqlsh>
cqlsh> USE Ecommerce;
cqlsh:ecommerce> CREATE TABLE Customer_Orders (
            ...     Order_Id text PRIMARY KEY,
            ...     Customer_Name text,
            ...     Product_Name text,
            ...     Quantity int,
            ...     Order_Date date
            ... );
cqlsh:ecommerce>
cqlsh:ecommerce> INSERT INTO Customer_Orders (Order_Id, Customer_Name, Product_Name, Quantity, Order_Date)
            ... VALUES ('O1','Ravi','Laptop',1,'2026-02-20');
cqlsh:ecommerce>
cqlsh:ecommerce> INSERT INTO Customer_Orders (Order_Id, Customer_Name, Product_Name, Quantity, Order_Date)
            ... VALUES ('O2','Sneha','Mobile',2,'2026-02-21');
cqlsh:ecommerce>
cqlsh:ecommerce> INSERT INTO Customer_Orders (Order_Id, Customer_Name, Product_Name, Quantity, Order_Date)
            ... VALUES ('O3','Arjun','Headphones',3,'2026-02-22');
cqlsh:ecommerce>
cqlsh:ecommerce> ALTER TABLE Customer_Orders
            ... ADD Delivery_Status text;
cqlsh:ecommerce> UPDATE Customer_Orders
            ... SET Delivery_Status = 'Shipped'
            ... WHERE Order_Id = 'O2';
cqlsh:ecommerce>
cqlsh:ecommerce> SELECT * FROM Customer_Orders;

 order_id | customer_name | delivery_status | order_date | product_name | quantity
----------+---------------+-----------------+------------+--------------+----------
       O1 |          Ravi |            null | 2026-02-20 |       Laptop |        1
       O2 |         Sneha |         Shipped | 2026-02-21 |       Mobile |        2
       O3 |         Arjun |            null | 2026-02-22 |   Headphones |        3

(3 rows)
cqlsh:ecommerce>
```

4. EMPLOYEE SKILLS

```
cqlsh> CREATE KEYSPACE Company
   ... WITH replication = {'class':'SimpleStrategy','replication_factor':1};
cqlsh> USE Company;
cqlsh:company>
cqlsh:company> CREATE TABLE Employee_Skills (
          ...     Emp_Id text PRIMARY KEY,
          ...     Emp_Name text,
          ...     Skills set<text>
          ... );
cqlsh:company>
cqlsh:company> INSERT INTO Employee_Skills (Emp_Id, Emp_Name, Skills)
          ... VALUES ('E1','Amit',{'Java','Python'});
cqlsh:company>
cqlsh:company> INSERT INTO Employee_Skills (Emp_Id, Emp_Name, Skills)
          ... VALUES ('E2','Priya',{'C++','AWS'});
cqlsh:company>
cqlsh:company> UPDATE Employee_Skills
          ... SET Skills = Skills + {'Docker'}
          ... WHERE Emp_Id = 'E1';
cqlsh:company>
cqlsh:company> SELECT Emp_Name, Skills FROM Employee_Skills;

 emp_name | skills
----------+-----------------------------
     Amit | {'Docker', 'Java', 'Python'}
    Priya |               {'AWS', 'C++'}

(2 rows)
cqlsh:company>
```

5. PRODUCT EXPIRY USING TTL

```
bmsce@bmsce-HP-Elite-Tower-600-G9-Desktop-PC:~$ cqlsh
Connected to Test Cluster at 127.0.0.1:9042
[cqlsh 6.1.0 | Cassandra 4.1.8 | CQL spec 3.4.6 | Native protocol v5]
Use HELP for help.
cqlsh> DROP KEYSPACE IF EXISTS Store;
cqlsh> CREATE KEYSPACE Store
   ... WITH replication = {'class':'SimpleStrategy','replication_factor':1};
cqlsh>
cqlsh> USE Store;
cqlsh:store>
cqlsh:store> CREATE TABLE Product_Info (
         ...     Product_Id text PRIMARY KEY,
         ...     Product_Name text,
         ...     Price decimal,
         ...     Category text
         ... );
cqlsh:store> INSERT INTO Product_Info (Product_Id, Product_Name, Price, Category)
         ... VALUES ('P1','Milk',45,'Dairy')
         ... USING TTL 20;
cqlsh:store> SELECT * FROM Product_Info;

 product_id | category | price | product_name
------------+----------+-------+--------------
         P1 |    Dairy |    45 |         Milk

(1 rows)
cqlsh:store> SELECT * FROM Product_Info;
```

6. Online Course Registration

```
bmsce@bmsce-HP-Elite-Tower-600-G9-Desktop-PC:~$ cqlsh
Connected to Test Cluster at 127.0.0.1:9042
[cqlsh 6.1.0 | Cassandra 4.1.8 | CQL spec 3.4.6 | Native protocol v5]
Use HELP for help.
cqlsh> CREATE KEYSPACE Online_Courses
   ... WITH replication = {
   ...   'class': 'SimpleStrategy',
   ...   'replication_factor': 1
   ... };
AlreadyExists: Keyspace 'online_courses' already exists
cqlsh>
cqlsh> USE Online_Courses;
cqlsh:online_courses> CREATE TABLE Course_Registration (
                  ...     Reg_Id int PRIMARY KEY,
                  ...     Student_Name text,
                  ...     Course_Name text,
                  ...     Registration_Date date
                  ... );
AlreadyExists: Table 'online_courses.course_registration' already exists
cqlsh:online_courses> INSERT INTO Course_Registration (Reg_Id, Student_Name, Course_Name, Registration_Date)
                  ... VALUES (1, 'Asha', 'Machine Learning', '2026-02-20');
cqlsh:online_courses>
cqlsh:online_courses> INSERT INTO Course_Registration (Reg_Id, Student_Name, Course_Name, Registration_Date)
                  ... VALUES (2, 'Rahul', 'Blockchain', '2026-02-21');
cqlsh:online_courses>
cqlsh:online_courses> INSERT INTO Course_Registration (Reg_Id, Student_Name, Course_Name, Registration_Date)
                  ... VALUES (3, 'Meera', 'Cloud Computing', '2026-02-22');
cqlsh:online_courses> ALTER TABLE Course_Registration
                  ... ADD Payment_Status text;
cqlsh:online_courses> UPDATE Course_Registration
                  ... SET Payment_Status = 'Paid'
                  ... WHERE Reg_Id = 1;
cqlsh:online_courses> INSERT INTO Course_Registration (Reg_Id, Student_Name, Course_Name, Registration_Date, Payment_Status)
                  ... VALUES (4, 'Kiran', 'AI', '2026-02-23', 'Pending')
                  ... USING TTL 30;
cqlsh:online_courses> SELECT * FROM Course_Registration;

 reg_id | course_name      | payment_status | registration_date | student_name
--------+------------------+----------------+-------------------+--------------
      1 | Machine Learning |           Paid |        2026-02-20 |         Asha
      2 |       Blockchain |           null |        2026-02-21 |        Rahul
      4 |               AI |        Pending |        2026-02-23 |        Kiran
      3 |  Cloud Computing |           null |        2026-02-22 |        Meera

(4 rows)
cqlsh:online_courses> SELECT * FROM Course_Registration;

 reg_id | course_name      | payment_status | registration_date | student_name
--------+------------------+----------------+-------------------+--------------
      1 | Machine Learning |           Paid |        2026-02-20 |         Asha
      2 |       Blockchain |           null |        2026-02-21 |        Rahul
      3 |  Cloud Computing |           null |        2026-02-22 |        Meera

(3 rows)
cqlsh:online_courses>
```

7. Restaurant Orders

```
bmsce@bmsce-HP-Elite-Tower-600-G9-Desktop-PC: ~
bmsce@bmsce-HP-Elite-Tower-600-G9-Desktop-PC:~$ cqlsh
Connected to Test Cluster at 127.0.0.1:9042
[cqlsh 6.1.0 | Cassandra 4.1.8 | CQL spec 3.4.6 | Native protocol v5]
Use HELP for help.
cqlsh> CREATE KEYSPACE Restaurant
   ... WITH replication = {
   ...   'class': 'SimpleStrategy',
   ...   'replication_factor': 1
   ... };
cqlsh>
cqlsh> USE Restaurant;
cqlsh:restaurant> CREATE TABLE Orders (
              ...     Order_Id int PRIMARY KEY,
              ...     Customer_Name text,
              ...     Item_Name text,
              ...     Quantity int,
              ...     Bill_Amount decimal
              ... );
cqlsh:restaurant> BEGIN BATCH
              ... INSERT INTO Orders (Order_Id, Customer_Name, Item_Name, Quantity, Bill_Amount)
              ... VALUES (101, 'Anil', 'Pizza', 2, 500);
              ...
              ... INSERT INTO Orders (Order_Id, Customer_Name, Item_Name, Quantity, Bill_Amount)
              ... VALUES (102, 'Sneha', 'Burger', 1, 150);
              ...
              ... INSERT INTO Orders (Order_Id, Customer_Name, Item_Name, Quantity, Bill_Amount)
              ... VALUES (103, 'Vikram', 'Pasta', 3, 450);
              ... APPLY BATCH;
cqlsh:restaurant> UPDATE Orders
              ... SET Bill_Amount = 550
              ... WHERE Order_Id = 101;
cqlsh:restaurant> DELETE FROM Orders
              ... WHERE Order_Id = 102;
cqlsh:restaurant> SELECT * FROM Orders;

 order_id | bill_amount | customer_name | item_name | quantity
----------+-------------+---------------+-----------+----------
      101 |         550 |          Anil |     Pizza |        2
      103 |         450 |        Vikram |     Pasta |        3

(2 rows)
cqlsh:restaurant>
```

8. Movie Ratings System

```
bmsce@bmsce-HP-Elite-Tower-600-G9-Desktop-PC:~$ cqlsh
Connected to Test Cluster at 127.0.0.1:9042
[cqlsh 6.1.0 | Cassandra 4.1.8 | CQL spec 3.4.6 | Native protocol v5]
Use HELP for help.
cqlsh> CREATE KEYSPACE Movies
   ... WITH replication = {
   ...   'class': 'SimpleStrategy',
   ...   'replication_factor': 1
   ... };
cqlsh>
cqlsh> USE Movies;
cqlsh:movies> CREATE TABLE Movie_Ratings (
          ...     Movie_Id int PRIMARY KEY,
          ...     Movie_Name text,
          ...     Ratings map<text, int>
          ... );
cqlsh:movies> INSERT INTO Movie_Ratings (Movie_Id, Movie_Name, Ratings)
          ... VALUES (1, 'Inception', {'user1': 5, 'user2': 4});
cqlsh:movies>
cqlsh:movies> INSERT INTO Movie_Ratings (Movie_Id, Movie_Name, Ratings)
          ... VALUES (2, 'Interstellar', {'user1': 5, 'user3': 5});
cqlsh:movies> UPDATE Movie_Ratings
          ... SET Ratings['user4'] = 4
          ... WHERE Movie_Id = 1;
cqlsh:movies> UPDATE Movie_Ratings
          ... SET Ratings['user2'] = 5
          ... WHERE Movie_Id = 1;
cqlsh:movies> SELECT Ratings FROM Movie_Ratings
          ... WHERE Movie_Id = 1;

 ratings
-------------------------------------
 {'user1': 5, 'user2': 5, 'user4': 4}

(1 rows)
cqlsh:movies>
```

9. Shopping Cart System

```
bmsce@bmsce-HP-Elite-Tower-600-G9-Desktop-PC:~$ cqlsh
Connected to Test Cluster at 127.0.0.1:9042
[cqlsh 6.1.0 | Cassandra 4.1.8 | CQL spec 3.4.6 | Native protocol v5]
Use HELP for help.
cqlsh> CREATE KEYSPACE Shopping
   ... WITH replication = {
   ...   'class': 'SimpleStrategy',
   ...   'replication_factor': 1
   ... };
cqlsh>
cqlsh> USE Shopping;
cqlsh:shopping> CREATE TABLE Cart (
            ...     Cart_Id int PRIMARY KEY,
            ...     User_Name text,
            ...     Items list<text>
            ... );
cqlsh:shopping> INSERT INTO Cart (Cart_Id, User_Name, Items)
            ... VALUES (1, 'Ravi', ['Laptop', 'Mouse', 'Keyboard']);
cqlsh:shopping> UPDATE Cart
            ... SET Items = Items + ['Headphones']
            ... WHERE Cart_Id = 1;
cqlsh:shopping> UPDATE Cart
            ... SET Items = Items - ['Mouse']
            ... WHERE Cart_Id = 1;
cqlsh:shopping> UPDATE Cart
            ... USING TTL 60
            ... SET User_Name = 'Ravi'
            ... WHERE Cart_Id = 1;
cqlsh:shopping> SELECT * FROM Cart;

 cart_id | items                                    | user_name
---------+------------------------------------------+-----------
       1 | ['Laptop', 'Keyboard', 'Headphones'] |      Ravi

(1 rows)
cqlsh:shopping> SELECT * FROM Cart;

 cart_id | items                                    | user_name
---------+------------------------------------------+-----------
       1 | ['Laptop', 'Keyboard', 'Headphones'] |      null

(1 rows)
cqlsh:shopping>
```

10. Bank Account Balance

```
bmsce@bmsce-HP-Elite-Tower-600-G9-Desktop-PC:~$ cqlsh
Connected to Test Cluster at 127.0.0.1:9042
[cqlsh 6.1.0 | Cassandra 4.1.8 | CQL spec 3.4.6 | Native protocol v5]
Use HELP for help.
cqlsh> CREATE KEYSPACE Banking
   ... WITH replication = {
   ...   'class': 'SimpleStrategy',
   ...   'replication_factor': 1
   ... };
cqlsh>
cqlsh> USE Banking;
cqlsh:banking> CREATE TABLE Account_Transactions (
           ...     Account_No int PRIMARY KEY,
           ...     Balance counter
           ... );
cqlsh:banking> UPDATE Account_Transactions
           ... SET Balance = Balance + 1000
           ... WHERE Account_No = 12345;
cqlsh:banking>
cqlsh:banking> UPDATE Account_Transactions
           ... SET Balance = Balance + 500
           ... WHERE Account_No = 12345;
cqlsh:banking> UPDATE Account_Transactions
           ... SET Balance = Balance - 300
           ... WHERE Account_No = 12345;
cqlsh:banking> SELECT * FROM Account_Transactions;

 account_no | balance
------------+---------
      12345 |    1200

(1 rows)
cqlsh:banking>
```