



Basic CQL Commands

Connecting and Authentication

Connecting to Cassandra:

```
cqlsh <host> <port> --user=<username> --password=<password>
```

Example:

```
cqlsh 127.0.0.1 9042 --user=cassandra --password=cassandra
```

Without Authentication:

```
cqlsh <host> <port>
```

Example:

```
cqlsh 127.0.0.1 9042
```

Keyspace Operations

Create Keyspace:

```
CREATE KEYSPACE <keyspace_name> WITH REPLICATION = { 'class' : '<replication_strategy>', 'replication_factor' : <number> };
```

Example:

```
CREATE KEYSPACE my_keyspace WITH REPLICATION = { 'class' : 'SimpleStrategy', 'replication_factor' : 3 };
```

Use Keyspace:

```
USE <keyspace_name>;
```

Example:

```
USE my_keyspace;
```

Alter Keyspace:

```
ALTER KEYSPACE <keyspace_name> WITH REPLICATION = { 'class' : '<replication_strategy>', 'replication_factor' : <number> };
```

Example:

```
ALTER KEYSPACE my_keyspace WITH REPLICATION = { 'class' : 'NetworkTopologyStrategy', 'dc1' : 3, 'dc2' : 2 };
```

Drop Keyspace:

```
DROP KEYSPACE <keyspace_name>;
```

Example:

```
DROP KEYSPACE my_keyspace;
```

Table Operations

Create Table:

```
CREATE TABLE <table_name> (column1_name data_type PRIMARY KEY, column2_name data_type, ...);
```

Example:

```
CREATE TABLE users (id UUID PRIMARY KEY, name text, age int);
```

Alter Table (Add Column):

```
ALTER TABLE <table_name> ADD <column_name> data_type;
```

Example:

```
ALTER TABLE users ADD email text;
```

Drop Table:

```
DROP TABLE <table_name>;
```

Example:

```
DROP TABLE users;
```

Truncate Table:

```
TRUNCATE TABLE <table_name>;
```

Example:

```
TRUNCATE TABLE users;
```

Data Manipulation

Insert Data

```
INSERT INTO <table_name> (column1, column2, ...) VALUES (value1, value2, ...);
```

Example:

```
INSERT INTO users (id, name, age, email) VALUES (UUID(), 'John Doe', 30, 'john.doe@example.com');
```

Update Data

```
UPDATE <table_name> SET column1 = value1, column2 = value2, ... WHERE condition;
```

Example:

```
UPDATE users SET age = 31 WHERE id = aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee;
```

Select Data

```
SELECT column1, column2, ... FROM <table_name> WHERE condition;
```

Example:

```
SELECT id, name, age FROM users WHERE name = 'John Doe';
```

```
SELECT * FROM users WHERE age > 25;
```

Delete Data

```
DELETE FROM <table_name> WHERE condition;
```

Example:

```
DELETE FROM users WHERE id = aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee;
```

```
DELETE age, email FROM users WHERE id = aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee;
```

Data Types

Common Data Types

text	Variable-length UTF-8 string.
ascii	US-ASCII string.
bigint	64-bit signed long.
blob	Arbitrary sequence of bytes (up to 2GB).
boolean	True or False.
decimal	Variable-precision decimal.
double	64-bit IEEE-754 floating point.

float	32-bit IEEE-754 floating point.
int	32-bit signed integer.
timestamp	Date and time (stored as milliseconds since epoch).
uuid	Type 4 universally unique identifier.
timeuuid	Type 1 time-based universally unique identifier.
inet	IPv4 or IPv6 address.

Advanced CQL Features

Indexes

Creating an Index:	<pre>CREATE INDEX IF NOT EXISTS <index_name> ON <table_name> (<column_name>);</pre> <p>Example:</p> <pre>CREATE INDEX IF NOT EXISTS users_name_idx ON users (name);</pre>
Dropping an Index:	<pre>DROP INDEX IF EXISTS <index_name>;</pre> <p>Example:</p> <pre>DROP INDEX IF EXISTS users_name_idx;</pre>

Batch Statements

Executing Multiple Statements Atomically:
<pre>BEGIN BATCH statement1; statement2; ... APPLY BATCH;</pre> <p>Example:</p> <pre>BEGIN BATCH INSERT INTO users (id, name, age) VALUES (UUID(), 'Alice', 28); UPDATE users SET age = 29 WHERE name = 'Alice'; APPLY BATCH;</pre>

Using Functions

Built-in Functions:	CQL provides various built-in functions for data manipulation.
Examples:	<pre>token(partition_key)</pre> : Returns the token of a partition key. <pre>now()</pre> : Returns the current timestamp. <pre>toDate(timestamp)</pre> : Converts a timestamp to a date.
Example Usage:	<pre>SELECT * FROM users WHERE token(id) > token(min_id) AND token(id) <= token(max_id);</pre>