# cqlsh Cassandra query shell

**START HERE**

**`cqlsh <host>`**
Connect to a node

**`DESCRIBE TABLE <table>`**
Show table definition

**`DESCRIBE KEYSPACES`**
List keyspaces

**`SELECT * FROM <table> LIMIT 10;`**
Peek at rows

**`USE <keyspace>;`**
Switch keyspace

**`CONSISTENCY QUORUM`**
Set read/write consistency

**`DESCRIBE TABLES`**
List tables

**`EXIT`**
Leave the shell

## CONCEPTS

Statements end with a semicolon; shell commands such as DESCRIBE or CONSISTENCY do not need one.

**Cluster** nodes, partitioner
**Keyspace** replication settings
**Table** partition key + clustering
**Row** columns

cqlsh talks to one node; the coordinator it picks applies the chosen consistency level to each request.

## CONNECT

**`cqlsh <host> <port>`**
Connect to a node (default port 9042)

**`cqlsh -u <user> -p <password> <host>`**
Authenticate on connect
*A password on the command line shows up in the process list; prefer a credentials file.*

**`cqlsh -k <keyspace> <host>`**
Start in a keyspace

**`cqlsh --ssl <host>`**
Connect over TLS

**`cqlsh -e "<statement>" <host>`**
Run one statement and quit

**`cqlsh -f <file.cql> <host>`**
Run a script file and quit

**`cqlsh --request-timeout=<sec> <host>`**
Raise the request timeout (default 10)

**`cqlsh --cqlshrc=<file> <host>`**
Use another cqlshrc

**`cqlsh --no-color <host>`**
Disable colour output

**`cqlsh --version`**
Print the cqlsh version

## EXPLORE SCHEMA

**`DESCRIBE CLUSTER`**
Cluster name and partitioner

**`DESCRIBE KEYSPACES`**
List all keyspaces

**`DESCRIBE KEYSPACE <keyspace>`**
DDL of one keyspace

**`DESCRIBE TABLES`**
Tables in current keyspace

**`DESCRIBE TABLE <table>`**
DDL of a table

## EXPLORE SCHEMA (cont.)

**`DESCRIBE SCHEMA`**
DDL for the whole schema

**`DESCRIBE INDEX <index>`**
Show a secondary index

**`DESCRIBE MATERIALIZED VIEW <view>`**
Show a materialized view

**`DESCRIBE TYPES`**
List user-defined types

**`DESC <object>`**
Short form of DESCRIBE

## SESSION CONTROL

**`USE <keyspace>;`**
Make a keyspace current

**`CONSISTENCY <level>`**
Set consistency level
*Levels: ANY, ONE, TWO, THREE, QUORUM, ALL, LOCAL_QUORUM, LOCAL_ONE, SERIAL, LOCAL_SERIAL.*

**`CONSISTENCY`**
Show current consistency level

**`SERIAL CONSISTENCY <level>`**
Consistency for IF (LWT) writes
*Takes SERIAL or LOCAL_SERIAL.*

**`TRACING ON`**
Print a trace after each query

**`TRACING OFF`**
Stop tracing

**`EXPAND ON`**
One column per line

**`EXPAND OFF`**
Back to table output

**`PAGING ON`**
Page results with a prompt

**`PAGING <rows>`**
Set the page size

**`PAGING OFF`**
Print all rows at once

**`EXIT`**
Quit the shell

## SERVER AND FILES

**`SHOW VERSION`**
cqlsh, Cassandra and protocol versions

**`SHOW HOST`**
Connected node and cluster name

**`SHOW SESSION <session-id>`**
Pretty-print a trace session

**`SHOW REPLICAS <token> <keyspace>`**
Replicas for a token

**`SOURCE '<file.cql>'`**
Run a CQL file from the prompt

## SERVER AND FILES (cont.)

**`CAPTURE '<file>'`**
Append query output to a file

**`CAPTURE OFF`**
Stop capturing output

**`HELP <topic>`**
Built-in help on a command

## BASIC CQL

**`CREATE KEYSPACE <ks> WITH replication = {'class': 'SimpleStrategy', 'replication_factor': 1};`**
Create a keyspace

**`CREATE TABLE <table> (<id> uuid PRIMARY KEY, <col> text);`**
Create a table

**`INSERT INTO <table> (<id>, <col>) VALUES (<id>, '<val>');`**
Insert a row

**`SELECT * FROM <table> WHERE <id> = <value>;`**
Read by partition key

**`SELECT * FROM <table> LIMIT <n>;`**
Limit rows returned

**`UPDATE <table> SET <col> = '<val>' WHERE <id> = <value>;`**
Change a column

**`DELETE FROM <table> WHERE <id> = <value>;`**
Delete a row

**`TRUNCATE <table>;`**
Remove all rows

**`ALTER TABLE <table> ADD <col> <type>;`**
Add a column

**`DROP TABLE <table>;`**
Delete a table

**`CREATE INDEX ON <table> (<col>);`**
Secondary index on a column

**`SELECT * FROM <table> WHERE <col> = '<val>' ALLOW FILTERING;`**
Filter without index (slow)
*Scans many partitions; avoid on large tables.*

## COPY IMPORT AND EXPORT

**`COPY <table> TO '<file.csv>' WITH HEADER = true;`**
Export a table to CSV

**`COPY <table> FROM '<file.csv>' WITH HEADER = true;`**
Import CSV into a table

## COPY IMPORT AND EXPORT (cont.)

**`COPY <table> (<col1>, <col2>) TO STDOUT;`**
Export chosen columns

**`COPY <table> FROM '<file.csv>' WITH INGESTRATE = <n>;`**
Throttle an import

**`COPY <table> FROM '<file.csv>' WITH ERRFILE = '<file>';`**
Write rejected rows to a file

**`COPY <table> TO '<file>' WITH PAGESIZE = <n>;`**
Rows per page when exporting

### COPY defaults

| Option | Default | Meaning |
|---|---|---|
| `PAGESIZE` | 1000 | Rows per page (TO) |
| `MAXREQUESTS` | 6 | Token ranges at once (TO) |
| `INGESTRATE` | 100000 | Rows per second (FROM) |
| `MAXBATCHSIZE` | 20 | Rows per batch (FROM) |
| `NUMPROCESSES` | 16 | Worker processes |
| `HEADER` | false | First line holds names |

## CONFIG

**`~/.cassandra/cqlshrc`**
Default config file

**`CQL_HISTORY=/dev/null`**
Env var: disable history

**`cqlsh --credentials=<file> <host>`**
Use another credentials file

The credentials file must be owned by you and unreadable by others, or cqlsh refuses it.

## RECIPES

### Debug a slow query

```
cqlsh <host> -k <keyspace>
TRACING ON
CONSISTENCY LOCAL_QUORUM
SELECT * FROM <table> WHERE
<id> = <value>;
# trace prints after the result
```

### Dump a table and reload it

```
COPY <table> TO 'dump.csv' WITH
HEADER = true;
TRUNCATE <table>;
COPY <table> FROM 'dump.csv'
WITH HEADER = true;
```

### Script a schema over TLS

```
cqlsh --ssl -u <user> -p
<password> -f schema.cql <host>
```

### Save results for later

```
CAPTURE '/tmp/out.txt'
EXPAND ON
SELECT * FROM <table> LIMIT 5;
CAPTURE OFF
```