Cassandra cli--thrift

```
Display the general help page with a list of available commands.;

[default@unknown] help ;
Getting around:
?                       Display this help.
help;                   Display this help.
help <command>;         Display command-specific help.
exit;                   Exit this utility.
quit;                   Exit this utility.

Commands:
assume                  Apply client side validation.
connect                 Connect to a Cassandra node.
consistencylevel        Sets consisteny level for the client to use.
count                   Count columns or super columns.
create column family    Add a column family to an existing keyspace.
create keyspace         Add a keyspace to the cluster.
del                     Delete a column, super column or row.
decr                    Decrements a counter column.
describe cluster        Describe the cluster configuration.
describe                Describe a keyspace and its column families or column family in current keyspace.
drop column family      Remove a column family and its data.
drop keyspace           Remove a keyspace and its data.
drop index              Remove an existing index from specific column.
get                     Get rows and columns.
incr                    Increments a counter column.
list                    List rows in a column family.
set                     Set columns.
show api version        Show the server API version.
show cluster name       Show the cluster name.
show keyspaces          Show all keyspaces and their column families.
show schema             Show a cli script to create keyspaces and column families.
truncate                Drop the data in a column family.
update column family    Update the settings for a column family.
update keyspace         Update the settings for a keyspace.
use                     Switch to a keyspace.

[default@unknown] create keyspace demo;
42a4dc44-f9fb-3777-b079-13e273693e93
[default@unknown] drop keyspace demoks;
Keyspace 'demoks' not found.
[default@unknown] drop keyspace demos;
Keyspace 'demos' not found.
[default@unknown] drop keyspace demo;
6403a0ff-f93b-3b1f-8c35-0a8dc85a5b66
[default@unknown] create keyspace demo;
b9ca0d78-0c9a-3e2d-9a28-48385d81ab2f
[default@unknown] describe keyspace demo;
Syntax error at position 9: no viable alternative at input 'keyspace'
[default@unknown]
[default@unknown] describe keyspace demo;
Syntax error at position 9: no viable alternative at input 'keyspace'
[default@unknown] describe demo;

WARNING: CQL3 tables are intentionally omitted from 'describe' output.
See https://issues.apache.org/jira/browse/CASSANDRA-4377 for details.

Keyspace: demo:
  Replication Strategy: org.apache.cassandra.locator.NetworkTopologyStrategy
```

```
    Options: [datacenter1:1]
  Column Families:
[default@unknown] describe cluster;
Cluster Information:
   Name: Test Cluster
   Snitch: org.apache.cassandra.locator.SimpleSnitch
   Partitioner: org.apache.cassandra.dht.Murmur3Partitioner
   Schema versions:
        b9ca0d78-0c9a-3e2d-9a28-48385d81ab2f: [127.0.0.1]

[default@unknown] use demo;
Authenticated to keyspace: demo
[default@demo] create column family test with comparator=UTF8Type and default_validation_class=UTF8Type;
1a45700d-ebd3-3532-938d-9f6e26aca13c
[default@demo] create column family users with comparator=UTF8Type
...       and column_metadata=[{column_name: full_name, validation_class: UTF8Type}, {column_name: birth_date, vali
ngType, index_type: KEYS}];
d44052c8-63d5-3f5d-9215-79d91a3c0e4f
[default@demo] drop column family test;
6ed8c0c4-527b-3c66-aade-ffc33e1863ab
[default@demo] update column family users with comparator=UTF8Type
...       and column_metadata=[{column_name: full_name, validation_class: UTF8Type},
...       {column_name: birth_date, validation_class: LongType, index_type: KEYS}, {column_name: state, validatior
, index_type: KEYS}];
5682ec52-67f6-3a03-b737-668c5f21ac81
[default@demo] set test[row1][col1] = 'val1';
test not found in current keyspace.
[default@demo] create column family test with comparator=UTF8Type and default_validation_class=UTF8Type;
effb19b6-5cdd-354e-b180-fb863d9999e8
[default@demo] set test[row1][col1] = 'val1';
org.apache.cassandra.serializers.MarshalException: cannot parse 'row1' as hex bytes
[default@demo] set test[0][0] = 'val1';
org.apache.cassandra.serializers.MarshalException: cannot parse '0' as hex bytes
[default@demo] help set;
set <cf>['<key>']['<col>'] = <value>;
set <cf>['<key>']['<super>']['<col>'] = <value>;
set <cf>['<key>']['<col>'] = <function>(<argument>);
set <cf>['<key>']['<super>']['<col>'] = <function>(<argument>);
set <cf>[<key>][<function>(<col>)] = <value> || <function>;
set <cf>[<function>(<key>)][<function>(<col>) || <col>] =
    <value> || <function> with ttl = <secs>;

Sets the column value for the specified column family and key.

Required Parameters:
- cf: Name of the column family to set columns in.

- col: Name of the column to set.

- key: Key for the row to set columns in.

Optional Parameters:
- function: Name of a function to call to parse the supplied argument to the
  specified type. Some functions will generate values without needing an
  argument.

  Valid options are:
    - ascii
    - bytes: if used without arguments generates a zero length byte array
```

```
    - int
    - integer
    - lexicaluuid: if used without arguments generates a new random uuid
    - long
    - timeuuid: if used without arguments generates a new time uuid
    - utf8

- secs: Time To Live for the column in seconds. Defaults to no ttl.

- super: Name of the super column to contain the column.

- value: The value to set the column to.

Examples:
set Super1[ascii('testkey')][ascii('my super')][ascii('test col')]='this is a test';
set Standard1['testkey']['test col']='this is also a test';
set Standard1[testkey][testcol] = utf8('this is utf8 string.');
set Standard1[testkey][timeuuid()] = utf8('hello world');
set Standard1[testkey][timeuuid()] = utf8('hello world') with ttl = 30;
set UseComposites[utf8('testkey')]['CompositeType(utf8(first),int(4))'] = utf8('inserts this string into a column
4');

[default@demo] set users[prothfuss][full_name] = 'Satya';
org.apache.cassandra.serializers.MarshalException: cannot parse 'prothfuss' as hex bytes
[default@demo] set users[5682ec52-67f6-3a03-b737-668c5f21ac81][full_name] = 'Satya';
org.apache.cassandra.serializers.MarshalException: cannot parse '5682ec52-67f6-3a03-b737-668c5f21ac81' as hex byte
[default@demo] show keyspaces
...     ;

WARNING: CQL3 tables are intentionally omitted from 'show keyspaces' output.
See https://issues.apache.org/jira/browse/CASSANDRA-4377 for details.

Keyspace: demo:
  Replication Strategy: org.apache.cassandra.locator.NetworkTopologyStrategy
  Durable Writes: true
    Options: [datacenter1:1]
  Column Families:
    ColumnFamily: test
      Key Validation Class: org.apache.cassandra.db.marshal.BytesType
      Default column value validator: org.apache.cassandra.db.marshal.UTF8Type
      Cells sorted by: org.apache.cassandra.db.marshal.UTF8Type
      GC grace seconds: 864000
      Compaction min/max thresholds: 4/32
      Read repair chance: 0.0
      DC Local Read repair chance: 0.1
      Caching: KEYS_ONLY
      Default time to live: 0
      Bloom Filter FP chance: default
      Index interval: default
      Speculative Retry: NONE
      Built indexes: []
      Compaction Strategy: org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy
      Compression Options:
        sstable_compression: org.apache.cassandra.io.compress.LZ4Compressor
    ColumnFamily: users
      Key Validation Class: org.apache.cassandra.db.marshal.BytesType
      Default column value validator: org.apache.cassandra.db.marshal.BytesType
      Cells sorted by: org.apache.cassandra.db.marshal.UTF8Type
      GC grace seconds: 864000
```

```
      Read repair chance: 0.0
      DC Local Read repair chance: 0.1
      Caching: KEYS_ONLY
      Default time to live: 0
      Bloom Filter FP chance: default
      Index interval: default
      Speculative Retry: NONE
      Built indexes: [users.users_birth_date_idx, users.users_state_idx]
      Column Metadata:
        Column Name: full_name
          Validation Class: org.apache.cassandra.db.marshal.UTF8Type
        Column Name: state
          Validation Class: org.apache.cassandra.db.marshal.UTF8Type
          Index Name: users_state_idx
          Index Type: KEYS
        Column Name: birth_date
          Validation Class: org.apache.cassandra.db.marshal.LongType
          Index Name: users_birth_date_idx
          Index Type: KEYS
      Compaction Strategy: org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy
      Compression Options:
        sstable_compression: org.apache.cassandra.io.compress.LZ4Compressor
Keyspace: system:
  Replication Strategy: org.apache.cassandra.locator.LocalStrategy
  Durable Writes: true
    Options: □
  Column Families:
    ColumnFamily: IndexInfo
    "indexes that have been completed"
      Key Validation Class: org.apache.cassandra.db.marshal.UTF8Type
      Default column value validator: org.apache.cassandra.db.marshal.BytesType
      Cells sorted by: org.apache.cassandra.db.marshal.UTF8Type
      GC grace seconds: 0
      Compaction min/max thresholds: 4/32
      Read repair chance: 0.0
      DC Local Read repair chance: 0.0
      Caching: KEYS_ONLY
      Default time to live: 0
      Bloom Filter FP chance: 0.01
      Index interval: default
      Speculative Retry: 99.0PERCENTILE
      Built indexes: □
      Compaction Strategy: org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy
      Compression Options:
        sstable_compression: org.apache.cassandra.io.compress.LZ4Compressor
    ColumnFamily: hints
    "hints awaiting delivery"
      Key Validation Class: org.apache.cassandra.db.marshal.UUIDType
      Default column value validator: org.apache.cassandra.db.marshal.BytesType
      Cells sorted by: org.apache.cassandra.db.marshal.CompositeType(org.apache.cassandra.db.marshal.TimeUUIDType,o
ndra.db.marshal.Int32Type)
      GC grace seconds: 0
      Compaction min/max thresholds: 4/32
      Read repair chance: 0.0
      DC Local Read repair chance: 0.0
      Caching: KEYS_ONLY
      Default time to live: 0
      Bloom Filter FP chance: 0.01
      Index interval: default
```

```
      Compaction Strategy: org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy
      Compression Options:
        sstable_compression: org.apache.cassandra.io.compress.LZ4Compressor
    ColumnFamily: hints
    "hints awaiting delivery"
      Key Validation Class: org.apache.cassandra.db.marshal.UUIDType
      Default column value validator: org.apache.cassandra.db.marshal.BytesType
      Cells sorted by: org.apache.cassandra.db.marshal.CompositeType(org.apache.cassandra.db.marshal.TimeUUIDType,
ndra.db.marshal.Int32Type)
      GC grace seconds: 0
      Compaction min/max thresholds: 4/32
      Read repair chance: 0.0
      DC Local Read repair chance: 0.0
      Caching: KEYS_ONLY
      Default time to live: 0
      Bloom Filter FP chance: 0.01
      Index interval: default
      Speculative Retry: 99.0PERCENTILE
      Built indexes: []
      Compaction Strategy: org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy
      Compaction Strategy Options:
        enabled: false
      Compression Options:
        sstable_compression: org.apache.cassandra.io.compress.LZ4Compressor
    ColumnFamily: schema_keyspaces
    "keyspace definitions"
      Key Validation Class: org.apache.cassandra.db.marshal.UTF8Type
      Default column value validator: org.apache.cassandra.db.marshal.BytesType
      Cells sorted by: org.apache.cassandra.db.marshal.UTF8Type
      GC grace seconds: 604800
      Compaction min/max thresholds: 4/32
      Read repair chance: 0.0
      DC Local Read repair chance: 0.0
      Caching: KEYS_ONLY
      Default time to live: 0
      Bloom Filter FP chance: 0.01
      Index interval: default
      Speculative Retry: 99.0PERCENTILE
      Built indexes: []
      Column Metadata:
        Column Name: durable_writes
          Validation Class: org.apache.cassandra.db.marshal.BooleanType
        Column Name: strategy_options
          Validation Class: org.apache.cassandra.db.marshal.UTF8Type
        Column Name: strategy_class
          Validation Class: org.apache.cassandra.db.marshal.UTF8Type
      Compaction Strategy: org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy
      Compression Options:
        sstable_compression: org.apache.cassandra.io.compress.LZ4Compressor
Keyspace: system_traces:
  Replication Strategy: org.apache.cassandra.locator.SimpleStrategy
  Durable Writes: true
    Options: [replication_factor:2]
  Column Families:
[default@demo] show cluster name;
Test Cluster
[default@demo] show api version
...     ;
19.39.0
```

```
See https://issues.apache.org/jira/browse/CASSANDRA-4377 for details.

    ColumnFamily: users
      Key Validation Class: org.apache.cassandra.db.marshal.BytesType
      Default column value validator: org.apache.cassandra.db.marshal.BytesType
      Cells sorted by: org.apache.cassandra.db.marshal.UTF8Type
      GC grace seconds: 864000
      Compaction min/max thresholds: 4/32
      Read repair chance: 0.0
      DC Local Read repair chance: 0.1
      Caching: KEYS_ONLY
      Default time to live: 0
      Bloom Filter FP chance: default
      Index interval: default
      Speculative Retry: NONE
      Built indexes: [users.users_birth_date_idx, users.users_state_idx]
      Column Metadata:
        Column Name: full_name
          Validation Class: org.apache.cassandra.db.marshal.UTF8Type
        Column Name: state
          Validation Class: org.apache.cassandra.db.marshal.UTF8Type
          Index Name: users_state_idx
          Index Type: KEYS
        Column Name: birth_date
          Validation Class: org.apache.cassandra.db.marshal.LongType
          Index Name: users_birth_date_idx
          Index Type: KEYS
      Compaction Strategy: org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy
      Compression Options:
        sstable_compression: org.apache.cassandra.io.compress.LZ4Compressor
[default@demo] assume
...     ;
Syntax error at position 7: mismatched input ';' expecting set null
[default@demo] assume users comparator as ascii;
Assumption for column family 'users' added successfully.
[default@demo] assume users  keys as ascii;
Assumption for column family 'users' added successfully.
[default@demo] set users[prothfuss][full_name] = 'Satya';
Value inserted.
Elapsed time: 28 msec(s).
[default@demo] set users[prothfuss][birth_date] = 1983
...     ;
Value inserted.
Elapsed time: 8.3 msec(s).
[default@demo] assume test  keys as ascii;
Assumption for column family 'test' added successfully.
[default@demo] assume test comparator as ascii;
Assumption for column family 'test' added successfully.
[default@demo] set test[row1][col1] = 'val1';
Value inserted.
Elapsed time: 1.81 msec(s).
[default@demo]  set test[row1][col2] = 'val2' with ttl=60;
Value inserted.
Elapsed time: 1.39 msec(s).
[default@demo] get test[row1];
=> (name=col1, value=val1, timestamp=1497217216786000)
=> (name=col2, value=val2, timestamp=1497217229504000, ttl=60)
Returned 2 results.
Elapsed time: 36 msec(s).
```

```
yntax error at position 7: mismatched input ';' expecting set null
default@demo] assume users comparator as ascii;
ssumption for column family 'users' added successfully.
default@demo] assume users  keys as ascii;
ssumption for column family 'users' added successfully.
default@demo] set users[prothfuss][full_name] = 'Satya';
alue inserted.
lapsed time: 28 msec(s).
default@demo] set users[prothfuss][birth_date] = 1983
..     ;
alue inserted.
lapsed time: 8.3 msec(s).
default@demo] assume test  keys as ascii;
ssumption for column family 'test' added successfully.
default@demo] assume test comparator as ascii;
ssumption for column family 'test' added successfully.
default@demo] set test[row1][col1] = 'val1';
alue inserted.
lapsed time: 1.81 msec(s).
default@demo]  set test[row1][col2] = 'val2' with ttl=60;
alue inserted.
lapsed time: 1.39 msec(s).
default@demo] get test[row1];
> (name=col1, value=val1, timestamp=1497217216786000)
> (name=col2, value=val2, timestamp=1497217229504000, ttl=60)
eturned 2 results.
lapsed time: 36 msec(s).
default@demo] get users where birth_date = 1945;

 Row Returned.
lapsed time: 39 msec(s).
default@demo] get users where state = 'UT' and birth_date > 1980;

 Row Returned.
lapsed time: 4.6 msec(s).
default@demo] get test[row1];
> (name=col1, value=val1, timestamp=1497217216786000)
> (name=col2, value=val2, timestamp=1497217229504000, ttl=60)
eturned 2 results.
lapsed time: 1.78 msec(s).
default@demo] get test[row1];
> (name=col1, value=val1, timestamp=1497217216786000)
eturned 1 results.
lapsed time: 2.59 msec(s).
default@demo]
```

```
cqlsh:tutorialspoint> CREATE KEYSPACE tutorialspoint WITH replication = {'class':'SimpleStrategy', 'replication_factor' : 3};
cqlsh:tutorialspoint> USE tutorialspoint;
cqlsh:tutorialspoint> CREATE TABLE emp(
                   ...    emp_id int PRIMARY KEY,
                   ...    emp_name text,
                   ...    emp_city text,
                   ...    emp_sal varint,
                   ...    emp_phone varint
                   ...    );
cqlsh:tutorialspoint>  ALTER TABLE emp
                   ...     ... ADD emp_email text;
SyntaxException: line 2:4 no viable alternative at input '.' (ALTER TABLE emp   .[.]...)
cqlsh:tutorialspoint>  ALTER TABLE emp
                   ... ADD emp_email text;
cqlsh:tutorialspoint>  ALTER TABLE emp DROP emp_email;
cqlsh:tutorialspoint> select * from emp;

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

(0 rows)
cqlsh:tutorialspoint>
cqlsh:tutorialspoint> DROP TABLE emp;
cqlsh:tutorialspoint> DESCRIBE COLUMNFAMILIES;

<empty>

cqlsh:tutorialspoint> CREATE TABLE emp(    emp_id int PRIMARY KEY,    emp_name text,    emp_city text,    emp_sal varint,    emp_p
hone varint    );
cqlsh:tutorialspoint> DESCRIBE COLUMNFAMILIES;

emp

cqlsh:tutorialspoint> BEGIN BATCH
                   ... INSERT INTO emp (emp_id, emp_city, emp_name, emp_phone, emp_sal) values(  4,'Pune','rajeev',9848022331, 3000
0);
                   ...  UPDATE emp SET emp_sal = 50000 WHERE emp_id =3;
                   ... DELETE emp_city FROM emp WHERE emp_id = 2;
                   ... APPLY BATCH;
cqlsh:tutorialspoint> INSERT INTO emp (emp_id, emp_name, emp_city,
```

```
cqlsh:tutorialspoint> DESCRIBE COLUMNFAMILIES;

emp

cqlsh:tutorialspoint> BEGIN BATCH
                  ... INSERT INTO emp (emp_id, emp_city, emp_name, emp_phone, emp_sal) values(  4,'Pune','rajeev',
0);
                  ...  UPDATE emp SET emp_sal = 50000 WHERE emp_id =3;
                  ... DELETE emp_city FROM emp WHERE emp_id = 2;
                  ... APPLY BATCH;
cqlsh:tutorialspoint> INSERT INTO emp (emp_id, emp_name, emp_city,
                  ...    emp_phone, emp_sal) VALUES(1,'ram', 'Hyderabad', 9848022338, 50000);
cqlsh:tutorialspoint> INSERT INTO emp (emp_id, emp_name, emp_city,
                  ...    emp_phone, emp_sal) VALUES(2,'robin', 'Hyderabad', 9848022339, 40000);
cqlsh:tutorialspoint> INSERT INTO emp (emp_id, emp_name, emp_city,
                  ...    emp_phone, emp_sal) VALUES(3,'rahman', 'Chennai', 9848022330, 45000);
cqlsh:tutorialspoint>  SELECT * FROM emp;

 emp_id | emp_city  | emp_name | emp_phone  | emp_sal
--------+-----------+----------+------------+---------
      1 | Hyderabad |      ram | 9848022338 |   50000
      2 | Hyderabad |    robin | 9848022339 |   40000
      4 |      Pune |   rajeev | 9848022331 |   30000
      3 |   Chennai |   rahman | 9848022330 |   45000

(4 rows)
cqlsh:tutorialspoint> UPDATE emp SET emp_city='Delhi',emp_sal=50000
                  ...    WHERE emp_id=2;
cqlsh:tutorialspoint> select * from emp;

 emp_id | emp_city  | emp_name | emp_phone  | emp_sal
--------+-----------+----------+------------+---------
      1 | Hyderabad |      ram | 9848022338 |   50000
      2 |     Delhi |    robin | 9848022339 |   50000
      4 |      Pune |   rajeev | 9848022331 |   30000
      3 |   Chennai |   rahman | 9848022330 |   45000

(4 rows)
cqlsh:tutorialspoint>
cqlsh:tutorialspoint> SELECT emp_name, emp_sal from emp;

 emp_name | emp_sal
----------+---------
      ram |   50000
    robin |   50000
   rajeev |   30000
   rahman |   45000

(4 rows)
cqlsh:tutorialspoint> CREATE INDEX ON emp(emp_sal);
cqlsh:tutorialspoint> SELECT * FROM emp WHERE emp_sal=50000;

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

```
(2 rows)
cqlsh:tutorialspoint>  DELETE FROM emp WHERE emp_id=3;
cqlsh:tutorialspoint> CREATE TABLE data(name text PRIMARY KEY, email list<text>);
cqlsh:tutorialspoint> INSERT INTO data(name, email) VALUES ('ramu',
                 ... ['abc@gmail.com','cba@yahoo.com'])
                 ... ;
cqlsh:tutorialspoint> SELECT * FROM data;

 name | email
------+---------------------------------
 ramu | ['abc@gmail.com', 'cba@yahoo.com']

(1 rows)
cqlsh:tutorialspoint> CREATE TABLE data2 (name text PRIMARY KEY, phone set<varint>);
cqlsh:tutorialspoint> INSERT INTO data2(name, phone)VALUES ('rahman',    {9848022338,9848022339});
cqlsh:tutorialspoint>  SELECT * FROM data2;

 name   | phone
--------+--------------------------
 rahman | {9848022338, 9848022339}

(1 rows)
cqlsh:tutorialspoint>
cqlsh:tutorialspoint> CREATE TABLE data3 (name text PRIMARY KEY, address
                 ... map<timestamp, text>);
cqlsh:tutorialspoint> INSERT INTO data3 (name, address)
                 ...    VALUES ('robin', {'home' : 'hyderabad' , 'office' : 'Delhi' } );
```