Example 5:
cqlsh
create columnfamily

Creating a column family

```
cassandra@cqlsh> use catalog;
```

Command to go to our keyspace

```
cassandra@cqlsh:catalog>
```

now the current keyspace is catalog

Creating a column family

```
cassandra@cqlsh> use catalog;

cassandra@cqlsh:catalog> CREATE COLUMNFAMILY product
  productId varchar,
  title text,
  brand varchar,
  publisher varchar,
  length int,
  breadth int,
  height int,
  PRIMARY KEY(productId)
 );
```

Command to create column family

Creating a column family

```
cassandra@cqlsh> use catalog;

cassandra@cqlsh:catalog> CREATE COLUMNFAMILY product
  productId varchar,
  title text,
  brand varchar,
  publisher varchar,
  length int,
  breadth int,
  height int,
  PRIMARY KEY(productId)
 );
```

columnfamily name

Creating a column family

```
cassandra@cqlsh> use catalog;

cassandra@cqlsh:catalog> CREATE COLUMNFAMILY product
 productId varchar,
 title text,
 brand varchar,
 publisher varchar,
 length int,
 breadth int,
 height int,
 PRIMARY KEY(productId)
);
```

column definitions

Creating a column family

```
cassandra@cqlsh> use catalog;

cassandra@cqlsh:catalog> CREATE COLUMNFAMILY product
  productId varchar,
  title text,
  brand varchar,
  publisher varchar,
  length int,
  breadth int,
  height int,
  PRIMARY KEY(productId)
 );
```

Set the productid column as the primary key

Creating a column family

Let's verify whether product column family is created

```
cassandra@cqlsh:catalog> describe columnfamilies;
```

command to list all column families in the current keyspace

Creating a column family

Let's verify whether product column family is created

```
cassandra@cqlsh:catalog> describe columnfamilies;
```

output

```
product
```

columnfamily product is created

Creating a column family

```
cassandra@cqlsh:catalog> describe product;
```

this command describes for this columnfamily

1. all columns
2. primary key
3. storage properties

Creating a column family

```
cassandra@cqlsh:catalog> describe product;

CREATE TABLE catalog.product (
    productid text PRIMARY KEY,
    brand text,
    breadth int,
    height int,
    length int,
    publisher text,
    title text
) WITH bloom_filter_fp_chance = 0.01
    AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
    AND comment = ''
    AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompact
ionStrategy', 'max_threshold': '32', 'min_threshold': '4'}
    AND compression = {'chunk_length_in_kb': '64', 'class': 'org.apache.cassandra.io
.compress.LZ4Compressor'}
    AND crc_check_chance = 1.0
    AND dclocal_read_repair_chance = 0.1
    AND default_time_to_live = 0
    AND gc_grace_seconds = 864000
    AND max_index_interval = 2048
    AND memtable_flush_period_in_ms = 0
    AND min_index_interval = 128
    AND read_repair_chance = 0.0
    AND speculative_retry = '99PERCENTILE';
```

There are many configuration settings

Lets go through it one at a time

Creating a column family

```
cassandra@cqlsh:catalog> describe product;

CREATE TABLE catalog.product (
    productid text PRIMARY KEY,
    brand text,
    breadth int,
    height int,
    length int,
    publisher text,
    title text
) WITH bloom_filter_fp_chance = 0.01
    AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
    AND comment = ''
    AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompact
ionStrategy', 'max_threshold': '32', 'min_threshold': '4'}
    AND compression = {'chunk_length_in_kb': '64', 'class': 'org.apache.cassandra.io
.compress.LZ4Compressor'}
    AND crc_check_chance = 1.0
    AND dclocal_read_repair_chance = 0.1
    AND default_time_to_live = 0
```

the column definitions with which we created the columnfamily product

Creating a column family

```
cassandra@cqlsh:catalog> describe product;
CREATE TABLE catalog.product (
    productid text PRIMARY KEY,
    brand text,
    breadth int,
    height int,
    length int,
    publisher text,
    title text
) WITH bloom_filter_fp_chance = 0.01
    AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
    AND comment = ''
    AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompact
ionStrategy', 'max_threshold': '32', 'min_threshold': '4'}
    AND compression = {'chunk_length_in_kb': '64', 'class': 'org.apache.cassandra.io
.compress.LZ4Compressor'}
    AND crc_check_chance = 1.0
    AND dclocal_read_repair_chance = 0.1
    AND default_time_to_live = 0
```

The probability that there might be a false positive when checking for the presence of a key in the internal bloom filter

Creating a column family

```
cassandra@cqlsh:catalog> describe product;

CREATE TABLE catalog.product (
    productid text PRIMARY KEY,
    brand text,
    breadth int,
    height int,
    length int,
    publisher text,
    title text
) WITH bloom_filter_fp_chance = 0.01
    AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
    AND comment = ''
    AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompact
ionStrategy', 'max_threshold': '32', 'min_threshold': '4'}
    AND compression = {'chunk_length_in_kb': '64', 'class': 'org.apache.cassandra.io
.compress.LZ4Compressor'}
    AND crc_check_chance = 1.0
    AND dclocal_read_repair_chance = 0.1
    AND default_time_to_live = 0
```

Each columnfamily uses a bloom filter to determine if a key is present in the columns

A bloom filter is a bit array

Creating a column family

```
cassandra@cqlsh:catalog> describe product
```

```
CREATE TABLE catalog.product (
    productid text PRIMARY KEY,
    brand text,
    breadth int,
    height int,
    length int,
    publisher text,
    title text
) WITH bloom_filter_fp_chance = 0.01
    AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
    AND comment = ''
    AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompact
ionStrategy', 'max_threshold': '32', 'min_threshold': '4'}
    AND compression = {'chunk_length_in_kb': '64', 'class': 'org.apache.cassandra.io
.compress.LZ4Compressor'}
    AND crc_check_chance = 1.0
    AND dclocal_read_repair_chance = 0.1
    AND default_time_to_live = 0
```

bloom filter

Hash values of the key set 1 or 0 in the bit array

A 1 bit indicates that a key is present and a 0 bit indicates it is absent

Creating a column family

```
CREATE TABLE catalog.product (
    productid text PRIMARY KEY,
    brand text,
    breadth int,
    height int,
    length int,
    publisher text,
    title text
) WITH bloom_filter_fp_chance = 0.01
    AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
    AND comment = ''
    AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompact
ionStrategy', 'max_threshold': '32', 'min_threshold': '4'}
    AND compression = {'chunk_length_in_kb': '64', 'class': 'org.apache.cassandra.io
.compress.LZ4Compressor'}
    AND crc_check_chance = 1.0
    AND dclocal_read_repair_chance = 0.1
    AND default_time_to_live = 0
```

bloom filter

Bloom filters can have false positives
but not false negatives

Creating a column family

false positive probability can be between 0 and 1

higher the false probability, smaller size of bloom filters

value = 1 disables the bloom filter

```
cassandra@cqlsh:catalog> describe product;

CREATE TABLE catalog.product (
    productid text PRIMARY KEY,
    brand text,
    breadth int,
    height int,
    length int,
    publisher text,
    title text
) WITH bloom_filter_fp_chance = 0.01
    AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
    AND comment = ''
    AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompact
ionStrategy', 'max_threshold': '32', 'min_threshold': '4'}
    AND compression = {'chunk_length_in_kb': '64', 'class': 'org.apache.cassandra.io
.compress.LZ4Compressor'}
    AND crc_check_chance = 1.0
    AND dclocal_read_repair_chance = 0.1
    AND default_time_to_live = 0
```

Creating a column family

'keys' refers to the key cache that we have seen earlier

'ALL' = store location data for all keys

'NONE' = store location data for no keys

```
cassandra@cqlsh:catalog> describe product;

CREATE TABLE catalog.product (
    productid text PRIMARY KEY,
    brand text,
    breadth int,
    height int,
    length int,
    publisher text,
    title text
) WITH bloom_filter_fp_chance = 0.01
    AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
    AND comment = ''
    AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompact
ionStrategy', 'max_threshold': '32', 'min_threshold': '4'}
    AND compression = {'chunk_length_in_kb': '64', 'class': 'org.apache.cassandra.io
.compress.LZ4Compressor'}
    AND crc_check_chance = 1.0
    AND dclocal_read_repair_chance = 0.1
    AND default_time_to_live = 0
```

Creating a column family

number of rows to be stored in cache
possible values: NONE, ALL, number(N)

by default no rows are stored in cache

```
cassandra@cqlsh:catalog> describe product;

CREATE TABLE catalog.product (
    productid text PRIMARY KEY,
    brand text,
    breadth int,
    height int,
    length int,
    publisher text,
    title text
) WITH bloom_filter_fp_chance = 0.01
    AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
    AND comment = ''
    AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompact
ionStrategy', 'max_threshold': '32', 'min_threshold': '4'}
    AND compression = {'chunk_length_in_kb': '64', 'class': 'org.apache.cassandra.io
.compress.LZ4Compressor'}
    AND crc_check_chance = 1.0
    AND dclocal_read_repair_chance = 0.1
    AND default_time_to_live = 0
```

Creating a column family

```
cassandra@cqlsh:catalog> describe product;
CREATE TABLE catalog.product (
    productid text PRIMARY KEY,
    brand text,
    breadth int,
    height int,
    length int,
    publisher text,
    title text
) WITH bloom_filter_fp_chance = 0.01
    AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
    AND comment = ''
    AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompact
ionStrategy', 'max_threshold': '32', 'min_threshold': '4'}
    AND compression = {'chunk_length_in_kb': '64', 'class': 'org.apache.cassandra.io
.compress.LZ4Compressor'}
    AND crc_check_chance = 1.0
    AND dclocal_read_repair_chance = 0.1
    AND default_time_to_live = 0
```

a brief description about the product columnfamily like what is its use , what data does it store

Creating a column family

data is stored on disk in SSTable file

specifies the strategy class to be used for compaction of SSTable files

```
cassandra@cqlsh:catalog> describe product;

CREATE TABLE catalog.product (
    productid text PRIMARY KEY,
    brand text,
    breadth int,
    height int,
    length int,
    publisher text,
    title text
) WITH bloom_filter_fp_chance = 0.01
    AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
    AND comment = ''
    AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompact
ionStrategy', 'max_threshold': '32', 'min_threshold': '4'}
    AND compression = {'chunk_length_in_kb': '64', 'class': 'org.apache.cassandra.io
.compress.LZ4Compressor'}
    AND crc_check_chance = 1.0
    AND dclocal_read_repair_chance = 0.1
    AND default_time_to_live = 0
```

Creating a column family

compaction is triggered when the number of SSTable files for a columnfamily reaches the value set in min_threshold

max_threshold is the maximum number of files that can be created while the files are being compacted

```
CREATE TABLE catalog.product (
    productid text PRIMARY KEY,
    brand text,
    breadth int,
    height int,
    length int,
    publisher text,
    title text
) WITH bloom_filter_fp_chance = 0.01
    AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
    AND comment = ''
    AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompact
ionStrategy', 'max_threshold': '32', 'min_threshold': '4'}
    AND compression = {'chunk_length_in_kb': '64', 'class': 'org.apache.cassandra.io
.compress.LZ4Compressor'}
    AND crc_check_chance = 1.0
    AND dclocal_read_repair_chance = 0.1
    AND default_time_to_live = 0
```

Creating a column family

```
cassandra@cqlsh:catalog> describe product;
CREATE TABLE catalog.product (
    productid text PRIMARY KEY,
    brand text,
    breadth int,
    height int,
    length int,
    publisher text,
    title text
) WITH bloom_filter_fp_chance = 0.01
    AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
    AND comment = ''
    AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompact
ionStrategy', 'max_threshold': '32', 'min_threshold': '4'}
    AND compression = {'chunk_length_in_kb': '64', 'class': 'org.apache.cassandra.io
.compress.LZ4Compressor'}
    AND crc_check_chance = 1.0
    AND dclocal_read_repair_chance = 0.1
    AND default_time_to_live = 0
```

specifies the algorithm to be used for compression

Creating a column family

probability that an uncompressed block is run against checksum algorithm

By default checksum algorithm is run for every uncompressed block to check that the compression was done correctly

```
CREATE TABLE catalog.product (
    productid text PRIMARY KEY,
    brand text,
    breadth int,
    height int,
    length int,
    publisher text,
    title text
) WITH bloom_filter_fp_chance = 0.01
    AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
    AND comment = ''
    AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompact
ionStrategy', 'max_threshold': '32', 'min_threshold': '4'}
    AND compression = {'chunk_length_in_kb': '64', 'class': 'org.apache.cassandra.io
.compress.LZ4Compressor'}
    AND crc_check_chance = 1.0
    AND dclocal_read_repair_chance = 0.1
    AND default_time_to_live = 0
```

Creating a column family

```
cassandra@cqlsh:catalog> describe product;
CREATE TABLE catalog.product (
    productid text PRIMARY KEY,
    brand text,
    breadth int,
    height int,
    length int,
    publisher text,
          :
    AND crc_check_chance = 1.0
    AND dclocal_read_repair_chance = 0.1
    AND default_time_to_live = 0
    AND gc_grace_seconds = 864000
    AND max_index_interval = 2048
    AND memtable_flush_period_in_ms = 0
    AND min_index_interval = 128
    AND read_repair_chance = 0.0
    AND speculative_retry = '99PERCENTILE';
```

read repair is a process that is done by cassandra in the background to make all replicas of data consistent

this property dictates the probability of read repair being invoked for requests to replicas in the same data center

Creating a column family

```
cassandra@cqlsh:catalog> describe product;
CREATE TABLE catalog.product (
    productid text PRIMARY KEY,
    brand text,
    breadth int,
    height int,
    length int,
    publisher text,
          :
    AND crc_check_chance = 1.0
    AND dclocal_read_repair_chance = 0.1
    AND default_time_to_live = 0
    AND gc_grace_seconds = 864000
    AND max_index_interval = 2048
    AND memtable_flush_period_in_ms = 0
    AND min_index_interval = 128
    AND read_repair_chance = 0.0
    AND speculative_retry = '99PERCENTILE';
```

default expiration time for a column family in seconds, Used for mapreduce jobs to create temporary tables

if you set it to 10 secs, all the inserted rows will be marked to be deleted in 10secs

Creating a column family

when we delete data from cassandra, internally cassandra marks it with a tombstone

```
cassandra@cqlsh:catalog> describe product;

CREATE TABLE catalog.product (
    productid text PRIMARY KEY,
    brand text,
    breadth int,
    height int,
    length int,
    publisher text,
          :
    AND crc_check_chance = 1.0
    AND dclocal_read_repair_chance = 0.1
    AND default_time_to_live = 0
    AND gc_grace_seconds = 864000
    AND max_index_interval = 2048
    AND memtable_flush_period_in_ms = 0
    AND min_index_interval = 128
    AND read_repair_chance = 0.0
    AND speculative_retry = '99PERCENTILE';
```

This property specifies how long this data sits around

During compaction, tombstone marked data is deleted permanently from disk

Creating a column family

```
cassandra@cqlsh:catalog> describe product;
CREATE TABLE catalog.product (
    productid text PRIMARY KEY,
    brand text,
    breadth int,
    height int,
    length int,
    publisher text,
          :
    AND crc_check_chance = 1.0
    AND dclocal_read_repair_chance = 0.1
    AND default_time_to_live = 0
    AND gc_grace_seconds = 864000
    AND max_index_interval = 2048
    AND memtable_flush_period_in_ms = 0
    AND min_index_interval = 128
    AND read_repair_chance = 0.0
    AND speculative_retry = '99PERCENTILE';
```

the default value is very large (10 days)

This can be tuned as per our requirement

Creating a column family

```
cassandra@cqlsh:catalog> describe product;
CREATE TABLE catalog.product (
    productid text PRIMARY KEY,
    brand text,
    breadth int,
    height int,
    length int,
    publisher text,
          :
    AND crc_check_chance = 1.0
    AND dclocal_read_repair_chance = 0.1
    AND default_time_to_live = 0
    AND gc_grace_seconds = 864000
    AND max_index_interval = 2048
    AND memtable_flush_period_in_ms = 0
    AND min_index_interval = 128
    AND read_repair_chance = 0.0
    AND speculative_retry = '99PERCENTILE';
```

The first component of a table's primary key is called the partition key

Partition keys are stored in the SSTable index file

Creating a column family

```
cassandra@cqlsh:catalog> describe product;

CREATE TABLE catalog.product (
    productid text PRIMARY KEY,
    brand text,
    breadth int,
    height int,
    length int,
    publisher text,
          :
          :
    AND crc_check_chance = 1.0
    AND dclocal_read_repair_chance = 0.1
    AND default_time_to_live = 0
    AND gc_grace_seconds = 864000
    AND max_index_interval = 2048
    AND memtable_flush_period_in_ms = 0
    AND min_index_interval = 128
    AND read_repair_chance = 0.0
    AND speculative_retry = '99PERCENTILE';
```

Partition keys may also be stored in a summary file is created by sampling the indexes

This specifies the sample interval so the size of the summary files can be controlled

Creating a column family

```
cassandra@cqlsh:catalog> describe product

CREATE TABLE catalog.product (
    productid text PRIMARY KEY,
    brand text,
    breadth int,
    height int,
    length int,
    publisher text,
          :
    AND crc_check_chance = 1.0
    AND dclocal_read_repair_chance = 0.1
    AND default_time_to_live = 0
    AND gc_grace_seconds = 864000
    AND max_index_interval = 2048
    AND memtable_flush_period_in_ms = 0
    AND min_index_interval = 128
    AND read_repair_chance = 0.0
    AND speculative_retry = '99PERCENTILE';
```

Data in cassandra is first written to commit log (on disk) and MemTable (in memory) and then flushed to SSTable file on disk

a force flush to SSTable is triggered using this property

Creating a column family

```
cassandra@cqlsh:catalog> describe product;
CREATE TABLE catalog.product (
    productid text PRIMARY KEY,
    brand text,
    breadth int,
    height int,
    length int,
    publisher text,
          :
    AND crc_check_chance = 1.0
    AND dclocal_read_repair_chance = 0.1
    AND default_time_to_live = 0
    AND gc_grace_seconds = 864000
    AND max_index_interval = 2048
    AND memtable_flush_period_in_ms = 0
    AND min_index_interval = 128
    AND read_repair_chance = 0.0
    AND speculative_retry = '99PERCENTILE';
```

by default no value is set for this property

if value is set for this property, then memtable is force flushed into disk after the time period set by this property has expired

Creating a column family

```
cassandra@cqlsh:catalog> describe product;
CREATE TABLE catalog.product (
    productid text PRIMARY KEY,
    brand text,
    breadth int,
    height int,
    length int,
    publisher text,
          :
    AND crc_check_chance = 1.0
    AND dclocal_read_repair_chance = 0.1
    AND default_time_to_live = 0
    AND gc_grace_seconds = 864000
    AND max_index_interval = 2048
    AND memtable_flush_period_in_ms = 0
    AND min_index_interval = 128
    AND read_repair_chance = 0.0
    AND speculative_retry = '99PERCENTILE';
```

Summary file for partition keys is created by sampling the indexes using this value as the minimum interval between the keys

Example 6:
cqlsh
modify columnfamily

Modifying a Columnfamily

Add columns to a columnfamily

Let's add a column ,"modelId", of type text

```
cassandra@cqlsh:catalog> ALTER COLUMNFAMILY product ADD modelId text;
```

Command to modify
column family

Modifying a Columnfamily

Add columns to columnfamily

Let's add a column ,"modelid", of type text

```
cassandra@cqlsh:catalog> ALTER COLUMNFAMILY product ADD modelId text;
```

operator to add a column

Modifying a Columnfamily

Add columns to columnfamily

Let's add a column ,"modelid", of type text

```
cassandra@cqlsh:catalog> ALTER COLUMNFAMILY product ADD modelId text;
```

definition of column modelId i.e name and type

Modifying a Columnfamily

Change the properties of a columnfamily

Lets change the gc_grace_seconds to 1 day (86400 seconds)

```
cassandra@cqlsh:catalog> ALTER COLUMNFAMILY product WITH gc_grace_seconds=86400;
```

operator for updating the property

Modifying a Columnfamily

Change the properties of a columnfamily

Lets change the gc_grace_seconds to 1 day (86400 seconds)

```
cassandra@cqlsh:catalog> ALTER COLUMNFAMILY product WITH gc_grace_seconds=86400;
```

property name and new value

Modifying a Columnfamily

Change the properties of a columnfamily

Lets change the gc_grace_seconds to 1 day (86400 seconds)

```
cassandra@cqlsh:catalog> ALTER COLUMNFAMILY product WITH gc_grace_seconds=86400;
```

Let's verify this with the describe command

Modifying a Columnfamily

Change the properties of a columnfamily

```
cassandra@cqlsh:catalog> describe product;

 AND crc_check_chance = 1.0
 AND dclocal_read_repair_chance = 0.1
 AND default_time_to_live = 0
 AND gc_grace_seconds = 86400
 AND max_index_interval = 2048
```