cassandra 2.0.X Cheat Sheet

by connygy via cheatography.com/32168/cs/9964/

nodetool

bin/nodetool status	get the status of nodes, UN: up + nomal
bin/nodetool info -h 127.0.0.1	detailed infomation of the node 127.0.0.1
bin/status ring	get the ring information with the

system

bin/cassandra	start the cassandra; with -f run in the foreground
$ ps aus | grep cass	get the cassandra pid
$ kill pid	close the cassandra service
conf/cassandra.yaml	configuration file
config/log4j-server.properties	where is log fille written, size the max file size

cql - crud

source 'filename.cql'	run a file with cql commands
insert into	
insert into <table> (xxx,xxx) values('xxx','xxx')	insert value to table
sstableloader tool	
select * from <table>	
select xxx,xxx from <table>	
copy from	import .csv file

cql - crud (cont)

copy to	export .csv file
copy <table> (xxx,xxx) from 'file path' with header = true and delimiter = '|'	copy csv file example, notice: if a record already there and duplicated with the primary key with the file, thus the record will be simply replaced
bin/cassandra-cli	start cli (thrift)
use <dbname>	cli command, use keyspace
list <tablename>	cli command, list how the table is stored

cassandra-cli

bin/cassandra-cli	start cassandra-cli tool
use <keyspace>	go into the keyspace
list <table>	show the storage of the table
bin/nodetool flush home_security	flush the memtable to disk
bin/sstable2json /var/lib/cassandra data/home_security/activity/home_security_activity-jb-1-Data.db	see the sstable, notice: use the Data.db file

data modeling

no join	no join in cassandra, the query should just work in one talbe
select * from <table> where <partition key> = 'xxx' and <primary key> = "xxx"	`where` need to include one partiiton key
secondary index	a index beyond the partition key and clustering columns, for each secondary index, cassandra creates a hidden talbe on each node in the cluster, it doesn't improve the speed
create a table for each query	this can improve the speed
create index <index_name> on <table> (code_used)	create a secondary index
composite partition key	a partition key with more than one column
create table <tablename> (XXX XXX, ..., primary key((xxx, xxx), xxx))	create a composite partition key

By **connygy**
cheatography.com/connygy/

Not published yet.
Last updated 26th November, 2016.

Cheatography

cassandra 2.0.X Cheat Sheet

by connygy via cheatography.com/32168/cs/9964/

cql	
datastax.com/documentation/cql	cql documentation
bin/cqlsh	start cql comments
describe cluster	describe cluster
help <commend>	help
exit	exit
describe keyspaces	list all the databases
describe keyspace <dbname>	details about the database
create keyspace <dbname> with replication = {'class':'NetworkTopologyStrategy', 'dc1':3, 'dc2':2}	create a database across multiple data center
create keyspace <dbname> with replication = {'class'='SimpleStrategy', 'replication_factor'=1}	create a database in one cluster
drop keyspace <tablename>	delete a database
create table <tablename> (home_id text, datetime timestamp, event text, code_used text primary key (home_id, datetime)) with clustering order by (datetime DESC)	create a table
drop table <tablename>	delete table
use <dbname>	use a keyspace

cql (cont)	
ascii, bigint, blob, boolean, counter, decimal, double, float, inet, int, list, map, set, text, timestamp, uuid, timeuuid, varchar, varint	cql data types
primary key	a way to uniquely identify a record in a table
partition key	first primary key, to determine which node store the record. (old name: row key) Partitioner hash the partition key
create table <tablename> (...) with clustering order by (datetime desc)	define the order of table, it default is ascend, if descend, than it takes longer to write, since the record is inserted at the start of a partition, but improves read performance. The order can not be changed by the command "alter <table>"

applications	
planetcassandra.org/client-drivers-tool	cassandra drivers
Cluster cluster = Cluster.builder().addContactPoints("127.0.0.1", "127.0.0.2").build();	build a cluster with java driver, it is better more than one contact point exist

update data	
update <table> set xxx='xxx', xxx='xxx' where xxx='xxx'	update record
update location using ttl 100 set XXX=XXX, XXX=XXX where XXX=XXX and XXX=XXX	updating with time to live
delete	delete a value in a column, or a row or rows
delete column from <table> where ...	delete the column value where ...
delete from <table> where ...	delete a row where ...
truncate	delete all of the rows in a talbe
drop	delete a table or keyspaces
drop table <table>	
drop keyspace <keyspace>	

tombstone	
gc_grace_seconds	the minimum existence of the deleted record, it is 864000(10 days) by default
compaction	data deleted, then reclaim the disk space from deleted data
bin/nodetool compact	manually do the compaction, but it is usually automatically
TTL	Time To Live, a way to specify an expiration date for data that is being inserted
insert into location(xxx, xxx) values ('xxx', 'xxx') using ttl 30	inserted data will live 30 seconds
sstable2json <sstable>	in the records, "d": deletion (after TTL), "e": expire (before TTL)