CQL:

Quy tắc command:

Kết thúc câu lệnh bằng dấu ‘ ; ’

Không phân biệt ký tự hoa, ký tự thường, đều lưu trữ dưới dạng lowercase. Nếu muốn lưu trữ dưới dạng ký tự in hoa => dùng dấu nháy kép.

Cách đặt tên keyspace, bảng, cột…: bắt đầu bằng ký tự chữ hoặc số, không chứa ký tự đặc biệt, ngoại trừ dấu gạch dưới ( underscore ). Nếu muốn đặt tên với các ký tự đặc biệt => dùng dấu nháy kép.

Khai báo chuỗi, ngày: dùng dấu nháy đơn ( ‘ngày’ , ‘ 2017 – 12 – 01’). Ngoài ta, có thể dùng $$ để khai báo chuỗi ( $$ ngày $$)

Ký hiệu comment: -- hoặc // hoặc /\* \*/

Kiểu dữ liệu:

|  |  |
| --- | --- |
| Kiểu dữ liệu | Ý nghĩa |
| Ascii | Biểu diễn một chuỗi ký tự ASCII |
| Bigint | Số nguyên có dấu dài 64 bit |
| Blob | Lưu trữ các byte tùy ý trong hệ thập lục phân |
| Boolean | Kiểu logic: true hoặc false |
| Counter | Số nguyên 64 bit, giá trị cột này không thể thiết lập, chỉ có hai hoạt động trên cột này là: tăng và giảm. Không áp dụng câu lệnh INSERT cho các cột này, chỉ có thể dùng UPDATE (  UPDATE bang  SET counter\_column = counter\_column + 1 WHERE id = 1 ) |
| Date | yyyy-mm-dd |
| Decimal | Một biến có giá trị thập phân chính xác ( phù hợp cho tiền tệ ) |
| Double | Giá trị dấu chấm động dài 64 bit |
| Float | Giá trị dấu chấm động 32 bit |
| Inet | Chuỗi địa chỉ IP ( IPv4 hoặc IPv6) |
| Itn | Số nguyên có dấu dài 32 bit |
| Smallint | Số nguyên 2 byte ( 16 bit ) |
| Text | Chuỗi UTF-8, lưu trữ ký tự không phải mã ASCII |
| Time | Giá trị thời gian ( hh:mm:ss) |
| Timestamp | Ngày và giờ, chính xác đến mili giây ( yyyy-mm-dd hh:mm:ss) |
| Tinyint | Số nguyên 1 byte ( 8 bit) |
| Timeuuid | Lưu trữ phiên bản 1 UUID ( lưu trữ thời gian bằng các khoảng 100 nano giây ( 00:00:00.00 UTC ( 60 bit)). |
| Uuid | UUID ở định dạng chuẩn |
| Varchar | Chuỗi UTF-8 ( giống text) |
| Variant | Số nguyên với độ chính xác tùy ý. |

So sánh với SQL:

Giống:

Tạo bảng: CREATE TABLE IF NOT EXISTS tenBang ( id INT PRIMARY KEY);

Tạo index: CREATE INDEX tenChiMuc ON tenBang(tenCot);

Cập nhật bảng: ALTER TABLE tenBang ADD tenCot KieuDuLieu;

Thêm dữ liệu: INSERT INTO tenBang(id, myField) VALUES (1,7);

Truy vấn: SELECT \* FROM tenBang WHERE điều kiện;

SELECT COUNT(\*) FROM tenBang;

Khác:

* Không hỗ trợ: JOIN, GROUP BY, FOREIGN KEY ( viết và truy vấn dữ liệu nhanh hơn, chi phí đọc cao)
* Câu lệnh UPDATE giống với SQL, tuy nhiên, cơ chế lại khác. CQL cập nhật dữ liệu nhưng không cần đọc, nếu dòng không tồn tại thì sẽ được tạo mới ( giảm chi phí update)
* Cơ chế INSERT cũng có sự khác biệt: cũng không đọc dữ liệu, nếu dữ liệu đã tồn tại thì sẽ được thay thế.
* Time to live (TTL): cho phép cài đặt TTL cho một dòng dữ liệu, sau thời gian này dòng dữ liệu sẽ không thể được truy vấn. ( TTL được tính theo giây)
* Có lệnh để theo dõi đường truy vấn qua các node ( vì vậy chi phí đọc cao)
* Modeling:
  + Lưu trữ dữ liệu dưới dạng không chuẩn hóa
  + Lưu trữ dữ liệu có cấu trúc hoặc không có cấu trúc

CQL Collection

Có 3 loại collection:

1. Set

* Lưu trữ các dữ liệu với giá trị duy nhất, giá trị trùng lắp sẽ không được lưu trữ. Khi truy vấn, các giá trị trả về sẽ được sắp xếp, dù không được lưu trữ có thứ tự.
* Dùng để lưu trữ dữ liệu có mối quan hệ một – nhiều với cột khác

Ví dụ: Create table University.Teacher

(

id int,

Name text,

Email set<text>,

Primary key(id)

);

1. List

* Dữ liệu lưu trữ có thể giống nhau, và theo một thứ tự nhất định, có thể insert và truy vấn dữ liệu bằng index
* Dùng để lưu trữ dữ liệu có mối quan hệ nhiều nhiều với cột khác

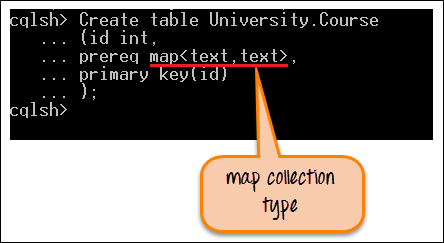
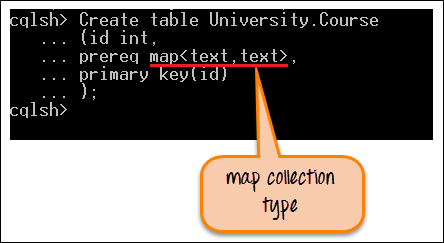
Ví dụ : cqlsh> **CREATE** **TABLE** cycling.upcoming\_calendar ( **year** int, **month** int, **events** **list**<text>, **PRIMARY** **KEY** ( **year**, **month**) );



1. Map

* Các dữ liệu được lưu trữ dưới dạng một cặp key-value. Một key chỉ có một giá trị và không có lưu trùng lắp
* Mỗi thành phần trong map là một cột Cassandra có thể bị thay đổi, thay thế, xóa, và truy vấn. Mỗi thành phần có một TTL riêng.

Ví dụ: map lưu tên môn học và tên môn học tiên quyết của môn học đó.



Khuyết:

Dung lượng lưu trữ: < 64KB;

Phải giữ một bộ collection nhỏ để tránh chi phí truy cập collection vì toàn bộ collection cần phải được duyệt qua.

Nếu lưu trữ > 64KB, thì chỉ có thể truy cập 64KB, nên có thế làm mất dữ liệu