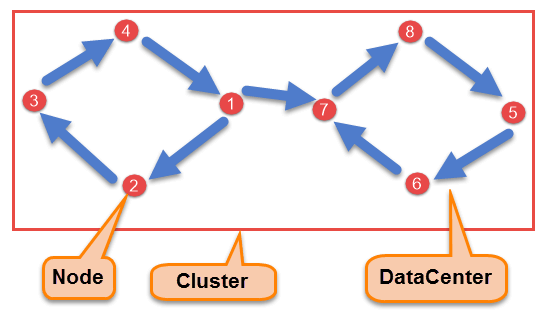
# Hoc cassandra

## Các component

There are following components in the Cassandra;

[](https://cdn.guru99.com/images/cassandra/021116_0524_CassandraAr1.png)

* **Node**

Node is the place where data is stored. It is the basic component of Cassandra.

* **Data Center**

A collection of nodes are called data center. Many nodes are categorized as a data center.

* **Cluster**

The cluster is the collection of many data centers.

* **Commit Log**

Every write operation is written to Commit Log. Commit log is used for crash recovery.

* **Mem-table**

After data written in Commit log, data is written in Mem-table. Data is written in Mem-table temporarily.

* **SSTable**

Khi Mem-TABLE đạt đến một ngưỡng nhất định, dữ liệu được flushed vào một tập tin đĩa SSTable

## DATA REPLICATION

– SAO CHEP DỮ LIỆU để không bị mất dữ liệu khi có sự cố

Khi sự cố phần cứng có thể xảy ra hoặc liên kết có thể bị hỏng bất cứ lúc nào trong quá trình xử lý dữ liệu, cần có giải pháp để cung cấp bản sao lưu khi sự cố xảy ra. Vì vậy, dữ liệu được nhân rộng để đảm bảo không có điểm duy nhất của thất bại.

Cassandra đặt bản sao dữ liệu trên các nút khác nhau dựa trên hai yếu tố này.

* Nơi đặt bản sao tiếp theo được xác định bởi ***replication strategy***.
* Trong khi tổng số bản sao được đặt trên các nút khác nhau được xác định bởi **replication factor**.

Một replication factor có nghĩa là chỉ có một bản sao dữ liệu trong khi ba replication factor có nghĩa là có ba bản sao của dữ liệu trên ba nút khác nhau.

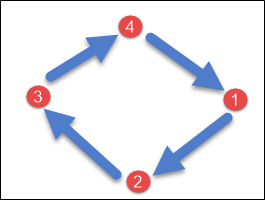
Để đảm bảo không có điểm lỗi duy nhất, replication factor phải là ba.

Có 2 loại replication strategies trong cassandra.

Simple strategy

SimpleStrategy được sử dụng khi bạn chỉ có một datacenter.

SimpleStrategy đặt bản sao đầu tiên trên nút được chọn bởi trình phân vùng. Sau đó, các bản sao còn lại được đặt theo chiều kim đồng hồ trong vòng Node. Đây là đại diện hình ảnh của SimpleStrategy



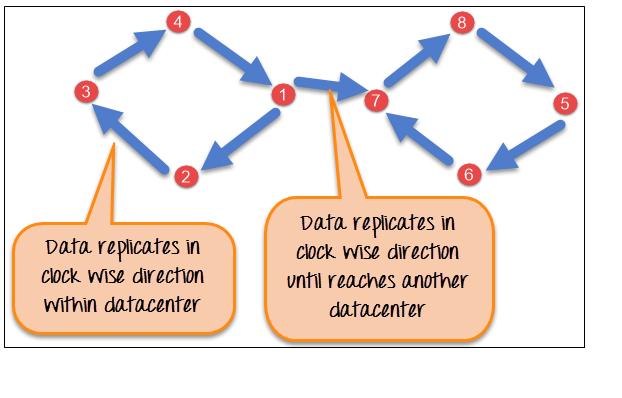
**NetworkTopologyStrategy**

**>= 2 datacenter**

Trong **NetworkTopologyStrategy**, bản sao được đặt riêng cho từng datacenter.

**NetworkTopologyStrategy** đặt các bản sao theo chiều kim đồng hồ trong vòng cho đến khi đạt đến nút đầu tiên trong một giá đỡ khác.

strategy này cố gắng đặt bản sao trên các giá khác nhau trong cùng một datacenter. Điều này là do lý do đôi khi thất bại hoặc vấn đề có thể xảy ra trong giá. Sau đó, bản sao trên các nút khác có thể cung cấp dữ liệu.



## Write operation

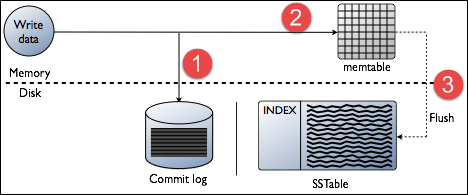
Điều phối viên gửi một yêu cầu ghi tới các bản sao. Nếu tất cả các bản sao đang lên, họ sẽ nhận được yêu cầu viết bất kể mức độ nhất quán của họ (consistency level)

Consistency level xác định có bao nhiêu nút sẽ phản hồi lại với sự thừa nhận thành công.

Nút sẽ phản hồi lại với sự thừa nhận thành công nếu dữ liệu được ghi thành công vào commit log and **memTable**

Ví dụ, trong một datacenter với replication bằng ba, ba bản sao sẽ nhận được yêu cầu viết. Nếu consistency level là một, chỉ có một bản sao sẽ phản hồi lại với sự thừa nhận thành công, và hai bản còn lại sẽ không hoạt động.

Giả sử nếu còn lại hai bản sao mất dữ liệu do các nút bị hỏng hoặc một số vấn đề khác, Cassandra sẽ làm cho hàng phù hợp với cơ chế sửa chữa tích hợp trong Cassandra.



**Read Operation**

Có 3 loại read request có thể gửi đến replication

* Yêu cầu trực tiếp
* Yêu cầu thông báo
* Đọc yêu cầu sửa chữa

Điều phối viên gửi yêu cầu trực tiếp đến một trong các bản sao.

Sau đó, điều phối viên gửi yêu cầu thông báo đến số lượng bản sao được chỉ định bởi consistency level và kiểm tra xem dữ liệu được trả về có phải là dữ liệu được cập nhật hay không.

Sau đó, điều phối viên gửi yêu cầu thông báo tới tất cả các bản sao còn lại. Nếu bất kỳ nút nào đưa ra giá trị ngày, yêu cầu sửa chữa đọc nền sẽ cập nhật dữ liệu đó. Quá trình này được gọi là cơ chế sửa chữa đọc.

## Tìm hiểu mô hình dữ liệu Cassandra

Mặc dù ngôn ngữ truy vấn Cassandra giống với ngôn ngữ SQL, các phương thức data model của chúng hoàn toàn khác nhau.

Trong Cassandra, một mô hình dữ liệu xấu có thể làm giảm hiệu suất, đặc biệt là khi người dùng cố gắng thực hiện các khái niệm RDBMS trên Cassandra. Tốt nhất là hãy ghi nhớ một số quy tắc được nêu chi tiết bên dưới

In this tutorial, you will learn-

### [Cassandra Data Model Rules](https://www.guru99.com/cassandra-data-model-rules.html#1)

Trong Cassandra, viết không đắt. Cassandra không hỗ trợ join, group by OR mệnh đề, aggregation, vv Vì vậy, bạn phải lưu trữ dữ liệu của bạn theo cách mà nó phải được hoàn toàn có thể phục hồi. Vì vậy, các quy tắc này phải được lưu ý trong khi tạo mô hình dữ liệu trong Cassandra.

1 Tối đa hóa số lượng write

Trong Cassandra, viết rất rẻ. Cassandra được tối ưu hóa cho hiệu suất ghi cao. Vì vậy, hãy cố gắng tối đa hóa bài viết của bạn để có hiệu suất đọc tốt hơn và tính khả dụng của dữ liệu. Có một sự cân bằng giữa ghi dữ liệu và đọc dữ liệu. Vì vậy, tối ưu hóa hiệu suất đọc dữ liệu bằng cách tối đa hóa số lượng dữ liệu được ghi.

2 Tối đa hóa sao chép dữ liệu

Dữ liệu không chuẩn hóa và sao chép dữ liệu là defacto của Cassandra. Dung lượng ổ đĩa (hiện tại bây h không quan trọng nữa) không đắt hơn bộ nhớ, xử lý CPU và hoạt động IO. Vì Cassandra là một cơ sở dữ liệu phân tán nên việc sao chép dữ liệu cung cấp dữ liệu sẵn có tức thời và không có điểm lỗi nào.

Data Modeling Goals

1 Lan rộng dữ liệu đều xung quanh Cluster

Bạn muốn có một lượng dữ liệu bằng nhau trên mỗi nút của CLuster Cassandra. Dữ liệu được lan truyền đến các nút khác nhau dựa trên các khóa phân vùng là phần đầu tiên của khóa chính. Vì vậy, hãy cố gắng chọn số nguyên làm khóa chính để phân tán dữ liệu đồng đều xung quanh cụm.

2 Giảm thiểu số lượng phân vùng đọc trong khi truy vấn dữ liệu

Phân vùng(partition) là một nhóm các bản ghi có cùng khóa phân vùng. Khi truy vấn đọc được phát hành, nó thu thập dữ liệu từ các nút khác nhau từ các phân vùng khác nhau.

Nếu có nhiều phân vùng, thì tất cả các phân vùng này cần phải được truy cập để thu thập dữ liệu truy vấn.

Nó không có nghĩa là không nên tạo phân vùng. Nếu dữ liệu của bạn rất lớn, bạn không thể giữ số lượng dữ liệu khổng lồ đó trên một phân vùng. Phân vùng đơn sẽ bị chậm lại. Vì vậy, hãy thử chọn số lượng phân vùng cân bằng.

**Good Primary Key**

Here is the table MusicPlaylist.

Create table MusicPlaylist

(

SongId int,

SongName text,

Year int,

Singer text,

Primary key(SongId, SongName)

);

Songid là khóa phân vùng và SongName là cột phân nhóm

Dữ liệu sẽ được nhóm lại trên cơ sở của SongName. Chỉ có một phân vùng sẽ được tạo bằng SongId. Sẽ không có bất kỳ phân vùng nào khác trong bảng MusicPlaylist.

Việc truy xuất dữ liệu sẽ bị chậm bởi mô hình dữ liệu này do khóa chính xấu.

Here is another table MusicPlaylist.

Create table MusicPlaylist

(

SongId int,

SongName text,

Year int,

Singer text,

Primary key((SongId, Year), SongName)

);

In above example, table MusicPlaylist,

Songid và Year là khóa phân vùng và SongName là cột phân nhóm.

Dữ liệu sẽ được nhóm lại trên cơ sở của SongName. Trong bảng này, mỗi năm, một phân vùng mới sẽ được tạo ra. Tất cả các bài hát của năm sẽ nằm trên cùng một nút. Khóa chính này sẽ rất hữu ích cho dữ liệu. Việc truy xuất dữ liệu của chúng tôi sẽ nhanh chóng bằng mô hình dữ liệu này.

### [Model Your Data in Cassandra](https://www.guru99.com/cassandra-data-model-rules.html#2)

(Mô hình dữ liệu của bạn trong Cassandra)

Xác định những truy vấn bạn muốn hỗ trợ Trước hết, hãy xác định những truy vấn bạn muốn. Ví dụ, bạn cần? join, group by,..

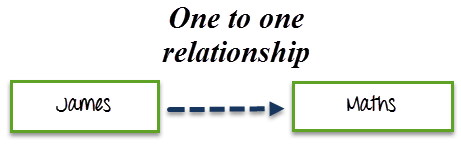
Tạo bảng theo truy vấn của bạn.

Tạo một bảng sẽ thỏa mãn các truy vấn của bạn. Cố gắng tạo một bảng theo cách sao cho số lượng phân vùng tối thiểu cần phải được đọc.

### [Handling One to One Relationship](https://www.guru99.com/cassandra-data-model-rules.html#3)

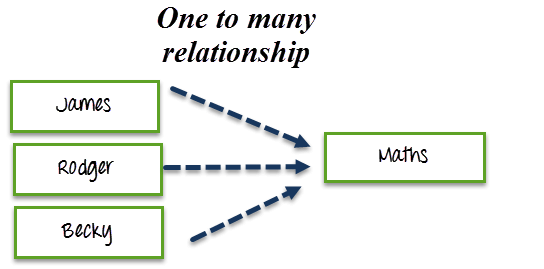
Có sự đối xững 1 1 giữa 2 bản

Vs 1 sinh viên có thể đk 1 khóa học, tôi muốn tìm kiếm một sinh viên mà khóa học đó đã được đk bởi sv nào.



### [Handling one to many relationships](https://www.guru99.com/cassandra-data-model-rules.html#4)

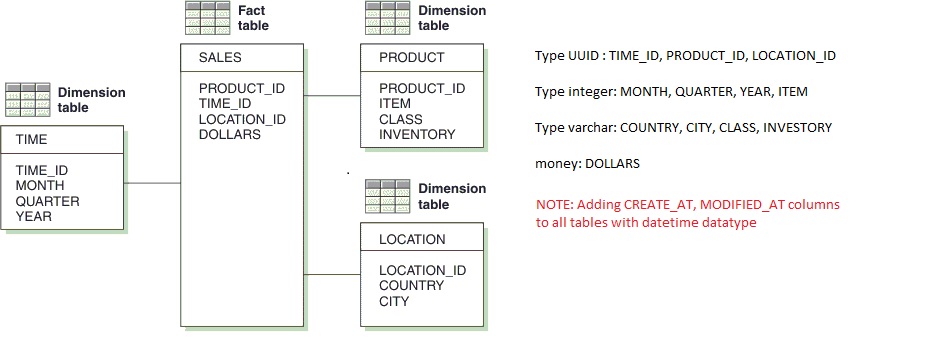
Vd 1 môn được đk bởi nhiều sv



### [Handling Many to Many Relationship](https://www.guru99.com/cassandra-data-model-rules.html#5)

Vd khóa học có thể được đk bới nhìu sv và sv cũng có thể đk nhìu môn cùng lúc

Create database



## Tạo keyspace (DB), xóa và sửa Keyspace

### Create keyspace

Keyspace là một đối tượng chứa các họ cột, các kiểu do người dùng định nghĩa. Trong Cassandra, Keyspace tương tự như Cơ sở dữ liệu RDBMS. Keyspace giữ các cột gia đình,  indexes, user defined types, data center awareness, strategy used in keyspace, replication factor, etc.

**Syntax**

Create keyspace KeyspaceName with replicaton={'class':strategy name,

'replication\_factor': No of replications on different nodes}

### Alter Keyspace

Thay đổi keyspace name, strategy name, replication\_factor,

**Syntax**

Alter Keyspace KeyspaceName with replication={'class':'StrategyName',

'replication\_factor': no of replications on different nodes}

with DURABLE\_WRITES=true/false

* DURABLE\_WRITES :DURABLE\_WRITES value can be altered by specifying its value true/false. By default, it is true. If set to false, no updates will be written to the commit log and vice versa.

### Drop Keyspace

**Syntax**

Drop keyspace KeyspaceName

# Cassandra Table: Create, Alter, Drop & Truncate

## *Tạo bảng: giống như RDBMS*

**Syntax**

Create table KeyspaceName.TableName

(

ColumnName DataType,

ColumnName DataType,

ColumnName DataType

.

.

.

Primary key(ColumnName)

) with PropertyName=PropertyValue;

**Primary key:**There are two types of primary key:

**Single Primary Key – Có 1 khóa chính**

**Syntax**

Primary key (ColumnName)

chỉ có một cột duy nhất. Cột đó còn được gọi là khóa phân vùng. Dữ liệu được phân đoạn trên cơ sở của cột đó. Dữ liệu được lan truyền trên các nút khác nhau trên cơ sở khóa phân vùng.

**Compound Primary Key –khóa hợp nhất**

**Syntax**

Primary key(ColumnName1,ColumnName2 . . .)

Trong cú pháp trên, ColumnName1 là khóa phân vùng và ColumnName2 là khóa Clustering. Dữ liệu sẽ được phân đoạn trên cơ sở ColumnName1 và dữ liệu sẽ được nhóm lại trên cơ sở ColumnName2. Clustering là quá trình sắp xếp dữ liệu trong phân vùng.

**Compound Partitioning key**

**syntax**

Primary Key((ColumnName1,ColumnName2),ColumnName3...))

Trong cú pháp trên, ColumnName1 và ColumnName2 là khóa phân vùng ghép. Dữ liệu sẽ được phân đoạn trên cơ sở của cả hai cột ColumnName1 và ColumnName2 và dữ liệu sẽ được nhóm lại trên cơ sở của ColumnName3. Nếu bạn có quá nhiều dữ liệu trên một phân vùng. Sau đó, phím phân vùng ghép được sử dụng. key phân vùng hợp chất được sử dụng để tạo nhiều phân vùng cho dữ liệu.

## Cassandra Alter table

Command 'Alter Table' được sử dụng để thả cột, thêm một cột mới, thay đổi tên cột, thay đổi kiểu cột và thay đổi thuộc tính của bảng

**Syntax**

Following is the syntax of command 'Alter Table.'

Alter table KeyspaceName.TableName +

Alter ColumnName TYPE ColumnDataype |

Add ColumnName ColumnDataType |

Drop ColumnName |

Rename ColumnName To NewColumnName |

With propertyName=PropertyValue

## Drop Table

Command 'Drop table' drops specified table including all the data from the keyspace. Before dropping the table, Cassandra takes a snapshot of the data not the schema as a backup.

**Syntax**

Drop Table KeyspaceName.TableName

## Truncate Table

Xóa tất cả dữ liệu của bảng

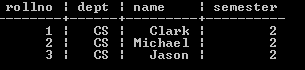
Command 'Truncate table' removes all the data from the specified table. Before truncating the data, Cassandra takes the snapshot of the data as a backup.

**Syntax**

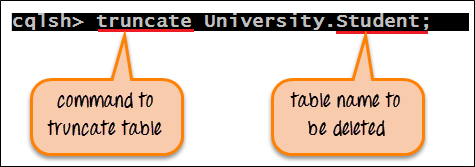
Truncate KeyspaceName.TableName

**Execution**

There are three records in the table Student. These are the records in the table.

[](https://cdn.guru99.com/images/cassandra/021116_1117_CassandraCQ10.png)

Here is the snapshot of the executed command 'Truncate table' that will remove all the data from the table Student.

[](https://cdn.guru99.com/images/cassandra/021116_1117_CassandraCQ11.png)

After successful execution of the command 'Truncate Table', all the data will be removed from the table Student.

Here is the snapshot of the database state where there are no records in the table Student.

[](https://cdn.guru99.com/images/cassandra/021116_1117_CassandraCQ12.png)