

Cassandra

1. Environnement

1.1. Linux

1.1.1. Installation et démarrage via les paquets Debian

cf <https://www.globo.tech/learning-center/install-cassandra-ubuntu-16/>

```
$ echo "deb http://debian.datastax.com/community stable main" |  
sudo tee -a /etc/apt/sources.list.d/cassandra.sources.list  
$ curl -L http://debian.datastax.com/debian/repo_key | sudo  
apt-key add -  
$ sudo apt-get update  
$ sudo apt-get install cassandra -y
```

Démarrage :

```
$ systemctl start cassandra  
$ systemctl enable cassandra
```

1.1.2. Installation et démarrage via Docker

cf <http://blog.ditullio.fr/2016/06/10/docker-docker-basics-cassandra/>

Dans un terminal :

```
$ curl -fsSL https://get.docker.com/ | sh  
$ sudo usermod -aG docker drussier
```

Fermer votre session et en ouvrir une autre pour que la dernière commande soit prise en compte.

Pour démarrer, dans un terminal :

```
$ docker run -d --name=cass1 --net=host cassandra
```

Pour effectuer les requêtes en langage cqlsh (dans le même terminal):

```
$ docker exec -ti cass1 cqlsh localhost
```

1.1.3. Arrêt (à la fin du TP)

Dans un terminal :

```
$ docker stop cass1
```

1.2. Mac

1.2.1. Installation

Ecrire les commandes ci-dessous ligne par ligne dans un terminal.

Installation de Homebrew :

```
$ ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

Installation de Python :

```
$ brew install python
$ echo 'export PATH="/usr/local/opt/python/libexec/bin:$PATH"' >> ~/.bash_profile
$ source ~/.bash_profile
$ pip install cql
$ pip install cassandra-driver
```

Installation de Cassandra :

```
$ brew install cassandra
$ mkdir -p ~/Library/LaunchAgents/
$ cp /usr/local/Cellar/cassandra/3.11.1/homebrew.mxcl.cassandra.plist
~/Library/LaunchAgents/
```

1.2.2. Démarrage

Démarrage de Cassandra :

```
$ launchctl load ~/Library/LaunchAgents/homebrew.mxcl.cassandra.plist
```

Cassandra shell pour faire les requêtes :

```
$ cqlsh
```

1.2.3. Arrêt (à la fin du TP)

Dans un terminal :

```
$ launchctl stop homebrew.mxcl.cassandra
```

1.3. Documentation

Cassandra :

<http://cassandra.apache.org/doc/latest/>

Le langage CQL est très proche du SQL:

http://docs.datastax.com/en/cql/3.1/cql/cql_intro_c.html

<https://www.datastax.com/dev/blog/a-deep-look-to-the-cql-where-clause>

Cependant, le filtrage sur des colonnes qui ne font pas partie de la clé primaire a un comportement particulier :

<https://www.datastax.com/dev/blog/allow-filtering-explained-2>

2. TP

2.1. Keyspace

Création d'une base de données ECOLE:

```
> CREATE KEYSPACE IF NOT EXISTS ecole WITH REPLICATION = { 'class'
: 'SimpleStrategy', 'replication_factor': 3 };
```

On sélectionne la base de données à utiliser pour les requêtes :

```
> USE ecole;
```

2.2. Création des tables

On va insérer des données “manuellement”, ie il faut copier/coller les instructions ci-dessous dans le terminal (cqlsh).

2.2.1. Table Cours

```
> CREATE TABLE Cours (
idCours INT, Intitule VARCHAR, Responsable INT, Niveau VARCHAR,
nbHeuresMax INT, Coeff INT,
PRIMARY KEY ( idCours )
);
> CREATE INDEX fk_Enseignement_Enseignant_idx ON Cours (
Responsable ) ;
```

2.2.2. Table Enseignant

```
> CREATE TABLE Enseignant (
idEnseignant INT, Nom VARCHAR, Prenom VARCHAR, status VARCHAR,
PRIMARY KEY ( idEnseignant )
);
```

On vérifie que les tables ont bien été créées :

```
> DESC ecole;
```

2.3. Table Enseignants

```
INSERT INTO Enseignant (idEnseignant,Nom,Prenom,status) VALUES
(1,'Travers','Nicolas','Vacataire');
INSERT INTO Enseignant (idEnseignant,Nom,Prenom,status) VALUES
(2,'Mourier','Pascale','Titulaire');
INSERT INTO Enseignant (idEnseignant,Nom,Prenom,status) VALUES
(3,'Boisson','Francois','Vacataire');
```

```

INSERT INTO Enseignant (idEnseignant,Nom,Prenom,status) VALUES
(4,'Mathieu','Eric','Titulaire');
INSERT INTO Enseignant (idEnseignant,Nom,Prenom,status) VALUES
(5,'Chu','Chengbin','Titulaire');
INSERT INTO Enseignant (idEnseignant,Nom,Prenom,status) VALUES
(6,'Boutin','Philippe','Titulaire');
INSERT INTO Enseignant (idEnseignant,Nom,Prenom,status) VALUES
(7,'Escribe','Julien','Vacataire');
INSERT INTO Enseignant (idEnseignant,Nom,Prenom,status) VALUES
(8,'Znaty','David','Vacataire');
INSERT INTO Enseignant (idEnseignant,Nom,Prenom,status) VALUES
(9,'Abal-Kassim','Cheik Ahamed','Vacataire');

```

2.4. Table Cours

```

INSERT INTO Cours
(idCours,Intitule,Responsable,Niveau,nbHeuresMax,Coeff) VALUES
(1,'Introduction aux Bases de Donnees',1,'M1',30,3);
INSERT INTO Cours
(idCours,Intitule,Responsable,Niveau,nbHeuresMax,Coeff) VALUES
(2,'Immeubles de Grandes Hauteurs',4,'M1',30,2);
INSERT INTO Cours
(idCours,Intitule,Responsable,Niveau,nbHeuresMax,Coeff) VALUES
(3,'Production et distribution de biens et de ser',5,'M1',30,2);
INSERT INTO Cours
(idCours,Intitule,Responsable,Niveau,nbHeuresMax,Coeff) VALUES
(4,'Bases de Donnees Avancees',1,'M2',30,5);
INSERT INTO Cours
(idCours,Intitule,Responsable,Niveau,nbHeuresMax,Coeff) VALUES
(5,'Architecture des Systemes Materiel',6,'M2',8,1);
INSERT INTO Cours
(idCours,Intitule,Responsable,Niveau,nbHeuresMax,Coeff) VALUES
(6,'IT Business / Introduction',7,'M2',20,3);
INSERT INTO Cours
(idCours,Intitule,Responsable,Niveau,nbHeuresMax,Coeff) VALUES
(7,'IT Business / Strategie et Management',8,'M2',10,1);

```

On vérifie le contenu de la table :

```
> SELECT * FROM Cours;
```

2.5. Requêtes

Pour interroger la base de données, la syntaxe CQL (pour Cassandra Query Language) est fortement inspirée de SQL. Pour la suite des exercices, exprimer en CQL la requête demandée :

1. Lister tous les cours

2. Liste des intitulés de cours
3. Nom de l'enseignant n°4
4. Intitulé des cours du responsable n° 1
5. Intitulé des cours dont le nombre d'heures maximum est égal à 30
6. Intitulé des cours dont le responsable est '1' et dont le niveau est 'M1'
Utiliser 'ALLOW FILTERING'
7. Intitulé des cours dont les responsables ont une valeur inférieure à 5
8. Compter le nombre de lignes retournées par la requête précédente
Utiliser 'COUNT(*)'

2.5. Requêtes avancées

9. La requête ci-dessous ne fonctionne pas, trouver une solution pour la faire fonctionner :

```
select nom, prenom from enseignant where status='Vacataire';
```

10. Donner les intitulés des cours dont le statut du responsable est 'Vacataire' ;
11. La jointure n'est pas possible avec CQL (à cause du stockage par hachage distribué). Nous allons créer une solution alternative en fusionnant les deux tables. Pour cela, il va nous falloir choisir d'intégrer la table 'Enseignant' dans la table 'Cours', nous appellerons cette table 'coursEnseignant'. Trois possibilités sont disponibles pour le faire : SET, LIST, MAP, SubType. Choisissez la solution qui permettra de faire un filtrage sur le statut de l'enseignant.

12. Insérer les données suivantes :

```
INSERT INTO CoursEnseignant
(idCours,Intitule,Responsable,Niveau,nbHeuresMax,Coeff) VALUES
(1,'Introduction aux Bases de
Donnees',{'idenseignant':'1','nom':'Travers','prenom':'Nicolas','s
tatus':'Vacataire'},'M1',30,3);
```

```
INSERT INTO CoursEnseignant
(idCours,Intitule,Responsable,Niveau,nbHeuresMax,Coeff) VALUES
(2,'Immeubles de Grandes
Hauteurs',{'idenseignant':'4','nom':'Mathieu','prenom':'Eric','sta
tus':'Titulaire'},'M1',30,2);
```

```
INSERT INTO CoursEnseignant
(idCours,Intitule,Responsable,Niveau,nbHeuresMax,Coeff) VALUES
(3,'Production et distribution de biens et de
ser',{'idenseignant':'5','nom':'Chu','prenom':'Chengbin','status':
'Titulaire'},'M1',30,2);
```

```
INSERT INTO CoursEnseignant
(idCours,Intitule,Responsable,Niveau,nbHeuresMax,Coeff) VALUES
(4,'Bases de Donnees
Avancees',{'idenseignant':'1','nom':'Travers','prenom':'Nicolas','
status':'Vacataire'},'M2',30,5);
```

```
INSERT INTO CoursEnseignant
(idCours,Intitule,Responsable,Niveau,nbHeuresMax,Coeff) VALUES
(5,'Architecture des Systemes
Materiel',{'idenseignant':'6','nom':'Boutin','prenom':'Philippe','
status':'Titulaire'},'M2',8,1);
```

```
INSERT INTO CoursEnseignant
(idCours,Intitule,Responsable,Niveau,nbHeuresMax,Coeff) VALUES
(6,'IT Business /
Introduction',{'idenseignant':'7','nom':'Escribe','prenom':'Julien
','status':'Vacataire'},'M2',20,3);
```

```
INSERT INTO CoursEnseignant
(idCours,Intitule,Responsable,Niveau,nbHeuresMax,Coeff) VALUES
(7,'IT Business / Strategie et
Management',{'idenseignant':'8','nom':'Znaty','prenom':'David','st
atus':'Vacataire'},'M2',10,1)
```

13. Créer un index sur le Niveau de la table CoursEnseignant

14. Donner les noms des responsables des cours (table CoursEnseignant) de niveau 'M1'

15. Donner l'intitulé des cours dont le responsable est vacataire

16. Nous allons inverser la fusion des tables en créant une table 'EnseignantCours' avec un sous-type 'Cours'. Pour pouvoir rajouter un ensemble de valeurs pour les cours d'un responsable, on peut utiliser : set, map ou list. Nous utiliserons le type map.

Créer le type 'Cours' et la table 'EnseignantCours'

17. Insérer les données suivantes :

```
INSERT INTO EnseignantCours (idEnseignant,Nom,Prenom,status,cours)
VALUES (1,'Travers','Nicolas','Vacataire',
{1:{idcours:1,intitule:'Introduction aux Bases de
Donnees',niveau:'M1',nbHeuresMax:30,Coeff:3},
4:{idcours:4,intitule:'Bases de Donnees
Avancees',niveau:'M2',nbHeuresMax:30,coeff:5}}});
```

```
INSERT INTO EnseignantCours (idEnseignant,Nom,Prenom,status,cours)
VALUES (2,'Mourier','Pascale','Titulaire', {});
```

```
INSERT INTO EnseignantCours (idEnseignant,Nom,Prenom,status,cours)
VALUES (3,'Boisson','Francois','Vacataire', {});
```

```
INSERT INTO EnseignantCours (idEnseignant,Nom,Prenom,status,cours)
VALUES (4,'Mathieu','Eric','Titulaire',
{4:{idcours:2,intitule:'Immeubles de Grandes
Hauteurs',niveau:'M1',nbHeuresMax:30,coeff:2}}});
```

```
INSERT INTO EnseignantCours (idEnseignant,Nom,Prenom,status,cours)
VALUES (5,'Chu','Chengbin','Titulaire', {});
```

```
INSERT INTO EnseignantCours (idEnseignant,Nom,Prenom,status,cours)
VALUES (6,'Boutin','Philippe','Titulaire',
{5:{idcours:5,intitule:'Architecture des Systemes
Materiel',niveau:'M2',nbHeuresMax:8,coeff:1}});
```

```
INSERT INTO EnseignantCours (idEnseignant,Nom,Prenom,status,cours)
VALUES (7,'Escribe','Julien','Vacataire',
{6:{idcours:6,intitule:'IT Business /
Introduction',niveau:'M2',nbHeuresMax:20,coeff:3}});
```

```
INSERT INTO EnseignantCours (idEnseignant,Nom,Prenom,status,cours)
VALUES (8,'Znaty','David','Vacataire', {7:{idcours:7,intitule:'IT
Business / Strategie et
Management',niveau:'M2',nbHeuresMax:10,coeff:1}});
```

```
INSERT INTO EnseignantCours (idEnseignant,Nom,Prenom,status,cours)
VALUES (9,'Abal-Kassim','Cheik Ahamed','Vacataire', {});
```

18. Créer un index sur le status de la table EnseignantCours

19. Donner les intitulés des cours dont le responsable est Vacataire