

**Master Mathématiques, Informatique, Décision, Organisation (MIDO)  
2ème année - MIAGE-IF App.**

**ANNEE 2022/ 2023**

**Désignation de l'enseignement** : NoSQL

**Nom du document** : TP sous Cassandra

**Rédacteur** : Maude Manouvrier

La reproduction de ce document par tout moyen que ce soit est interdite conformément  
aux articles L111-1 et L122-4 du code de la propriété intellectuelle

## I. Tester Cassandra via le tutoriel

Cassandra dispose de plusieurs tutoriels en ligne dont :

- <https://www.datastax.com/learn/cassandra-fundamentals/inserts-updates-deletes>
- <https://www.datastax.com/learn/cassandra-fundamentals/queries>

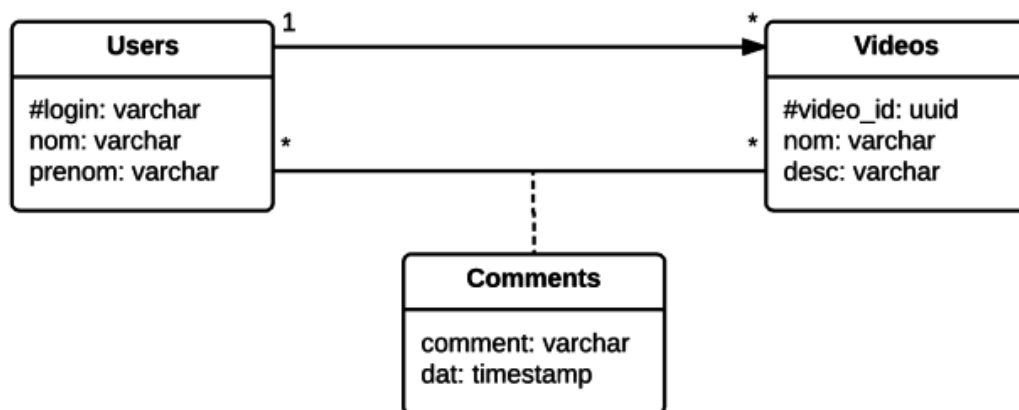
**Attention il faut un compte *github* pour manipuler ces tutoriels sur *gitpod*.**

**Vous pouvez tester chacun de ces tutoriels pour vous imprégner des différentes commandes possibles.**

- Sélectionner *Local deployment of open-source Apache Cassandra*
- Pour tester les commandes indiquées dans chaque tutorial, il suffit de cliquer sur les instructions du cadre gris dans la partie gauche de la fenêtre, et elles seront automatiquement exécutées dans la partie droite.
- Vous pouvez tester vos propres commandes en les saisissant directement dans la partie droite de l'écran (*Terminal*).
- Le détail de chaque commande de *Cassandra Query Language* (CQL) est disponible sur : <https://cassandra.apache.org/doc/latest/cql/>

## II. A vous de jouer

On souhaite modéliser la base de données suivante<sup>1</sup> (vue en cours) :



- Les utilisateurs possèdent un login unique, ainsi que leur nom et leur prénom.
- Un utilisateur possède plusieurs vidéos
- Une vidéo possède un id, un nom et une description et est publiée par un utilisateur.
- Les vidéos sont commentées par plusieurs utilisateurs.

<sup>1</sup> Repris et adapté de <https://stph.scenari-community.org/contribs/nos/Cassandra1/co/exercice1.html>

***Dans la suite du TP, vous allez pratiquer le CQL (Cassandra Query Language) en testant vos propres commandes. La session est temporaire, pensez donc à copier vos instructions dans un fichier texte si vous souhaitez les conserver. Vous pouvez exécuter une requête précédente en utilisant les flèches du clavier.***

*Pour la suite, vous pouvez vous inspirer des commandes du tutoriel précédent ou consulter la documentation en ligne :*

*<https://cassandra.apache.org/doc/latest/cassandra/cql/index.html>.*

1. Utiliser la console à droite du tutoriel pour y saisir vos propres commandes, ou créez-vous une base de données Astra en lançant <https://astra.datastax.com/> (Attention il faut un email mais c'est gratuit - il faut sélectionner la version gratuite).

2. Si vous utilisez la console du tutoriel :

- a. Relancer le tutoriel, relancez cassandra et la console cql
- b. Créer un nouveau *keyspace* et placez-vous dans ce *keyspace* :

```
CREATE KEYSPACE IF NOT EXISTS User_Video WITH
replication = { 'class':
'NetworkTopologyStrategy', 'DC-Houston': 1 };
USE User_Video;
```

3. Si vous souhaitez créer une base de données dans <https://astra.datastax.com/> :

- a. Créer une base de données *UserVideo* et un *KEYSPACE*, puis aller dans CQL console une fois la base créée (cela peut prendre du temps).
- b. Dans la console CQL placez-vous dans le *KEYSPACE* que vous avez créé avec la commande : `USE nom_du_keyspace ;` Si cela ne fonctionne pas, recréer éventuellement un autre *keyspace*.

4. Créer<sup>2</sup> les relations correspondantes (les types CQL sont disponibles à l'adresse <https://cassandra.apache.org/doc/latest/cql/types.html> )

```
CREATE TABLE videos (video_id int, video_name varchar, video_uploaded
timestamp, user_login varchar, PRIMARY KEY(video_id));
```

```
CREATE TABLE users (login varchar, firstname varchar, lastname
varchar, PRIMARY KEY(login));
```

```
CREATE TABLE user_videos_index (login varchar, video_id int,
video_name varchar, PRIMARY KEY(login, video_id));
```

```
CREATE TABLE comments_by_video (video_id int, user_login varchar,
comment_dat timestamp, comment varchar, PRIMARY KEY(video_id,
user_login));
```

```
CREATE TABLE comments_by_user (user_login varchar, video_id int,
comment_dat timestamp, comment varchar, PRIMARY KEY(user_login,
video_id));
```

---

<sup>2</sup> Sous <https://astra.datastax.com/>, pour copier vous devez utiliser le menu accessible via la souris ; le Ctrl V ne fonctionne pas.

## 5. Exécuter les insertions suivantes :

```

INSERT INTO users(login, firstname, lastname) VALUES ('cschmidt',
'chris', 'SCHMIDT');

INSERT INTO users(login, firstname, lastname) VALUES ('saraj00n',
'Sarah', 'JUNE');

INSERT INTO videos (video_id, video_name, video_uploaded, user_login)
VALUES (1,'Keyboard Cat', dateof(now()), 'cshmidt');

INSERT INTO videos (video_id, video_name, video_uploaded, user_login)
VALUES (2,'nyan cat', dateof(now()), 'saraj00n');

INSERT INTO user_videos_index (login, video_id, video_name) VALUES
('cschmidt',1, 'Keyboard Cat');

INSERT INTO user_videos_index (login, video_id, video_name) VALUES
('saraj00n',2, 'nyan cat');

INSERT INTO comments_by_user (user_login, video_id, comment_dat,
comment) VALUES ('saraj00n', 1, '2017-01-15 12:00+0000', 'first');

INSERT INTO comments_by_video (video_id, user_login, comment_dat,
comment) VALUES (1, 'saraj00n', '2017-01-15 12:00+0000', 'first');

INSERT INTO comments_by_user (user_login, video_id, comment_dat,
comment) VALUES ('MasterTroll', 1, '2017-01-15 12:05+0000', 'thats
lame');

INSERT INTO comments_by_video (video_id, user_login, comment_dat,
comment) VALUES (1, 'MasterTroll', '2017-01-15 12:05+0000', 'thats
lame');

```

## 6. Faire une requête pour visualiser le contenu de chaque table.

Pour information, les couleurs, lors de l’affichage des attributs d’une table, ont une sémantique. Sur l’affichage ci-dessous, par exemple :

```

cqlsh:uservideo> SELECT * FROM user_videos_index;

```

login	video_id	video_name
cschmidt	1	Keyboard Cat
saraj00n	2	nyan cat

L’attribut dont le nom apparaît en rouge correspond à la clé partition, celui dont le nom apparaît en bleu à la clé de *clustering*, et le violet est associé aux autres attributs.

- Qu’identifiez-vous au niveau des tables `comments_by_video` et `comments_by_user` par rapport à la table `users` ? Corriger le problème.
- Faire une requête permettant d’afficher les attributs de la vidéo numéro 1.
- Faire une requête permettant d’afficher tous les id des vidéos insérées par la personne de login `cshmidt`.
- Créer un index sur l’attribut `user_login` de la table `videos` et refaire la requête de la question précédente.
- Que se passe-t-il lorsque l’on fait l’insertion suivante :

```

INSERT INTO users (login, firstname, lastname) VALUES ('saraj00n', 'Sarah', 'Junior');

```

12. Modifier la table `videos` pour ajouter une colonne `production` de type `SET<TEXT>` (cf. <https://cassandra.apache.org/doc/latest/cql/ddl.html#alter-table>).

Pour visualiser le schéma de la table : `DESCRIBE TABLE videos ;`

13. Exécuter l'insertion suivante :

```
UPDATE videos
SET production = { 'Walt Disney Pictures',
                  'Roth Films' }
WHERE video_id = 1;
```

14. Ajouter un nouveau producteur à la vidéo numéro 1.

15. Faire une requête sur la table `videos` pour sélectionner les vidéos dont l'attribut `production` contient (mot-clé `CONTAINS`) le producteur que vous venez d'ajouter (cf. [https://docs.datastax.com/en/cql-oss/3.1/cql/ddl/ddl\\_query\\_collection\\_c.html](https://docs.datastax.com/en/cql-oss/3.1/cql/ddl/ddl_query_collection_c.html))

16. Modifier la table `users` pour ajouter un attribut `children` de type `MAP<TEXT, INT>`, représentant des couples (prénom d'enfant, âge), puis exécuter l'instruction suivante :

```
UPDATE users
SET children = {'Tom':4, 'Sofia':12}
WHERE login='cschmidt';
```

17. Faire une requête permettant de récupérer l'utilisateur ayant comme enfant 'Tom' (cf. [https://docs.datastax.com/en/cql-oss/3.1/cql/ddl/ddl\\_query\\_collection\\_c.html](https://docs.datastax.com/en/cql-oss/3.1/cql/ddl/ddl_query_collection_c.html))

18. Faire une requête permettant de récupérer l'utilisateur ayant un enfant de 12 ans (cf. [https://docs.datastax.com/en/cql-oss/3.1/cql/ddl/ddl\\_query\\_collection\\_c.html](https://docs.datastax.com/en/cql-oss/3.1/cql/ddl/ddl_query_collection_c.html))

19. Créer le type suivant et ajouter un attribut de ce type dans la table `users` :

```
CREATE TYPE ADDRESS (
    street TEXT,
    city TEXT,
    state TEXT,
    postal_code TEXT
);
```

20. Ajouter un attribut `address` de type `ADDRESS` dans la table `users` puis exécuter l'instruction suivante :

```
UPDATE users
SET address = { street: '1100 Congress Ave',
               city: 'Austin',
               state: 'Texas',
               postal_code: '78701' }
WHERE login='cschmidt';
```

21. Tester les 2 requêtes suivantes :

```
SELECT address.city from users;
```

```
SELECT lastname FROM users WHERE address = { street: '1100  
Congress Ave', city: 'Austin', state: 'Texas',  
postal_code: '78701' };
```

22. Exécuter les instructions suivantes puis exécuter à nouveau les requêtes de la question précédentes (en remplaçant address par address2).

```
ALTER TABLE users ADD address2 frozen<ADDRESS>;
```

```
UPDATE users
```

```
SET address2 = { street: '1100 Congress Ave',  
                city: 'Austin',  
                state: 'Texas',  
                postal_code: '78701' }
```

```
WHERE login='cschmidt';
```

23.

24. Faites le tutoriel de Cassandra suivant :

<https://www.datastax.com/learn/cassandra-fundamentals/tunable-consistency>

25. Tester votre compréhension de la cohérence avec Cassandra sur le quizz suivant :

[https://stph.scenari-community.org/contribs/nos/Cassandra3/co/3-Parametrer-Son-Cluster-Cassandra\\_2.html](https://stph.scenari-community.org/contribs/nos/Cassandra3/co/3-Parametrer-Son-Cluster-Cassandra_2.html)