

Fiche de TD/TP N°10- BDD NoSQL : Orienté Colonne avec Cassandra

Partie 1 : Installation et mise-en-œuvre :

1. Installer : cassandra for python: `python -m pip install cassandra-driver`
2. Lancer un container Cassandra avec docker en utilisant docker-compose
 - a. `docker-compose up -d`

docker-compose.yml

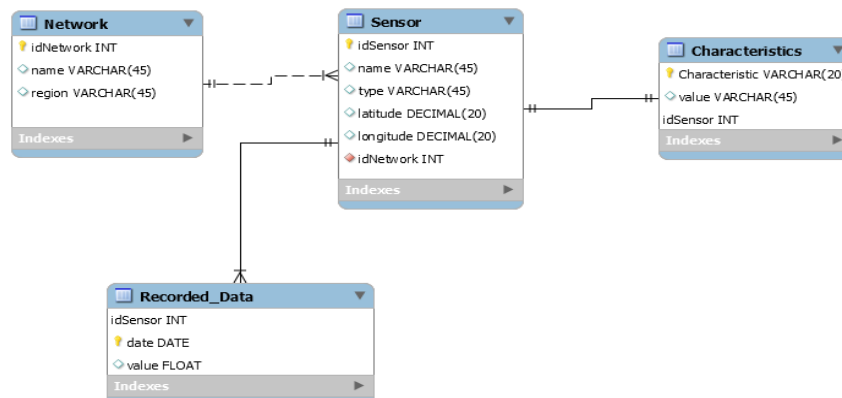
```
version: '3'
services:
  cass:
    image: datastax/dse-server:6.8.16-ubi7
    container_name: cass-dse
    ports:
      - "9042:9042"
    environment:
      - DS_LICENSE=accept
      - MAX_HEAP_SIZE=1000000000
```

Partie 2: Démo

1. Télécharger le notebook “**demo_cassandra_2023_esi-sba**”¹ et essayer de pratiquer les différents blocks.
 - a. Lancer l’invité de commande et positionner le dans le dossier dans lequel les notebooks sont enregistrés. Exécuter la commande `jupyter notebook`

Partie 3: Gestion d’un Sensor Network avec Cassandra/Python²

Supposons qu'on a un système IoT qui stocke les données capturées par différents Sensors en utilisant un modèle RELATIONNEL, tel que est décrit dans le schéma ER ci-dessous :



Imaginons que nous souhaitons analyser ces données à travers 3 requêtes analytiques:

- **Query1** → retourner les informations(name, nbSensors) de tous les networks d'une région donnée
- **Query2** → retourner les informations (name, type, localisation, characteristics) de tous les Sensors d'un Network donné.

¹ https://github.com/malkiAbdelhamid/CH3_Lab10_Cassandra_22-23.git

² https://github.com/malkiAbdelhamid/CH3_Lab10_Cassandra_22-23.git

- **Query3** → retourner le *AVG_measurements* par *Sensor*, par *année* et par *mois*. Ordonner le résultat par *année* (desc) et par *mois* (asc)
1. Créer les trois tables qui correspondent aux requêtes précédentes, en indiquant pour chacune: *Primary Key*, *Partition Key* et *Clustering Column*.
 2. Une fois vous créez les différentes tables, essayez de les peupler en se basant sur le contenu de la BDD relationnelle :

Network Table

idNetwork	name	region
1	esi-net	ouest
2	udi-net	ouest
3	alger-net	centre

Sensor Table

idSensor	name	type	latitude	longitude	idNetwork
1	temp-11	temp	31	-96	1
2	temp-12	temp	31	-97	1
3	temp-21	temp	32	-97	2
4	temp-31	temp	44	-111	3
5	temp-32	temp	45	-111	3

Characteristics Table

Characteristic	value	idSensor
accuracy	medium	1
sensitivity	high	1
accuracy	high	2
sensitivity	high	2
accuracy	low	3
sensitivity	medium	3
accuracy	medium	4
sensitivity	medium	4
accuracy	high	5
sensitivity	low	5

Recorded data Table

idSensor	date	value
1	2021-10-01	10
1	2021-10-31	12
1	2021-12-01	5
1	2021-12-31	-1
1	2022-10-01	14
1	2022-10-31	18
1	2022-12-01	10
1	2022-12-31	8
2	2021-10-01	11
2	2021-10-31	13
2	2021-12-01	6
2	2021-12-31	0
2	2022-10-01	14.5
2	2022-10-31	18.5
2	2022-12-01	11.5
2	2022-12-31	7.5

idSensor	date	value
3	2021-10-01	10
3	2021-10-31	15
3	2021-12-01	4
3	2021-12-31	1
3	2022-10-01	14
3	2022-10-31	18
3	2022-12-01	10
3	2022-12-31	7
4	2021-10-01	14
4	2021-10-31	18
4	2021-12-01	7
4	2021-12-31	6
4	2022-10-01	16
4	2022-10-31	20
4	2022-12-01	12
4	2022-12-31	10

idSensor	date	value
5	2021-10-01	11
5	2021-10-31	17
5	2021-12-01	5
5	2021-12-31	4
5	2022-10-01	12
5	2022-10-31	21
5	2022-12-01	10
5	2022-12-31	7

3. Valider votre modèle en exécutant les requêtes suivantes. Proposer une solution pour celles qui ne sont pas optimisées.
 - a. Retourner le nombre de sensors dans la région ouest"
 - b. Retourner les sensors qui sont placés dans le network "esi-net"
 - c. Retourner la température maximale enregistrée par le sensor "temp-11" pendant l'année 2021
 - d. Retourner la température maximale enregistrée par le sensor "temp-11" pendant le mois d'octobre de toutes les années