

Bases de données  
documentaires et distribuées,  
<http://b3d.bdpedia.fr>

Cassandra: CQL ... et ses limites

# Le langage CQL

Cassandra propose un langage, nommé CQL, inspiré de SQL, mais fortement restreint par l'absence de jointure.

CQL ne permet d'interroger qu'une seule table !

De plus, d'autres types de restrictions s'appliquent ! Elles s'expliquent toutes par la volumétrie potentielle des tables.

Nous allons étudier tout cela.

# Premiers pas

Sélectionnons tous les films.

```
select * from movies;
```

Ou seulement les 20 premiers.

```
select * from movies limit 20;
```

Retour en JSON

```
select JSON * from movies;
```

Accédons à un objet imbriqué.

```
select title, director.last_name from movies;
```

En revanche, on ne peut pas accéder aux éléments d'un ensemble.

```
select title, actors.last_name from movies;
```

# La clause where

Sélectionnons un film.

```
select * from movies where id='movie:33';
```

Ou plusieurs

```
select * from movies
  where id in ('movie:33', 'movie:44214', 'movie:29845');
```

Et si ce n'est pas la clé ?

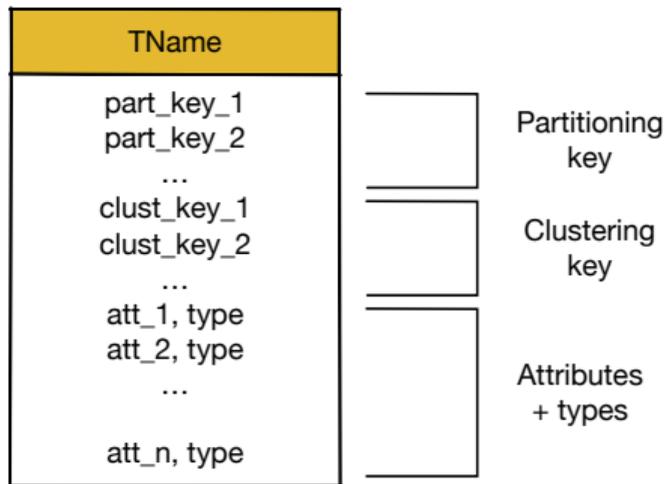
```
select * from movies where title='Elle' ;
```

Unable to execute CQL script. Cannot execute this query as it might involve data filtering and thus may have unpredictable performance.

Essayons de comprendre.



# Structure d'un schéma



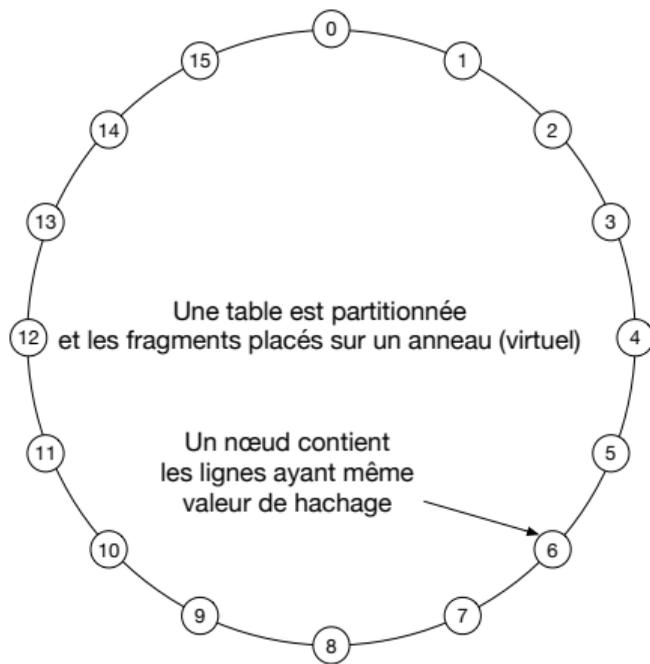
```
create table Tname (  
    part_key_1 type,  
    part_key_2 type,  
    ... type,  
    clust_key_1 type,  
    ... type,  
    att_1 type,  
    att_2 type,  
    ... type,  
    primary key (  
        (part_key_1, part_key2, ...),  
        (clust_key_1, clust_key_2, ...)  
    )  
)
```

# Exemple : les rôles d'un film

```
create table Roles (  
    id_film int,  
    id_artiste int,  
    role text,  
    artist frozen<artist>,  
    primary key (id_film, id_artiste)  
)
```

- La clé de partitionnement est l'identifiant du film
- La clé de regroupement est l'identifiant de l'artiste

# Clé de partitionnement = placement



- Pour chaque ligne on applique une fonction de hachage aux valeurs de la clé de **partitionnement**
- La valeur de hachage détermine le placement dans le système distribué
- $\Rightarrow$  une requête incluant comme critère une clé de partitionnement ne concerne qu'un seul serveur

# Clé de regroupement = contiguité

Roles		
id_film	id_artiste	Autre attributs
369	102	Gabin, rôle
369	104	Delon, rôle
...	...	...
386	34	Deneuve, rôle
386	104	Delon, rôle
386	234	...
386	276	
...	...	
392	12	
392	234	
...	...	

- Dans un fragment (sur un serveur) les lignes sont triées sur la clé primaire
- Pour une même valeur de partitionnement, les lignes sont **consécutives** et ordonnées sur la clé de regroupement
- $\Rightarrow$  une requête sur une préfixe de la clé primaire (part. + regroupement) correspond à un parcours **séquentiel** sur un seul nœud

# Restrictions sur les requêtes

Toute recherche sur des critères autres qu'un préfixe de la clé entraîne un parcours séquentiel

Cassandra est conçu pour de très grandes bases de données, et le rejet de ces requêtes séquentielles est une précaution.

Principe : Cassandra limite les requêtes à celles dont le coût est proportionnel à la taille du résultat.

On peut outrepasser avec la clause `allow filtering`.

```
select * from movies
      where country='US' and year=2020 allow filtering;
```