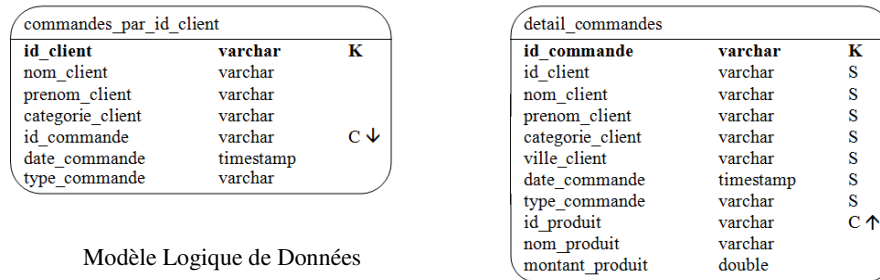


⑥ AB-Orders - NoSQL : Cassandra

On s'intéresse à l'entreprise AB-Orders™ qui a mis en place une Base de Données Cassandra afin de gérer les commandes de ses clients.

Le Modèle Logique de Données de la BD Cassandra vous est communiqué ci-dessous :



- Avec la structure des tables du Modèle Logique de Données indiqué, si on ne rajoute pas d'index secondaire ou de vue matérialisée, et qu'on n'utilise pas non plus la clause `ALLOW FILTERING` dans les requêtes, pensez-vous qu'il est possible de réaliser les requêtes suivantes en CQL (on vous demande de répondre aux questions sur votre feuille sans pour le moment écrire le code des requêtes) ?

 - identifiant des commandes passées par le client 'C1'. oui ☐ non ☐
 - identifiant des commandes passées par le client 'Delune'. oui ☐ non ☐
 - le nom des produits commandés par le client 'C1'. oui ☐ non ☐
 - identifiant des commandes qui contiennent le produit 'P1'. oui ☐ non ☐
 - le montant du produit 'P1' dans la commande '001'. oui ☐ non ☐
 - l'identifiant des commandes passées par le client 'C3' ou 'C4'. oui ☐ non ☐
 - identifiant de tous les clients (sans doublon). oui ☐ non ☐
 - identifiant et le nom de tous les clients (sans doublon). oui ☐ non ☐
 - les différentes villes des clients (sans doublon). oui ☐ non ☐
 - date de la commande '001'. oui ☐ non ☐
 - identifiant et le nom et le prénom des clients, classés par leur id. oui ☐ non ☐
 - identifiant, date et type des commandes passées par le client 'C1', classées de la plus vieille à la plus récente. oui ☐ non ☐
 - identifiant, date et type des commandes passées par le client 'C1', classées par identifiant de commande. oui ☐ non ☐
 - identifiant, date et type des commandes, classées par identifiant de commande. oui ☐ non ☐
- Attendez que votre binôme favori termine la question 1. Une fois que cela est fait, comparez vos réponses avec les siennes. Si vous n'avez pas répondu la même chose à certaines requêtes, essayez de convaincre votre binôme que c'est vous qui avez raison.
- Exécutez sur Cassandra les requêtes de la question 1 afin de voir si vos hypothèses sont exactes. Pour vous connecter à Cassandra, vous pouvez utiliser IntelliJ ou DataGrip.

 - Pour se connecter au serveur Cassandra de l'IUT, indiquez 162.38.222.152 pour le Host et 9034 pour le Port. Indiquez votre "Login" (nom + première lettre du prénom), votre "Password" (code INE ou sinon, date de la rentrée) et le "Keyspace" (qui est le même que votre login).
 - Si cela n'est pas déjà fait, ouvrez une nouvelle console et vérifiez que vous êtes bien sur le keyspace qui a pour nom votre nom de login. Si ce n'est pas le cas, sélectionnez le keyspace qui convient dans le menu déroulant en haute à droite de la console ou tapez la commande « *use votreLogin* » pour vous positionner sur votre keyspace.
 - Copiez et exécutez le code du fichier AB-Orders.cql qui se trouve sur le Moodle. Cela créera les tables avec des données à l'intérieur.
 - Réalisez les requêtes de la question 1 afin de vérifier vos hypothèses.

commandes_par_id_client		
id_client	varchar	K
nom_client	varchar	
prenom_client	varchar	
categorie_client	varchar	
id_commande	varchar	C ↓
date_commande	timestamp	
type_commande	varchar	

Modèle Logique de Données

detail_commandes		
id_commande	varchar	K
id_client	varchar	S
nom_client	varchar	S
prenom_client	varchar	S
categorie_client	varchar	S
ville_client	varchar	S
date_commande	timestamp	S
type_commande	varchar	S
id_produit	varchar	C ↑
nom_produit	varchar	
montant_produit	double	

4. Ecrire une requête qui interroge la table `commandes_par_id_client` afin de trouver l'identifiant des commandes passées par les clients de la catégorie 'Client patibulaire'. Cela ne devrait pas marcher car `categorie_client` n'est pas la clé de partition.
- Réalisez cette requête en créant un index secondaire dans la table `commandes_par_id_client`.
 - Toutefois, les index secondaires sont déconseillés car ils dégradent les performances notamment lorsque les données sont stockées sur plusieurs nœuds (de la même façon qu'il faut éviter d'utiliser l'instruction `ALLOW FILTERING` pour réaliser une requête). Afin d'améliorer les performances, on va donc plutôt faire la requête grâce à une vue matérialisée. Ecrire le code d'une vue matérialisée contenant les données permettant de faire la requête précédente.
 - On rappelle que la vue matérialisée doit avoir dans sa clé (clé de partition ou colonne de clustering) toutes les colonnes qui étaient dans la clé de la table d'origine, même si elles ne sont pas forcément utiles pour la requête qu'on veut réaliser.
 - Faire une requête qui utilise la vue matérialisée (dans son `FROM`) et vérifiez qu'il y a bien deux commandes dans le résultat.
 - Une fois la vue matérialisée créée, vérifiez qu'elle est bien mise à jour lorsqu'on modifie les données de la table `commandes_par_id_client` ; par exemple, il ne doit plus y avoir qu'une seule commande dans la vue matérialisée après avoir lancé la requête suivante :


```
UPDATE commandes_par_id_client
SET categorie_client = 'XXX'
WHERE id_client = 'C7'
AND id_commande = '013';
```