

## Générer les données compréhensibles pour ArangoDB à partir des données fournies :

### **1. Partie JSON :**

#### *a. Collection Order :*

Une unique modification est nécessaire : ajouter un champ key qui correspond au champ OrderId afin de pouvoir accéder à la collection (clé primaire).

Pour ce faire, un script **AddKeyToOrder.java** est fourni. Il suffit de modifier la variable *from\_path* [ligne 27] (contenant le chemin vers le fichier d'origine *Order.json*) et la variable *to\_path* [ligne 27] (contenant le chemin vers le fichier de sortie) et la conversion se fera automatiquement.

#### *b. Collection Product :*

Pour cette collection, la même modification est réalisée ainsi qu'une seconde : on va ajouter le champ brand à la fin de chaque ligne. On ajoute donc le nom de la marque associée à chaque produit qui est initialement fournie dans le fichier *BrandByProduct.csv*. (Il s'agit en résumé d'une opération de merge sur le fichier Product). Le script est **AddBrandToProduct.java**.

Il faudra donc renseigner cette fois ci la variable *from\_path\_product* [ligne 56] avec le chemin du fichier initial *Product.csv* ainsi que la variable *from\_path\_brand* [ligne 57] avec le chemin vers le fichier *BrandByProduct.csv*. Pour finir le chemin vers lequel vous souhaitez écrire le fichier est à renseigner dans la variable *to\_path* [ligne 59].

### **2. Partie Graphes :**

#### *a. CollectionTag :*

Le fichier *tag\_0\_0.csv* n'étant pas fourni il faut le générer. Pour ce faire, le script **generateTag.java** utilise les « edges collection » *person\_hasInterest\_tag* et *post\_hasTag\_tag* afin de récupérer les ids et ainsi créer le fichier *tag\_0\_0.csv*. Comme dans les cas précédents il faut changer les variables *from\_path\_hasInterest* [ligne 45] avec le chemin d'accès du fichier *person\_hasInterest\_tag\_0\_0.csv* et *from\_path\_hasTag* [ligne 46] avec le chemin d'accès du fichier *post\_hasTag\_tag\_0\_0.csv*. Et pour finir la variable *to\_path\_Tag* [ligne 48] spécifie le chemin de création de fichier.

#### *b. Le reste :*

**GraphToJSON.java** est le dernier script java fournit pour cette partie.

Il faut modifier la variable *from\_path* [ligne 64] avec le chemin d'accès vers le fichier à convertir (le nom de ces derniers sont spécifiés dans une variable au-dessus afin de modifier rapidement). La variable *to\_path* [ligne 65] correspond au chemin d'écriture de fichier.

**ATTENTION** : cette fois ci il faut aussi modifier à la ligne 75 le boolean de la méthode writeJSON ! **true** lorsque la collection est une « edge collection », **false** dans le cas contraire.

## Importer les données dans la base de données :

Vous trouverez ci-dessous les lignes de commandes utilisées pour importer les collections générées dans la base de données :

### **1. Partie JSON :**

#### *a. Collection Order :*

```
arangoinport --server.database "multiModelDB" --file "Order.json" --type json --collection "Order" --create-collection true --ignore-missing
```

#### *b. Collection Product :*

```
arangoinport --server.database "multiModelDB" --file "Product.csv" --type csv --collection "Product" --create-collection true --ignore-missing
```

### **2. Partie Graphes :**

#### Vertices

#### *a. Collection Post :*

```
arangoinport --server.database "multiModelDB" --file "post_0_0.json" --type json --collection "Post" --create-collection true --ignore-missing
```

#### *b. Collection Person :*

```
arangoinport --server.database "multiModelDB" --file "person_0_0.json" --type json --collection "Person" --create-collection true --ignore-missing
```

#### *c. Collection Tag :*

```
arangoinport --server.database "multiModelDB" --file "tag_0_0.json" --type json --collection "Tag" --create-collection true --ignore-missing
```

#### Edges

#### *a. Collection hasTag:*

```
arangoinport --server.database "multiModelDB" --file "post_hasTag_tag_0_0.json" --type json --collection "post_hasTag_tag" --create-collection true --create-collection-type edge --ignore-missing
```

#### *b. Collection hasInterest :*

```
arangoinport --server.database "multiModelDB" --file "person_hasInterest_tag_0_0.json" --type json --collection "person_hasInterest_tag" --create-collection true --create-collection-type edge --ignore-missing
```

c. *Collection hasCreator :*

```
arangoimport --server.database "multiModelDB" --file  
"post_hasCreator_person_0_0.json" --type json --collection "post_hasCreator_person"  
--create-collection true --create-collection-type edge --ignore-missing
```

d. *Collection knows:*

```
arangoimport --server.database "multiModelDB" --file  
"person_knows_person_0_0.json" --type json --collection "person_knows_person" --  
create-collection true --create-collection-type edge --ignore-missing
```

### **IMPORTANT :**

Pour que les lignes de commandes s'exécutent bien faites attention à l'option `--file` !

Il faut mettre les fichiers au même niveau que l'exécutable `arangoimport.exe` (c'est le cas du paramètre actuel) sinon n'oubliez pas d'ajouter le nom de dossier contenant le fichier.