

- Création des nœuds Users :

```
LOAD CSV with headers FROM "file:/utilisateurs.csv" as row fieldterminator
';' CREATE (u:Users {userId: toInteger(row.Id), userName: row.DisplayName})
return u
```

- Création des indexes en Users :

```
CREATE INDEX ON :Users (userId)
```

- Création d'un Utilisateur anonyme (pour traiter les cas anonymes)

```
CREATE (u:Users {userId:0, userName: "anonymous"}) return u
```

- Création de la relation Comment entre les utilisateurs:

```
:auto using Periodic Commit LOAD CSV with headers FROM "file:/comment.csv"
AS row FIELDTERMINATOR ';' match (n:Users {userId:
coalesce(toInteger(row.OwnerUserId),0)}), (b:Users {userId:
coalesce(toInteger(row.UserId),0)}) CREATE (b) -[:Comment {NbrOfComments:
toInteger(row.NbrOfComments)}]- > (n)
```

- Création de la relation Answer entre les utilisateurs:

```
:auto using Periodic Commit LOAD CSV with headers FROM
"file:/nouveauRanswer.csv" AS row FIELDTERMINATOR ';' match (n:Users
{userId: coalesce(toInteger(row.OwnerUserId),0)}), (b:Users {userId:
coalesce(toInteger(row.UserId),0)}) CREATE (b) -[:Answer {NbrOfAnswers:
toInteger(row.NbrOfAnswers)}]- > (n)
```

- Création de la relation UpVote entre les utilisateurs:

```
:auto using Periodic Commit LOAD CSV with headers FROM "file:/upvote.csv"
AS row FIELDTERMINATOR ';' match (n:Users {userId:
coalesce(toInteger(row.OwnerUserId),0)}), (b:Users {userId:
coalesce(toInteger(row.UserId),0)}) CREATE (b) -[:Upvote {NbrOfUpvotes:
toInteger(row.NbrOfUpvotes)}]- > (n)
```

- Création de la relation DownVote entre les utilisateurs:

```
: auto using Periodic Commit LOAD CSV with headers FROM
"file:/downvote.csv" AS row FIELDTERMINATOR ';' match (n:Users {userId:
coalesce(toInteger(row.OwnerUserId),0)}), (b:Users {userId:
coalesce(toInteger(row.UserId),0)}) CREATE (b) -[:Downvote {NbrOfDownvotes:
toInteger(row.NbrOfDownvotes)}]- > (n)
```

- Création de la relation FavoriteVote entre les utilisateurs:

```
:auto using Periodic Commit LOAD CSV with headers FROM
"file:/favoritevote.csv" AS row FIELDTERMINATOR ';' match (n:Users {userId:
coalesce(toInteger(row.OwnerUserId),0)}), (b:Users {userId:
coalesce(toInteger(row.UserId),0)}) CREATE (b) -[:Favoritevote
{NbrOfFavoritevotes: toInteger(row.NbrOfFavoritevotes)}}- > (n)
```

- Création de la relation Relatedlink entre les utilisateurs:

```
:auto using Periodic Commit LOAD CSV with headers FROM
"file:/linksReladedTo.csv" AS row FIELDTERMINATOR ';' match (n:Users
{userId: toInteger(row.OwnerUserId)}), (b:Users {userId:
toInteger(row.UserId)}) MERGE (b)-[:Relatedlink {NbrOfLinks:
toInteger(row.NbrOfLinks)}}]- (n)
```

- Création des nœuds Tags :

```
LOAD CSV with headers FROM "file:/tags.csv" as row fieldterminator ';'
create (u:Tags {tagId: toInteger(row.Id), tagName: row.TagName}) return u
```

- Création des indexs en Tags :

```
CREATE INDEX ON :Tag(idtag)
```

- Création de la relation LieeA entre les Tags:

```
:auto using Periodic COMMIT LOAD CSV with headers FROM "file:/newtag.csv"
AS row FIELDTERMINATOR ';' match (n: Tags {tagId: toInteger(row.Tag)}), (b:
Tags {tagId: toInteger(row.TagId)}) MERGE (b)-[:LieeeA {NbrOfOccurence:
toInteger(row.NbrOfTags)}}]- (n)
```