TP n°4 – MapReduce avec CouchDB

1. Présentation de CouchDB

1.1 Définition de CouchDB

CouchDB est un système de gestion de bases de données NoSQL reposant sur un modèle orienté documents. Les données y sont enregistrées sous forme de fichiers JSON, ce qui facilite leur manipulation et leur échange. Pour l'analyse et le traitement des données, CouchDB s'appuie sur des vues MapReduce, permettant d'exécuter des calculs sur de grands ensembles d'informations.

CouchDB se distingue notamment par les points suivants :

- Une mise en place rapide et une prise en main intuitive.
- Un projet open source sous licence Apache.
- Une interaction basée sur une interface REST, exploitant les méthodes HTTP suivantes :
    - GET pour consulter des données,
    - PUT pour créer de nouveaux éléments,
    - POST pour envoyer ou mettre à jour des informations,
    - DELETE pour retirer des données.

1.2 Mise en place de CouchDB

Installation à l'aide de Docker

L'exécution de CouchDB peut se faire facilement grâce à Docker, sans nécessiter d'installation manuelle complexe. La commande ci-dessous permet de lancer un conteneur CouchDB en arrière-plan, tout en définissant les informations d'authentification et le port d'accès :

```
C:\Users\Kaoutar>docker run -d --name couchdbdemo -e COUCHDB_USER=youssef -e COUCHDB_PASSW
ORD=samir -p 5984:5984 couchdb
Unable to find image 'couchdb:latest' locally
latest: Pulling from library/couchdb
87d5a41c1790: Pull complete
f6c9c88e767d: Pull complete
4b62383183a1: Pull complete
ae4ce04d0e1c: Pull complete
51b27282aaec: Pull complete
2897b671832f: Pull complete
cd726490d7d0: Pull complete
716c80c273f8: Pull complete
8856ade7c27a: Pull complete
609e5bdfa7b1: Pull complete
7f82cbc9af1f: Pull complete
Digest: sha256:a2c8839c86304962ae60df41c9aa51907925b7ca022b69acfcd45663a89daa77
Status: Downloaded newer image for couchdb:latest
ad5cff46df736d8fdca68a11d89cc30d930b9056840c90553ce9582fddf975cf

C:\Users\Kaoutar>
```

Une fois le service lancé, l'interface d'administration graphique (Fauxton) est accessible depuis un navigateur à l'adresse :

`http://localhost:5984/_utils`

1.3 Vérification du bon fonctionnement de CouchDB

Pour confirmer que CouchDB est bien démarré et joignable, on peut effectuer un test simple en envoyant une requête HTTP depuis le terminal. La commande ci-dessous permet de vérifier l'accessibilité du service et d'obtenir une réponse au format JSON :

```
C:\Users\Kaoutar>curl -X GET http://youssef:samir@localhost:5984/
{"couchdb":"Welcome","version":"3.5.1","git_sha":"44f6a43d8","uuid":"8f7635d527696b1e91963
20e55b5fc66","features":["access-ready","partitioned","pluggable-storage-engines","reshard
","scheduler"],"vendor":{"name":"The Apache Software Foundation"}}

C:\Users\Kaoutar>
```

2. Gestion des bases de données

2.1 Création d'une base

Pour créer une base de données intitulée films, on envoie une requête HTTP utilisant la méthode PUT vers la ressource correspondante :

```
C:\Users\Kaoutar>curl -X PUT http://youssef:samir@localhost:5984/films
{"ok":true}
```

2.2 Ajout de documents

L'insertion d'un document peut se faire de manière unitaire en indiquant directement son identifiant dans l'URL, puis en transmettant son contenu (au format JSON) dans la requête:

```
C:\Users\Kaoutar>curl -X PUT "http://youssef:samir@localhost:5984/films/doc" -H "Content-Type: appli
cation/json" -d "{\"cle\":\"valeur\"}"
{"error":"conflict","reason":"Document update conflict."}

C:\Users\Kaoutar>curl -X PUT "http://youssef:samir@localhost:5984/films/doc1" -H "Content-Type: appl
ication/json" -d "{\"nom\":\"youssef\"}"
{"ok":true,"id":"doc1","rev":"1-beed6da013fd2a82e42f0c9e07a79a74"}
```

Il est aussi possible d'ajouter plusieurs documents en une seule requête en utilisant le mécanisme d'insertion en masse (bulk insert), ce qui permet de charger un ensemble de données plus rapidement en une seule opération :

```
C:\Users\Kaoutar>curl -X POST "http://youssef:samir@localhost:5984/films/_bulk_docs" -H "Content-Type: application/json" --data-binary "@C:\Users\Kaoutar\Downloads\films_couchdb.json"
[{"ok":true,"id":"movie:11","rev":"1-c404285f85cc593f351855821ebe5fc7"},{"ok":true,"id":"movie:24","rev":"1-7f851a642ab7108adad8354952d4c560"},{"ok":true,"id":"movie:28","rev":"1-5ef74f3007d597da5c1a41d73e00f308"},{"ok":true,"id":"movie:33","rev":"1-210992fbbd105dd91ceb02a1f6b1811d"},{"ok":true,"id":"movie:38","rev":"1-902184f7cc63bc4f802f3b33ffd2eb27"},{"ok":true,"id":"movie:59","rev":"1-2ca4990b59fbaee2006e0b0ef74481d0"},{"ok":true,"id":"movie:62","rev":"1-89d7541cf67625fbeda283dadf7294bb"},{"ok":true,"id":"movie:74","rev":"1-ea1b40608799bd603ebc7f82cf4511ac"},{"ok":true,"id":"movie:75","rev":"1-522355c47de05179064d6218542b6ca7"},{"ok":true,"id":"movie:77","rev":"1-85291b834cfda40e18739d1b37d6deef"},{"ok":true,"id":"movie:78","rev":"1-c2b126bd26e20d4256ea573e5bc5a11a"},{"ok":true,"id":"movie:85","rev":"1-ae348bef3600f3a445ed329201ccd191"},{"ok":true,"id":"movie:87","rev":"1-8981cf7f2d13f2d0004ae8c4a8440868"},{"ok":true,"id":"movie:89","rev":"1-373a9c699285665f1e822c33f2bf9768"},{"ok":true,"id":"movie:98","rev":"1-8c8ccdbdcb518ac13921b4f0492a3c10"},{"ok":true,"id":"movie:103","rev":
```

2.3 Lecture d'un document

Pour consulter un document spécifique à partir de son identifiant, on envoie une requête HTTP utilisant la méthode GET. Par exemple, afin de récupérer le document dont l'ID est movie:490132 dans la base films, la requête s'écrit comme suit :

```
C:\Users\Kaoutar>curl -X GET http://youssef:samir@localhost:5984/films/movie:490132
{"_id":"movie:490132","_rev":"1-d99c07bfb5be11f8e296fd86b28994f0","title":"Green Book : Sur les routes du sud","year":2018,"genre":"Drame","summary":"En 1962, alors que règne la ségrégation, Tony Lip (Viggo Mortensen), un videur italo-américain du Bronx, est engagé pour conduire et protéger le Dr Don Shirley (Mahershala Ali), un pianiste noir de renommée mondiale, lors d'une tournée de concerts. Durant leur périple de Manhattan jusqu'au Sud profond, ils s'appuient sur le Green Book pour dénicher les établissements accueillant les personnes de couleur, où l'on ne refusera pas de servir Shirley et où il ne sera ni humilié ni maltraité.  Dans un pays où le mouvement des droits civiques commence à se faire entendre, les deux hommes vont être confrontés au pire de l'âme humaine, dont ils se guérissent grâce à leur générosité et leur humour. Ensemble, ils vont devoir dépasser leurs préjugés, oublier ce qu'ils considéraient comme des différences insurmontables, pour découvrir leur humanité commune.","country":"US","director":{"_id":"artist:7396","last_name":"Farrelly","first_name":"Peter","birth_date":1956},"actors":[{"last_name":"Mortensen","first_name":"Viggo","birth_date":1958},{"last_name":"Cardellini","first_name":"Linda","birth_date":1975},{"last_name":"Ali","first_name":"Mahershala","birth_date":1974}]}
```

3. MapReduce avec CouchDB

3.1 Définition

MapReduce est un modèle de traitement qui permet d'exécuter des calculs sur de grandes quantités de données, en s'appuyant sur deux étapes principales :

- Map : traite les documents un par un et émet des couples *(clé, valeur)* via `emit(key, value)`.

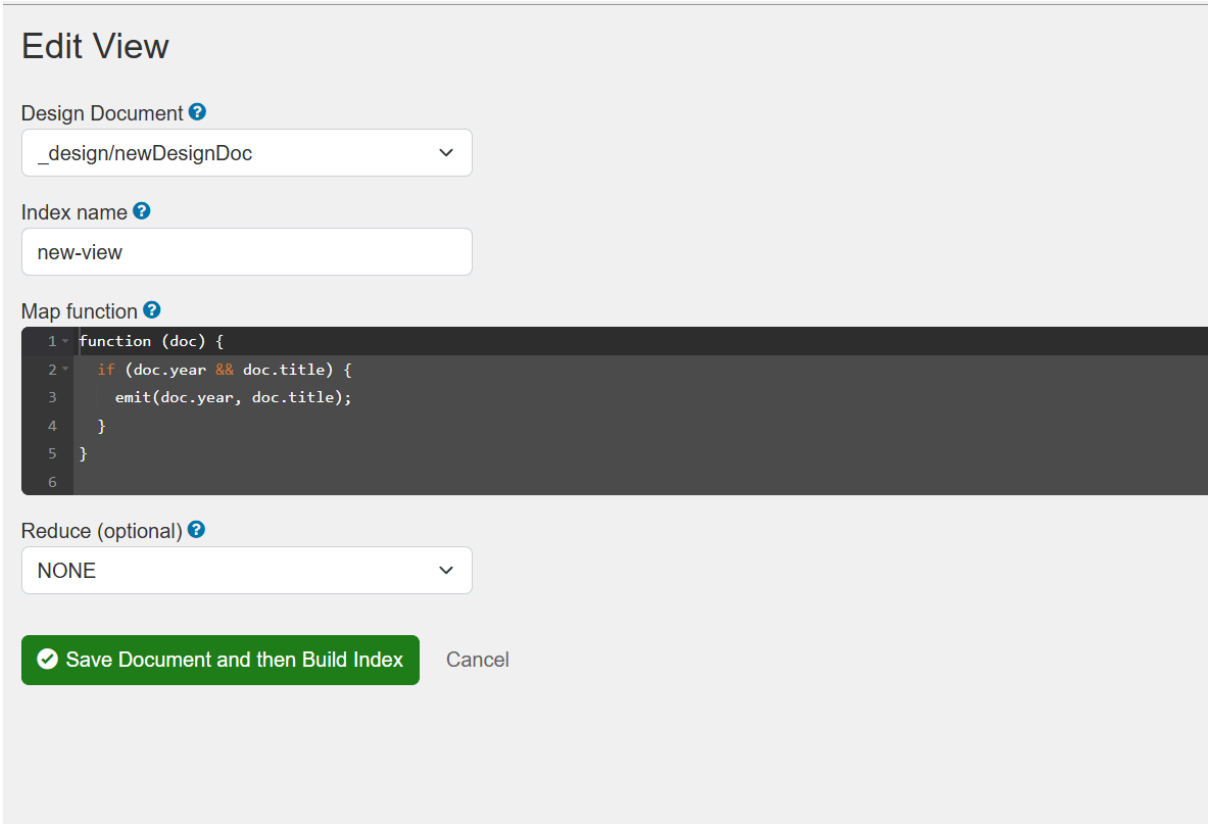
- Reduce : regroupe toutes les valeurs ayant la même clé et applique une fonction d'agrégation (ex. somme, comptage).

Dans CouchDB, le résultat de la phase Map peut être consulté directement ; la phase Reduce est optionnelle et doit être activée dans les options de la vue.

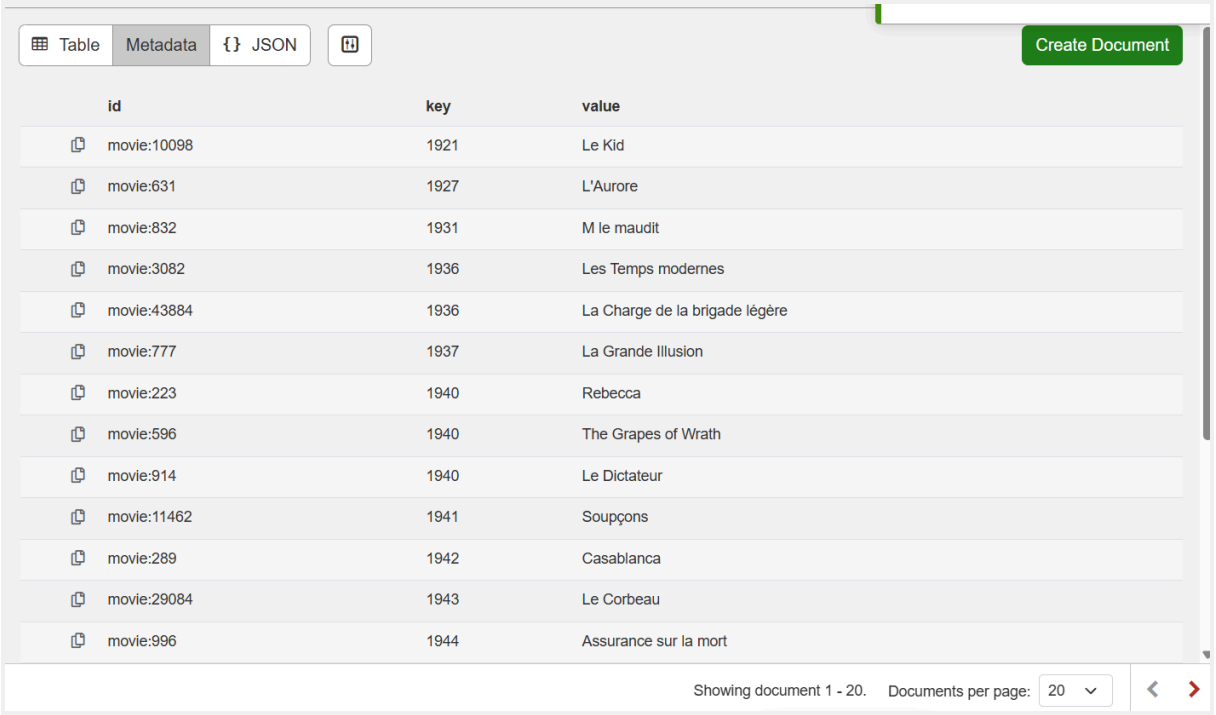
3.2 Exemple : nombre de films par année

Fonction Map :

```
function (doc) {
  if (doc.year && doc.title) {
    emit(doc.year, doc.title);
  }
}
```



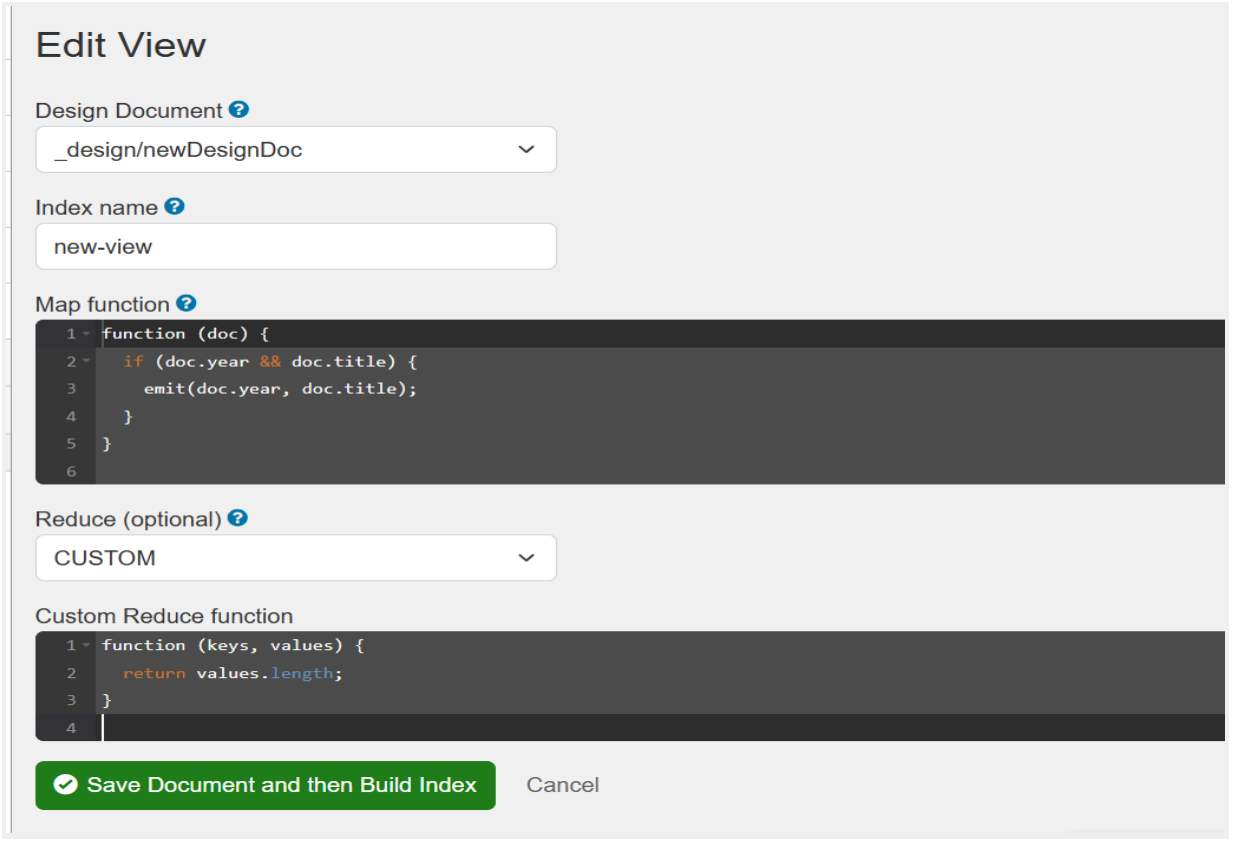
Résultat intermédiaire (Map) :



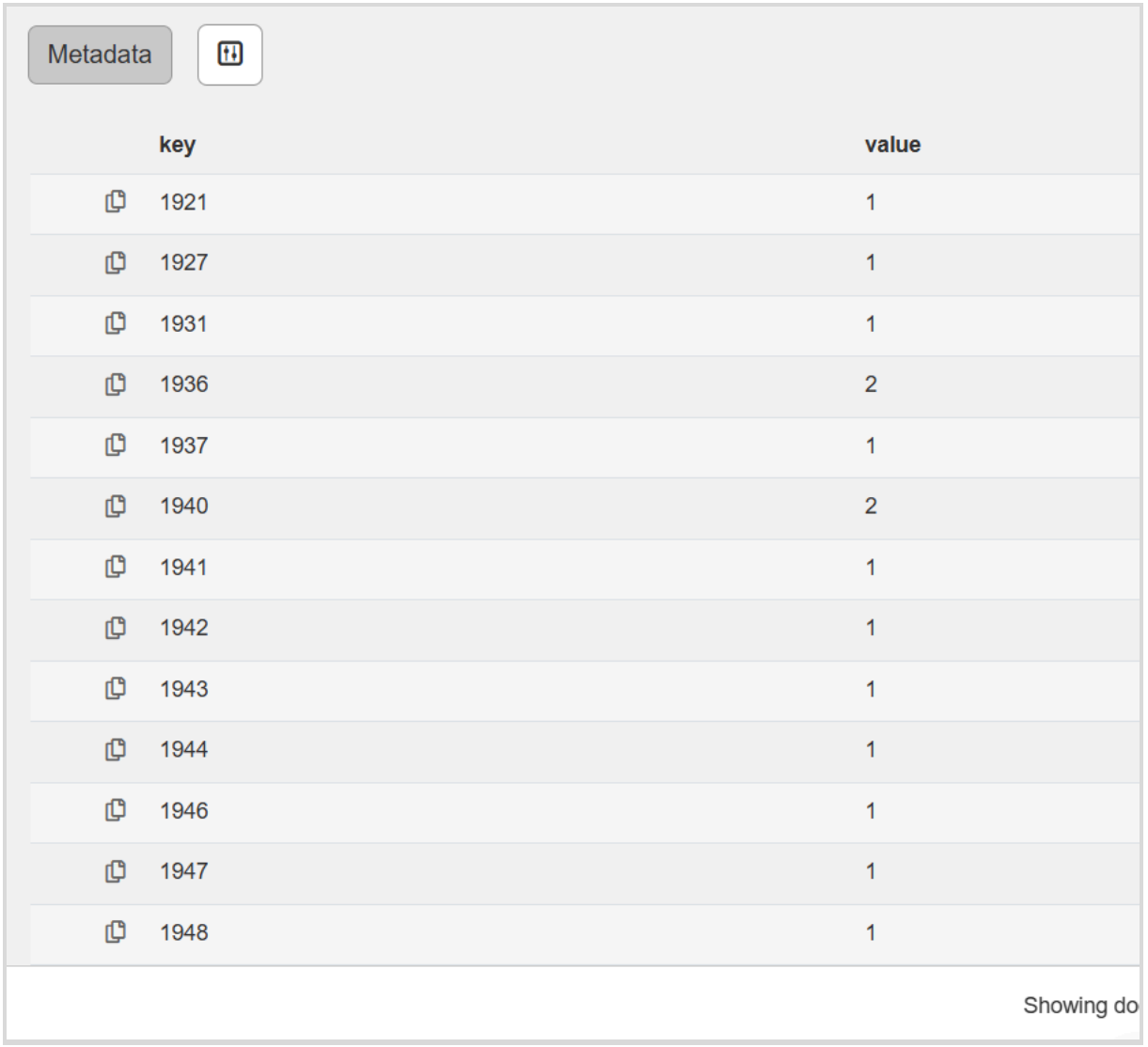
Fonction Reduce :

```
function(keys, values) {

  return values.length;
}
```



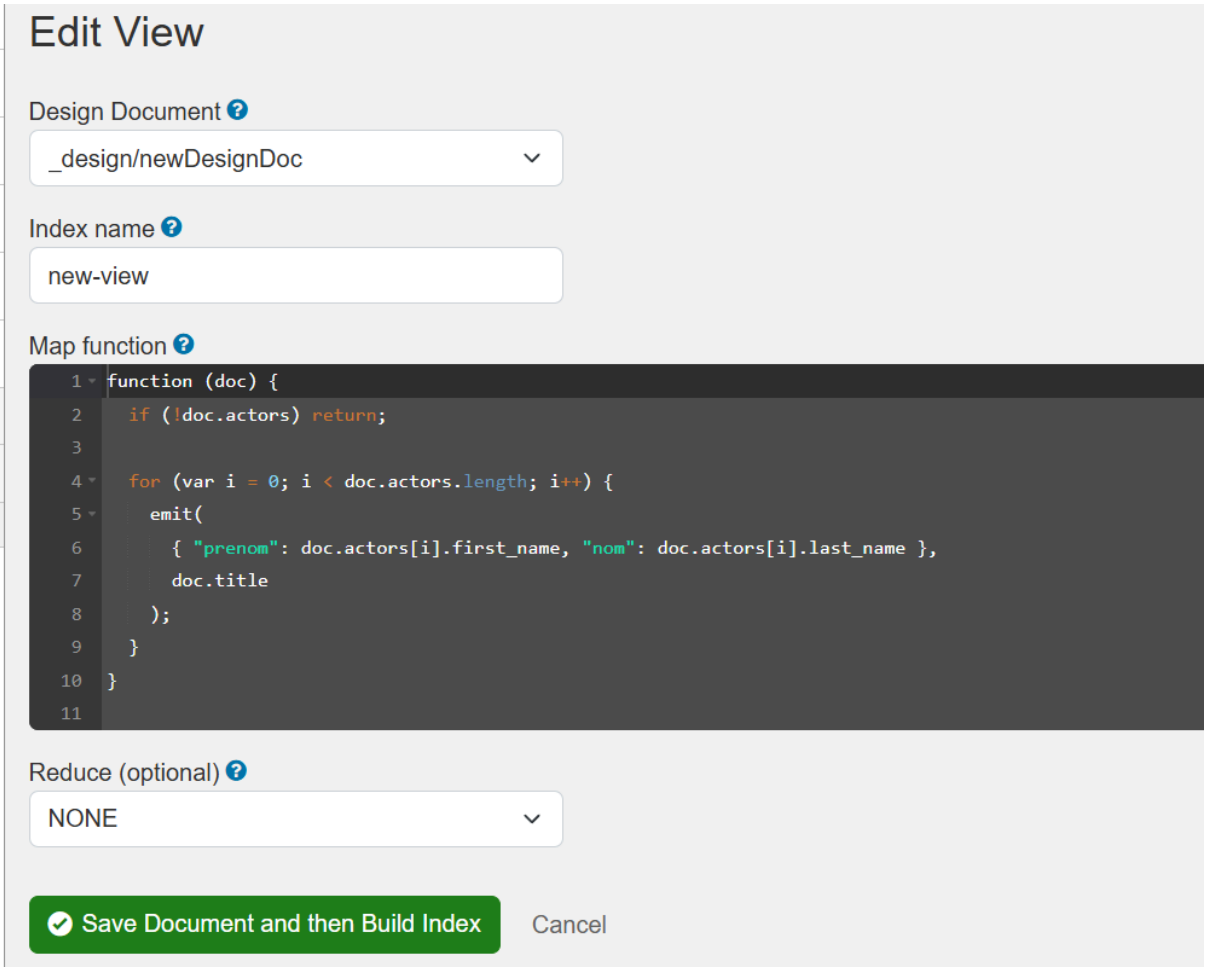
Résultat final (Reduce + group) :



3.3 Exemple : Nombre de films par acteur

Fonction Map

Pour déterminer combien de films sont associés à chaque acteur, on définit une fonction *Map* qui parcourt la liste des acteurs d'un film et émet une paire *(acteur, 1)* pour chacun d'eux :

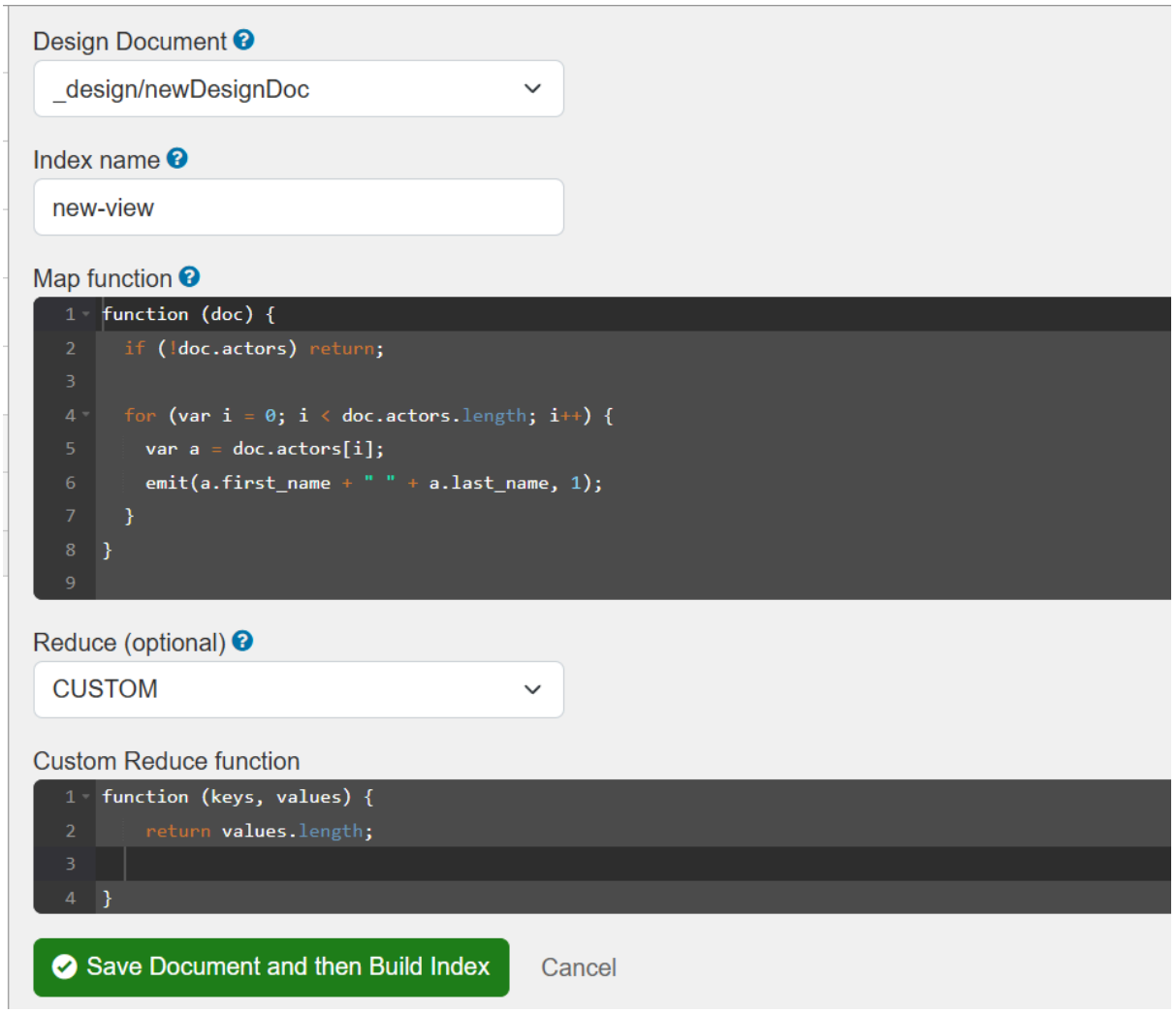


Le résultat:

	id	key	value
	movie:1443	{ "prenom": "A.J.", "nom": "Cook" }	Virgin suicides
	movie:155	{ "prenom": "Aaron", "nom": "Eckhart" }	The Dark Knight : Le Chevalier noir
	movie:46738	{ "prenom": "Abdelghafour", "nom": "Elaaziz" }	Incendies
	movie:773	{ "prenom": "Abigail", "nom": "Breslin" }	Little Miss Sunshine
	movie:140607	{ "prenom": "Adam", "nom": "Driver" }	Star Wars : Le Réveil de la Force
	movie:181808	{ "prenom": "Adam", "nom": "Driver" }	Star Wars : Les Derniers Jedi
	movie:181812	{ "prenom": "Adam", "nom": "Driver" }	Star Wars, épisode IX
	movie:245703	{ "prenom": "Adam", "nom": "Driver" }	Midnight Special
	movie:857	{ "prenom": "Adam", "nom": "Goldberg" }	Il faut sauver le soldat Ryan
	movie:21575	{ "prenom": "Adel", "nom": "Bencherif" }	Un prophète
	movie:531428	{ "prenom": "Adèle", "nom": "Haenel" }	Portrait de la jeune fille en feu
	movie:4034	{ "prenom": "Adolfo", "nom": "Celi" }	L'Homme de Rio
	movie:975	{ "prenom": "Adolphe", "nom": "Menjou" }	Les sentiers de la gloire

Fonction Reduce:

Pour compter le nombre de films par acteur :



Le résultat:

Metadata

Crea

key	value
A.J. Cook	1
Aaron Eckhart	1
Abdelghafour Elaaziz	1
Abigail Breslin	1
Adam Driver	2
Adam Goldberg	1
Adel Bencherif	1
Adèle Haenel	1
Adolfo Celi	1
Adolphe Menjou	1
Adrian Rawlins	1
Adrien Brody	1
Agnès Jaoui	1