

# TP2 CouchDB

## 1. Création et alimentation d'une base de données vaccination

---

### 1.1 Sans schéma et structure JSON

L'objectif du premier exercice est de réfléchir en amont à la structuration des documents JSON. Cette structuration est essentielle puisque une base de données CouchDB est dite sans schéma. Il est donc impératif de veiller en amont à disposer d'une auto-description des documents qui puisse faciliter ensuite l'organisation des documents et la consultation des données.

- Vous créez une nouvelle base de données nommée **vaccination** en la préfixant avec votre nom de famille (tout en minuscules). Réalisez cette opération en ligne de commandes avec curl, et en modifiant le nombre de partitions pour la base ;

```
curl -X PUT $COUCH3/zzz_vaccination?q=8
```

Listing 1 – Exemple changement nombre de partitions (ici 8)

- Un diagramme de classes est donné, qui reprend la structure qui sera à retrouver. Pour des raisons d'anonymat des patients, les doses (dose1 pour la première injection, dose2 pour la seconde injection, dose3 pour la troisième injection) pour chaque vaccin ne sont comptabilisées qu'au niveau du département, et non de la commune, à une date donnée (jour). Les vaccins sont à prendre parmi Pfizer, Moderna, AstraZeneca, Janssen et TousVaccins (la somme des 4). Vous noterez le champ **type** qui est un début de définition d'un schéma, et ce, quelle que soit la structuration (fichier) choisie.

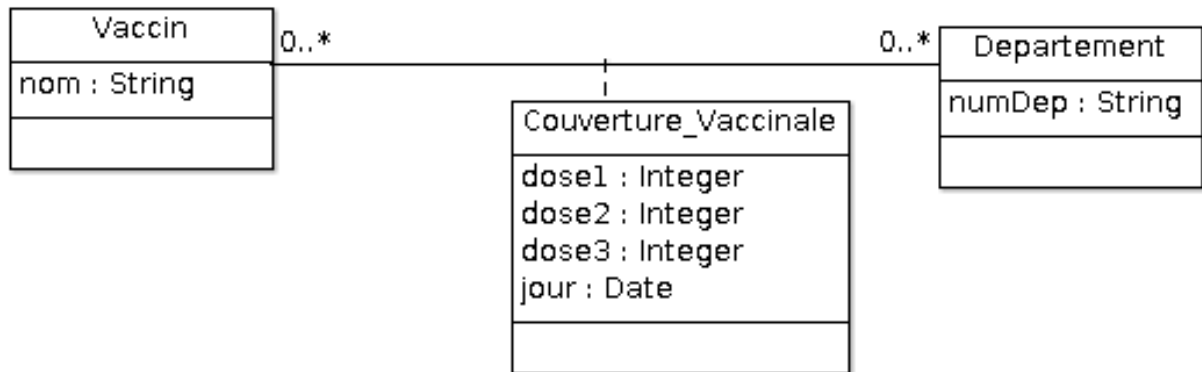


FIGURE 1 – Diagramme de classes pour l'application

Deux exemples distincts de documents sont fournis, qui illustrent les deux structures de document définies,

```

{
  "_id": "12_27-DEC-20_AstraZeneka",
  "dep": "12",
  "jour": "27-DEC-20" ,
  "type": "couverture_vaccinale" ,
  "doses": [ {"vaccin": "AstraZeneka" } , {"dose1": 0 } , { "dose2": 0 } , {
"dose3": 0} ]
}
  
```

Listing 2 – Exemple un (vaccin1.json)

```

{
  "jour": "04-JAN-21",
  "dep": "12",
  "type": "couverture_vaccinale",
  "vaccinations": [
    {"vaccin": "AstraZeneka", "doses": [ {"dose1": 0 }, { "dose2": 0 }, {
      "dose3":
0} ]} , {"vaccin": "Janssen", "doses": [ {"dose1": 0 }, { "dose2": 0 }, {
"dose3": 0} ]} , {"vaccin": "Moderna", "doses": [ {"dose1": 0 }, { "dose2":
0 },
{ "dose3": 0} ]} , {"vaccin": "Pfizer", "doses": [ {"dose1": 1 }, {
"dose2": 0 },
{ "dose3": 0} ]} , {"vaccin": "TousVaccins", "doses": [ {"dose1": 1 }, {
"dose2":
0 }, { "dose3": 0} ]} ]
}
  
```

```
}

```

Listing 3 – Exemple deux (vaccin2.json)

- Vous ferez un choix entre les deux structures proposées. Vous expliquerez les raisons de ce choix (en accord avec les transparents du cours sur les bonnes pratiques). Les deux structures proposées sont discutables et pourraient être améliorées. Vous ferez des propositions d'amélioration.
- Vous exploiterez le mode "ajout par lots" via curl pour insérer les documents du fichier vaccin1.json ou vaccin2.json en fonction de votre choix concernant la structuration.

## 2. Appropriation de la base de données

---

### 2.1 Ajout de documents de type "departement"

Vous ajouterez également les quatres documents du fichier `departements.json`, et vous construirez la fonction `map/reduce` suivante :

```
function (doc) {
  if (doc.type=='departement')
    emit(doc.type,doc.population);
}

function (keys, values, rereduce) {
  return Math.min.apply({}, values);
}
```

Listing 4 – Exemple map/reduce

- que renvoie cette requête (au niveau `map`, puis au niveau `reduce` et enfin au niveau `rereduce`) ?

### 2.2 Jeu de questions à définir en fonction du modèle choisi

Vous répondrez aux questions suivantes en vous aidant de requêtes CURL :

1. lister les informations générales concernant la base vaccination, à l'aide du mécanisme GET. Pouvez vous connaître le nombre de documents contenus dans la base ?
2. lister tous les documents de la BD
3. faire afficher le contenu d'un document.

## 3. Définition de vues

---

Vous définirez des vues (en javascript) dans votre base pour consulter les documents décrivant les couvertures vaccinales. Ce travail peut se faire soit en ligne de commande avec curl, soit avec l'interface Web Fauxton (depuis un navigateur à l'adresse `prodpeda-couchdb3-2.infra.umontpellier.fr:5984/_utils/`).

### 3.1 MAP et MAP/REDUCE

1. renvoyez pour tous les documents de type couverture\_vaccinale du département de l'Hérault (34), l'identifiant du document (clé doc.id) et le jour de vaccination (valeur doc.jour)
2. donnez le nombre de documents de type couverture\_vaccinale du département de l'Hérault (34) (clé doc.dep, valeur 1)
3. donnez le nombre de documents de type couverture\_vaccinale du département de l'Hérault (34) pour chaque année écoulée. Un exemple de manipulation de date est fourni.

```
var d = new Date(doc.jour) ;  
-- fonctions pour retourner annee ou mois  
-- d.getFullYear() ou d.getMonth()
```

Listing 5 – Manipulation date

4. donnez le nombre de documents de type couverture\_vaccinale par département, par année et par mois écoulés.
5. donnez pour les documents de type couverture\_vaccinale pour le vaccin Pfizer, l'identifiant du document (clé doc.id) et la date et le département de vaccination (valeur : doc.jour et doc.dep)
6. donnez le nombre de documents de type couverture\_vaccinale pour le vaccin Pfizer par département
7. donnez le nombre de documents de type couverture\_vaccinale pour le vaccin Pfizer par département, par an et par mois
8. donnez la somme de dose1 (et autres statistiques) pour le vaccin Pfizer, par département, par an et par mois

## 4. Distribution de la base de données

---

Vous répondrez aux questions suivantes en vous aidant de requêtes CURL :

1. Combien de partitions sont définies (avant recopie) ? Quel est le nombre de copies ? Combien de partitions répliquées sont définies ?
2. Comment savoir dans quelle(s) partition(s) se trouve un des documents de la base ?
3. Est ce que des copies de toutes les partitions sont présentes sur tous les nœuds ?