

NoSQL avec MongoDB

Cours 4 - Insertion, Modification, et Supression

Steve Lévesque, Tous droits réservés © où applicables

Table des matières

1 Principe des Actions Destructrices

2 Insertion

- Singulière
- Multiple

3 Modification

- Singulière
- Multiple

4 Suppression

- Singulière
- Multiple

Principe des Actions Destructrices

Une action est considérée **destructrice** lorsqu'elle menace l'intégrité de la base de données.

Dans cette définition large, l'insertion est considérée comme une menace à l'intégrité de la base de données, et donc une action destructrice sur son intégrité.

L'interrogation serait la seule manipulation préservant l'état d'une base de données.

Insertion

L'insertion permet d'ajouter de nouveaux documents (entrées) dans la base de données et d'augmenter le corpus.

En général, plus nous avons de données, plus les entraînements de modèles d'Intelligence Artificielle sont efficaces.

Insertion - Singulière

Une insertion singulière permet de sauvegarder un document (au respect des contraintes s'il y a un schéma JSON de validation) dans la base de données.

La commande est :
collection.insert_one(document),
où le document est un seul objet.



Figure: **collection.insert_one(document)**

Insertion - Singulière

Listing: <https://pymongo.readthedocs.io/en/stable/index.html>

```
1  # Steve Levesque, All rights reserved where applicable
2  from pymongo import MongoClient
3
4  client = MongoClient()
5  db = client.admin
6  collection = db.Vegetables
7
8  # Insert 1 Vegetable.
9  collection.insert_one({
10     "name": "New Vegetable 1",
11     "protein (g)": 1.1
12 })
```

Insertion - Multiple

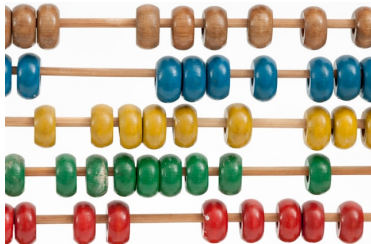


Figure: `collection.insert_many([document])`

Une insertion multiple permet de sauvegarder plusieurs documents d'un seul coup (au respect des contraintes s'il y a un schéma JSON de validation) dans la base de données.

La commande est : **`collection.insert_many([document])`**, où le tableau peut contenir un nombre arbitraire de documents.

Insertion - Multiple

Listing: <https://pymongo.readthedocs.io/en/stable/index.html>

```
1  # Steve Levesque, All rights reserved where applicable
2  from pymongo import MongoClient
3
4  client = MongoClient()
5  db = client.admin
6  collection = db.Vegetables
7
8  # Insert multiple Vegetables (2 in total).
9  collection.insert_many([
10     {
11         "name": "New Vegetable 2",
12         "protein (g)": 1.2
13     },
14     {
15         "name": "New Vegetable 3",
16         "protein (g)": 1.3
17     }
18 ])
```


Modification

La modification permet de mettre à jour des documents (entrées) qui peuvent être d'une représentation fautive ou non optimale.

Par exemple, il est important en Intelligence Artificielle que les données soient **propres, normalisées (entre 0 et 1) et absentes de valeurs considérées comme des vides.**

Modification - Singulière

Une modification singulière permet de changer une ou plusieurs valeurs d'un objet spécifique dans la base de données.

La commande est : **collection.update_one({"attribute": "value"}, {"\$a_rule", {"attribute", "value"}})**, où le document est un seul objet.



Figure: `collection.update_one(document)`

Modification - Singulière

Listing: <https://pymongo.readthedocs.io/en/stable/index.html>

```
1  # Steve Levesque, All rights reserved where applicable
2  from pymongo import MongoClient
3
4  client = MongoClient()
5  db = client.admin
6  collection = db.Vegetables
7
8  # Find the Vegetable named "Beet greens, raw" and put its protein value at 42.
9  name = "Beet greens, raw"
10 column = "protein (g)"
11 collection.update_one({"name": name}, {"$set": {column: 42}})
```

Modification - Multiple



Figure: `collection.update_many({"attribute": {"$a_regex": "attribute"}}, {"$a_rule": {"attribute", "value"}})`

Une modification multiple permet de changer des valeurs de plusieurs documents respectant les contraintes de recherche, le tout d'un seul coup dans la base de données.

La commande est : **`collection.update_many({"attribute": {"$a_regex": "attribute"}}, {"$a_rule": {"attribute", "value"}})`**, où le tableau peut contenir un nombre arbitraire de documents.

Modification - Multiple

Listing: <https://pymongo.readthedocs.io/en/stable/index.html>

```
1  # Steve Levesque, All rights reserved where applicable
2  from pymongo import MongoClient
3
4  client = MongoClient()
5  db = client.admin
6  collection = db.Vegetables
7
8  # Finds all Vegetables with "et gree" in its name, and increment protein by 1.
9  name = "et gree"
10 column = "protein (g)"
11 collection.update_many({"name": {"$regex": name}}, {"$inc": {column: 1}})
```

Suppression

La suppression permet d'enlever des documents incohérents, inutiles et nuisant au corpus et à l'intégrité des données.

Par exemple, si une classe dans un problème de classification en Intelligence Artificielle n'est plus utile pour le domaine d'affaires et que celle-ci fausse les résultats, il serait pertinent de la supprimer de la base de données.

Suppression - Singulière

Une suppression singulière permet d'enlever un document spécifique dans la base de données.

La commande est : **collection.delete_one({"attribute": "value"}).**

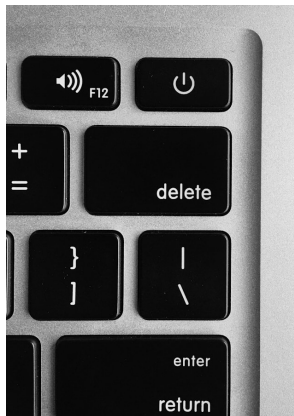


Figure: `collection.delete_one({"attribute": "value"})`

Suppression - Singulière

Listing: <https://pymongo.readthedocs.io/en/stable/index.html>

```
1  # Steve Levesque, All rights reserved where applicable
2  from pymongo import MongoClient
3
4  client = MongoClient()
5  db = client.admin
6  collection = db.Vegetables
7
8  # Delete one Vegetable with number of protein is 43.
9  # If multiple documents with the same number,
10 # only removes the first in order.
11 collection.delete_one({"protein (g)": 43})
```


Suppression - Multiple



Une suppression multiple permet d'enlever plusieurs documents respectant les contraintes de recherche, le tout d'un seul coup dans la base de données.

La commande est : **collection.update_many({"attribute": {"\$a_regex": "attribute"}}, {"\$a_rule": "attribute"}).**

Figure: `collection.update_many({"attribute": {"$a_regex": "attribute"}}, {"$a_rule": {"attribute", "value"}})`

Suppression - Multiple

Listing: <https://pymongo.readthedocs.io/en/stable/index.html>

```
1  # Steve Levesque, All rights reserved where applicable
2  from pymongo import MongoClient
3
4  client = MongoClient()
5  db = client.admin
6  collection = db.Vegetables
7
8  # Removes all Vegetables with a specific query.
9  collection.delete_many({"protein (g)": {"$lte": 1}})
```

Bibliographie

- `https://pymongo.readthedocs.io/en/stable/index.html`