Installer the MongoDB database.

Installer l’outil [Mongo Database Tool](https://docs.mongodb.com/database-tools/installation/installation-windows/), en plus du mongo compass.

En utilisant mongo Shell :

* Créer une base de données nommé : « starter ».
* Créer une collections products.

Importer le fichier ci-joint dans votre base de données mongo, le fichier csv ci-joint. Pour ce faire :

Ajouter son chemin d’installation dans la variable Path.

Depuis votre console, exécuter la requête mongoimport pour importer votre fichier csv dans la base de données « starter » et dans la collection « products » :

**C:\Program Files\MongoDB\Server\4.4\bin>mongoimport -d starter -c products --type csv --file C:\Users\Youcode\Desktop\products.csv --headerline**

Faire les requêtes suivantes :

Afficher les données de la collection en utilisant db.products.find() ;

**db.products.find() ;**

Combien de documents sont inclus dans la collection « products ».

**db.products.find().count()**

**77**

Modifier la taille du batch de MongoDB pour afficher juste 10 éléments.

**db.products.find().batchSize(10).pretty()**

Exécuter les deux commandes suivantes l’une après l’autre

db.products.find({supplierID:{$eq:19}}).pretty();

db.products.find({SupplierID:{$eq:19}}).pretty();

Que remarquez-vous ?

* **La première commande ne fonctionne pas car supplierID en ‘S’ minuscule par contre la deuxième commande qui contient SupplierID en ‘S’ en majuscule**
* **La deuxième commande cherche Les documents qui on la supplierID = 19**

Sélectionner les documents qui ont un SupplierID égale à 19 et UnitPrice moins que 10.

**db.products.find({SupplierID:{$eq:19},UnitPrice:{$lt:10}}).pretty()**

**{**

**"\_id" : ObjectId("6005a91920fb9f3712740512"),**

**"ProductID" : 41,**

**"ProductName" : "Jack's New England Clam Chowder",**

**"SupplierID" : 19,**

**"CategoryID" : 8,**

**"QuantityPerUnit" : "12 - 12 oz cans",**

**"UnitPrice" : 9.65,**

**"UnitsInStock" : 85,**

**"UnitsOnOrder" : 0,**

**"ReorderLevel" : 10,**

**"Discontinued" : 0**

**}**

Dans le shell, créer la variable suivante :

var allProducts=db.products.find();

**var allProducts=db.products.find();**

Exécuter le bout de code suivant dans votre shell

while(ap.hasNext()){ printjson(ap.next()); };

**while(allProducts.hasNext()){ printjson(allProducts.next()); };**

Afficher tous les éléments du curseur : db.products.find() en utilisant la fonction forEach.

**db.products.find().forEach(printjson);**

Dans le shell Mongo, executer la commande suivante

db.products.find({CategoryID:4},{ProductName:1}).pretty();

**db.products.find({CategoryID:4},{ProductName:1}).pretty();**

En utilisant le shell, créer un index de type « text » pour « ProductName » :

**db.products.createIndex( { ProductName: "text" } );**

db.products.find({CategoryID:4},{ProductName:1}).pretty();

Ensuite, executer la commande suivante:

db.products.find({CategoryID:4},{ProductName:1}).sort({ProductName:-1}).pretty();

Afficher les documents de products ayant un « CategoryID » égal à un, par ordre ascendant de ProductName.

**db.products.find().sort({CategoryID:1}).pretty();**

Limiter le nombre des documents à afficher à 2 ;

**db.products.find().limit(2).pretty();**

db.products.find().sort({ProductID:-1}).limit(1).pretty();

Qu’affiche la commande ci-hauts.

**db.products.find().sort({ProductID:-1}).limit(1).pretty();**

**{**

**"\_id" : ObjectId("6005a91920fb9f3712740536"),**

**"ProductID" : 77,**

**"ProductName" : "Original Frankfurter grÃ¼ne SoÃŸe",**

**"SupplierID" : 12,**

**"CategoryID" : 2,**

**"QuantityPerUnit" : "12 boxes",**

**"UnitPrice" : 13,**

**"UnitsInStock" : 32,**

**"UnitsOnOrder" : 0,**

**"ReorderLevel" : 15,**

**"Discontinued" : 0**

**}**

Exécuter la commande suivante :

db.products.insert(

{

"\_id" : ObjectId("60054a6cd020a8ea8dfd8064"),

"ProductID" : 78,

"ProductName" : "Raclette Marocaine",

"SupplierID" : 28,

"CategoryID" : 4,

"QuantityPerUnit" : "5 kg pkg.",

"UnitPrice" : 65,

"UnitsInStock" : 179,

"UnitsOnOrder" : 10,

"ReorderLevel" : 0,

"Discontinued" : 0

}

) ;

Que remarquez-vous ?

**insrted**

Maintenant, exécuter la commande suivante :

db.products.insert(

{

"ProductID" : 78,

"ProductName" : "Raclette Marocaine",

"SupplierID" : 28,

"CategoryID" : 4,

"QuantityPerUnit" : "5 kg pkg.",

"UnitPrice" : 65,

"UnitsInStock" : 179,

"UnitsOnOrder" : 10,

"ReorderLevel" : 0,

"Discontinued" : 0

}

) ;

Maintenant, exécuter la commande suivante :

db.products.insertMany(

[{

"ProductID" : 79,

"ProductName" : "Sirop d'érable Marocain",

"SupplierID" : 29,

"CategoryID" : 2,

"QuantityPerUnit" : "24 - 500 ml bottles",

"UnitPrice" : 28.5,

"UnitsInStock" : 113,

"UnitsOnOrder" : 0,

"ReorderLevel" : 25,

"Discontinued" : 0

}

,

{

"ProductID" : 80,

"ProductName" : "Moroccan Steeleye Stout",

"SupplierID" : 16,

"CategoryID" : 1,

"QuantityPerUnit" : "24 - 12 oz bottles",

"UnitPrice" : 18,

"UnitsInStock" : 20,

"UnitsOnOrder" : 0,

"ReorderLevel" : 15,

"Discontinued" : 0

}

]

);

Exécuter la commande suivante :

db.products.insert(

{

"ProductID" : 81,

"ProductName" : "Raclette Marocaine Spéciale",

"SupplierID" : 28,

"CategoryID" : 4,

"QuantityPerUnit" : "5 kg pkg.",

"UnitPrice" : 65,

"UnitsInStock" : 179,

"UnitsOnOrder" : 10,

"ReorderLevel" : 0,

"Discontinued" : 0,

"DateFirstShip": new Date()

}

) ;

Quel est la différence entre ce document et les documents déjà inséré.

Exécuter la commande suivante :

db.products.find({DateFirstShip:{$exists:true}}) ;

Que renvoi-t-elle?

**db.products.find({DateFirstShip:{$exists:true}}).pretty();**

**{**

**"\_id" : ObjectId("6007409a7c76594081756939"),**

**"ProductID" : 81,**

**"ProductName" : "Raclette Marocaine Spéciale",**

**"SupplierID" : 28,**

**"CategoryID" : 4,**

**"QuantityPerUnit" : "5 kg pkg.",**

**"UnitPrice" : 65,**

**"UnitsInStock" : 179,**

**"UnitsOnOrder" : 10,**

**"ReorderLevel" : 0,**

**"Discontinued" : 0,**

**"DateFirstShip" : ISODate("2021-01-19T20:27:06.550Z")**

**}**

Exécuter la commande suivante:

db.shutdownServer();

Faites la modification, pour que la méthode précédente s’exécute bien.

db.shutdownServer();

shutdown command only works with the admin database; try 'use admin'

**db.shutdownServer();**

**shutdown command only works with the admin database; try 'use admin'**

**> use admin**

**switched to db admin**

**> db.shutdownServer();**

**server should be down...**

Exécuter quit() ;

**quit();**

**C:\Program Files\MongoDB\Server\4.4\bin>**