UNIVERSITÉ PARIS DESCARTES | USPC Université Sorbonne Paris Cité	DUT STID **BASES NoSQL** TP Mises à jour *F. Role*

Il est facile de modifier la structure des documents, ce qui convient bien à une démarche agile.

On travaille sur la collection créée avec les instructions ci-dessous :

```
db.meteo.insert( {"pays" : "France", "ville" : "Marseille" ,
"temperature" : 25 } )

db.meteo.insert( {"pays" : "France", "ville" : "Toulouse" ,
"temperature" : 26 } )

db.meteo.insert( {"pays" : "Italie", "ville" : "Rome" ,
"temperature" :30 } )

db.meteo.insert( {"pays" : "Allemagne", "ville" : "Berlin" ,
"temperature" : 15 } )
```

Exercice 1

On exécute les deux commandes ci-dessous :

```
db.meteo.update( {"pays" : "France", "ville" : "Paris"  }  ,
{ $set : { conditions : ["ensoleillé", "venteux"]} }  )

db.meteo.update( {"ville" : "Moscou"  }  , { $set : { conditions :
undefined} }  )
```

Que souhaite-t-on faire avec ces commandes ? Que constatez- vous ?

Exercice 2

Réexecutez les deux commandes précédentes mais cette fois en donnant comme troisième argument à `update` l'argument ci-dessous :

```
{upsert: true}
```

Que constatez- vous ?

Exercice 3

Essayez de mettre toutes les températures de la France à 20 avec l'instruction :

```
db.meteo.update( {"pays" : "France"  }  , { $set :
{ temperature : 20}
```

Que constatez-vous ?

Exercice 4

Réexecutez la commande précédente mais cette fois en donnant comme troisième argument à update l'argument ci-dessous :

```
{multi : true }
```

Que constatez- vous ?

Exercice 5 (suppression d'un enregistrement)

Supprimez l'enregistrement correspondant à Moscou en utilisant la méthode .remove() à qui vous passez l'id de l'enrgistrement comme dans :

```
db.meteo.remove({ "_id" : ObjectId("55200536d212000000007fee") }) ;
```

Exercice 6 (suppression d'un champ)

Le second argument de update doit être un objet de la forme `{ $unset : { nomduchamp : 1} }`

Enlevez le champ pays de l'enregistrement correspondant à Rome.

Exercice 7

Retrouvez les enregistrements possédant un champ `pays`.

Exercice 8

Enlevez tous les enregistrements de la collection `meteo`.

Exercice 9

Soit la liste `[ [ "Paris" ,"France", 11 ] , ["Shangai" , "Chine" , 16] , ["Rome" , "Italie", 18 ] , ["Moscou" , "Russie", 2 ]]`

Ecrivez un programme qui utilise cette liste pour créer automatiquement dans la base une collection `meteo2` contenant les enregistrements :

```
{pays : "France", ville : "Paris", temperature : 11 }

{pays : "Chine", ville, : "Shangai", temperature : 16 }

{pays : "Italie", ville : "Rome", temperature : 18 }

{pays : "Russie", ville: "Moscou", temperature : 2}
```

Indication : l'instruction `for (var i in l )` permet d'avoir dans `i` l'index courant dans la liste `l`. L'indexation des listes suit la même sytnaxe qu'en Python.

Exercice 10

Utilisez la liste `[ [ "Paris" ,"oceanique"] , ["Shangai" , "tropical"] , ["Rome" , "mediterraneen"] , ["Moscou" , "continental" ]]`

pour ajouter à chaque ville son type de climat. On veut donc que les enregistrements de meteo2 soient :

```
{ "ville" : "Paris", "pays" : "France", "temperature" : 11, "climat" : "oceanique" }

{ "ville" : "Shangai", "pays" : "Chine", "temperature" : 16, "climat" : "tropical" }

{ "ville" : "Rome", "pays" : "Italie", "temperature" : 18, "climat" : "mediterraneen" }

{ "ville" : "Moscou", "pays" : "Russie", "temperature" : 2, "climat" : "continental" }
```