
Laboratório 6 – Bases de Dados NoSQL (SGBD MongoDB)

Objetivos:

- Introdução ao SGBD MongoDB.
 - Criar base de dados e *collections*.
 - Operações CRUD

Materiais:

- Guia Instalação do MongoDB, disponível no Moodle.
- Manual MongoDB
 - <https://www.tutorialspoint.com/mongodb/index.htm>
 - <https://docs.mongodb.com/manual/>

Entrega e Discussão:

A partir da Etapa 2 o trabalho desenvolvido neste laboratório representa um conhecimento que deverá ser integrado no projeto da UC. Relembramos que a 2ª Fase do projeto contempla uma componente de MongoDB que será objeto de discussão (cotação 10%).

ETAPA 0: Utilização da Mongo Shell

- Abrir uma janela de *prompt* de comandos (CMD)
- Ir para diretoria de instalação do MongoDB, exemplo:
`cd C:\Program Files\MongoDB\Server\4.2\bin`
- Abrir a Mongo Shell
`Mongo – port 27017 (ver o número do port no Compass)`
- Criar uma base de dados
`use cbd`
- Ver as bases de dados existentes
`show dbs`
- Criar uma coleção (*Collection*)
`db.createCollection('Customers')`

ETAPA 1: Operações CRUD em MongoDB

1. Inserir documentos

Nota: Substituir #Customer1# pela informação constante do ficheiro Customer.json (disponível no moodle).

- Inserir um documento
`db.Customer.insert(#Customer1#)`
- Inserir vários documentos
`db.Customer.insert(#Customers2#)`

2. Executar consultas sobre documentos

- Obter toda a informação dos documentos
`db.Customer.find()`
- Pesquisa sobre campos dos documentos
`db.Customer.find({_id:1})`

```
db.Customer.find({_id:1}).pretty()  
db.Customer.find({phone:2455550173}).pretty()
```

- Pesquisa sobre campos dos documentos aninhados em outros documentos

```
db.Customer.find({"addresses.city" : "Toronto"}).pretty()  
db.Customer.find({"addresses.city" : "Toronto"}).count()
```

- Operadores lógicos nas pesquisas de documentos

```
db.Customer.find({$and: [{"addresses.city":"Bothell"},{firstname:"Donna"}]})  
db.Customer.find({$or: [{_id:2},{ phone:2455550173}]})
```

- Pesquisa sobre campos não existentes (idêntico ao is null)

```
db.Customer.find({phone: {$exists:false}}).pretty()
```

- Selecionar apenas determinados campos dos documentos (projecção)

```
db.Customer.find({}, {firstname:1})  
db.Customer.find({}, {firstname:1, _id:0})
```

- Ordenar os documentos

```
db.Customer.find({}, {firstname:1, _id:0}).sort({firstname:-1})  
db.Customer.find({}, {firstname:1, _id:0}).sort({firstname:1})
```

- Criar agregação sobre campos de documentos

```
db.Customer.aggregate([{$group : {_id : "$addresses.city", num_cust : {$sum : 1}}}] )
```

3. Atualizar documentos

- Alterar valores de campos de documentos

```
db.Customer.update({_id:1},{ $set:{ emailaddress: "orlando@adventure-works.com" } })
```

- Adicionar um valor a um campo do tipo *array*

```
db.Customer.update({ _id: 1 },{ $push: { phone: 210000089 } })
```

4. Remover documentos

```
db.Customer.insert({_id:5, firstname:"Arnold"})  
db.Customer.find({_id:5})  
db.Customer.remove({_id:5})  
db.Customer.find({_id:5})
```

ETAPA 2: Criar uma Base de Dados no MongoDB

Por forma a obter o feedback dos clientes sobre os seus produtos, a empresa decidiu criar uma base de dados NoSQL no servidor MongoDB. Esta base de dados deve permitir aos clientes classificar e adicionar comentários sobre os produtos que compraram. A informação que caracteriza os comentários/avaliações é a seguinte:

- Identificador do produto
- Nome
- Categoria
- lista de comentários com os seguintes campos:
 - Email do cliente
 - Texto do comentário
 - Classificação (0 a 5)
 - Data do comentário

1. Criar em SQL Server as consultas necessárias para gerar os comandos MongoDB de forma a inserir a informação relativa a:

- Para Produtos da categoria ‘Bikes’:
 - Identificador do produto, Nome e Categoria

Exemplo:

```
db.products.insert({_id:749, name:"Road-150 Red, 62", category:"Road Bikes"})
db.products.insert({_id:750, name:"Road-150 Red, 44", category:"Road Bikes"})
...
```

- Para Clientes:
 - Email e País

Com base no output das consultas crie os ficheiros Produtos.txt e Clientes.txt.

Nota: Utilizar com fonte de informação a base de dados normalizada do AdventureWorksLT2019

2. Criar no MongoDB uma nova base de dados – AdventureWorksMDB e as coleções necessárias para suportar a informação referida no ponto 1.
3. Importar os ficheiros de comandos gerados no ponto 1, para a base de dados AdventureWorksMDB.
4. Inserir um conjunto de comentários ilustrativo da informação manipulada por esta base de

dados e que permita responder às questões listadas na etapa 3.

ETAPA 3: Pesquisas e Inserções de Dados NoSQL

Defina os comandos necessários para:

1. Listar o modelo e categoria do produto com os respetivos comentários
2. Listar os produtos de uma determinada categoria
3. Listar os comentários de um dado utilizador
4. Listar os produtos que não têm comentários
5. Obter o nº de comentários por cliente
6. Obter a classificação média por produto
7. Adicionar um comentário a um determinado produto
8. Remover os comentários de um dado utilizador
9. Obter a média de classificações dos clientes por País (Nome País, Média das Classificações).

ETAPA 4: Integração e Normalização de Informação

Com o objetivo de integrar a informação constante da base de dados de comentários e avaliações na base de dados do AdventureWorksLT2019 existente no SQL Server, deve realizar as seguintes tarefas:

1. Exportar a informação relativa aos comentários para um ficheiro JSON (Comentarios.json).
Exemplo:
`mongoexport --db cbd --collection Customer --pretty --out D:\EST\2019-2020\CBD\projeto\output.json`
2. Importar no SQL Server o ficheiro json para uma nova(s) tabela(s).
3. Proceder à integração e normalização da informação importada para a base de dados criada no âmbito projeto.
4. Criar uma view com a média de classificações dos clientes por País (Nome País, Média das Classificações). Compare em termos de desempenho e complexidade, com o processo desenvolvido em MongoDB (Etapa 3 ponto 9).

(fim de enunciado)