

RESIDÊNCIA



LISTA DE EXERCÍCIOS MONGODB - NOSQL BANCO DE DADOS

ALUNO: JOÃO PAULO MARQUES DE OLIVEIRA

NATAL - RN

1 - Explorando o Compass

- 1.1. Crie o seu cluster no mongo atlas
- 1.2. Conecte-se a seu cluster usando COMPASS
- 1.3. Conecte-se a seu cluster usando o Mongo Shell
- 1.4. Carregue a coleção de tourinfo.restaurants usando o MongoShell com o seguinte comando
\$ sudo mongoimport --uri
"mongodb+srv://jpmarques:akin2010@cluster0.pnqhe.mongodb.net/tourinfo?authSource=admin"
--collection restaurants --file restaurants.json

```
jpmarques@~/Downloads/scripts/db:~$ sudo mongoimport --uri "mongodb+srv://jpmarques:akin2010@cluster0.pnqhe.mongodb.net/tourinfo?authSource=admin" --collection restaurants --file restaurants.json
connected to localhost
2021-02-15T14:30:02.154-0300 [##.....] tourinfo.restaurants 1.04MB/11.3MB (9.2%)
2021-02-15T14:30:05.154-0300 [#####.....] tourinfo.restaurants 6.19MB/11.3MB (54.8%)
2021-02-15T14:30:08.154-0300 [#####.....] tourinfo.restaurants 10.5MB/11.3MB (92.8%)
2021-02-15T14:30:09.072-0300 [#####.....] tourinfo.restaurants 11.3MB/11.3MB (100.0%)
imported 25359 documents
```

- 1.5. Explore a coleção no COMPASS
 - 1.5.1. Document structure
 - 1.5.2. Visualização de schema
 - 1.5.2.1 Visualização de dados geográficos expandindo o campo address

2 - Coleção movieDetails e Linguagem de consulta

- 2.1. Carregue a coleção de videos.movieDetails usando o Mongo Shell
 - 2.1.1. Caso a base video.movieDetails exista, apague-a usando o COMPASS local
 - 2.1.2. Abra um command prompt e vá para a pasta /dbs na pasta da aula
 - 2.1.3. Execute mongo e dê os seguintes comandos
\$ use video
\$ load ('loadMovieDetailsDataset.js')

```
MongoDB Enterprise atlas-z888c8-shard-0:PRIMARY> use video
switched to db video
MongoDB Enterprise atlas-z888c8-shard-0:PRIMARY> load ('loadMovieDetailsDataset.js')
true
MongoDB Enterprise atlas-z888c8-shard-0:PRIMARY> █
```

- 2.2. Explore a coleção no COMPASS
 - 2.2.1. Estrutura dos documentos
 - 2.2.2. Visualizações do esquema
- 2.3. Questões
 - 2.3.1 - \$ db.movieDetails.find({'genres':'Comedy'}).count();
 - 2.3.2 - \$ db.movieDetails.find({'rated':'PG-13', 'awards.nominations':10}).count();
 - 2.3.3 - \$ db.movieDetails.find({'genres':'Family'}).count();
 - 2.3.4 - \$ db.movieDetails.find({'genres.1':'Western'}).count();
 - 2.3.5 - \$ db.movieDetails.find({'writers':{'\$in':['Ethan Coen','Joel Coen']}}).count();
 - 2.3.6 - \$ db.movieDetails.find({'rated':{'\$in':['PG-13','R']}}).count();
 - 2.3.7 - \$ db.movieDetails.find({'year':2013,'rated':'PG-13','awards.wins':0},{title:1,_id:0});
 - 2.3.8 - \$ db.movieDetails.find({'countries.1':'Sweden'}).count();

3 - Coleção Grades (versão flat) e Linguagem de consulta

3.1. Carregue a coleção de students.grades_flat usando o Mongo Shell, com o seguinte comando:

```
$ sudo mongoimport --uri  
"mongodb+srv://jpmarques:akin2010@cluster0.pnqhe.mongodb.net/students?authSource=admin"  
--collection grades --file grades_flat.json
```

```
joao@joao:~/Downloads/Scripts/dbs$ sudo mongoimport --uri "mongodb+srv://jpmarques:akin2010@cluster0.pnqhe.mongodb.net/students?authSource=admin" --collection grades --file grades_flat.json  
2021-02-15T16:36:04.411-0300 connected to: localhost  
2021-02-15T16:36:05.540-0300 imported 800 documents
```

3.2. Explore a coleção no COMPASS

3.2.1. Estrutura dos documentos

3.2.2. Visualizações do esquema

3.2.3. Confirme que você carregou corretamente o seu arquivo.

Utilizar o seguinte comando deve retornar 800:

```
$ use students  
$db.grades.count()
```

```
MongoDB Enterprise atlas-z888c8-shard-0:PRIMARY> use students  
switched to db students  
MongoDB Enterprise atlas-z888c8-shard-0:PRIMARY> db.grades.count()  
800  
MongoDB Enterprise atlas-z888c8-shard-0:PRIMARY> █
```

3.2.4. Esta próxima consulta informará o student_id com a pontuação média mais alta:

```
$ db.grades.aggregate(  
  [  
    { '$group': { '_id': '$student_id', 'average': { '$avg': '$score' } } },  
    { '$sort': { 'average': -1 } }, { '$limit': 1 }  
  ]  
)
```

3.3.1. Para responder esta questão, foi utilizado o seguinte código:

```
$ db.grades.find(  
  { 'score': { '$gt': 65 }, 'type': 'exam' },  
  { 'student_id': 1, '_id': 0 }  
) .sort({ 'score': 1 });
```

```
MongoDB Enterprise atlas-z888c8-shard-0:PRIMARY> db.grades.find({'score':{'$gt':65},'type':'exam'},{'student_id':1, '_id':0}).sort({'score':1});  
{ "student_id" : 22 }  
{ "student_id" : 115 }  
{ "student_id" : 48 }  
{ "student_id" : 57 }  
{ "student_id" : 87 }  
{ "student_id" : 194 }  
{ "student_id" : 116 }  
{ "student_id" : 111 }  
{ "student_id" : 199 }  
{ "student_id" : 96 }  
{ "student_id" : 163 }  
{ "student_id" : 67 }  
{ "student_id" : 166 }  
{ "student_id" : 120 }  
{ "student_id" : 64 }  
{ "student_id" : 189 }  
{ "student_id" : 104 }  
{ "student_id" : 190 }  
{ "student_id" : 180 }  
{ "student_id" : 184 }  
Type "it" for more  
MongoDB Enterprise atlas-z888c8-shard-0:PRIMARY> █
```

3.3.2. Código disponibilizado em anexo

4 - Coleção Students_200_hw e Linguagem de consulta

4.1. Carregue a coleção de school.students_200_hw usando o Mongo Shell utilizando o seguinte comando:

```
$ sudo mongoimport --uri  
"mongodb+srv://jpmarques:akin2010@cluster0.pnqhe.mongodb.net/students?authSource=admin"  
--collection grades_full --file students_200_hw.json
```

```
joao@joao:~/Downloads/Scripts/db$ sudo mongoimport --uri "mongodb+srv://jpmarques:akin2010@cluster0.pnqhe.mongodb.net/students?authSource=admin" --collection grades_full --file students_200_hw.json  
[sudo] password for joao: connected to localhost  
2021-02-15T20:28:26.246-0300 [#####] students.grades_full 51.6KB/51.6KB (100.0%)  
2021-02-15T20:28:27.321-0300
```

4.2. Explore a coleção no COMPASS

4.2.1. Estrutura dos documentos

4.2.2. Visualizações do esquema

4.2.3. Confirme que temos, de fato, 200 documentos utilizando o seguinte comando

```
$ db.grades_full.count()
```

```
MongoDB Enterprise atlas-z888c8-shard-0:PRIMARY> db.grades_full.count()  
200  
MongoDB Enterprise atlas-z888c8-shard-0:PRIMARY> █
```

4.3. Utilizando o código (script em anexo) obtemos o seguinte resultado para o _id 137:

```
▼  
_id: 137  
name: "Tamika Schildgen"  
scores: Array  
  0: Object  
    type: "exam"  
    score: 4.433956226109692  
  1: Object  
    type: "quiz"  
    score: 65.50313785402548  
  2: Object  
    type: "homework"  
    score: 89.5950384993947
```

4.4. Utilizando o código (script em anexo) obtemos a seguinte resposta:

```
joao@joao:~/Documents/BANCO DE DADOS/NOSQL$ python3 Exercicio_4.3.4.py  
{'_id': 13, 'average': 91.98315917172745}
```

5 - Coleção posts_hw e Consultas com Agregação

- 5.1. Carregue a coleção de blog.posts_hw usando o Mongo Shell
- 5.2. Explore a coleção no COMPASS
- 5.2.2. Visualizações do esquema
- 5.2.3. Confirme que temos, de fato, 1000 documentos

```
MongoDB Enterprise atlas-z888c8-shard-0:PRIMARY> db.posts.find().count()  
1000
```

- 5.2.4. Qual o autor com o maior número de comentários?

```
$ db.posts.aggregate(  
    [{ $unwind: "$comments" },  
    { $group: { "_id": "$comments.author", "count": { $sum: 1 } } },  
    { $sort: { "count": 1 } } ]  
)
```

```
MongoDB Enterprise atlas-z888c8-shard-0:PRIMARY> db.posts.aggregate([{$unwind: "$comments"},{$group: {"_id": "$comments.author", "count": {$sum: 1}}},{$sort: {'count': 1}}])  
{ "_id": "Mariela Sherer", "count": 387 }  
{ "_id": "Tawana Oberg", "count": 396 }  
{ "_id": "Kayce Kenyon", "count": 400 }  
{ "_id": "Maren Scheider", "count": 401 }  
{ "_id": "Tanika Schildgen", "count": 404 }  
{ "_id": "Kaila Deibler", "count": 406 }  
{ "_id": "Jessica Dagenais", "count": 406 }  
{ "_id": "Zachary Langlais", "count": 406 }  
{ "_id": "Marcus Blohm", "count": 408 }  
{ "_id": "Tandra Meadows", "count": 410 }  
{ "_id": "Fleta Duplantis", "count": 411 }  
{ "_id": "Shin Albright", "count": 413 }  
{ "_id": "Joaquina Arbuckle", "count": 413 }  
{ "_id": "Merissa Mann", "count": 415 }  
{ "_id": "Salena Olmos", "count": 415 }  
{ "_id": "Edgar Sarkis", "count": 416 }
```

6 - Coleção small_zip e Consultas com Agregação

- 6.1. Carregue a coleção de zips.small_zips usando o Mongo Shell utilizando o seguinte comando:
\$ sudo mongoimport --uri
"mongodb+srv://jpmarques:akin2010@cluster0.pnqhe.mongodb.net/zip?authSource=admin"
--collection small_zips --file small_zips.json

- 6.2. Explore a coleção no COMPASS

- 6.2.1. Estrutura dos documentos
- 6.2.2. Visualizações do esquema
- 6.2.3. Confirme que temos, de fato, 200 documentos

```
MongoDB Enterprise atlas-z888c8-shard-0:PRIMARY> db.small_zips.find().count()  
200  
MongoDB Enterprise atlas-z888c8-shard-0:PRIMARY> █
```

```
6.3 - db.small_zips.aggregate([  
    { "$match": { "pop": { "$gt": 25000 } } },  
    { "$match": { "state": { "$in": [ "NY", "CT" ] } } },  
    { "$group": { "_id": null, "population": { "$avg": "$pop" } } }  
)
```

```
MongoDB Enterprise atlas-z888c8-shard-0:PRIMARY> db.small_zips.aggregate([{"$match": { "pop": { "$gt": 25000 } }}, {"$match": {"state": {"$in": ["NY", "CT"]}}}, {"$group": {"_id": "$city", "population": {"$avg": "$pop"}}}])  
{ "_id": "WESTBURY", "population": 38926 }  
{ "_id": "WILSTON", "population": 46676 }  
{ "_id": "WINDSOR", "population": 27815 }  
{ "_id": "WATERBURY", "population": 25128 }  
{ "_id": "FLUSHING", "population": 51947 }  
{ "_id": "CHESHIRE", "population": 25684 }  
{ "_id": "GROTON", "population": 32435 }  
{ "_id": "JAMAICA", "population": 50404 }  
{ "_id": "MIDDLETOWN", "population": 42846 }  
{ "_id": "BRIDGEPORT", "population": 23219 }  
{ "_id": "BINGHAMTON", "population": 30741 }  
{ "_id": "BROOKLYN", "population": 36852 }  
{ "_id": "DANBURY", "population": 37764 }  
{ "_id": "ENDWELL", "population": 46236 }  
{ "_id": "ROCHESTER", "population": 30655 }  
{ "_id": "EAST HARTFORD", "population": 22754 }  
{ "_id": "WEST HAVEN", "population": 53642 }  
MongoDB Enterprise atlas-z888c8-shard-0:PRIMARY> db.small_zips.aggregate([{"$match": { "pop": { "$gt": 25000 } }}, {"$match": {"state": {"$in": ["NY", "CT"]}}}, {"$group": {"_id": null, "population": {"$avg": "$pop"}}}])  
{ "_id": null, "population": 38319.470588235294 }
```

7 - Coleção grades e Consultas com Agregação

7.1 - Carregue a coleção de `grades` usando o Mongo Shell utilizando o seguinte comando:

```
$ sudo mongoimport --uri
"mongodb+srv://jpmarques:akin2010@cluster0.pnqhe.mongodb.net/grades?authSource=admin"
--collection grades_complete --file grades.json
```

7.2. Explore a coleção no COMPASS

7.2.1. Estrutura dos documentos

7.2.2. Visualizações do esquema

7.2.3. Confirme que temos, de fato, 280 documentos

```
MongoDB Enterprise atlas-z888c8-shard-0:PRIMARY> db.grades_complete.find().count()
280
MongoDB Enterprise atlas-z888c8-shard-0:PRIMARY> █
```

7.6. Qual é o `class_id` que tem o maior desempenho médio dos alunos?

```
db.grades_complete.aggregate([
  {$unwind: "$scores"},
  {$match: {'scores.type': {'$in': ['exam', 'homework']}}},
  {$group: {'_id': '$student_id', '$student_id': '$student_id', 'class_id': '$class_id',
    aaa: {'$avg': '$scores.score'}}},
  {$group: {'_id': '$_id.class_id', 'avg': {'$avg': '$aaa'}}}, {$sort: {'_id': 1}}
])
```

8 - Coleção zips e Consultas com Agregação

8.1. Carregue a coleção de `zipcode.zips` usando o Mongo Shell com o seguinte código:

```
$ sudo mongoimport --uri
"mongodb+srv://jpmarques:akin2010@cluster0.pnqhe.mongodb.net/zip?authSource=admin"
--collection zips --file zips.json
```

8.2. Explore a coleção no COMPASS

8.2.1. Estrutura dos documentos

8.2.2. Visualizações do esquema

8.2.3. Confirme que temos, de fato, 29353 documentos.

```
MongoDB Enterprise atlas-z888c8-shard-0:PRIMARY> db.zips.find().count()
29353
MongoDB Enterprise atlas-z888c8-shard-0:PRIMARY> █
```

8.5 -

```
$ db.zips.aggregate([
  {$group: {'_id': '$city', 'pop': {'$sum': '$pop'}}},
  {$project: {'first_char': {'$substr': ['$_id', 0, 1]}, 'pop': '$pop'}},
  {$match: {'first_char': {'$in': ['B', 'D', 'O', 'G', 'N', 'M']}}},
  {$sort: {'pop': -1}}
])
```

```
MongoDB Enterprise atlas-z888c8-shard-0:PRIMARY> db.zips.aggregate([{$group: {'_id': '$city', 'pop': {'$sum': '$pop'}}}, {$project: {'first_char': {'$substr': ['$_id', 0, 1]}, 'pop': '$pop'}}, {$match: {'first_char': {'$in': ['B', 'D', 'O', 'G', 'N', 'M']}}}, {$sort: {'pop': -1}}])
{"_id": "BROOKLYN", "first_char": "B", "pop": 2341387 }
{"_id": "NEW YORK", "first_char": "N", "pop": 1476790 }
{"_id": "BRONX", "first_char": "B", "pop": 1289548 }
{"_id": "DALLAS", "first_char": "D", "pop": 959042 }
{"_id": "DETROIT", "first_char": "D", "pop": 927468 }
{"_id": "MIAMI", "first_char": "M", "pop": 848436 }
{"_id": "BALTIMORE", "first_char": "B", "pop": 738846 }
{"_id": "MEMPHIS", "first_char": "M", "pop": 643485 }
{"_id": "MILWAUKEE", "first_char": "M", "pop": 597324 }
{"_id": "NEWARK", "first_char": "N", "pop": 504651 }
{"_id": "NEW ORLEANS", "first_char": "N", "pop": 490937 }
{"_id": "DENVER", "first_char": "D", "pop": 472054 }
{"_id": "GREENVILLE", "first_char": "G", "pop": 427483 }
{"_id": "BUFFALO", "first_char": "B", "pop": 400942 }
{"_id": "NASHVILLE", "first_char": "N", "pop": 399946 }
{"_id": "GLENDALE", "first_char": "G", "pop": 391979 }
{"_id": "OAKLAND", "first_char": "O", "pop": 390529 }
{"_id": "MADISON", "first_char": "M", "pop": 382916 }
{"_id": "OKLAHOMA CITY", "first_char": "O", "pop": 368141 }
{"_id": "ORAMA", "first_char": "O", "pop": 363830 }
Type "it" for more
MongoDB Enterprise atlas-z888c8-shard-0:PRIMARY> █
```