

Exercícios MongoDB

Para as atividades a seguir você poderá escolher usar a sua base de dados em um cluster do mongo ou na sua base local.

1. Instalando MongoDB para Uso Local

- Instale o MongoDB Server Community Edition.
Leia instruções em <https://docs.mongodb.com/manual/installation/>
- Instale o MongoDB Shell (mongosh).
Leia instruções em <https://www.mongodb.com/docs/mongodb-shell/install/>
- Instale o MongoDB Compass.
Leia instruções em <https://www.mongodb.com/docs/compass/master/install/>

2. Coleção `movieDetails` e Linguagem de Consulta

- 2.1. Carregue a coleção de `hw.movieDetails` usando o Mongo Shell
 - 2.1.1. Caso a base exista, apague-a usando o COMPASS local
 - 2.1.2. Descompacte o arquivo **exemplos.zip** disponível no SIGAA
 - 2.1.3. Abra um command prompt e vá para a pasta `/dbs` que está no arquivo descompactado acima
 - 2.1.4. Execute o Mongo Shell `mongosh` e dê os seguintes comandos

```
use hw
```

```
load("loadMovieDetailsDataset.js")
```

- 2.2. Explore a coleção no COMPASS
 - 2.2.1. Estrutura dos documentos
 - 2.2.2. Visualizações do esquema
 - 2.2.3. Confirme que você carregou corretamente o seu arquivo. Você deveria ter 2295 documentos.
- 2.3. Obtenha as seguintes respostas usando COMPASS ou Mongo shell. Sua resposta deve conter a resposta à pergunta e a consulta que gerou a resposta.
 - 2.3.1. Quantos filmes são do gênero “Comedy”?
 - 2.3.2. Quantos filmes são ranqueados com PG-13 e tem exatamente 10 nomeações de prêmios?
 - 2.3.3. Quantos filmes tem “Family” como um de seus gêneros?
 - 2.3.4. Quantos filmes tem “Western” como segundo elemento de seus gêneros?
 - 2.3.5. Usando o operador `$in`, filtre a coleção `video.movieDetails` para determinar quantos filmes listam "Ethan Coen" ou "Joel Coen" entre seus escritores. Seu filtro deve corresponder a todos os filmes que listam qualquer um dos irmãos Coen como escritores independentemente de quantos outros escritores também estão listados. Qual a quantidade de filmes que satisfazem a este filtro das opções abaixo.
 - 2.3.6. Quantos filmes são ranqueados como “R” ou “PG-13”?
 - 2.3.7. Qual o título do filme do ano 2013 que é classificado como PG-13 e ganhou nenhum prêmio?

2.3.8. Quantos filmes listam “Sweden” como segundo na lista de países?

3. Coleção grades (versão flat) e Linguagem de Consulta

3.1. Carregue a coleção de hw.grades_flat usando o Mongo Shell

3.1.1.Caso a base exista, apague-a usando o COMPASS local

3.1.2. Arquivo fonte: dbs/grades_flat.json

3.1.3. Abra um command prompt e vá para a pasta /dbs mencionada acima

3.1.4. Execute o seguinte comando

```
mongoimport --drop --file grades_flat.json --db hw --collection grades_flat
```

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. Você deveria ter 800 documentos.

3.2.4.Esta próxima consulta, que usa a estrutura de agregação, informará o student_id com a pontuação média mais alta:

```
use hw

db.grades_flat.aggregate(

  [ { '$group': {'_id': '$student_id', 'average': { '$avg': '$score' } } },
    { '$sort': {'average': -1 } }, { '$limit': 1 }

  ]

)
```

Resultado: [{ _id: 164, average: 89.29771818263373 }]

3.3. Obtenha as seguintes respostas tanto usando COMPASS ou Mongo shell.

3.3.1.Encontre todos os resultados dos exames maiores ou iguais a 65, e classifique os resultados do menor ao mais alto. **Qual é o student_id do menor índice de exame acima de 65?**

3.3.2.Escreva um programa na linguagem de programação de sua escolha (usando o respectivo driver de MongoDB) que remova a nota do tipo "homework" com a pontuação mais baixa para cada aluno do conjunto de dados. Como cada documento é uma nota, ele deve remover um documento por aluno. Este exercício usará o mesmo conjunto de dados que a última questão. **Qual a identidade do aluno com a média mais alta na classe com a seguinte consulta que usa a estrutura de agregação?** A resposta aparecerá no campo _id do documento resultante. **Anexe o arquivo fonte Exercício_4.3.2.EXT à sua resposta (EXT corresponde à extensão dos arquivos fontes da linguagem de programação de escolha).**

```
db.grades.aggregate(

  { '$group' : { '_id' : '$student_id', 'average' : { $avg : '$score' } } },
  { '$sort' : { 'average' : -1 } },
  { '$limit' : 1 }

)
```

4. Coleção `students_200_hw` e Linguagem de Consulta

4.1. Carregue a coleção de `hw.students_200_hw` usando o Mongo Shell

4.1.1. Caso a base exista, apague-a usando o COMPASS local

4.1.2. Arquivo fonte: `dbfs/students_200_hw.json`

4.1.3. Abra um command prompt e vá para a pasta `/dbfs` mencionada acima

4.1.4. Execute o seguinte comando

```
mongoimport --drop --file students_200_hw.json --db hw --collection students_200
```

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

4.3. Escreva um programa na linguagem de programação de sua escolha (usando o respectivo driver de MongoDB) que irá remover a nota de “homework” mais baixa para cada aluno. Uma vez que existe um único documento para cada aluno contendo uma série de pontuações, você precisará atualizar o array de pontuação e remover o “homework”. Lembre-se, basta remover uma pontuação de “homework”. Não remova um questionário ou um exame!

4.3.1. Dica: Uma maneira é encontrar o “homework” mais baixo no código e, em seguida, atualizar o array de pontuação o removendo.

4.3.2. No final, ao contar o número de documentos, o resultado deveria continuar sendo 200

4.3.3. No final, as notas de “Tamika Schildgen's” são as seguintes

```
{
  "_id" : 137,
  "name" : "Tamika Schildgen",
  "scores" : [
    {
      "type" : "exam",
      "score" : 4.433956226109692
    },
    {
      "type" : "quiz",
      "score" : 65.50313785402548
    },
    {
      "type" : "homework",
      "score" : 89.5950384993947
    }
  ]
}
```

4.3.4. Qual a identidade (na forma de seu `_id`) do aluno com a média mais alta na classe com a seguinte consulta que usa a estrutura de agregação? A resposta aparecerá no campo `_id` do documento resultante. **Anexe o arquivo fonte Exercício_5.3.4.EXT à sua resposta (EXT corresponde à extensão dos arquivos fontes da linguagem de programação de escolha).**

```
db.students.aggregate( [
  { '$unwind': '$scores' },
  {
    '$group':
    {
```

```

        '_id': '$_id',
        'average': { $avg: '$scores.score' }
    }
},
{ '$sort': { 'average' : -1 } },
{ '$limit': 1 } ] )

```

5. Coleção `posts_hw` e Consultas com Agregação

5.1. Carregue a coleção de `hw.posts_hw` usando o Mongo Shell

5.1.1. Caso a base exista, apague-a usando o COMPASS local

5.1.2. Arquivo fonte: `dbs/posts_hw.json`

5.1.3. Abra um command prompt e vá para a pasta `/dbs` mencionada acima

5.1.4. Execute o seguinte comando

```
mongoimport --drop --file posts_hw.json --db hw --collection posts
```

5.2. Explore a coleção no COMPASS

5.2.1. Estrutura dos documentos

5.2.2. Visualizações do esquema

5.2.3. Confirme que temos, de fato, 1000 documentos

5.3. Qual o autor com o maior número de comentários? Você precisará usar a estrutura de agregação para responder a esta pergunta. Para ajudá-lo a verificar seu trabalho antes de enviar, o autor com o menor número de comentários é Mariela Sherer e ela comentou 387 vezes.

6. Coleção `small_zips` e Consultas com Agregação

6.1. Carregue a coleção de `hw.small_zips` usando o Mongo Shell

6.1.1. Caso a base exista, apague-a usando o COMPASS local

6.1.2. Arquivo fonte: `dbs/small_zips.json`

6.1.3. Abra um command prompt e vá para a pasta `/dbs` mencionada acima

6.1.4. Execute o seguinte comando

```
mongoimport --drop --file small_zips.json --db hw --collection small_zips
```

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

6.3. Qual a população média das cidades na Califórnia (CA) e Nova York (NY) (tomadas em conjunto) com populações acima de 25.000? Para esse problema, presuma que um nome da cidade que aparece em mais de um estado representa duas cidades separadas. Arredonde a resposta para um número inteiro.

6.4. Dica: A resposta para CT e NJ (usando este conjunto de dados) é 38177.

6.5. Observe:

6.5.1. Diferentes estados podem ter o mesmo nome da cidade.

6.5.2. Uma cidade pode ter múltiplos códigos postais.

- 6.5.3. Para esse problema, usamos um subconjunto dos dados que usamos anteriormente em `zips.json`, não o conjunto completo. Para este conjunto, existem apenas 200 documentos (e 200 códigos postais), e todos estão em Nova York, Connecticut, Nova Jersey e Califórnia.
- 6.5.4. Este é um dado bruto do Serviço Postal dos Estados Unidos. Se você notar que durante a importação, há algumas duplicatas não se preocupe, isso é esperado e não afetará sua resposta.
- 6.5.5. Depois de gerar sua consulta de agregação e encontrar sua resposta, diga qual o resultado.

7. Coleção `grades` e Consultas com Agregação

- 7.1. Carregue a coleção de `hw.grades` usando o Mongo Shell
 - 7.1.1. Caso a base exista, apague-a usando o COMPASS local
 - 7.1.2. Arquivo fonte: `db/grades.json`
 - 7.1.3. Abra um command prompt e vá para a pasta `/db` mencionada acima
 - 7.1.4. Execute o seguinte comando

```
mongoimport --drop --file grades.json --db hw --collection grades
```

- 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
- 7.3. Existem documentos para cada aluno (`student_id`) em uma variedade de classes (`class_id`). Observe que nem todos os alunos da mesma classe têm o mesmo número exato de avaliações. Alguns alunos têm três trabalhos de casa, etc.
- 7.4. Sua tarefa é calcular a classe com o melhor desempenho médio dos alunos. Isso envolve o cálculo de uma média para cada aluno em cada classe de todas as avaliações não-quiz e, em seguida, a média desses números para obter uma média de classe. Para ser claro, a média de cada aluno inclui apenas exames e notas de lição de casa. Não inclua as suas pontuações em quiz nos cálculos.
- 7.5. Dica / Estratégia: você precisa agrupar duas vezes para resolver esse problema. Você deve descobrir a média que cada aluno alcançou em uma aula e, em seguida, a média desses números para obter uma média de classe. Depois disso, você só precisa classificar. A classe com a média mais baixa é a classe com `class_id = 2`. Esses alunos alcançaram uma média de classe de 37,6
- 7.6. Qual é o `class_id` que tem o maior desempenho médio dos alunos?

8. Coleção `zips` e Consultas com Agregação

- 8.1. Carregue a coleção de `hw.zips` usando o Mongo Shell
 - 8.1.1. Caso a base exista, apague-a usando o COMPASS local
 - 8.1.2. Arquivo fonte: `db/zips.json`
 - 8.1.3. Abra um command prompt e vá para a pasta `/db` mencionada acima
 - 8.1.4. Execute o seguinte comando

```
mongoimport --drop --file zips.json --db hw --collection zips
```

- 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

- 8.3. Neste problema, você calculará o número de pessoas que vivem em um CEP nos EUA onde o nome da cidade começa com algumas letras específicas.
- 8.4. O estágio \$project pode extrair o primeiro caractere de qualquer campo. Por exemplo, para extrair a primeira letra do campo da cidade, você poderia escrever esse pipeline:

```
db.zips.aggregate([
  // Project the first character and the population
  {$project:
    {
      first_char: {$substr : ["$city",0,1]},...
    }
  },
  ...
])
```

- 8.5. **Qual a soma total de pessoas que vivem em um CEP onde a cidade começa com uma das seguintes letras: B, D, O, G, N ou M?** Você precisará usar agregação para responder a esta pergunta.
- 8.6. Você precisará provavelmente mudar sua projeção para enviar mais informações do que apenas essa primeira letra. Além disso, você precisará de um passo de filtragem para se livrar de todos os documentos onde a cidade não começa com o conjunto selecionado de caracteres iniciais.