

Disciplina:

Banco de Dados Não Relacional

Professor:

Thiago Gonçalves Mendes

Tecnologia:

MongoDB

Integrantes
Eryck Gabryel Ribeiro Lino
João Augusto S. Pacolla
Luana Marques Gomes
Maikon Fabrício Gino

Relatório de combustível do primeiro semestre de 2020.

O combustível deste relatório trata-se do etanol, todo o banco de dados estará disponível no GitHub link: [JPacolla376/MongoDB Replica SET: Tem como objeto a análise dos preços de combustíveis no Brasil, utilizando o MongoDB e seu framework de agregação. \(github.com\)](https://github.com/JPacolla376/MongoDB_Replica_SET:_Tem_como_objeto_a_analise_dos_precos_de_combustiveis_no_Brasil_utilizando_o_MongoDB_e_seu_framework_de_agregacao)

O objetivo desse relatório é a utilização dos conceitos de busca no banco de dados não relacional em específico o conceito de Replica SET.

Remodelagem do Banco – Array

O script MongoDB abaixo é um pipeline de agregação que reestrutura os dados na coleção Etanol_remodelado. Sua finalidade é categorizar as informações por estado e região, gerando um array que contém documentos detalhados para cada combinação de estado e região. Este processo facilita a análise e visualização das informações agrupadas, permitindo uma melhor compreensão dos dados segmentados geograficamente.

```
db.Etanol_remodelado([
  {
    $unwind: "$dados"
  },
  {
    $addFields: {
      valorVendaCorrigido: {
        $cond: {
          if: { $isNumber: "$dados.Valor de Venda" },
          then: { $toString: "$dados.Valor de Venda" },
          else: "$dados.Valor de Venda"
        }
      }
    }
  },
  {
    $addFields: {
      valorVendaNumerico: {
        $toDouble: {
          $replaceAll: {
            input: "$valorVendaCorrigido",
            find: ",",
            replacement: "."
          }
        }
      }
    }
  }
])
```

```

    }
  }
}
},
{
  $group: {
    _id: null,
    mediaPrecoTotal: { $avg: "$valorVendaNumerico" }
  }
},
{
  $project: {
    _id: 0,
    mediaPrecoTotal: 1
  }
}
});

```

Média do Etanol no 1º semestre de 2020

```

db.Etanol_remodelado.aggregate([
  {
    $unwind: "$dados"
  },
  {
    $addFields: {
      valorVendaCorrigido: {
        $cond: {
          if: { $isNumber: "$dados.Valor de Venda" },
          then: { $toString: "$dados.Valor de Venda" },
          else: "$dados.Valor de Venda"
        }
      }
    }
  },
  {
    $addFields: {
      valorVendaNumerico: {
        $toDouble: {
          $replaceAll: {
            input: "$valorVendaCorrigido",
            find: ",",
            replacement: "."
          }
        }
      }
    }
  }
]);

```

```

    }
  }
},
{
  $group: {
    _id: null,
    mediaPrecoTotal: { $avg: "$valorVendaNumerico" }
  }
},
{
  $project: {
    _id: 0,
    mediaPrecoTotal: 1
  }
}
]);

```

```

>_MONGOSH
> db.Etanol_remodelado.aggregate([
  {
    $unwind: "$dados"
  },
  {
    $addFields: {
      valorVendaCorrigido: {
        $cond: {
          if: { $isNumber: "$dados.Valor de Venda" },
          then: { $toString: "$dados.Valor de Venda" },
          else: "$dados.Valor de Venda"
        }
      }
    }
  },
  {
    $addFields: {
      valorVendaNumerico: {
        $toDouble: {
          $replaceAll: {
            input: "$valorVendaCorrigido",
            find: ",",
            replacement: "."
          }
        }
      }
    }
  }
]);

```

>_MONGOSH

```
,
{
  $addFields: {
    valorVendaNumerico: {
      $toDouble: {
        $replaceAll: {
          input: "$valorVendaCorrigido",
          find: ",",
          replacement: "."
        }
      }
    }
  }
},
{
  $group: {
    _id: null,
    mediaPrecoTotal: { $avg: "$valorVendaNumerico" }
  }
},
{
  $project: {
    _id: 0,
```

```

        }
    }
}
},
{
    $group: {
        _id: null,
        mediaPrecoTotal: { $avg: "$valorVendaNumerico" }
    }
},
{
    $project: {
        _id: 0,
        mediaPrecoTotal: 1
    }
}
]);
< {
    mediaPrecoTotal: 3.194551413419622
}

```

1. Descompactação dos Dados:

- Utiliza **\$unwind** para descompactar o array de dados, permitindo que cada elemento do array se torne um documento separado.

2. Correção do Valor de Venda:

- Adiciona um campo **valorVendaCorrigido** que converte o valor de venda para uma string se não for numérico.

3. Conversão para Número:

- Adiciona um campo **valorVendaNumerico** que converte o valor de venda corrigido para um número de ponto flutuante.

4. Agregação e Cálculo da Média:

- Agrupa os documentos em uma chave única (**_id: null**) e calcula a média dos valores de venda utilizando **\$avg**.

5. Projeção do Resultado:

- Projeta apenas o campo **mediaPrecoTotal**, removendo o **_id**, para apresentar somente a média total do preço de venda de etanol.

Esse código realiza uma operação de agregação no banco de dados para calcular a média total do preço de venda de etanol, corrigindo valores e realizando os cálculos necessários.

Média de preço no primeiro semestre de 2020.

```
db.Etanol_remodelado.aggregate([
  {
    $unwind: "$dados"
  },
  {
    $addFields: {
      valorVendaCorrigido: {
        $cond: {
          if: { $eq: [{ $type: "$dados.Valor de Venda" }, "string"] },
          then: "$dados.Valor de Venda",
          else: { $toString: "$dados.Valor de Venda" }
        }
      },
      valorVendaNumerico: {
        $toDouble: {
          $replaceAll: {
            input: { $cond: { if: { $eq: [{ $type: "$dados.Valor de Venda" }, "string"] }, then: "$dados.Valor de Venda", else: { $toString: "$dados.Valor de Venda" } } },
            find: ",",
            replacement: "."
          }
        }
      }
    }
  },
  {
    $project: {
      _id: 0,
      mediaPrecoTotal: { $avg: "$valorVendaNumerico" }
    }
  }
])
```

```

        $toDate: {
            $multiply: ["$dados.Data da Coleta", 86400000]
        }
    },
    {
        $group: {
            _id: { $month: "$dataColeta" },
            mediaPreco: { $avg: "$valorVendaNumerico" }
        }
    },
    {
        $project: {
            _id: 0,
            mes: "$_id",
            media: "$mediaPreco"
        }
    },
    {
        $sort: { mes: 1 }
    }
});

```



```

>_MONGOSH
{
  mes: 1,
  media: 3.429366532645748
}
{
  mes: 2,
  media: 3.4508444613821143
}
{
  mes: 3,
  media: 3.4415685531058373
}
{
  mes: 4,
  media: 3.1301980262585865
}
{
  mes: 5,
  media: 2.860194072987663
}
{
  mes: 6,
  media: 2.9134954003407154
}

```

1. Desagrupando array

```

{
  $unwind: "$dados"
}

```

Nessa parte do código, estamos desmembrando o array **dados** em documentos individuais. Isso facilita a manipulação de cada entrada separadamente.

2. Adicionar e ajustar campos

```

{
  $addFields: {
    valorVendaCorrigido: {
      $cond: {
        if: { $eq: [{ $type: "$dados.Valor de Venda" }, "string"] },
        then: "$dados.Valor de Venda",
        else: { $toString: "$dados.Valor de Venda" }
      }
    },
    valorVendaNumerico: {
      $toDouble: {
        $replaceAll: {
          input: "$valorVendaCorrigido",
          find: ",",
          replacement: "."
        }
      }
    },
    dataColeta: {
      $toDate: {
        $multiply: ["$dados.Data da Coleta", 86400000]
      }
    }
  }
}

```

- **valorVendaCorrigido:** Verifica se o campo "Valor de Venda" é uma string. Se não for, converte para string.
- **valorVendaNumerico:** Converte "Valor de Venda" em número, substituindo vírgulas por pontos decimais.
- **dataColeta:** Converte "Data da Coleta" para um formato de data, multiplicando o valor em dias pelo número de milissegundos em um dia para obter a data correta.

3. Agrupação por mês.

```

{

```

```

    $group: {
      _id: { $month: "$dataColeta" },
      mediaPreco: { $avg: "$valorVendaNumerico" }
    }
  }
}

```

Agrupar os dados pelo mês da dataColeta e calcular a média dos preços de venda (mediaPreco) para cada mês

4. Campos projetados

```

{
  $project: {
    _id: 0,
    mes: "$_id",
    media: "$mediaPreco"
  }
}

```

Seleciona e renomeia os campos para a saída final: **mes** e **media**.

5. Ordenar os resultados por mês

```

{
  $sort: { mes: 1 }
}

```

Ordena os resultados em ordem crescente mês.

Combustível mais alto.

```

db.Etanol_remodelado.aggregate([
  {
    $unwind: "$dados"
  },
  {
    $addFields: {
      valorVendaCorrigido: {
        $cond: {

```

```

    if: { $eq: [{ $type: "$dados.Valor de Venda" }, "string"] },
    then: {
      $replaceAll: {
        input: "$dados.Valor de Venda",
        find: ",",
        replacement: "."
      }
    },
    else: {
      $replaceAll: {
        input: { $toString: "$dados.Valor de Venda" },
        find: ",",
        replacement: "."
      }
    }
  }
}
}
}
},
{
  $addFields: {
    valorVendaNumerico: { $toDouble: "$valorVendaCorrigido" }
  }
},
{
  $match: {
    valorVendaNumerico: { $ne: null }
  }
},
{
  $sort: { valorVendaNumerico: -1 }
},
{
  $limit: 1
},
{
  $project: {
    _id: 0,
    estado: "$Estado - Sigla",
    valorVenda: "$valorVendaNumerico"
  }
}
})

```

```

        valorVendaNumerico: { $ne: null }
    }
},
{
    $sort: { valorVendaNumerico: -1 }
},
{
    $limit: 1
},
{
    $project: {
        _id: 0,
        estado: "$Estado - Sigla",
        valorVenda: "$valorVendaNumerico"
    }
}
])
< {
    estado: 'RJ',
    valorVenda: 5.099
}

```

1. Desdobrar os dados:

O estágio **\$unwind** é usado para desdobrar o array **dados**, transformando cada elemento do array em um documento separado. Isso facilita a manipulação dos dados.

2. Adicionar e ajustar campos:

Os estágios **\$addFields** são usados para criar novos campos e ajustar os valores existentes. No caso, estamos corrigindo o campo "Valor de Venda" para garantir que seja uma string e, em seguida, convertendo-o para um valor numérico.

3. Filtrar valores nulos:

O estágio **\$match** é usado para filtrar documentos onde o campo "Valor de

Venda" numérico seja diferente de nulo, garantindo que estejamos trabalhando apenas com valores válidos.

4. Ordenar por preço de venda decrescente:

O estágio **\$sort** é usado para classificar os documentos com base no valor de venda numérico em ordem decrescente, colocando os valores mais altos no topo.

5. Limitar o resultado ao combustível mais caro:

O estágio **\$limit** é usado para limitar o resultado a apenas um documento, ou seja, o documento com o valor de venda mais alto.

6. Projetar os campos desejados:

O estágio **\$project** é usado para selecionar e renomear os campos que desejamos incluir na saída final. Neste caso, estamos selecionando o estado onde o valor de combustível mais alto foi encontrado e o próprio valor.

Combustível com o valor mais baixo.

```
db.Etanol_remodelado.aggregate([
  {
    $unwind: "$dados"
  },
  {
    $addFields: {
      valorVendaCorrigido: {
        $cond: {
          if: { $eq: [{ $type: "$dados.Valotr de Venda" }, "string"] },
          then: {
            $replaceAll: {
              input: "$dados.Valor de Venda",
              find: ",",
              replacement: "."
            }
          },
          else: {
            $replaceAll: {
              input: { $toString: "$dados.Valor de Venda" },
              find: ",",

```

```

        replacement: "."
      }
    }
  }
}
},
{
  $addFields: {
    valorVendaNumerico: { $toDouble: "$valorVendaCorrigido" }
  }
},
{
  $match: {
    valorVendaNumerico: { $ne: null }
  }
},
{
  $sort: { valorVendaNumerico: 1 }
},
{
  $limit: 1
},
{
  $project: {
    _id: 0,
    estado: "$dados.Estado",
    valorVenda: "$valorVendaNumerico"
  }
}
])

```

```

        valorVendaNumerico: { $ne: null }
    }
},
{
    $sort: { valorVendaNumerico: 1 }
},
{
    $limit: 1
},
{
    $project: {
        _id: 0,
        estado: "$Estado - Sigla",
        valorVenda: "$valorVendaNumerico"
    }
}
])
< {
    estado: 'SP',
    valorVenda: 1.799
}

```

1. Desdobrar os dados:

O estágio **\$unwind** é usado para desdobrar o array **dados**, transformando cada elemento do array em um documento separado. Isso facilita a manipulação dos dados.

2. Adicionar e ajustar campos:

Os estágios **\$addFields** são usados para criar novos campos e ajustar os valores existentes. Neste caso, estamos corrigindo o campo "Valor de Venda" para garantir que seja uma string e, em seguida, convertendo-o para um valor numérico.

3. Filtrar valores nulos:

O estágio **\$match** é usado para filtrar documentos onde o campo "Valor de

Venda" numérico seja diferente de nulo, garantindo que estejamos trabalhando apenas com valores válidos.

4. **Ordenar por preço de venda crescente:**

O estágio **\$sort** é usado para classificar os documentos com base no valor de venda numérico em ordem crescente, colocando os valores mais baixos no topo.

5. **Limitar o resultado ao combustível mais barato:**

O estágio **\$limit** é usado para limitar o resultado a apenas um documento, ou seja, o documento com o valor de venda mais baixo.

6. **Projetar os campos desejados:**

O estágio **\$project** é usado para selecionar e renomear os campos que desejamos incluir na saída final. Neste caso, estamos selecionando o estado onde o valor de combustível mais baixo foi encontrado e o próprio valor.