

**INSTITUTO
FEDERAL**
Santa Catarina

Banco de Dados 2



Bancos de Dados não relacionais

1. Introdução
2. MongoDB
3. Operações DML
4. Operações DQL
- 5. Schema Design**
- 6. Otimização**
- 7. Agregação**



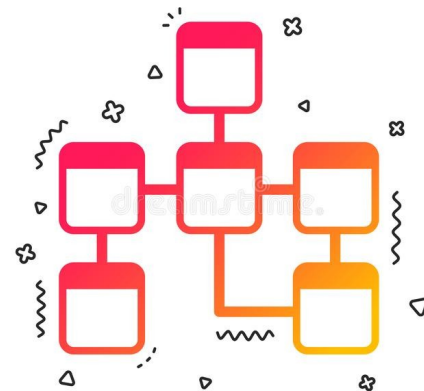
5/8

Schema Design



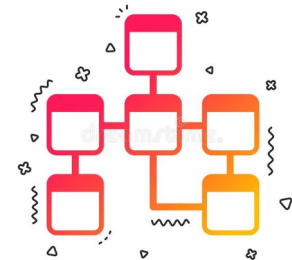
Normalização

- Revistando o conceito:
- São diretrizes que podem ser usadas para determinar a qualidade do projeto do esquema de relação:
 - Garantir que a semântica dos atributos seja clara no esquema;
 - Reduzir a informação redundante nas tuplas;
 - Reduzir os valores NULL nas tuplas;
 - Reprovar a possibilidade de gerar tuplas falsas.
- Porém o MongoDB não possui operações de JOIN



Desnormalização

- Este conceito está relacionado ao agrupamento de todos os dados em uma única coleção.
- Como vantagem, podemos ter consultas mais rápidas. A normalização faz a escrita ser mais rápida e a desnormalização faz a leitura ser mais rápida
- E desvantagens?



Schema Design – qual caminho tomar?

Diretor

```
{
  "_id":1,
  "nome":"Joseph Kosinski",
  "país":"EUA"
}
```

Atores

```
[
  {
    "_id":1,
    "nome":"Tom Cruise"
  },
  {
    "_id":2,
    "nome":"Miles Teller"
  },
  {
    "_id":3,
    "nome":"Jennifer Connelly"
  }
]
```

Filme

```
{
  "titulo":"Top Gun",
  "duracao":121,
  "ano": 2022,
  "diretor":1,
  "elenco":[1,2,3]
}
```

Schema Design – qual caminho tomar?

Filme

```
{  
  "titulo":"Top Gun",  
  "duracao":121,  
  "ano": 2022,  
  "diretor":{  
    "nome":"Joseph Kosinski",  
    "país":"EUA"  
  },  
  "elenco":["Tom Cruise", "Miles Teller", "Jennifer  
Connelly"]  
}
```

6/8

Otimização



- Para criar um índice, utilizamos:

```
db.produtos.createIndex( { "nome": 1 } )
```

- Para criar um índice com valor único:

```
db.produtos.createIndex( { "nome": 1 }, {"unique", true} )
```

- Verificando os índices:

```
db.produtos.getIndexes();
```

7/8

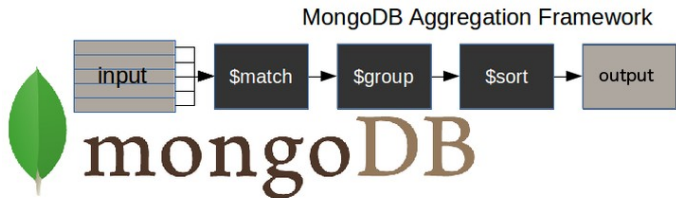
Agregação



Aggregation Framework

- Assim como no SQL, podemos fazer agregações como soma e média de valores. Uma das formas é utilizando o “Aggregation Framework” do MongoDB.
- Fornece várias funções para processamento de dados.

```
db.orders.aggregate( [  
  // Stage 1: Filter pizza order documents by pizza size  
  {  
    $match: { size: "medium" }  
  },  
  // Stage 2: Group remaining documents by pizza name and  
  // calculate total quantity  
  {  
    $group: { _id: "$name", totalQuantity: { $sum: "$quantity" } }  
  }  
])
```



Map Reduce

- É uma ferramenta que permite resolver problemas mais complexos utilizando programação javascript como linguagem para os comandos.
- No entanto, possui um problema de performance em processamento em tempo real.
- Quando possível utilize preferencialmente o “Aggregation Framework”

```

Collection
↓
db.orders.mapReduce(
  map   → function() { emit( this.cust_id, this.amount ); },
  reduce → function(key, values) { return Array.sum( values ); },

  query → { query: { status: "A" },
  output → out: "order_totals"
}
)
  
```

Map Reduce - exemplo

