

TECNOLOGIAS DE PROGRAMAÇÃO DE SISTEMAS DE
INFORMAÇÃO

MongoDB

SISTEMAS GESTORES DE BASES DE DADOS II | Prof. Magno
Andrade | Prof. Jorge Louro

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



- **Conteúdo da apresentação - Parte 2**
 - **Análise de dados**
 - *MapReduce*
 - *Agregações únicas*

ANÁLISE DE DADOS - MAPREDUCE

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



- **Introdução**

- O *MapReduce* é um paradigma de processamento de dados, para condensar grandes volumes de dados em resultados agregados importantes.
- Para efectuar o *MapReduce*, o MongoDB disponibiliza para esse efeito o comando *mapReduce*.

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



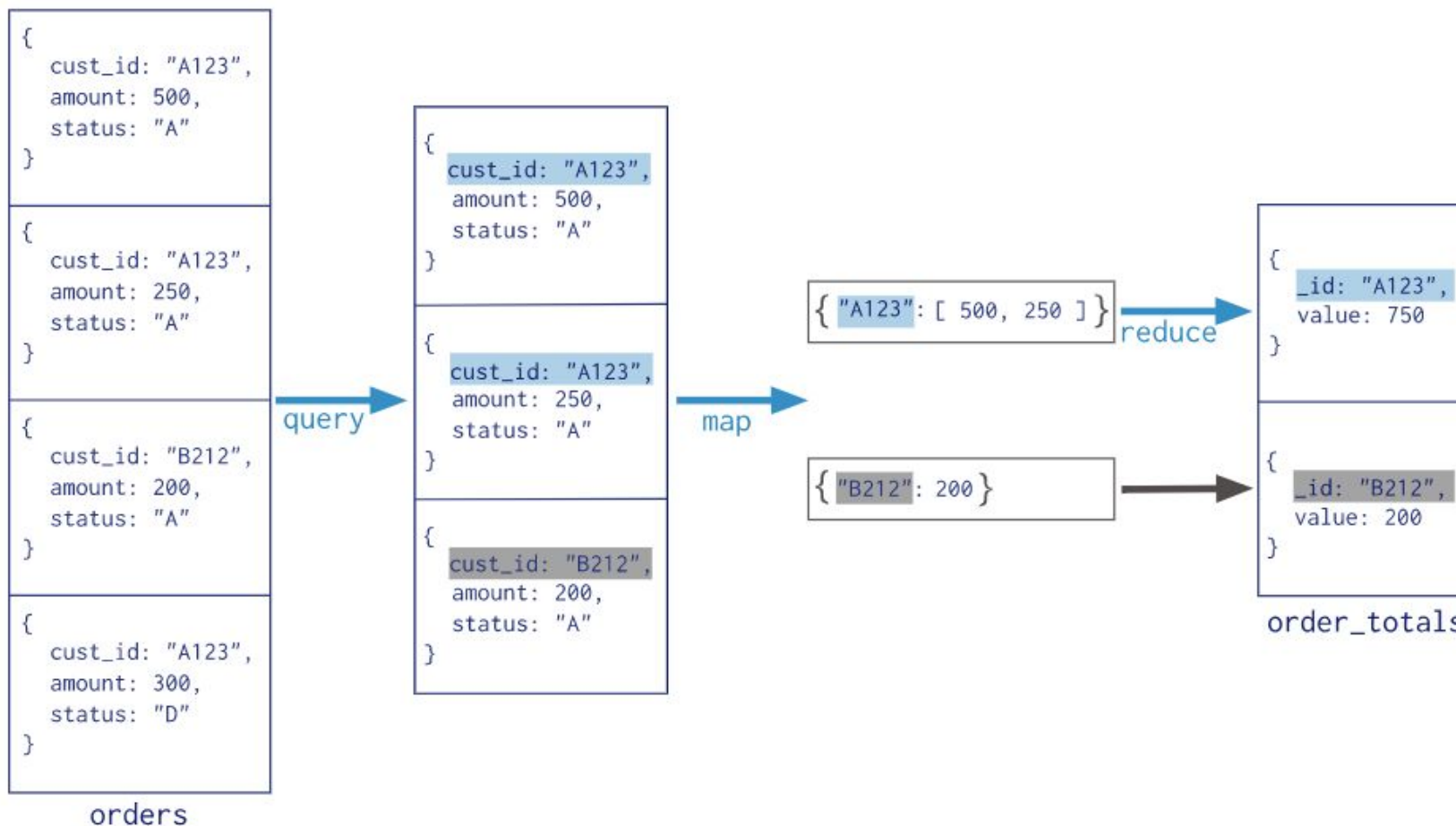
mongoDB®

- Exemplo

```
Collection
↓
db.orders.mapReduce(
  map    → function() { emit( this.cust_id, this.amount ); },
  reduce → function(key, values) { return Array.sum( values ) },
  query  → {
  output →   query: { status: "A" },
            out: "order_totals"
            }
)
```


UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II

- Exemplo (cont.)



UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



- **Funcionamento - comando *mapReduce***
 - É o comando utilizado na mongo *shell* para realizar as operações *mapReduce*.
 - **Nota:** as *views* não suportam operações *map-reduce*

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II

- **Funcionamento - comando *mapReduce***

- Tem a seguinte sintaxe:

```
db.collection.mapReduce(  
    <map>,  
    <reduce>,  
    {  
        out: <collection>,  
        query: <document>,  
        sort: <document>,  
        limit: <number>,  
        finalize: <function>,  
        scope: <document>,  
        jsMode: <boolean>,  
        verbose: <boolean>,  
        bypassDocumentValidation: <boolean>  
    }  
)
```



mongoDB®

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II

- **Funcionamento - comando *mapReduce***
 - **Parâmetros**



Parâmetro	Tipo	Descrição
map	função	Uma função <i>javascript</i> , que associa ou faz o mapeamento de um valor com uma chave e emite o par chave e valor.
reduce	função	Uma função <i>javascript</i> , que reduz para um único objecto, todos os valores associados a uma chave em particular.
options	documento	Um documento que especifica parâmetros adicionais.
bypassDocumentValidation	booleano	Opcional. Permite ultrapassar a validação de documentos durante a operação. Isto permite assim inserir documentos que não satisfazem as validações.

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II

- Funcionamento - comando *mapReduce*
 - Alguns argumentos



Campo	Tipo	Descrição
out	<i>string</i> ou documento	Especifica a localização do resultado da operação <i>map-reduce</i> . O <i>output</i> pode ser para uma colecção ou só apresentar os resultados.
query	documento	Especifica o critério de selecção usando operadores de <i>query</i> para determinar os documentos de entrada, para a função <i>map</i> .
sort	documento	Ordena os documentos de entrada. Esta opção é importante na optimização. Por exemplo, especificamos a chave de ordenação (<i>sort key</i>) para ser a mesma da <i>emit key</i> , fazendo com que haja menos operações de <i>reduce</i> .

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II

- Funcionamento - comando *mapReduce*
 - Alguns argumentos (cont.)



Campo	Tipo	Descrição
limit	número	Especifica um número máximo de documentos para a entrada da função <i>map</i> .
finalize	função	Opcional. É realizada logo a seguir ao <i>reduce</i> e modifica o <i>output</i> .
scope	documento	Especifica as variáveis globais que estão acessíveis nas funções <i>map</i> , <i>reduce</i> e <i>finalize</i> .
verbose	booleano	Especifica se deve incluir as informações de tempo nas informações dos resultados. Definir como true para incluir as informações de tempo. Por omissão é false .

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II

- Funcionamento - comando *mapReduce*
 - Requisitos da função *map*

- A função *map* é responsável por transformar cada documento de entrada em zero ou mais documentos. Consegue aceder às variáveis definidas no parâmetro *scope*.
- Tem a seguinte sintaxe base:

```
function() {  
    ...  
    emit(key, value);  
}
```



UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



- Funcionamento - comando *mapReduce*
 - Requisitos da função *map* - *Exemplo*
 - A seguinte função *map*, chama *emit(key,value)* 0 ou 1 vezes, dependendo do valor do campo *status* no documento de entrada.

```
function() {  
    if (this.status == 'A')  
        emit(this.cust_id, 1);  
}
```


UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



- **Funcionamento - comando *mapReduce***
 - **Requisitos da função *map* - Exemplo**
 - A seguinte função ***map***, chama *emit(key,value)* múltiplas vezes, dependendo do número de elementos nos itens do documento de entrada.

```
function() {  
    this.items.forEach(function(item){ emit(item.sku, 1); });  
}
```

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II

- Funcionamento - comando *mapReduce*

- Requisitos da função *reduce*

- A função *reduce* é responsável por fazer a redução para um único objecto, a todos os valores associados com uma chave em particular.
- Tem a seguinte sintaxe base:

```
function(key, values) {  
    ...  
    return result;  
}
```



UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



- **Funcionamento - comando *mapReduce***
 - **Requisitos do campo *out***
 - Especificar a localização do resultado da operação *map-reduce*.
 - Tem a seguinte sintaxe:
 - `out: <collectionName>`
 - Para fazer *output* em linha:
 - `out: { inline: 1 }`

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



- **Funcionamento - Exemplo**

- Na operação de *MapReduce*, o MongoDB aplica a fase de ***map***, para cada um dos documentos de entrada (ex: os documentos da colecção que satisfazem os critérios definidos na *query*).
- A função *map* emite pares chave-valor.

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



- **Funcionamento - Exemplo**

- Para estas chaves que têm múltiplos valores, é aplicado a fase de ***reduce***, que recolhe e condensa os dados agregados.
- Os resultados são então guardados numa colecção.
- Opcionalmente, o *output* da função ***reduce*** pode passar por uma função ***finalize*** para condensar e processar ainda mais os resultados da agregação.

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



- **Framework e MapReduce**

- Para a maioria das operações de agregação, a *framework* de agregação do MongoDB fornece melhor desempenho e uma interface mais coerente.
- No entanto, as operações *map-reduce* disponibilizam uma flexibilidade que não existe, nem está presente no *pipeline* de agregação.

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



- **Funções de *JavaScript Map-Reduce***

- As operações *map-reduce* utilizam funções customizadas de *Javascript* para o mapeamento, ou associar, valores a uma chave.
- Se uma chave tem múltiplos valores mapeados para si própria, a operação *reduce*, reduz os valores da chave para um único objecto.

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II

- **Map-Reduce**
 - **Exemplo 1**

- Tendo em conta uma colecção **orders**, que tem múltiplos documentos do seguinte tipo:

```
{
  _id: ObjectId("50a8240b927d5d8b5891743c"),
  cust_id: "abc123",
  ord_date: new Date("Oct 04, 2012"),
  status: 'A',
  price: 25,
  items: [ { sku: "mmm", qty: 5, price: 2.5 },
            { sku: "nnn", qty: 5, price: 2.5 } ]
}
```



UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



- **Map-Reduce - Exemplo 1**
 - Obter o preço total (*total price*) para cada cliente (*cust_id*).
 - Realizar a operação de *map-reduce* na colecção *orders*, para agrupar por *cust_id*, e calcular a soma do campo *price* para cada *cust_id*.
 - Definir a função *map* para processar cada documento de entrada.

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



- **Map-Reduce - Exemplo 1**
 - Obter o preço total (*total price*) para cada cliente (*cust_id*).
 - Função *map*:

```
var mapFunction1 = function() {  
    emit(this.cust_id, this.price);  
};
```


UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



- **Map-Reduce - Exemplo 1**
 - **Obter o preço total (*total price*) para cada cliente (*cust_id*).**
 - Nesta função, ***this*** refere-se ao documento que a operação *map-reduce* está a processar.
 - A função “mapea” o valor do campo ***price*** para o ***cust_id*** para cada documento e emite o par *cust_id* e *price*.

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



mongoDB®

- **Map-Reduce - Exemplo 1**
 - Obter o preço total (*total price*) para cada cliente (*cust_id*).
 - Definir a função **reduce** com dois argumentos **keyCustId** e **valuesPrices**.
 - O argumento **valuesPrices** é um *array* cujos elementos são os valores **price** emitidos pela função **map** e agrupados pela **keyCustId**.
 - A função reduz o *array* **valuesPrice**, para a soma dos seus elementos.

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



- **Map-Reduce - Exemplo 1**
 - Obter o preço total (*total price*) para cada cliente (*cust_id*).
 - Função *reduce*:

```
var reduceFunction1 = function(keyCustId, valuesPrices) {  
    return Array.sum(valuesPrices);  
};
```

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



mongoDB®

- **Map-Reduce - Exemplo 1**
 - Obter o preço total (*total price*) para cada cliente (*cust_id*).
 - Função final:

```
db.orders.mapReduce(  
    mapFunction1,  
    reduceFunction1,  
    { out: "map_reduce_example" }  
)
```

ANÁLISE DE DADOS - AGREGAÇÃO ÚNICA

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



mongoDB®

- **Agregação Única - Introdução**

- O MongoDB proporciona os seguintes comandos para agregação única:
 - `db.collection.estimatedDocumentCount()`
 - `db.collection.count()`
 - `db.collection.distinct()`

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



- **Agregação Única - Introdução**

- Todas estas operações agregam documentos de uma única colecção.
- Enquanto que estas operações fornecem um acesso simples, para efectuar processos de agregação, falta flexibilidade e capacidades de agregação, como existe no *pipeline* e *map-reduce*.

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II

- **Agregação Única - Exemplo**

Collection
↓
`db.orders.distinct("cust_id")`

<pre>{ cust_id: "A123", amount: 500, status: "A" }</pre>
<pre>{ cust_id: "A123", amount: 250, status: "A" }</pre>
<pre>{ cust_id: "B212", amount: 200, status: "A" }</pre>
<pre>{ cust_id: "A123", amount: 300, status: "D" }</pre>

orders

distinct → ["A123", "B212"]



mongoDB®

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



- **Agregação Única**
 - `db.collection.estimatedDocumentCount()`
 - Retorna a contagem de todos os documentos numa coleção ou *view*.
 - Tem a seguinte sintaxe:
 - `db.collection.estimatedDocumentCount(<options>)`

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



- **Agregação Única**
 - `db.collection.estimatedDocumentCount()` - Exemplo
 - `db.orders.estimatedDocumentCount({})`
 - Com o comando acima, é obtida a contagem total de todos os documentos na colecção ***orders***.

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



- **Agregação Única**

- `db.collection.count()`

- Retorna a contagem de documentos que correspondia a uma query *find()* para uma colecção ou *view*.
 - O método não realiza a operação *find()*, em vez disso, conta e retorna o número de resultados que correspondem a uma *query*.

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



- **Agregação Única**
 - `db.collection.count()`
 - Tem a seguinte sintaxe:
 - `db.collection.count(query, options)`
 - Este comando é equivalente a:
 - `db.collection.find(query).count()`

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



- **Agregação Única**

- `db.collection.count()` - Exemplo 1

- Contar todos os documentos de uma coleção.

- `db.orders.count()`

- O comando acima é equivalente a:

- `db.orders.find().count()`

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II

- **Agregação Única**

- `db.collection.count()` - Exemplo 2

- Contar o número de documentos na coleção ***orders***, com o campo ***ord_dt*** maior do que ***new Date('01/01/2012')***.

- Utilizamos o seguinte comando:

- `db.orders.count({ ord_dt: { $gt: new Date('01/01/2012') } })`
 - equivalente a
 - `db.orders.find({ ord_dt: { $gt: new Date('01/01/2012') } }).count()`



UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



- **Agregação Única**

- `db.collection.distinct()`
 - Encontra os valores distintos para um campo específico numa única colecção ou *view* e retorna os resultados num *array*.
 - Os resultados não podem exceder o tamanho máximo de 16 *MB*.
 - Se os resultados excederem o tamanho máximo, devemos utilizar a função ***aggregate*** com a etapa ***\$group***, para obter os valores distintos.

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



- **Agregação Única**
 - `db.collection.distinct()`
 - Tem a seguinte sintaxe:
 - `db.collection.distinct(field, query, options)`

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



- **Agregação Única**
 - `db.collection.distinct()` - Parâmetros

Parâmetro	Tipo	Descrição
field	string	O campo para o qual retornar valores distintos.
query	documento	Uma <i>query</i> que especifica os documentos, a partir do qual se obtém os valores distintos.
options	documento	Opcional. Um documento que especifica as opções.

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



- **Agregação Única**

- `db.collection.distinct()` - Exemplo

- Tendo em conta os seguintes documentos:

- `{ "_id": 1, "dept": "A", "item": { "sku": "111", "color": "red" }, "sizes": ["S", "M"] }`
 - `{ "_id": 2, "dept": "A", "item": { "sku": "111", "color": "blue" }, "sizes": ["M", "L"] }`
 - `{ "_id": 3, "dept": "B", "item": { "sku": "222", "color": "blue" }, "sizes": "S" }`
 - `{ "_id": 4, "dept": "A", "item": { "sku": "333", "color": "black" }, "sizes": ["S"] }`

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



- **Agregação Única**

- `db.collection.distinct()` - Exemplo 1

- O exemplo seguinte retorna os valores distintos para o campo ***dept*** de todos os documentos da colecção ***inventory***.
 - Usando o seguinte comando:
 - `db.inventory.distinct("dept")`
 - Obtemos o seguinte resultado em *array*:
 - `["A", "B"]`

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



- **Agregação Única**
 - `db.collection.distinct()` - Exemplo 2
 - O exemplo seguinte retorna os valores distintos para o campo ***sku***, que está embebido no campo ***item***, de todos os documentos na colecção ***inventory***.

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



mongoDB®

- **Agregação Única**

- `db.collection.distinct()` - Exemplo 2
 - Usando o seguinte comando:
 - `db.inventory.distinct("item.sku")`
 - Obtemos o seguinte resultado em *array*:
 - `["111", "222", "333"]`

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



- **Agregação Única**

- `db.collection.distinct()` - Exemplo 3

- O exemplo seguinte retorna os valores distintos para o campo ***sizes*** de todos os documentos na coleção ***inventory***.
 - Usando o seguinte comando:
 - `db.inventory.distinct("sizes")`
 - Obtemos o seguinte resultado em *array*:
 - `["M", "S", "L"]`

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



- **Agregação Única**
 - `db.collection.distinct()` - Exemplo 4
 - O seguinte exemplo retorna os valores distintos para o campo ***sku***, embebido no campo ***item***, dos documentos em que o campo ***dept*** é igual a **"A"**.

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



mongoDB®

- **Agregação Única**

- `db.collection.distinct()` - Exemplo 4

- Usando o seguinte comando:

- `db.inventory.distinct("item.sku", { dept: "A" })`

- Obtemos o seguinte resultado em *array*:

- `["111", "333"]`

UNIDADE CURRICULAR : SISTEMAS GESTORES DE BASES DE DADOS II



- **Referências**

- <https://docs.mongodb.com/manual/aggregation/>
- <https://docs.mongodb.com/manual/tutorial/perform-incremental-map-reduce/>
- <https://docs.mongodb.com/manual/tutorial/map-reduce-examples/>



CTeSP

CURSOS TÉCNICOS
SUPERIORES PROFISSIONAIS