

Agregacja

Aggregation Pipeline

Mechanizm agregacji wbudowany w Mongo jest alternatywą dla MapReduce

Wynikiem agregacji jest dokument lub zbiór dokumentów

Działa natywnie w mongod (vs JS w mapReduce)

Agregacje predefiniowane komendy

```
db.collection.count(<query>)
```

```
db.collection.distinct("field")
```

```
db.collection.group(  
  {  
    key: <document>, //po których polach grupować  
    cond: <document>, //zapytanie  
    reduce: <function>, //obsługa kolejnych dokumentów  
    initial: <document> // początkowy dokument dla każdej grupy  
  })
```

Pipeline

Wzorowany na operatorze | w komendach unixowych

Wynik z każdego etapu jest przekazywany do przetworzenia do kolejnego etapu

Każdy etap może zmieniać ilość dokumentów wynikowych

Etapy mogą występować dowolną ilość razy

Agregacja - wykonywanie

```
db.collection.aggregate(  
  [  
    //...pipeline  
  ]  
)
```

Agregacja - wykonywanie

Jedna kolekcja

Ograniczanie ilości dokumentów na jak najwcześniejszym etapie

Indeksy

Mongo stara się optymalizować różne sposoby agregacji (zmiana kolejności, łączenie)

Wyrażenia, operatory np.:

"newField": "\$existingSubdocument.existingField"

Agreagacja - operator

\$match (= WHERE)

\$group (= GROUP BY)

\$match (= HAVING)

\$project (= SELECT)

Agreagacja - operator

\$sort (= ORDER BY)

\$limit (= LIMIT)

\$sum (= SUM)

\$sum (= COUNT)

\$unwind (~ JOIN)

Agregacja operator \$match

```
{ $match: { <query> } }
```

Powinien być użyty jak najwcześniej.

Jeśli jest pierwszym etapem to może korzystać z indeksów.

Nie można używać \$where.

\$text tylko jeśli to pierwszy etap (potrzebuje indeksu).

Agregacja operator \$group

```
{ $group:
  {
    _id: <expr>,
    <field>: { <accumulator>: <expr> },
    //...
  }
}
```

Grupowanie po polach w _id.

Dla _id: null nie grupuje po niczym.

Max 100mb ramu.

Agregacja operator \$project

```
{ $project: { <specifications> } }
```

Przekazuje dalej jedynie określone właściwości dokumentów

Agregacja operator \$sort

```
{ $sort:
  {
    <field1>: <sort order>,
    <field2>: <sort order>
    // ...
  }
}
```

Projekcja istniejących oraz dodawanie nowych pól (na podstawie wyrażeń).

Jeśli jest od razu przed \$limit to trzyma tylko wymagana liczbę dokumentów w pamięci.

Max 100mb ramu.

Agregacja operatorzy \$limit, \$sum

\$limit:

```
{ $limit: <positive integer> }
```

\$sum:

```
{ $sum: <expression> }
```

Agregacja operator \$unwind

```
{ $unwind: <field path> }
```

Rozbija tablice (iloczyn kartezjański wartości z tablicy z resztą pól z dokumentu).

Działa tylko dla pól z tablicą lub pól nieistniejących.

Agregacja operator \$sample

```
{ $sample:{size:N }}
```

Zwraca N losowych elementów

Dostępny od Mongo 3.2

Agregacja operator \$lookup

Dostępny od Mongo 3.2

Wykonuje „left outer join”

Działa tylko dla kolekcji bez shardingu

Działa tylko w ramach jednej bazy

Agregacja operator \$lookup

```
{ $lookup:
  {
    from: <collectionToJoin>,
    localField: <fieldA>,
    foreignField: <fieldB>,
    as: <nameOfReturnedArray>
  }
}
```

Agregacja operator \$lookup

From – nazwa kolekcji z tej samej bazy, która zostanie dołączona

localField – pole z dokumentu bazowego, po którym nastąpi złączenie

foreignField – pole z dołączonego dokumentu, po którym nastąpi złączenie

As – nazwa zwróconej tablicy ze złączonymi dokumentami

Agregacja - ograniczenia

Pojedynczy zwracany dokument max 16mb

Wynik może być zapisany do kolekcji lub zwrócony jako kursor

Każdy pojedynczy etap ma limit 100mb RAM