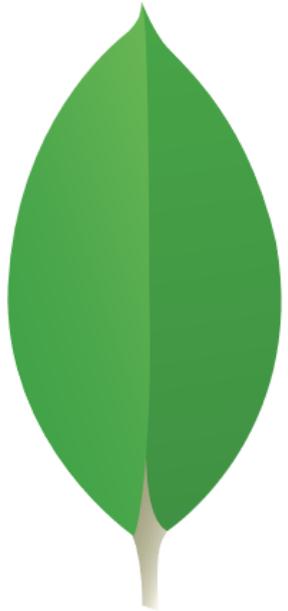


Grundlagen



mongoDB

Lernziele



mongoDB

- Aggregation Framework
 - Pipelining
 - Expressions

Aggregation Framework



- Bisher gab es die „group()“ Funktion welche auch zur Daten Aggregation verwendet wurde. Jedoch war die Funktionalität dieser Funktion beschränkt:
 - Funktionierte nicht auf einer mittels Sharding verteilten Umgebung
 - War auf 20.000 Dokumente im Ergebnisdokument beschränkt
- Im Gegensatz dazu bietet das Aggregation Framework eine erweiterte Funktionalität, welche in ca. 80% aller Anwendungsfälle ausreichend sein dürfte.
- Der Funktionsumfang ist in etwa einer GROUP...BY Funktion in SQL vergleichbar, jedoch sind im Aggregation Framework noch zusätzliche Funktionen wie Berechnen der Summe/Mittelwert / Min-/Maximum möglich.
- Das Aggregation Framework ist deklarativ → d.h. man muss kein JavaScript schreiben

Aggregation Framework



- Das Aggregation Framework baut auf 2 Grundkonzepten auf:

1. Pipelining

- Es werden eine Kette von vordefinierten Operationen festgelegt, welche sequenziell auf Dokumente angewendet werden.
- Es wird bereits ein sinnvolles Set an Operationen zum Gruppieren, Filtern und Sortieren von Daten mitgeliefert.

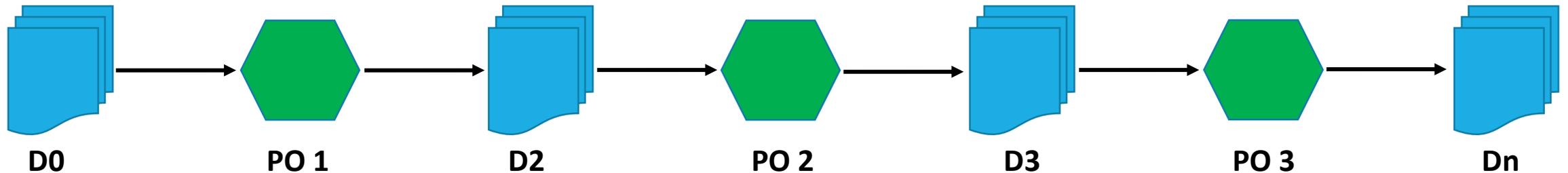
2. Expressions

- Mit Hilfe der Expressions kann, innerhalb einer Stufe der Pipeline, eine Manipulation auf Dokumenten durchgeführt werden.
- Sie können nur auf dem aktuell in der Pipeline verarbeiteten Dokument agieren.
- Es ist nicht möglich, zusätzliche Daten aus anderen Dokumenten hinzuzufügen.

Aggregation Framework



Pipelining



- Es werden Dokumente von einer vordefinierten Operation zur nächsten Pipeline-Operation weitergeleitet
- Diese Operationen verarbeiten den Dokumentenstrom wie beim Auslesen einer Collection.
- Sie können miteinander kombiniert oder wiederholt werden um so neue Dokumente zu erzeugen
- Jeder Pipeline Operator PO_i erzeugt dabei als Ausgabe eine **neue** Menge von Dokumenten.
- D.h. man ist nicht darauf beschränkt, dass für jedes eingehende Dokument ein ausgehendes Dokument erzeugt wird → es können innerhalb einer Stufe neue Dokumente angelegt oder gefiltert werden

Aggregation Framework



Pipeline-Operatoren:

Operator	Beschreibung
\$match	Dient zum Filtern von Dokumenten innerhalb der Pipeline
\$project	Ermöglicht die Manipulation von Dokumenten durch Umbenennen, Hinzufügen und Entfernen von Feldern
\$limit	Begrenzt die Anzahl an Dokumenten, die in der Pipeline weitergereicht werden
\$skip	Überspringt die angegebene Anzahl an Dokumenten
\$group	Gruppiert Dokumente um auf diesen Berechnungen mittels Expressions durchzuführen
\$sort	Sortiert Dokumente innerhalb der Pipeline
\$unwind	Löst ein Array auf und erzeugt für jedes Element im Array ein eigenes Dokument
\$Redact	Dynamisches Filtern in Abhängigkeit vom Inhalt des Dokuments
\$out	Legt das Ergebnis der Aggregation in einer spezifizierbaren Collection ab

Aggregation Framework



Expressions

Können nur auf dem aktuell in der Pipeline verarbeiteten Dokument agieren. Sie bieten dazu zahlreiche Möglichkeiten:

- Aggregierende Funktionen (\$min, \$avg, \$sum...)
- Einfache mathem. Funktionen (\$add, \$subtract, \$multiply,...)
- Verknüpfung von Bedingungen (\$and, \$or, \$not ...)
- Vergleichen von Daten (\$eq, \$lt, \$gt,....)
- Steuerung des Datenflusses (\$cond, \$ifNull)
- Manipulationen von Strings (\$toLower, \$substr,)
- Zerlegen von Datumswerten (\$dayOfYear, \$year, \$week...)

Aggregation Framework



Eine Aggregation starten:

Die Operationen des Aggregation Framework sind mittels der Methode „aggregate()“ aufzurufen.

Dieser Befehl dient als Wrapper des Datenbankkommandos „aggregate“ und nimmt die Stufen der Aggregationspipeline entweder als Argumente oder als Array entgegen.

Somit sind folgende Aufrufe äquivalent:

„`db.collection.aggregate (<pipeline>)`“ oder

„`db.collection.aggregate([pipeline])`“

Aggregation Framework



Beispiel: Geben Sie die top 3 (nach Anzahl der Follower) deutschsprachigen Twitter User aus.

Vorgehen:

1. Aggregationframework auf der entsprechenden Collection starten
2. Einschränken auf deutschsprachige User
3. Gruppieren nach User
 1. Die meisten Follower herausfinden
4. Absteigend sortieren
5. Die ersten 3 User ausgeben

Aggregation Framework



Beispiel: Gib die top 3 (nach Followern) deutschsprachigen Twitter User aus.

Befehl:

Aggregation starten

```
„db.tweets.aggregate(  
  {$match: {„user.lang“ : „de“}},  
  {$group: {_id: {  
    name: „$user.name“},  
    follower_count: {$max: „$user.followers_count“}  
  }},  
  {$sort: {follower_count: -1}},  
  {$limit: 3}  
);
```

Operator	Beschreibung
\$match	Dient zum Filtern von Dokumenten innerhalb der Pipeline
\$project	Ermöglicht die Manipulation von Dokumenten durch Umbenennen, Hinzufügen und Entfernen von Feldern
\$limit	Begrenzt die Anzahl an Dokumenten, die in der Pipeline weitergereicht werden
\$skip	Überspringt die angegebene Anzahl an Dokumenten
\$group	Gruppiert Dokumente um auf diesen Berechnungen mittels Expressions durchzuführen
\$sort	Sortiert Dokumente innerhalb der Pipeline
\$unwind	Löst ein Array auf und erzeugt für jedes Element im Array ein eigenes Dokument
\$Redact	Dynamisches Filtern in Abhängigkeit vom Inhalt des Dokuments
\$out	Legt das Ergebnis der Aggregation in einer spezifizierbaren Collection ab

Aggregation Framework



Beispiel:

Gib die top 3 (nach Followern) deutschsprachigen Twitter User aus.

Befehl:

```
„db.tweets.aggregate(  
  {$match : {„user.lang“ : „de“}},  
  {$group : {_id : {  
    name : „$user.name“,  
    follower_count : {$max: „$user.followers_count“}  
  }},  
  {$sort : {follower_count : -1}},  
  {$limit: 3}  
);
```

1. Schritt: deutschsprachige Dokumente herausfiltern, welche als Eingabe für den nächsten Schritt dienen

2. Schritt: nach Name gruppieren und die max. Anzahl an Followern selektieren. Umbenennung von „user.folloers_count“ in „follower_count“

3. & 4. Schritt: mittels „\$sort“ nach dem neuen Feldnamen „follower_count“ absteigend sortieren. Das Ergebnis wird auf 3 Ergebnisdokumente limitiert.

Aggregation Framework



Beispiel:

Gib die top 3 (nach Followern) deutschsprachigen Twitter User aus.

Befehl:

```
„db.tweets.aggregate(  
  {$match : {„user.lang“ : „de“}},  
  {$group : { _id : {  
    name : „$user.name“},  
    follower_count : {$max: „$user.followers_count“}  
  }},  
  {$sort : {follower_count : -1}},  
  {$limit: 3}  
);
```

```
  "place" : null,  
  "user" : {  
    "friends_count" : 11933,  
    "profile_sidebar_fill_color" : "efe6ce",  
    "location" : "NYC",  
    "verified" : true,  
    "follow_request_sent" : null,  
    "favourites_count" : 52,  
    "profile_sidebar_border_color" : "e0d0b1",  
    "profile_image_url" : "http://a1.twimg.com/pr  
    "geo_enabled" : true,  
    "created_at" : "Fri May 04 15:32:48 +0000 200  
    "description" : "Wine guy, host of Wine Libra  
    "time_zone" : "Eastern Time (US & Canada)",  
    "url" : "http://garyvaynerchuk.com/post/10730  
    "screen_name" : "garyvee",  
    "notifications" : null,  
    "profile_background_color" : "2a3d37",  
    "listed_count" : 10140,  
    "lang" : "en",  
    "profile_background_image_url" : "http://a1.t  
    "statuses_count" : 27072,  
    "following" : null,  
    "profile_text_color" : "996633",  
    "protected" : false,  
    "show_all_inline_media" : false,  
    "profile_background_tile" : false,  
    "name" : "Gary Vaynerchuk",  
    "contributors_enabled" : false,  
    "profile_link_color" : "268ccd",  
    "followers_count" : 854199,  
    "id" : 5768872,  
    "profile_use_background_image" : true,  
    "utc_offset" : -18000  
  },  
  "favorited" : false,  
  "in_reply_to_user_id" : 69380109,  
  "id" : NumberLong("22820297900")  
}
```

Aggregation Framework



```
> db.tweets.aggregate(  
...     {$match : {"user.lang" : "de"}},  
...     {$group : {_id:{  
...         name : "$user.name"},  
...         follower_count: {$max:"$user.followers_count"}  
...     }},  
...     {$sort: {follower_count : -1}},  
...     {$limit: 3}  
... );  
{ "_id" : { "name" : "Tom Schraud" }, "follower_count" : 6135 }  
{ "_id" : { "name" : "Rene Nachtnebel 🌫️" }, "follower_count" : 5776 }  
{ "_id" : { "name" : "safer-print" }, "follower_count" : 4780 }  
>
```

Aggregation Framework



Dokumente filtern: \$match

- Jene Dokumente, die dem Filterkriterium entsprechen werden weiter gegeben, alle anderen herausgenommen.
- Wir hatten diesen Operator bereits in unserem vorangegangenen Beispiel verwendet.
- Dass dieser an erster Stelle der Pipeline stand war kein Zufall.... Warum?
- Dadurch, dass bereits am Anfang der Pipeline eine Teilmenge definiert wurde, reduziert sich die Anzahl der Dokumente für jede weitere Operation innerhalb der Pipeline → **PERFORMANCE**

```
db.tweets.aggregate(  
  {$match : {„user.lang“ : „de“}},  
  {$group : {_id : {  
    name : „$user.name“},  
    follower_count : {$max: „$user.followers_count“}  
  }},  
  {$sort : {follower_count : -1}},  
  {$limit: 3}  
);
```

Aggregation Framework



Dokumente filtern: \$match

Beispiel:

- Es sollen jene Benutzer gefunden werden, deren Followerzahl im einstelligen Bereich liegt.

Befehl:

```
„db.tweets.aggregate( {$match : {„user.followers_count“ :  
                                {$gt : 0, $lt : 10}}}).pretty();“
```

Aggregation Framework



Dokumente filtern: \$match

```
> db.tweets.aggregate({$match: {"user.followers_count": {$gte : 0, $lt :10 }}}).pretty();
```

```
place : null,  
"user" : {  
  "friends_count" : 16,  
  "profile_sidebar_fill_color" : "e0ff92",  
  "location" : "ÜT: -6.29195,107.128883",  
  "verified" : false,  
  "follow_request_sent" : null,  
  "favourites_count" : 0,  
  "profile_sidebar_border_color" : "87bc44",  
  "profile_image_url" : "http://a3.twimg.com/profile_images/1114904603/IMG00085-20100828-0755_normal.jpg",  
  "geo_enabled" : true,  
  "created_at" : "Mon Aug 02 08:17:09 +0000 2010",  
  "description" : "Bandung,west java",  
  "time_zone" : null,  
  "url" : "http://www.facebook.com/yogi.sugiana",  
  "screen_name" : "gigiyogi",  
  "notifications" : null,  
  "profile_background_color" : "9ae4e8",  
  "listed_count" : 0,  
  "lang" : "en",  
  "profile_background_image_url" : "http://s.twimg.com/a/1283397887/images/themes/theme1/bg.png",  
  "statuses_count" : 83,  
  "following" : null,  
  "profile_text_color" : "000000",  
  "protected" : false,  
  "show_all_inline_media" : false,  
  "profile_background_tile" : false,  
  "name" : "yogi sugiana",  
  "contributors_enabled" : false,  
  "profile_link_color" : "0000ff",  
  "followers_count" : 0,  
  "id" : 13753500,  
  "profile_use_background_image" : true,  
  "utc_offset" : null
```

Aggregation Framework



Dokumente manipulieren: \$project

Mit \$project kann man Dokumenten Felder hinzufügen, entfernen oder umbenennen.

Syntax: „`db.collection.aggregate ({$project : fieldname : (0/1)});`“

Beispiel: „`db.tweets.aggregate ({$project : {text : 1}});`“

Was wird Ihrer Meinung nach das Ergebnis dieses Befehls sein?

Aggregation Framework



Verwendungszweck von \$project

- Hinzufügen von Feldern, die mittels Expressions berechnet wurden
- Was ist das Resultat des folgenden Befehls?
 - Befehl: „`db.tweets.aggregate(`
 `{ $project : { _id: 0,`
 `inhalt_des_tweets : „$text“,`
 `anzahl_freunde : { $add: ([„$user.friends_count“, 10]) } } }`“

```
... { $project : { _id : 0,  
...   inhalt_des_tweets : "$text",  
...   anzahl_freunde : { $add: (["$user.friends_count", 10]) } } };
```

```
{ "inhalt_des_tweets" : "I can't wait for #BoardwalkEmpire", "anzahl_freunde" : 214 }
```

Aggregation Framework



Verwendungszweck von \$project

- Hinzufügen von Subdokumenten
- Was bewirkt der folgende Befehl?

- Befehl: „`db.tweets.aggregate(
 { $project : { _id : 0,
 inhalt_des_tweets : „$text“,
 benutzer: {
 name: „$user.name“,
 anzahl_freunde :
 {$add: ([„$user.friends_count“, 10])}
 }
 }
});“`

Aggregation Framework



Verwendungszweck von \$project

- Hinzufügen von Subdokumenten

```
> db.tweets.aggregate(
... {$project : {_id: 0,
...           inhalt_des_tweets : "$text",
...           benutzer : {
...             name: "$user.name",
...             anzahl_freunde: {$add: (["user.friends_count", 10])}}}});
{
  "inhalt_des_tweets" : "I can't wait for #BoardwalkEmpire",
  "benutzer" : {
    "name" : "Travis Siebrass",
    "anzahl_freunde" : 214
  }
}
```

Aggregation Framework



Expressions

- Wie bereits erwähnt, kann man innerhalb einer Operation der Aggregationspipeline Manipulationen an Dokumenten vornehmen.
- Wir haben einige davon bereits kennen gelernt. (\$sum, \$gte...)
- Expressions können in folgende Gruppen eingeteilt werden:
 - Expressions für \$group
 - Arithmetische Expressions
 - Boolesche Expressions
 - Vergleichende Expressions
 - Konditionale Expressions
 - Expressions auf Strings
 - Expressions auf Datumswerten
 - Expressions auf Arrays

Genauere Erörterung aller Expressions findet man auf der MongoDB homepage:
[MongoDB.com](https://www.mongodb.com)

