

La agregación pipeline

La agregación pipeline o tuberías de agregación se basa en someter una colección a un conjunto de operaciones o etapas, que se irán convirtiendo y transformando en un conjunto de documentos pertenecientes a la colección hasta obtener un conjunto de documentos con el resultado deseado. Las etapas son las siguientes:

Etapas	Descripción	Multip.
<i>\$project</i>	Cambia la forma del documento. La proyección permite modificar la representación de los datos, por lo que en general se emplea para darles una forma con la que resulte más cómodo trabajar.	1:1
<i>\$match</i>	Filtra los resultados. Permite filtrar los documentos para que en el resultado de la etapa solo estén aquellos que cumplen ciertos criterios. Se puede filtrar antes o después de agregar los resultados, en función del orden en que se define esta etapa.	n:1
<i>\$group</i>	Agrupación. Agrupa distintos documentos según compartan el valor de uno o varios de sus atributos, y realizar operaciones de agregación sobre los elementos de cada uno de los grupos. Se utiliza en las funciones sum, max, min, avg, etc.	n:1
<i>\$sort</i>	Ordenación de documentos.	1:1
<i>\$skip</i>	Salta N elementos.	n:1
<i>\$limit</i>	Elige N elementos para el resultado.	n:1
<i>\$unwind</i>	Normaliza arrays.	1:n
<i>\$out</i>	Envía el resultado a una salida, se almacena en la BD como una nueva colección.	1:1

La multiplicidad se refiere a cuántos documentos obtenemos como resultado después de aplicar la etapa, por ejemplo, 1:1 se aplica a un documento y se obtiene uno, n:1 se aplica a n documentos y se obtiene 1, y así sucesivamente.

El formato es el siguiente:

db.mi_coleccion.aggregate

```
([
    { $etapa1: { ... } }, { $etapa2: { ... } }, .....
])
```

Condiciones de agregación:

Nombre	Descripción
<i>\$cond</i>	Evalúa una expresión y dependiendo del resultado, devuelve el valor de una de las dos expresiones. Recibe tres expresiones en una lista ordenada o tres parámetros con nombre. { \$cond: [<expresion_booleana>, <caso_true>, <caso_false>] }
<i>\$ifNull</i>	Devuelve o bien el resultado no nulo de la primera expresión o el resultado de la segunda expresión si la primera da como resultado nulo. Acepta dos expresiones como argumentos. El resultado de la segunda expresión puede ser nulo. { \$ifNull: [<expresion>, <expresionsiesnull>] }

Las funciones para arrays son entre otras:

Nombre	Descripción
<i>\$arrayElemAt</i>	Devuelve el elemento especificado en el índice
<i>\$concatArrays</i>	Devuelve un array concatenado en una cadena
<i>\$filter</i>	Selecciona elementos de un array y devuelve otro array con esos elementos
<i>\$isArray</i>	Determina si un operando es array o no. Devuelve true o false.
<i>\$size</i>	Devuelve el numero de elementos del array.
<i>\$slice</i>	Devuelve un subset de elementos del array, es especifica en número.

Relaciones entre documentos MongoDB

Los métodos para referenciar son:

- Referencias manuales.

En las que se guarda el campo `_id` de un documento como referencia en otro documento. Similar al concepto de clave ajena en el modelo relacional.

```
db.emple.insertOne({_id:'emp1',nombre:"Juan", salario:1000,
fechaalta:"10/10/1999"})
db.emple.insertOne({_id:'emp2',nombre:"Alicia", salario:1400,
fechaalta:"07/08/2000", oficio: "Profesora"})
db.emple.insertOne({_id:'emp3',nombre:"María Jesús", salario:1500,
fechaalta: "05/01/2005", oficio: "Analista", comisión:100})
db.emple.insertOne({_id:'emp4',nombre:"Alberto", salario:1100,
fechaalta:"15/11/2001"})

db.depart.insertOne({_id:'dep1',nombre:"Informática",loc:'Madrid',
emple:['emp1', 'emp2']})
db.depart.insertOne({_id:'dep2',nombre:"Gestión",loc:'Talavera',
emple:['emp3', 'emp4' ]})
```

- Mostrar los empleados que pertenecen al departamento1

```
departrabajo=db.depart.findOne({_id:'dep1'})
```

- Recuperar los empleados cuyo _id se encuentre enlazado a este departamento

```
emplesdep = db.emple.find({_id: {$in : departrabajo.emple}})
```

- Idem anterior, pero devolviendo el array de documentos

```
emplesdep = db.emple.find({_id: {$in : departrabajo.emple}}).toArray()
```

- Devolver los empleados del departamento dep2 que tienen el salario mayor de 1400.

```
emplesdep = db.emple.find({_id: {$in : departrabajo.emple}, salario: {$gt:1400}}).toArray()
```

- DBRefs

Son referencias en un documento a otro utilizando el valor del campo _id del primer documento, el nombre de la colección y opcionalmente el nombre de la DB. Ver transparencias 80-85.