

PROYECTO VENTAS



Nombre: José Miguel

Apellidos: Rodríguez Muñoz

Curso: 1º Asir A

FECHA: 06/01/2022

Explicación de cada aggregate

```
db.ventas.aggregate(  
  [  
    {$limit:4},{$skip:2}  
  ]  
) .pretty()
```

En este aggregate se quiere saber cuales son los productos vendidos a Decathlon, cuya categoría pertenece a la de calistenia.

En primer lugar tenemos un \$limit que limita el número de documentos se pasan a la siguiente etapa en la tubería, y por último tenemos un \$skip para que omita la cantidad especificada de documentos, así solo muestre las _id:3 e _id:4 que son los pertenecientes a dicha categoría.

```
db.ventas.aggregate(  
  [  
    {  
      $group:  
      {  
        _id: {"empresa":"$cliente"},  
        precioMaximo: {$max:"$precio_de_venta"},  
        precioMinimo: {$min:"$precio_de_venta"},  
      }  
    }  
  ]  
) .pretty()
```

Este aggregate es muy simple ya que su función es encontrar el precio máximo y mínimo vendido a cada empresa.

Primero tenemos un \$group que agrupa los documentos de entrada por la _id expresión especificada, en este caso al campo cliente se ha renombrado como empresa, después tenemos un \$max que encontrará el precio máximo de cada cliente y un \$min para encontrar el mínimo.

```

db.páginaweb.aggregate(
  [
    {
      $lookup:
      {
        from: "opiniondeusuario",
        localField: "Apartado",
        foreignField: "Subapartado",
        as: "valoraciones"
      }
    }
  ]
).pretty()

```

Este aggregate es algo más complejo ya que une dos colecciones ,haciendo que una de ellas se convierta en algo parecido a una subcategoría.

Al principio nos encontramos con un \$lookup que realiza una combinación externa izquierda a una colección no fragmentada en la misma base de datos para filtrar los documentos de la colección "unida" para su procesamiento.A continuación ,tenemos el documento que toma \$lookup con estos campos;

-from,especifica la colección en la misma base de datos para realizar la unión(opinión de usuario).

-localField,especifica el campo desde la entrada de documentos hasta la \$lookup etapa(Apartado).

-foreignField,especifica el campo de los documentos de la from colección(Subapartado).

-as,especifica el nombre del nuevo campo de matriz para agregar a los documentos de entrada(valoraciones).

```

db.ventas.aggregate(
  [
    { $match: { $expr: { $gt: [{ $month: "$fecha_venta" }, 07] } } },
    {
      $group:
      {
        _id: {mes:{ $month: "$fecha_venta" },
          venta_total: { $sum: { $multiply: ["$precio_de_venta",
"$unidades"] } } } }
      }
    }
  ]
).pretty()

```

El fin de este aggregate es saber cuales son las ventas totales,a partir del mes de Agosto hacia delante.

Se usa \$match que filtra los documentos para pasar solo los documentos que coinciden con las condiciones especificadas a la siguiente etapa de canalización.En este caso que la expresión la fecha sea mayor al mes 7.

Luego tenemos un \$group que agrupa los documentos de entrada por la _id expresión especificada,en este caso en la id tenemos que cuando nos salga los meses tenga como nombre "mes"para no obtener a secas el número y la venta.

Después,tenemos un \$sum que calcula y devuelve la suma colectiva de valores numéricos,pero antes se produce el \$multiply que multiplica los números del precio de la venta y las unidades.

```
db.ventas.aggregate(  
  [  
    { $match : { producto : "fondos" } },  
    {  
      $project:{ descuentodelproducto: { $round:[{$multiply:  
[{$divide:["$Descuento","$precio_de_venta"]},100]},0]}}  
    }  
  ]  
).pretty()
```

El objetivo de este aggregate es calcular el descuento de un producto en específico, en este caso unos fondos.

Primero se usa un \$match que filtra los documentos para pasar solo los documentos que coinciden con las condiciones especificadas a la siguiente etapa de canalización, en este caso queremos que nos filtre por el producto de fondos.

Más adelante, tenemos un \$project para pasar los documentos con los campos solicitados a la siguiente etapa del proceso. Después tenemos un \$round que redondea un número a un entero, dado que la cuenta del descuento sale en decimal. Luego tenemos un \$multiply que multiplica el resultado de la división por 100. Y finalmente tenemos un \$divide que divide el descuento entre el precio de venta.

```

db.ventas.aggregate(
  [
    {
      $project:
        {
          id:"$producto",
          producto:1,
          beneficios: {$subtract:[{ $multiply: [
"$precio_de_venta", "$unidades" ]},{ $multiply: [ "$precio_de_coste",
"$unidades" ]}]}}
        }
    ]
  ).pretty()

```

En este aggregate queremos mostrar los beneficios generados por cada compra que haya hecho el cliente.

En primer lugar ,tenemos un \$project que pasa los documentos con los campos solicitados a la siguiente etapa del proceso,en este caso con “producto”.Luego, nos encontramos con \$subtract que restará el resultado dado por los dos \$multiply en los cuales se multiplican las unidades por el precio de venta y de coste.

```

db.ventas.aggregate(
  [
    {
      $project: { tiempodellegadaAvg: { $avg: "$tiempodellegada"}
    }
  ]
  ).pretty()

```

Este aggregate es bastante simple ,puesto su función es calcular el tiempo medio de llegada.

Primero tenemos \$project el cual ya se ha explicado anteriormente su función ,y por último un \$avg que devuelve el valor medio del tiempo de llegada.

```

db.ventas.aggregate(
  [
    {
      $group:
      {
        _id: "$producto",
        venta_total: { $sum: { $multiply: ["$precio_de_venta",
"$unidades"] } }
      }
    }
  ]
).pretty()

```

En este aggregate se calcula el importe total de la venta de cada producto.

Se usa un \$group que agrupa los documentos de entrada por la `_id` expresión especificada, en este caso “producto”. Después, tenemos un \$sum que calcula y devuelve la suma colectiva de valores numéricos, pero antes se produce el \$multiply que multiplica los números del precio de la venta y las unidades. Dándonos así la venta total de cada producto.

```

db.ventas.aggregate(
  [
    { $project: { _id: { mes: { $month: "$fecha_venta"
}, empresa: "$cliente",
mejores: { $gte: [{ $round: [{ $multiply: [{
$divide: [{ $subtract: ["$precio_de_venta", "$precio_de_coste"] },
"$precio_de_venta"] }, 100] }], 50] } } }
    }
  ]
).pretty()

```

Este aggregate es muy complejo siendo el mayor tiempo me ha llevado, enseña cuales son los mejores y peores meses de venta con cada empresa.

En primer lugar tenemos un \$project ya dicho su función anteriormente, en este caso he filtrado la id con los meses de

las fechas llamándolo “mes”, luego se ha renombrado el cliente por “empresa”, después, se ha implementado un \$gte porque los meses que se consideran mejores en venta tienen que igualar o sobrepasar el %50, un \$round para que aproxime el número decimal, luego un \$multiply para multiplicar el resultado dado de la división y de la resta por 100, en penúltimo lugar un \$divide que divide el resultado de la resta entre el precio de venta, y por último un \$subtract que resta el precio de venta con el de coste.