

AggregateTec (ProyectoFinal)



mongoDB

Miguel Reyes García
IES PUNTA DEL VERDE
1º ASIR

¿De que trata mi proyecto?

Sencillamente se trata de la gestión sobretodo numérica de dos tiendas de tecnología tratando sus productos. Manejo los precios de producción y venta además de que tienda vende el producto y el vendedor, comprador y sus datos necesarios para la compra.

En las consultas realizadas con el gestor de base de datos mongodb tratamos de sacar varias cuestiones importantes a la hora de vender dichos productos.

La carpeta preparada con todo contiene el readme con la introducción básica a todo, la carpeta datos donde hay un .pdf con la muestra de uso de import y export de datos en mongo tanto local como en la nube o mongo atlas y una carpeta con los export de local y la nube. La carpeta Doc contiene el manual de uso de lo comentado Mongo Atlas y este mismo pdf. SRC trae el contenido más importante contando con 4 .js, la base son los Inserts, después jugamos con los datos en las consultas, con out separamos productos y comprador-vendedor y después unimos con lookup, por último tenemos operadores nuevos investigados del manual de mongo u otras fuentes de interés.

Como ya he dicho me he apoyado para hacer todo tanto del manual de mongo como de páginas de internet que tratan el mismo tema además de los ejercicios hechos en clase y las explicaciones recopiladas de cada operador.

----- → **Siguiente página explicación de consultas**

Explicación de todas las consultas

1º

```
db.tecnologia.aggregate([
  {
    $unwind: {
      path: "$productos"
    }
  },
  { $match: { $expr: { $gte: [{ $year: "$fechaVenta" }, 2020] } } },
  {
    $group: {
      _id: { $year: "$fechaVenta" },
      Unidades: { $sum: "$productos.unidadesVendidas" },
    },
  },
  {
    $project: {
      Año: "$_id",
      _id: 0,
      Unidades: "$Unidades",
      PorcentajeSobreTotal: { $multiply: [{ $divide: ["$Unidades", 61] }, 100] }
    }
  }
])
).pretty()
```

1º Unwind “desmenuza” el campo productos haciéndolos individuales (Lo he tenido que usar en casi todas las consultas)

2º Match hace que todas búsquedas tengan una fecha mayor o igual a 2020

3º Group agrupa por fecha y por las unidades vendidas sumadas con sum del año

4º proyectamos en vez de id año y quitamos id, incluimos las unidades y calculamos el porcentaje sobre el total de unidades vendidas esos 3 años de cada año

2º

```
db.tecnologia.aggregate(  
  [  
    {  
      $unwind: {  
        path: "$productos"  
      }  
    },  
    {  
      $match: {empresaVendedora:"SevillaTec SL"}  
    },  
    {  
      $group:  
      {  
        _id: "$vendedor" ,  
        TotalDeVentas: { $sum: { $multiply: ["$productos.precioVenta", "$productos.unidadesVendidas"] } }  
      }  
    },  
    {  
      $project: {  
        Vendedor: "$_id",  
        _id: 0,  
        totalventas: "$TotalDeVentas",  
        IVA: { $multiply: ["$TotalDeVentas", 0.21] },  
        TotalDeVentasConIVA: { $multiply: ["$TotalDeVentas", 1.21] },  
        Redondeo: { $round: [{ $multiply: ["$TotalDeVentas", 1.21] }, 0] }  
      }  
    }  
  ]  
) .pretty()
```

1º El unwind de todas las consultas

2º match filtra por la tienda de tecnología en Sevilla

3º agrupamos por vendedor y calculamos su total de venta multiplicando precio de venta por unidades vendidas y sumamos todo lo que ha vendido en total

4º Proyectamos el vendedor quitando el id, el total de ventas y empezamos a calcular el total + el IVA, sacando el IVA sobre el total y después la suma de los dos además de redondear el resultado sin ningún decimal

3º

```
db.tecnologia.aggregate([
  {
    $unwind: {
      path: "$productos",
    }
  },
  {
    $match: { "comprador.empresa": false }
  },
  {
    $group: {
      _id: "$comprador.nombre",
      GastoAVG: { $avg: { $multiply: [ "$productos.precioVenta", "$productos.unidadesVendidas" ] } },
      UnidadesAVG: { $avg: "$productos.unidadesVendidas" }
    }
  },
  {
    $project: {
      Comprador: "$_id",
      _id: 0,
      GastoAVG: "$GastoAVG",
      UnidadesAVG: "$UnidadesAVG"
    }
  }
])
).pretty()
```

1º Unwind de productos.

2º Filtramos por particulares

3º Agrupamos por nombre del comprador y calculamos el AVG de gasto del cliente multiplicando precios de venta y unidades compradas después avg hace el resto calculando su promedio, también calculamos el avg de compras del cliente

4º Proyectamos al comprador, su gasto promedio y las compras promedio

4º

```
db.tecnologia.aggregate([
  {
    $unwind: {
      path: "$productos"
    }
  },
  {
    $group: {
      _id: "$vendedor" ,
      MayorVenta: { $max: { $multiply: ["$productos.precioVenta", "$productos.unidadesVendidas"] } }
    }
  },
  {
    $project: {
      vendedor: "$_id",
      _id: 0,
      MayorVenta: "$MayorVenta"
    }
  },
  {
    $sort: {
      MayorVenta: -1
    }
  }
]).pretty()
```

1º Unwind a productos

2º Agrupamos por vendedor y buscamos la mayor venta de cada vendedor con el operador max

3º Proyectamos al vendedor y las venta mas alta

5º He hecho otro igual pero usando el operador contrario MIN y filtrando por la tienda en Málaga

```
db.tecnologia.aggregate([
  {
    $unwind: {
      path: "$productos"
    }
  },
  {
    $match: {empresaVendedora: "MalagaTec SL"}
  },
  {
    $group: {
      _id: "$vendedor" ,
      MayorVenta: { $min: { $multiply: ["$productos.precioVenta", "$productos.unidadesVendidas"] } }
    }
  },
  {
    $project: {
      vendedor: "$_id",
      _id: 0,
      MenorVenta: "$MayorVenta"
    }
  },
  {
    $sort: {
      MenorVenta: 1
    }
  }
]).pretty()
```

6º

```
db.tecnologia.aggregate([
  {
    $unwind: {
      path: "$productos"
    }
  },
  {
    $group: {
      _id: { $year: "$fechaVenta" },
      PrecioProduccion: { $sum: { $multiply: ["$productos.precioProduccion", "$productos.unidadesVendidas"] } },
      Venta: { $sum: { $multiply: ["$productos.precioVenta", "$productos.unidadesVendidas"] } }
    }
  },
  {
    $project: {
      Año: "$_id",
      _id: 0,
      Beneficio: { $round: [{$subtract: ["$Venta", "$PrecioProduccion"]}, 2] }
    }
  },
  {
    $sort: {
      Año: -1
    }
  },
  {
    $limit: 3
  }
]).pretty()
```

1º Unwind a productos

2º Agrupamos el año de la fecha de venta, calculamos el precio de producción total calculando precioproduccion con las unidades vendidas y sumamos todas las del año, calculamos precio de venta

3º Luego proyectamos el año y calculamos el beneficio restando la venta menos el precio de producción y redondeamos el resultado con dos decimales.

4º Ordenamos de mayor a menor el año.

7º

```
db.tecnologia.aggregate([
  {
    $match: {
      "comprador.metodoPago": "Visa"
    }
  },
  {
    $count: "Visa como metodo de pago"
  }
])
```

Es una consulta sencilla donde filtramos el metodo de pago Visa y contamos cuantos compradores lo usan.

8º

```
db.tecnologia.aggregate([
  {
    $unwind: {
      path: "$productos"
    }
  },
  {
    $match: { "comprador.nombre": "Mariscos Recio SA" }
  },
  {
    $group:
    {
      _id: { Día: { $dayOfMonth: "$fechaVenta"}, mes: { $month: "$fechaVenta" }, año: { $year: "$fechaVenta" } },
      TotalaPagar: { $sum: { $multiply: [ "$productos.unidadesVendidas", "$productos.precioVenta" ] } },
      NúmeroDePedidos: { $sum: 1 }
    }
  }
])
).pretty()
```

1º Unwind a productos

2º Filtramos por la empresa Mariscos Recios

3º agrupamos por la fecha de su compra completa, el total a pagar de la empresa y la suma de todos sus pedidos

9º

```
db.tecnologia.aggregate(  
  [  
    { $sort: { productos: -1 } },  
    {  
      $group:  
      {  
        _id: "$vendedor" ,  
        ListadoDeCompra: { $push: { producto: "$productos.nombre", precios: "$productos.precioVenta" } }  
      }  
    }  
  ]  
)
```

1º Ordena del primer pedido a ultimo

2º Agrupamos por vendedor y creamos un listado de compras usando push que crea un array con los productos que ha vendido por pedido y abajo otro array con sus precios de venta al público por cada pedido

-----> **Siguiente página aportación personal**

Aportación Personal

1º

```
db.tecnologia.aggregate([
  {
    $unwind: {
      path: "$productos"
    }
  },
  {
    $group: {
      _id: { $year: "$fechaVenta" },
      PrecioProduccion: { $sum: { $multiply: ["$productos.precioProduccion", "$productos.unidadesVendidas"] } },
      Venta: { $sum: { $multiply: ["$productos.precioVenta", "$productos.unidadesVendidas"] } }
    }
  },
  {
    $project: {
      Año: "$_id",
      _id: 0,
      Beneficio: { $round: [{$subtract: ["$Venta", "$PrecioProduccion"]}, 2] }
    }
  },
  {
    $sort: {
      Año: -1
    }
  },
  {
    $limit: 3
  }
]).pretty()
```

Es la misma consulta anterior solo que incluyendo el operador \$limit que limita los documentos en este caso a 3.

2º

```
db.tecnologia.aggregate([
  {
    $sort: {
      "comprador.nombre": 1
    }
  },
  {
    $group: {
      _id: {Empresa: { $eq: [ "$comprador.empresa", true] }},
      NumeroDeEmpresasoParti: {$sum: 1},
      PrimeraEmpresaoParti: {$first: "$comprador.nombre"},
      UltimaEmpresaoParti: {$last: "$comprador.nombre"}
    }
  }
]).pretty()
```

1º Primero ordenamos los nombres de compradores alfabéticamente

2º Mediante eq hacemos que a la hora de buscar identifique lo que queremos ver como true en este caso queremos que sea una Empresa no un Particular, luego sumamos el número de empresas o particulares, después con \$first busca la primera empresa en orden alfabético y la ultima con \$last

3º

```
db.tecnologia.aggregate([
  {
    $group: {
      _id: "$empresaVendedora",
      personal: {$addToSet: "$vendedor"}
    }
  },
  {
    $project: {
      Empresa: "$_id",
      _id: 0,
      personal: "$personal"
    }
  }
]).pretty()
```

1º Agrupamos por empresa que vende el producto y con addtoSet creamos un array con toda la plantilla de vendedores de Malaga y Sevilla

4º

```
db.tecnologia.aggregate( [
  {
    $unwind: {
      path: "$productos"
    }
  },
  {
    $bucketAuto: {
      groupBy: "$productos.precioVenta",
      buckets: 5
    }
  }
] ).pretty()
```

1º Unwind a productos

2º BucketAuto crea según le digas como por ejemplo en la imagen 5 rango cogiendo el mínimo de precio de venta de los productos y el máximo y va metiendo cada producto en un segmento como por ejemplo en la imagen siguiente mete 5 productos vendidos entre 15,99 y 20 euros:

```
/*
{ "_id" : { "min" : 15.99, "max" : 20 }, "count" : 5 }
{ "_id" : { "min" : 20, "max" : 449 }, "count" : 5 }
{ "_id" : { "min" : 449, "max" : 529 }, "count" : 4 }
{ "_id" : { "min" : 529, "max" : 3251 }, "count" : 6 }
{ "_id" : { "min" : 3251, "max" : 3251 }, "count" : 1 }
*/
```

5º

```
db.tecnologia.aggregate(
[
  {
    $unwind: {
      path: "$productos"
    }
  },
  {
    $project:
    {
      "productos.nombre":1,
      _id:0,
      "productos.precioVenta":1,
      descuento:
      {
        $cond: { if: { $gte: [ "$productos.precioVenta", 1000 ] }, then: 60, else: 40 }
      },
    }
  },
  {
    $project:
    {
      "productos.nombre":1,
      _id:0,
      descuento:"$descuento",
      "productos.precioVenta":1,
      RebajaDe: { $divide: [{ $multiply: ["$productos.precioVenta", "$descuento"] }, 100] }
    }
  },
  {
    $project:
    {
      "productos.nombre":1,
      _id:0,
      descuento:"$descuento",
      "productos.precioVenta":1,
      RebajaDe: "$RebajaDe",
      PrecioFinal: {$round: [{ $subtract: ["$productos.precioVenta", "$RebajaDe"] }, 2]}
    }
  }
]
)
```

1º Unwind a productos

2º Proyectamos los nombres de productos y sus precios de venta y metemos un condicional en el que si el precio es mayor o igual a 1000 se le aplica un descuento de 60 y en caso contrario 40

3º Quiero calcular la rebaja del porcentaje al producto multiplicando precioventa x descuento entre 100

4º Le resto el resultado anterior al precio original y lo redondeo en 2 decimales

6º

```
db.tecnologia.aggregate( [
  {
    $unwind: {
      path: "$productos"
    }
  },
  {
    $project:
    {
      "productos.nombre" : 1,
      "productos.otrosDatos.valoraciones" : 1,
      "calificacion":
      {
        $switch:
        [
          {
            branches: [
              {
                case: { $gte : [ { $avg : "$productos.otrosDatos.valoraciones" }, 4.75 ] },
                then: "Excelente!"
              },
              {
                case: { $and : [ { $gte : [ { $avg : "$productos.otrosDatos.valoraciones" }, 4 ] },
                                { $lt : [ { $avg : "$productos.otrosDatos.valoraciones" }, 4.75 ] } ] },
                then: "Muy bueno!"
              },
              {
                case: { $lt : [ { $avg : "$productos.otrosDatos.valoraciones" }, 4 ] },
                then: "Regular/Malo"
              }
            ],
            default: "Ninguna Valoracion"
          }
        ]
      }
    }
  },
  {
    $project:
    {
      _id:0,
      Producto:"$productos.nombre",
      Valoraciones:"$productos.otrosDatos.valoraciones",
      calificacion: "$calificacion"
    }
  }
] )
```

1º Unwind a productos

2º proyectamos el nombre del producto y la valoracion

3º Switch tiene una estructura donde branches hace una especie de arbol con varias opciones a declarar donde por ejemplo la primera si la media de valoracion es por encima de 4.75 la calificación es excelente, la segunda contiene un and con dos requisitos ser mayor o igual a 4 y menor a 4,75 en ese caso sería muy bueno y menor a 4 regular/malo

\$Out y \$Lookup

Vamos a separar los datos de productos con los datos del vendedor y comprador y luego vamos a unirlos con Lookup

1º Vamos separar los datos de vendedores-compradores

```
db.tecnologia.aggregate(  
  [  
    {  
      $project:{  
        _id:0,  
        productos:0,  
      }  
    },  
    {  
      $out:{db:"test",coll:"vendedorcomprador"}  
    }  
  ]  
)
```

Quitamos los productos y dejamos el resto de información que es la que queremos separar, con out lo enviamos a la database test y la colección vendedorcomprador

```
id: 1  
empresaVendedora: "SevillaTec SL"  
vendedor: "Paco Pálido"  
fechaVenta: 2020-11-23T00:00:00.000+00:00  
✓ comprador: Object  
  empresa: true  
  nombre: "Mariscos Recio SA"  
  CIF: "M345980"  
  metodoPago: "Visa"
```

2º Vamos a hacer lo mismo pero con la otra parte de productos:

```

db.tecnologia.aggregate([
  {
    $unwind: {
      path: "$productos",
    },
  },
  {
    $project: {
      _id: 0,
      id: 1,
      Nombre: "$productos.nombre",
      PrecioVenta: "$productos.precioVenta",
      PrecioProduccion: "$productos.precioProduccion",
      UnidadesVendidas: "$productos.unidadesVendidas",
      OtrosDatos: "$productos.otrosDatos"
    },
  },
  {
    $out: { db: "test", coll: "productos" }
  }
])

```

Aquí haremos un unwind a los productos y proyectaremos editando los nombres que teníamos y los enviamos a la db test y a la colección productos

```

id: 1
Nombre: "Samsung Z Fold 3"
PrecioVenta: 1760.42
PrecioProduccion: 800
UnidadesVendidas: 13
OtrosDatos: Object
  pulgadas: 7.6
  colores: Array
  valoraciones: Array
  peso: 271
  garantia: true

```

3º Ahora vamos a producir la conexión entre estas dos colecciones con el siguiente aggregate:


```

db.productos.aggregate([
  {
    $lookup: {
      from: 'vendedorcomprador',
      localField: 'id',
      foreignField: 'id',
      as: 'venta'
    }
  },
  {
    $addFields: {
      TotalVenta: { $multiply: ["$PrecioVenta", "$UnidadesVendidas"] },
      TotalProduccion: { $multiply: ["$PrecioProduccion", "$UnidadesVendidas"] },
      vendedor: { $arrayElemAt: [ '$venta', 0] }
    }
  },
  {
    $project: {
      venta: 0
    }
  }
]).pretty()

```

En el que con lookup cogemos vendedorcomprador desde productos seleccionamos la conexión en este caso id de la colección local y el id de la colección externa y el campo que se creara en principio se llamará venta, con addfields que podemos añadir campos añadimos el total de la venta y total del coste de producción y la parte importante vendedor que con arrayelemat añadimos todos los datos del array venta que creamos de forma ordenada. Le quitamos el identificador venta para que vendedor sea el nombre de la array.

Un ejemplo rápido:

```

{id" : 9,
"Nombre" : "Nintendo Switch",
"PrecioVenta" : 299,
"PrecioProduccion" : 150,
"UnidadesVendidas" : 1,
"OtrosDatos" : {
  "colores" : [
    "azul",
    "rojo",
    "gris"
  ],
  "valoraciones" : [
    3.25,
    4
  ],
  "peso" : 389,
  "garantia" : true
},
"TotalVenta" : 299,
"TotalProduccion" : 150,
"vendedor" : {
  "_id" : ObjectId("6223971524edd2093dcd069b"),
  "id" : 9,
  "empresaVendedora" : "MalagaTec SL",
  "vendedor" : "Miguel Reyes",
  "fechaVenta" : ISODate("2021-01-02T00:00:00Z"),
  "comprador" : {
    "empresa" : false,
    "nombre" : "Lebron James",
    "NIF" : "67859856T",
    "metodoPago" : "Efectivo"
  }
}
}

```

Por supuesto todo este contenido está subido a mi mongo atlas:

MongoDB Compass - cluster0.1onet.mongodb.net/test

Connect View Help

Local

1 DBS3 COLLECTIONS

☆ FAVORITE

HOSTS

cluster0-shard-00-00.1one...
cluster0-shard-00-01.1one...
cluster0-shard-00-02.1one...

CLUSTER

Replica Set (atlas-l0pz2g-s...
3 Nodes

EDITION

MongoDB 5.0.6 Enterprise

Q Filter your data

test

productos

tecnologia

vendedorcomprador

Collections

Create collectionView

Sort byCollection Name

products

Storage size: 20.48 kB	Documents: 21	Avg. document size: 268.00 B	Indexes: 1	Total index size: 20.48 kB
---------------------------	------------------	---------------------------------	---------------	-------------------------------

tecnologia

Storage size: 20.48 kB	Documents: 10	Avg. document size: 737.00 B	Indexes: 1	Total index size: 20.48 kB
---------------------------	------------------	---------------------------------	---------------	-------------------------------

vendedorcomprador

Storage size: 20.48 kB	Documents: 10	Avg. document size: 211.00 B	Indexes: 1	Total index size: 20.48 kB
---------------------------	------------------	---------------------------------	---------------	-------------------------------