

Practica Mongo Aggregation Framework – BDIA

Alumno: Juan Pablo Schamun

1. Realizar una consulta que devuelva la siguiente información: Región y cantidad total de productos vendidos a clientes de esa Región.

```
db.facturas.aggregate([
  {$unwind: { path: "$item" }},
  {$group:
  {
    _id: "$cliente.region",
    cantidad: { $sum: "$item.cantidad" }
  }
  },
  {
    $sort: { "cantidad": -1 }
  }
])
```

```
< {
  _id: 'NEA',
  cantidad: 420
}
{
  _id: 'CABA',
  cantidad: 282
}
{
  _id: 'CENTRO',
  cantidad: 180
}
{
  _id: 'NOA',
  cantidad: 14
}
```

2. Basado en la consulta del punto 1, mostrar sólo la región que tenga el menor ingreso

```
db.facturas.aggregate( [  
    {$unwind: { path: "$item" }},  
    {$group:  
    {  
        _id: "$cliente.region",  
        cantidad: { $sum: "$item.cantidad" },  
        ingreso: { $sum: { $multiply: ["$item.cantidad",  
"$item.precio"] } }  
    }  
    },  
    {$sort: { "ingreso": 1 }},  
    {$limit:1}  
]  
)
```

```
{  
  _id: 'NOA',  
  cantidad: 14,  
  ingreso: 9800  
}
```

3. Basado en la consulta del punto 1, mostrar sólo las regiones que tengan una cantidad de productos vendidos superior a 10000

```

db.facturas.aggregate( [
    {$unwind: { path: "$item" }},
    {$group:
    {
        _id: "$cliente.region",
        cantidad: { $sum: "$item.cantidad" }
    }
    },
    {$match: {"cantidad": {$gt: 10000}}},
    {$sort: { "cantidad": -1 }}
]
)

```

Sin Resultados

4. Se requiere obtener un reporte que contenga la siguiente información, nro. cuit, apellido y nombre y región y cantidad de facturas, ordenado por apellido

```

db.facturas.aggregate( [
    {$group:

```

```

{
  _id: {
    cuit:"$cliente.cuit",
    apellido:"$cliente.apellido",
    nombre:"$cliente.nombre",
    region:"$cliente.region"
  },
  cantidadFacturas: { $sum: 1 }
}

},

{$sort:{"_id.apellido":1}}
]
)

```

```

< {
  _id: {
    cuit: 2729887543,
    apellido: 'Lavagno',
    nombre: 'Soledad',
    region: 'NOA'
  },
  cantidadFacturas: 14
}
{
  _id: {
    cuit: 2740488484,
    apellido: 'Malinez',
    nombre: 'Marina',
    region: 'CENTRO'
  },
  cantidadFacturas: 15
}
{
  _id: {
    cuit: 2029889382,
    apellido: 'Manoni',
    nombre: 'Juan Manuel',
    region: 'NEA'
  },
  cantidadFacturas: 42
}

```

5. Basados en la consulta del punto 4 informar sólo los clientes con número de CUIT mayor a 2700000000 (saco un cero a la consigna original para que traiga algo)

```
db.facturas.aggregate( [
```

```

    {$group:
    {
      _id: {
        cuit:"$cliente.cuit",
        apellido:"$cliente.apellido",
        nombre:"$cliente.nombre",
        region:"$cliente.region"
      },
      cantidadFacturas: { $sum: 1 }
    }
  },
  {$match: {"_id.cuit": {$gt: 2700000000}}},
  {$sort:{"_id.apellido":1}}
]
)

```

```

{
  _id: {
    cuit: 2729887543,
    apellido: 'Lavagno',
    nombre: 'Soledad',
    region: 'NOA'
  },
  cantidadFacturas: 14
}
{
  _id: {
    cuit: 2740488484,
    apellido: 'Malinez',
    nombre: 'Marina',
    region: 'CENTRO'
  },
  cantidadFacturas: 15
}

```

6. Basados en la consulta del punto 5 informar solamente la cantidad de clientes que cumplen con esta condición.

```
db.facturas.aggregate( [
```

```

        {$group:
    {
      _id: {
        cuit:"$cliente.cuit",
        apellido:"$cliente.apellido",
        nombre:"$cliente.nombre",
        region:"$cliente.region"
      },
      cantidadFacturas: { $sum: 1 }
    }
  },
  {$match: {"_id.cuit": {$gt: 27000000000}}},
  {$sort:{"_id.apellido":1}},
  {"$group" : {_id:null, count:{$sum:1}}}
]
)

```

```

{
  _id: null,
  count: 2
}

```

7. Se requiere realizar una consulta que devuelva la siguiente información:
*producto y cantidad de facturas en las que lo compraron, ordenado por
 cantidad de facturas descendente*

```

db.facturas.aggregate( [
  {$unwind: { path: "$item" }},

```

```

        {$group:
    {
        _id: "$item.producto",
        facturas: { $sum: 1 }
    }
    },
    {$sort: { "facturas": -1 }}
]
)

```

```

{
  _id: 'TALADRO 12mm',
  facturas: 43
}
{
  _id: 'CORREA 10mm',
  facturas: 29
}
{
  _id: 'TUERCA 5mm',
  facturas: 28
}
{
  _id: 'TUERCA 2mm',
  facturas: 28
}
{
  _id: 'SET HERRAMIENTAS',
  facturas: 28
}
{
  _id: 'CORREA 12mm',
  facturas: 15
}

```

8. Obtener la cantidad total comprada así como también los ingresos totales para cada producto.

```

db.facturas.aggregate( [
    {$unwind: { path: "$item" }},
    {$group:

```

```

{
  _id: "$item.producto",
  cantidad: { $sum: "$item.cantidad" },
  ingreso: { $sum: { $multiply: ["$item.cantidad", "$item.precio"]} }
},
{$sort: { "cantidad": -1 }}
]
)

```

```

{
  _id: 'TUERCA 5mm',
  cantidad: 350,
  ingreso: 31500
}
{
  _id: 'CORREA 10mm',
  cantidad: 198,
  ingreso: 26532
}
{
  _id: 'CORREA 12mm',
  cantidad: 165,
  ingreso: 2970
}
{
  _id: 'TUERCA 2mm',
  cantidad: 112,
  ingreso: 6720
}
{
  _id: 'TALADRO 12mm',
  cantidad: 43,
  ingreso: 21070
}
{
  _id: 'SET HERRAMIENTAS',
  cantidad: 28,
  ingreso: 19600
}

```

9. Idem el punto anterior, ordenar por ingresos en forma ascendente, saltar el 1ro y mostrar 2do y 3ro

```

db.facturas.aggregate( [
  {$unwind: { path: "$item" }},
  {$group:

```



```

{
  _id: "$item.producto",
  cantidad: { $sum: "$item.cantidad" },
  ingreso: { $sum: { $multiply: ["$item.cantidad", "$item.precio"]} }
}
},
{$sort: { "ingreso": 1 }},
{$skip:1},
{$limit:2}
]
)

```

```

{
  _id: 'TUERCA 2mm',
  cantidad: 112,
  ingreso: 6720
}
{
  _id: 'SET HERRAMIENTAS',
  cantidad: 28,
  ingreso: 19600
}

```

10. Obtener todos productos junto con un array de las personas que lo compraron. En este array deberá haber solo strings con el nombre completo de la persona. Los documentos entregados como resultado deberán tener la siguiente forma: {producto: "<nombre>", personas:["...", ...]}

```

db.facturas.aggregate( [
  {$unwind: { path: "$item" }},
  {$project: { _id:0, producto:"$item.producto", cliente:{$concat:["$cliente.nombre"," ",
"$cliente.apellido"]}}}

```

```
{ $group : { _id: "$producto", personas: { $addToSet: "$cliente" } } }  
]  
)
```

```
{  
  _id: 'CORREA 12mm',  
  personas: [  
    'Marina Malínez'  
  ]  
}  
{  
  _id: 'TUERCA 2mm',  
  personas: [  
    'Juan Manuel Manoni',  
    'Martín Zavasi'  
  ]  
}  
{  
  _id: 'TUERCA 5mm',  
  personas: [  
    'Juan Manuel Manoni'  
  ]  
}  
{  
  _id: 'SET HERRAMIENTAS',  
  personas: [  
    'Juan Manuel Manoni',  
    'Soledad Lavagno'  
  ]  
}
```