

Proyecto 2 Trimestre

Lucas Coronilla Salmerón

¿Que Pretende el proyecto?

El proyecto pretende mostrar a pequeña escala como estaría estructurada y desarrollada parte de una base de datos de un establecimiento el cual tiene que lidiar con muchos pedidos y variedades de productos, organizarlas y procesarlas.

Operadores utilizados Interesantes

\$cond: Evalúa una expresión booleana para devolver una de las dos expresiones de retorno especificadas.

Ejemplo de sintaxis: `{ $cond: [{ $gte: ["$Productos_comprados", 2] }, 0.95, 1] }`

\$\$NOW: Obtiene la fecha actual.

Ejemplo de sintaxis: `fecha_actual: "$$NOW",`

\$push: agrega un valor especificado a un array.

Ejemplo de sintaxis: `plantas_fuera_de_stock: { $push: { plantas: "$cultivo.planta" } },`

\$unwind: Deconstruye un campo array a partir de los documentos de entrada para generar un documento de salida para cada elemento. Cada documento de salida es el documento de entrada con el valor del campo array reemplazado.

Ejemplo de sintaxis:

```
{
  $unwind: "$cultivo"
},
```

\$lookup: Une 2 colecciones a través de un campo en común(clave).

Ejemplo de sintaxis:

```
{
  $lookup:
  {
    from: "cultivo",
    localField: "pedidos",
    foreignField: "id_p",
    as: "cultivo"
  },
},
```

\$group: agrupa los elementos de la colección a través de un campo `_id` en un campo único y por cada agrupacion distinta se genera un documento de salida con sus datos.

Ejemplo de sintaxis:

```
{
  $group:
  {
    _id: <expression>, // Group By Expression
    <field1>: { <accumulator1> : <expression1> },
    ...
  }
}
```

Consultas

1-

```
db.proveedores.aggregate([
  {
    $lookup:
    {
      from: "cultivo",
      localField: "id_pr",
      foreignField: "cod_proveedor",
      as: "cultivo"
    },
  },
  {
    $unwind:"$cultivo"
  },
  {
    $match:{"cultivo.stock":false}
  },
  {
    $group:{
      _id:{proveedor:"$Nombre",id:"$id_pr"},
      numero_fs:{$sum:1},
      plantas_fuera_de_stock:{$push:{plantas:"$cultivo.planta"}},
    },
  },
  {
    $project:{
      _id:1,
      numero_fs:1,
      plantas_fuera_de_stock:1,
      sin_stock:{$cond:[{$lt:["$plantas_out",2]},"no necesitamos nuevas unidades","Sin stock llamar al distribuidor"]},
    }
  }
]).pretty();
```

Activar

Esta consulta obtiene los datos de la colección cliente y la relaciona con la colección cultivo(productos) mediante el \$lookup, sacamos la información de la colección cultivo del array creado en el lookup y filtramos a través del match los productos que están en stock,

En el group establecemos como id el Nombre del proveedor y su dni, establecemos un contador con {\$sum:1} que cuenta el número de plantas fuera de stock las cuales mostramos en un array.

En el project indicamos los campos que queremos que sean visibles y añadimos una condicion de que si el contador es mayor a 2 es decir tenemos más de 2 productos fuera de stock avise que necesitamos llamar al distribuidor.

2-

```
db.clientes.aggregate([
  {
    $match : { recibo : false }
  },
  {
    $group:{
      _id:{dni:"$dni",fecha_de_envio:"$fecha_de_envio"},
      dateDifference:{sum:{multiply:[{$subtract: [ "$$NOW", "$fecha_de_envio" ] },0.0000000115741]}},
      Cliente:{push:{Nombre:"$Nombre",Apellidos:"$Apellidos"}}
    }
  },
  {
    $project:{
      "_id.dni":1,
      cliente:1,
      fecha_de_envio:"$ _id.fecha_de_envio",
      fecha_actual:"$$NOW",
      diferencia_dias:{round:["$dateDifference",0]},
      retraso:{cond:[{$gte:[{$round:["$dateDifference",0]},20]},"Retraso de envio","sin retraso"]}
    },
  },
]) .pretty();
```

Esta consulta pretende obtener el tiempo que lleva el producto en envío y de cuanto se está retrasando.

Empezamos con el match el cual filtra los clientes que aún no han recibido su producto mediante el match.

En el group filtramos mediante el dni y la fecha de envío calculamos la diferencia de tiempo restando la fecha actual y la fecha de envío transformando el tiempo de ms a días.

En el Project indicamos los campos que queramos que sean visible además de redondear los días y establecemos una condición para que nos indique de que un envío se está retrasando a partir de una duración de 20 días sin haber llegado a su destino.

3-

```
db.clientes.aggregate([
  {
    $lookup:
    {
      from: "cultivo",
      localField: "pedidos",
      foreignField: "id_p",
      as: "cultivo"
    },
  },
  { $match : { recibo : true } },
  {
    $unwind: "$cultivo"
  },
  {
    $project: {
      _id: 0,
      Cliente: { $concat: [ "$Nombre", " ", "$Apellidos" ] },
      importe: { $multiply: [ "$gramos", "$cultivo.precio" ] },
      iva_incluido: { $multiply: [ "$gramos", "$cultivo.precio", 1.21 ] },
      dni: "$dni",
      planta: "$cultivo.planta",
      genetica: "$cultivo.genetica",
      precio: "$cultivo.precio",
      fecha: { $dateToString: { format: "%Y-%m-%d", date: "$fecha_de_envio" } }
    }
  },
  {
    $group: {
      _id: { cliente: "$Cliente", dni: "$dni" },
      importe: { $sum: "$importe" },
      iva_incluido: { $sum: "$iva_incluido" },
      Productos_comprados: { $push: { planta: "$planta", genetica: "$genetica", precio_gramo: "$precio", fecha_de_compra: "$fecha" } }
    }
  },
  {
    $sort: { "iva_incluido": 1 }
  }
]).pretty();
```

Activa
Vea Co

Esta consulta obtiene los datos de la colección cliente y la relaciona con la colección cultivo(productos) mediante el \$lookup, sacamos la información de la colección cultivo del array creado en el lookup y filtramos a través del match los productos que los clientes ya han recibido,

En el Project concatenamos 2 campos que son nombre y apellido para tener el nombre completo, calculamos el importe bruto (sin iva) de la compra y el importe total(con iva) además declaramos los datos que vamos a querer utilizar en el group posterior y convertimos la fecha de envío a cadena para que en el push del group no aparezca ISOdate.

En el group retornamos con el \$sum el importe bruto y el importe total y creamos un array con un push que nos muestra la planta su genetica el precio que tiene cada gramo del producto y la fecha de compra.

Al final ordenamos el resultado según el importe total mediante el sort.

4-

```
}).pretty();
```

Esta consulta obtiene los datos de la colección cliente y la relaciona con la colección cultivo(productos) mediante el \$lookup, sacamos la información de la colección cultivo del array creado en el lookup y filtramos a través del match los productos que los clientes no han recibido,

En el Project concatenamos 2 campos que son nombre y apellido para tener el nombre completo, calculamos el importe bruto (sin iva) de la compra y el importe total(con iva) además declaramos los datos que vamos a querer utilizar en el group posterior.

En el group retornamos con el \$sum el importe bruto y el importe total y creamos un contador para ver cuántos productos a comprado cada cliente.

En el segundo project indicamos los datos del group que queramos que sean visibles y aplicamos un descuento del 5% a los clientes que hayan comprado más de 2 productos.