

Proyecto final

Gestión de base de datos No Sql

Índice

Proyecto final.....	1
2.Estructura de la base de datos.....	2
3.Validar Base de datos.....	3
3.1.Uso del \$out.....	4
3.2.Primer consulta.....	5
3.3. Segunda Consulta.....	6
3.4. Tercera Consulta.....	7
3.4. Cuarta Consulta.....	9
3.4. Quinta Consulta.....	11
3.4. Sexta Consulta.....	12
3.5. Séptima Consulta.....	13
.....	14
.....	14
3.5. Octava Consulta.....	15
Consulta Extra.....	16
Mongo Atlas.....	17
Mongo DataBaseTool.....	35
Mongo Export / Import.....	36

1.Introducción

He creado una base de datos donde recojo información de un total de 6 ventas con sus respectivos datos de cada venta y recogiendo información acerca del día de la venta del vendedor y del cliente.

2.Estructura de la base de datos

```

S inserts.js > ...
db.camiones.drop();
db.camiones.insertMany([
  {
    _id: 01,
    item: [
      {
        marca: "scania",
        modelo: "s",
        preciofabricacion: 270000,
        precioventa: 524000,
        unidadesvendidas: 2,
        país: "suecia",
        extras: ["cama", "portavasos", "climatizacion"],
        características: {
          automatico: true,
          caballosvapor: [440, 430, 500, 610, 750],
          peso: 7220,
          carburante: "hvo"
        }
      }
    ],
  },
  {
    marca: "scania",
    modelo: "l",
    preciofabricacion: 120000,
    precioventa: 248000,
    unidadesvendidas: 1,
    país: "suecia",
    extras: ["gps", "portavasos", "radio"],
    características: {
      automatico: false,
      caballosvapor: [420, 430, 510, 620, 710],
      peso: 6290,
      carburante: "diesel"
    }
  },
],

```

Como podemos ver tiene un campo documento que a su vez hay un array con documentos donde se recoge información en datos numericos, arrays de tipo string, el campo características es de tipo documento donde hay booleano array de numeros

3. Validar Base de datos

Para saber que la base de datos ha sido validada correctamente usaremos lo siguiente

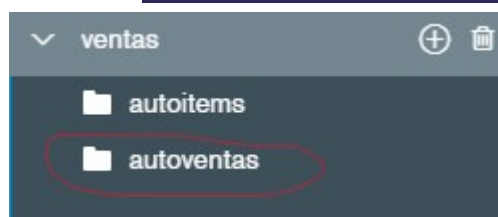
```
test> db.camiones.validate()
{
  ns: 'test.camiones',
  nInvalidDocuments: 0,
  nrecords: 6,
  nIndexes: 1,
  keysPerIndex: { _id_: 6 },
  indexDetails: { _id_: { valid: true } },
  valid: true,
  repaired: false,
  warnings: [],
  errors: [],
  extraIndexEntries: [],
  missingIndexEntries: [],
  corruptRecords: [],
  ok: 1
}
```

3.1. Uso del \$out

Usamos un project para que no nos muestre ni el identificador ni el item, creamos una base de datos ventas y creamos una colección llamada autoventas por ejemplo

```
//Separación de ventas
db.camiones.aggregate([
  {
    $project: {
      _id: 0,
      item: 0,
    }
  },
  {
    $out: {
      db: "ventas",
      coll: "autoventas"
    }
  }
])
```

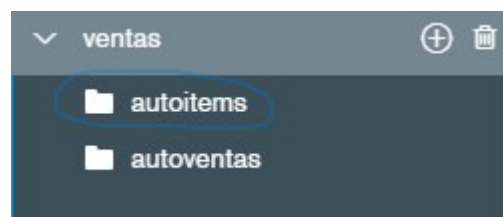
Comprobacion en MongoDB



Y luego usamos otro out para separar en autoitems dentro de la base de datos ventas

```
//Separacion de items
db.camiones.aggregate([
  {
    $project: {
      _id: 1,
      item: 1
    }
  },
  {
    $out: {
      db: "ventas",
      coll: "autoitems"
    }
  }
])
```

Comprobacion en MongoDB



3.2.Primer consulta

```
//Numero de ventas realizadas por cada vendedor
db.camiones.aggregate([
  {
    $group: {
      _id: "$vendedor",
      numerodeventas: {
        $count: {},
      }
    }
  },
  {
    $sort: {
      numerodeventas: -1
    }
  }
])
```

En esta consulta con el group agrupamos el id del vendedor y el numero de ventas y luego usamos un \$sort para ordenarlas de mayor a menor

Finalidad de la consulta: En esta consulta intentamos buscar el numero de ventas de cada vendedor en este caso añadimos el \$Sort pero no tiene mucho sentido ya que en cada venta he puesto una empresa diferente

Resultado de la consulta:

```
[
  { _id: 'Empresite', numerodeventas: 1 },
  { _id: 'Mascus', numerodeventas: 1 },
  { _id: 'Europages', numerodeventas: 1 },
  { _id: 'Autoline', numerodeventas: 1 },
  { _id: 'Truckscout', numerodeventas: 1 },
  { _id: 'Autocamiones', numerodeventas: 1 }
]
test>
```

3.3. Segunda Consulta

```
//Media de dinero generado por cada vendedor

db.camiones.aggregate([
  {
    $unwind: {
      path: "$item",
    },
  },
  {
    $group: {
      _id: "$vendedor",
      mediaponderada: {
        $avg: {
          $multiply: ["$item.preciofabricacion", "$item.precioventa"],
        },
      },
    },
  },
])
```

Con esta consulta **la finalidad** es saber la media generada por cada vendedor, para ellos usamos el \$unwind que deconstruye un campo de matriz a partir de los documentos de entrada para generar un documento para cada elemento. Luego usamos el operador \$Group en el que indicamos el id del vendedor que en este caso es el nombre de la empresa que vende los camiones y luego multiplicamos el precio de venta por el precio de fabricxacion y nos da la media de beneficios

Resultado de la consulta:

```
[
  { _id: 'Truckscout', mediaponderada: 32057500000 },
  { _id: 'Autocamiones', mediaponderada: 13608750000 },
  { _id: 'Autoline', mediaponderada: 44400000000 },
  { _id: 'Empresite', mediaponderada: 55430000000 },
  { _id: 'Mascus', mediaponderada: 25170000000 },
  { _id: 'Europages', mediaponderada: 10703187500 }
]
test>
```

3.4. Tercera Consulta

```

//Se necesita conocer los beneficios netos realizados por cada vendedor
db.camiones.aggregate([
  {
    $unwind: {
      path: "$item",
    },
  },
  {
    $project: {
      vendedor: 1,
      netIncome: {
        $subtract: [
          {
            $multiply: [
              "$item.precioventa",
              "$item.unidadesvendidas"
            ],
          },
          {
            $multiply: [
              "$item.preciofabricacion",
              "$item.unidadesvendidas"
            ],
          },
        ],
      },
    },
  },
  {
    $group: {
      _id: "$vendedor",
      total: {
        $sum: "$netIncome",
      },
    },
  },
])

```

La **finalidad** de la consulta es conocer los beneficios realizados por cada vendedor para ellos usamos el \$unwind explicado anteriormente, luego un \$project en el que especificamos con uno al vendedor para que nos aparezca, luego usamos el \$subtract para que nos haga la diferencia de la multiplicacion entre precio de venta y unidades vendidas y luego precio de fabricacion por unidades vendidas, luego lo agrupamos y hacemos una suma de los ingresos netos

Resultado de la consulta:


```
[
  { _id: 'Truckscout', total: 1115000 },
  { _id: 'Autocamiones', total: 648000 },
  { _id: 'Empresite', total: 1380000 },
  { _id: 'Mascus', total: 862000 },
  { _id: 'Europages', total: 926420 },
  { _id: 'Autoline', total: 668000 }
]
test>
```

3.4. Cuarta Consulta

```
//Soy el gerente y quiero ver las compras realizadas totales

db.camiones.aggregate([
  {$unwind: '$item'}
])
```

En esta consulta **la finalidad** es ver el numero total de ventas realizadas en nuestra base de datos con el operador \$unwind y que nos muestre todos los items

```

test> db.camiones.aggregate([
...   {$unwind: '$item'}
... ])
[
  {
    _id: 1,
    item: {
      marca: 'scania',
      modelo: 's',
      preciofabricacion: 270000,
      precioventa: 524000,
      unidadesvendidas: 2,
      país: 'suecia',
      extras: [ 'cama', 'portavasos', 'climatizacion' ],
      características: {
        automatico: true,
        caballosvapor: [ 440, 430, 500, 610, 750 ],
        peso: 7220,
        carburante: 'hvo'
      }
    },
    fechaventa: ISODate("2021-12-16T00:00:00.000Z"),
    cliente: { nombre: 'TIR Sevilla', codigo: 'B34187888' },
    vendedor: 'Empresite'
  },
  {
    _id: 1,
    item: {
      marca: 'scania',
      modelo: 'l',
      preciofabricacion: 120000,
      precioventa: 248000,
      unidadesvendidas: 1,
      país: 'suecia',
      extras: [ 'gps', 'portavasos', 'radio' ],
      características: {
        automatico: false,
        caballosvapor: [ 420, 430, 510, 620, 710 ],
        peso: 6290,
        carburante: 'diesel'
      }
    },
    fechaventa: ISODate("2021-12-16T00:00:00.000Z"),
    cliente: { nombre: 'TIR Sevilla', codigo: 'B34187888' },
    vendedor: 'Empresite'
  },
  {
    _id: 1,
    item: {
      marca: 'man',
      modelo: 'tgx',
      preciofabricacion: 160000,
      precioventa: 248000,
      unidadesvendidas: 3,
      país: 'alemania',
      extras: [ 'nevera', 'portavasos', 'radio' ],
      características: {
        automatico: false,

```

Como vemos nos muestra todas las ventas que era lo que queríamos saber

3.4. Quinta Consulta

```
db.camiones.aggregate([
  {
    $unwind:{
      path:"$item"
    }
  },
  {
    $group:{
      _id:"$cliente",
      camionescomprados:{
        $sum:"$item.unidadesvendidas"
      }
    }
  }
])
```

La finalidad de esta consulta es una lista con el numero de articulos comprados por los diferentes clientes ordenados de mayor a menor, usamos el \$unwind explicado anteriormente y luego con el \$group seleccionamos el id del cliente y luego hacemos la suma de las unidadesvendidas y luego usamos un \$sort con numero en negativo para ordenarlos de mayor a menor

Resultado de la consulta:

```
[
  {
    _id: { nombre: 'Garcia', codigo: 'C44183218' },
    camionescomprados: 10
  },
  {
    _id: { nombre: 'Primafrío', codigo: 'B34187338' },
    camionescomprados: 10
  },
  {
    _id: { nombre: 'Heineken', codigo: 'B34185688' },
    camionescomprados: 12
  },
  {
    _id: { nombre: 'Nanta', codigo: 'BC3465888' },
    camionescomprados: 11
  },
  {
    _id: { nombre: 'TIR Sevilla', codigo: 'B34187888' },
    camionescomprados: 10
  },
  {
    _id: { nombre: 'Garciden', codigo: 'B34387888' },
    camionescomprados: 10
  }
]
test>
```

3.4. Sexta Consulta

```
// Se necesita saber el maximo de unidades vendidas y el minimo de unidades vendidas
db.camiones.aggregate(
  [
    { $group:
      {
        _id: null,
        maxundsvendidas: { $max: "$item.unidadesvendidas" },
        minundsvendidas: { $min: "$item.unidadesvendidas" }
      }
    }
  ]
)
```

La **finalidad** de esta consulta es saber el recuento maximo de unidades vendidas y el minimo de unidades vendidas con el operador \$max y el \$min

Resultado de la consulta:

```
[
  {
    _id: null,
    maxundsvendidas: [ 6, 2, 3, 1 ],
    minundsvendidas: [ 2, 1, 3, 4 ]
  }
]
test>
```

3.5. Séptima Consulta

```
//se precisa de una lista de los ingresos netos por mes ordenados de mayor a menor
db.camiones.aggregate([
  {
    $unwind:{
      path:"$item"
    }
  },
  {
    $project:{
      month:{
        $month:"$fechaventa"
      },
      netIncome:{
        $subtract: [
          {
            $multiply: [
              "$item.precioventa",
              "$item.unidadesvendidas"
            ],
          },
          {
            $multiply: [
              "$item.preciofabricacion",
              "$item.unidadesvendidas"
            ],
          },
        ]
      }
    },
  },
  {
    $group:{
      _id:"$month",
      avg:{
        $avg:"$netIncome"
      }
    }
  },
])
```

```

{
  $project:{
    month:"$_id",
    ingresosnetos:{
      $round:[
        "$avg",
        2
      ]
    },
    _id:0
  }
},
{
  $sort:{
    avgNetIncome:-1
  }
}
]
)

```

La **finalidad** de esta consulta es mostrar una lista de los ingresos netos de cada mes ordenados de mayor a menor, para ello usamos un \$project en el que proyectaremos el mes, y luego un \$subtract en el que multiplicamos precio de venta y las unidades vendidas y luego otro multiply del precio de fabricacion y las unidades vendidas, a continuacion usamos el \$group en el usamos el \$avg que nos muestra los valores numéricos y luego un \$sort para ordenar de mayor a menor

Resultado de la consulta:

```

[
  { month: 7, ingresosnetos: 231605 },
  { month: 5, ingresosnetos: 162000 },
  { month: 12, ingresosnetos: 345000 },
  { month: 3, ingresosnetos: 167000 },
  { month: 2, ingresosnetos: 278750 },
  { month: 4, ingresosnetos: 215500 }
]
test>

```

3.5. Octava Consulta

```
//Dinero total que ha suministrado cada empresa de mayor a menor

db.camiones.aggregate([
  {
    $unwind:{
      path:"$item"
    }
  },
  {
    $project:{
      _id:0,
      vendedor:"$cliente.nombre",
      dinerototal:{
        $multiply: ["$item.unidadesvendidas","$item.precioventa"] }
    },
    {
      $group: {
        _id:"$vendedor",
        dinerototal:{ $sum:"$dinerototal"}
      }
    },
    {
      $sort: {
        dinerototal:-1
      }
    }
  }
])
```

La **finalidad** de esta consulta es ver el dinero que se ha gastado cada empresa en la compra de los camiones para ello usamos el \$unwind explicado anteriormente, luego con el \$project hacemos que el id sea 0 , mostramos el cliente.nombre que en este caso es el nombre de la empresa hacemos un \$multiply de unidades vendidas y el precio de venta, luego usamos un \$group y hacemos la suma total del dinero, para finalizar usamos un \$sort en -1 para ordenar de mayor a menor

Resultado de la consulta:

```
[
  { _id: 'TIR Sevilla', dinerototal: 2760000 },
  { _id: 'Nanta', dinerototal: 2630000 },
  { _id: 'Heineken', dinerototal: 2132000 },
  { _id: 'García', dinerototal: 1892000 },
  { _id: 'Garciden', dinerototal: 1651920 },
  { _id: 'Primafrío', dinerototal: 1403000 }
]
test>
```

Consulta Extra

```
//Muestra aleatoriamente 2 ventas
```

```
db.camiones.aggregate([{$sample : {size : 2}}])
```

La **finalidad** de esta consulta es encontrar a traves del operador \$sample mostrarnos aleatoriamente en este caso dos ventas

Resultado de la consulta:

```
test> db.camiones.aggregate([{$sample : {size : 2}}])
[
  {
    _id: 3,
    item: [
      {
        marca: 'renault',
        modelo: 's',
        preciofabricacion: 60000,
        precioventa: 98000,
        unidadesvendidas: 6,
        pais: 'francia',
        extras: [ 'extintor', 'alarma', 'gps' ],
        características: {
          automatico: true,
          caballosvapor: [ 510, 620, 710, 750 ],
          peso: 9300,
          carburante: 'hvo'
        }
      },
      {
        marca: 'iveco',
        modelo: 'l',
        preciofabricacion: 220000,
        precioventa: 298000,
        unidadesvendidas: 2,
        pais: 'españa',
        extras: [ 'herramientas', 'botiquín', 'gps' ],
        características: {
          automatico: false,
          caballosvapor: [ 400, 480, 550, 680, 710 ],
          peso: 2220,
          carburante: 'diesel'
        }
      },
      {
        marca: 'nissan',
        modelo: 'cabstar',
        preciofabricacion: 148000,
        precioventa: 200000,
        unidadesvendidas: 3,
        pais: 'japon',
        extras: [ 'extintor', 'nevera', 'herramientas' ],
        características: {
          automatico: true,
          caballosvapor: [ 320, 410, 480 ],
          peso: 2500,
          carburante: 'diesel'
        }
      }
    ]
  },
  {
    _id: 4,
    item: [
      {
        marca: 'toyota',
        modelo: 'hilux',
        preciofabricacion: 100000,
        precioventa: 150000,
        unidadesvendidas: 1,
        pais: 'japon',
        extras: [ 'extintor', 'gps' ],
        características: {
          automatico: true,
          caballosvapor: [ 100, 150, 200 ],
          peso: 1500,
          carburante: 'gasolina'
        }
      },
      {
        marca: 'toyota',
        modelo: 'proace',
        preciofabricacion: 100000,
        precioventa: 150000,
        unidadesvendidas: 1,
        pais: 'japon',
        extras: [ 'extintor', 'gps' ],
        características: {
          automatico: true,
          caballosvapor: [ 100, 150, 200 ],
          peso: 1500,
          carburante: 'gasolina'
        }
      }
    ]
  }
]
```

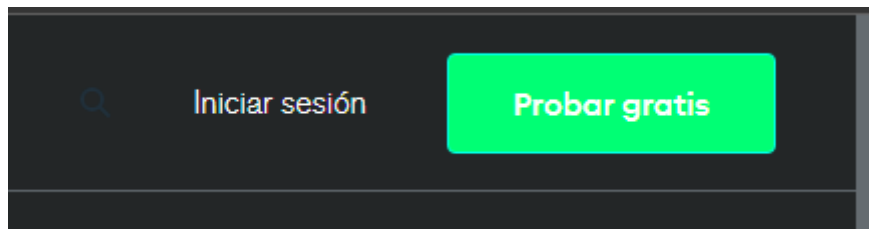

Mongo Atlas



En 1 lugar nos dirigiremos al enlace de Mongo Atlas:

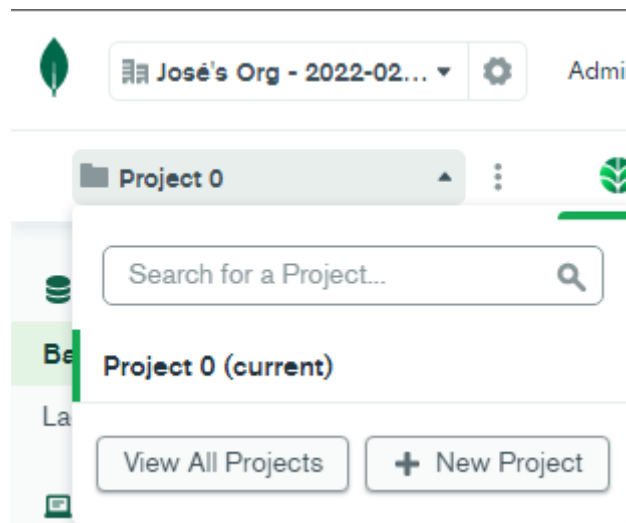
<https://docs.atlas.mongodb.com/>

Ahi nos registraremos con la cuenta del instituto por ejemplo:



Al yo tener la del instituto inicio sesion directamente.

Creamos un nuevo proyecto



Le damos nombre al nuevo proyecto

ORG DE JOSÉ - 2022-02-15 > PROYECTOS

Crear un proyecto

Nombra tu proyecto

Añadir miembros

Nombra tu proyecto

Los nombres de los proyectos deben ser únicos dentro de la organización (y otras restricciones).

proyectopruebamongo

Cancelar

Próximo

Agregamos a los usuarios que queramos

Crear un proyecto

✓ **Nombra tu proyecto**

Añadir miembros

Agregar miembros y establecer permisos

pepe

Otorgue a sus miembros permisos de acceso a continuación.

jguedia2908@g.educaand.es
(tú)

Propietario del proyecto ▼

Cancelar

← Regresa

Próximo

Procedemos a crear una base de datos



crear una base de datos

Elija su proveedor de nube, región y especificaciones.

construir una base de datos

Una vez que su base de datos esté en funcionamiento, migre en vivo una base de datos MongoDB existente a Atlas con nuestro [Servicio de migración en vivo](#).

Crear un clúster compartido

¡Bienvenido a MongoDB Atlas ! Hemos recomendado algunas de nuestras opciones más populares, pero no dude en personalizar su clúster según sus necesidades. Para más información, consulta nuestra [documentación](#) .

AVANCE sin servidor

Dedicado

GRATIS Compartido

Para aprender y explorar MongoDB en un entorno sandbox. Controles básicos de configuración.



No se requiere tarjeta de crédito para comenzar. Actualice a clústeres dedicados para una funcionalidad completa. Explore con conjuntos de datos de muestra. Límite de un clúster gratuito por proyecto.

Proveedor de nube y región

AWS, París (eu-west-3) ▼

aws

Google Cloud

Azure




























★Región recomendada ⓘ ⓘRegión de nivel pagado ⓘ

GRATIS

¡Siempre libre! Su clúster M0 es ideal para experimentar en un espacio aislado limitado. Puede actualizar a un clúster de producción en cualquier momento.

[Atrás](#)

Crear clúster

NORTEAMÉRICA	EUROPA	AUSTRALIA
 Oregón (us-west-2) ★	 Fráncfort (eu-central-1) ★	 Sidney (ap-sureste-2) ★
 Virginia del Norte (us-east-1) ★	 Estocolmo (eu-norte-1) ★	ASIA
 Ohio (us-este-2) ★ 	 París (eu-west-3) ★	 Hong Kong (ap-este-1) ★
 Norte de California (us-west-1) 	 Irlanda (eu-west-1) ★	 Bombay (ap-sur-1)
 Montreal (ca-central-1) 	 Londres (eu-west-2) ★ 	 Singapur (ap-sureste-1) ★
SUDAMERICA	 Milán (eu-sur-1) ★ 	 Tokio (ap-noreste-1) ★
 Sao Paulo (sa-este-1)	ORIENTE MEDIO	 Seúl (ap-noreste-2)
	 Bahréin (me-sur-1) ★	 Osaka (ap-noreste-3) ★ 
	ÁFRICA	
	 Ciudad del Cabo (af-sur-1) ★	

Lo dejamos todo predeterminado, ya que en este caso yo elijo la opcion gratuita, aunque puedes elegir las de pago si lo deseas

Creamos un usuario

Nombre de usuario y contraseña

Certificado

Cree un usuario de la base de datos utilizando un nombre de usuario y una contraseña. Los usuarios tendrán [privilegios de lectura y escritura en cualquier base de datos](#) de forma predeterminada. Puede actualizar estos permisos y/o crear usuarios adicionales más adelante. Asegúrese de que estas credenciales sean diferentes a su nombre de usuario y contraseña de MongoDB Cloud.

Nombre de usuario


Clave 

 Autogenerar contraseña segura

 Dupdo


Crear usuario

Configuramos en mi caso que solo quiero acceder a esta base de datos desde mi direccion ip



Mi entorno local

Use esto para agregar direcciones IP de red a la lista de acceso IP. Esto se puede modificar en cualquier momento.



Entorno de nube

Úselo para configurar el acceso a la red entre Atlas y su nube o entorno local. Específicamente, configure listas de acceso IP, interconexión de redes y puntos finales privados.

Agregue entradas a su lista de acceso IP

Solo una dirección IP que agregue a su lista de acceso podrá conectarse a los clústeres de su proyecto. Puede administrar las entradas de IP existentes a través de la [página de acceso a la red](#).

Dirección IP

Descripción

Lista de acceso IP

Descripción



Mi dirección IP

le clúster M0...
de 3 a 5 minutos.



Ahora procedemos a conectar la base de datos

Cluster0



R 0

W 0

Last 2 minutes

100.0/s



Connections 0

Last 2 minutes

100.0



In 0.0 B/s

Out 0.0 B/s

Last 2 minutes

100.0 B/s



[

L

5

VERSION

5.0.6

REGION

AWS / Paris (eu-west-3)

CLUSTER TIER

M0 Sandbox (General)

TYPE

Replica Set - 3 nodes

BACKUPS

Inactive

LINKED REALM APP

None Linked

Conectamos con MongoDB Shell



Connect with the MongoDB Shell

Interact with your cluster using MongoDB's interactive Javascript interface



Si no tenemos instalado Mongo Shell dirígete a este enlace:

<https://www.mongodb.com/try/download/shell>

Nos dirigimos a Mongo Atlas y en el apartado de Mongo db Shell copiamos el enlace y lo pegamos en la powershell

```
PS C:\Users\jgd02> mongosh "mongodb+srv://cluster0.gccf5.mongodb.net/myFirstDatabase" --apiVersion 1 --username pepe059
Enter password: ****
Current Mongosh Log ID: 62253a9f4cd4790fe71e700d
Connecting to:      mongodb+srv://cluster0.gccf5.mongodb.net/myFirstDatabase?appName=mongosh+1.2.2
Using MongoDB:      5.0.6 (API Version 1)
Using Mongosh:      1.2.2

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

Warning: Found ~/.mongorc.js, but not ~/.mongoshrc.js. ~/.mongorc.js will not be loaded.
You may want to copy or rename ~/.mongorc.js to ~/.mongoshrc.js.
Atlas atlas-larypp-shard-0 [primary] myFirstDatabase>
```

Luego después de esos copiamos nuestros inserts y los pegamos en la powershell:


```

Atlas atlas-1arypp-shard-0 [primary] myFirstDatabase>
Atlas atlas-1arypp-shard-0 [primary] myFirstDatabase> db.camiones.updateOne(
...     {
...         _id:01
...     },
...     {
...         $push:{
...             item:{
...                 marca:"mercedes",
...                 modelo:"benz",
...                 preciofabricacion:210.000,
...                 precioventa:534.000,
...                 unidadesvendidas:3,
...                 pais:"suecia",
...                 extras:["nevera","portavasos","extintor"],
...                 caracteristicas:{
...                     automatico:true,
...                     caballosvapor:[420,480,510,580,710],
...                     peso:7220,
...                     carburante:"hvo"
...                 }
...             }
...         }
...     }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-1arypp-shard-0 [primary] myFirstDatabase>

```

A continuacion pondremos en la powershell nuestros \$out:

```

Atlas atlas-larypp-shard-0 [primary] myFirstDatabase> db.camiones.aggregate([
...   {
...     $project: {
...       _id: 0,
...       item: 0,
...     },
...   },
...   {
...     $out: {
...       db: "ventas",
...       coll: "autoventas"
...     }
...   }
... ])

Atlas atlas-larypp-shard-0 [primary] myFirstDatabase> db.camiones.aggregate([
...   {
...     $project: {
...       _id: 1,
...       item: 1
...     },
...   },
...   {
...     $out: {
...       db: "ventas",
...       coll: "autoitems"
...     }
...   }
... ])

Atlas atlas-larypp-shard-0 [primary] myFirstDatabase>

```

Al no salirnos nada es que esta todo correcto

A continuacion procederemos a poner nuestraas consultas

1º consulta:

```

Atlas atlas-larypp-shard-0 [primary] myFirstDatabase> db.camiones.aggregate([
...   {
...     $group: {
...       _id: "$vendedor",
...       numerodeventas: {
...         $count: {},
...       }
...     },
...   },
...   {
...     $sort: {
...       numerodeventas: -1
...     }
...   }
... ])
[
  { _id: 'Autoline', numerodeventas: 1 },
  { _id: 'Mascus', numerodeventas: 1 },
  { _id: 'Empresite', numerodeventas: 1 },
  { _id: 'Autocamiones', numerodeventas: 1 },
  { _id: 'Europages', numerodeventas: 1 },
  { _id: 'Truckscout', numerodeventas: 1 }
]
Atlas atlas-larypp-shard-0 [primary] myFirstDatabase>

```

2º consulta:

```
Atlas atlas-larypp-shard-0 [primary] myFirstDatabase> db.camiones.aggregate([
...   {
.....     $unwind: {
.....       path: "$item",
.....     },
.....   },
...   {
.....     $group: {
.....       _id: "$vendedor",
.....       mediaponderada: {
.....         $avg: {
.....           $multiply: ["$item.preciofabricacion", "$item.precioventa"],
.....         },
.....       },
.....     },
.....   },
... ])
[
  { _id: 'Autoline', mediaponderada: 44400000000 },
  { _id: 'Truckscout', mediaponderada: 32057500000 },
  { _id: 'Mascus', mediaponderada: 25170000000 },
  { _id: 'Autocamiones', mediaponderada: 13608750000 },
  { _id: 'Empresite', mediaponderada: 44344022428 },
  { _id: 'Europages', mediaponderada: 10703187500 }
]
Atlas atlas-larypp-shard-0 [primary] myFirstDatabase>
```

3º consulta:

4º consulta:

```
Atlas atlas-larypp-shard-0 [primary] myFirstDatabase> db.camiones.aggregate([
...   {$unwind: '$item'}
... ])
[
  {
    _id: 1,
    item: {
      marca: 'scania',
      modelo: 's',
      preciofabricacion: 270000,
      precioventa: 524000,
      unidadesvendidas: 2,
      pais: 'suecia',
      extras: [ 'cama', 'portavasos', 'climatizacion' ],
      caracteristicas: {
        automatico: true,
        caballosvapor: [ 440, 430, 500, 610, 750 ],
        peso: 7220,
        carburante: 'hvo'
      }
    },
    fechaventa: ISODate("2021-12-16T00:00:00.000Z"),
  }
]
```

```
    },
    fechaventa: ISODate("2022-05-26T00:00:00.000Z"),
    cliente: { nombre: 'Primafrío', codigo: 'B34187338' },
    vendedor: 'Autocamiones'
  }
]
Type "it" for more
Atlas atlas-larypp-shard-0 [primary] myFirstDatabase>
```

5º consulta:

```
Atlas atlas-larypp-shard-0 [primary] myFirstDatabase> db.camiones.aggregate([
...   {
.....     $unwind:{
.....       path:"$item"
.....     }
.....   },
...   {
.....     $group:{
.....       _id:"$cliente",
.....       camionescomprados:{
.....         $sum:"$item.unidadesvendidas"
.....       }
.....     }
.....   }
... ]
[
  {
    _id: { nombre: 'Heineken', codigo: 'B34185688' },
    camionescomprados: 12
  },
  {
    _id: { nombre: 'Garcia', codigo: 'C44183218' },
    camionescomprados: 10
  },
  {
    _id: { nombre: 'TIR Sevilla', codigo: 'B34187888' },
    camionescomprados: 13
  },
  {
    _id: { nombre: 'Garciden', codigo: 'B34387888' },
    camionescomprados: 10
  },
  {
    _id: { nombre: 'Primafrío', codigo: 'B34187338' },
    camionescomprados: 10
  },
  {
    _id: { nombre: 'Nanta', codigo: 'BC3465888' },
    camionescomprados: 11
  }
]
Atlas atlas-larypp-shard-0 [primary] myFirstDatabase>
```

6º consulta:

```
Atlas atlas-1arypp-shard-0 [primary] myFirstDatabase> db.camiones.aggregate(
...   [
...     {$group:
...       { _id:null,
...         maxundsvendidas:{$max:"$item.unidadesvendidas"},
...         minundsvendidas:{$min:"$item.unidadesvendidas"}
...       }
...     ]
...   )
[
  {
    _id: null,
    maxundsvendidas: [ 6, 2, 3, 1 ],
    minundsvendidas: [ 2, 1, 3, 4, 3 ]
  }
]
Atlas atlas-1arypp-shard-0 [primary] myFirstDatabase>
```

7º consulta:

```
... {
..... $project:{
.....     month:{
.....         $month:"$fechaventa"
.....     },
.....     netIncome:{
.....         $subtract: [
.....             {
.....                 $multiply: [
.....                     "$item.precioventa",
.....                     "$item.unidadesvendidas"
.....                 ],
.....             },
.....             {
.....                 $multiply: [
.....                     "$item.preciofabricacion",
.....                     "$item.unidadesvendidas"
.....                 ],
.....             },
.....         ]
.....     },
..... },
..... {
.....     $group:{
.....         _id:"$month",
.....         avg:{
.....             $avg:"$netIncome"
.....         }
.....     },
..... },
..... {
.....     $project:{
.....         mes:"$_id",
.....         ingresosnetos:{
.....             $round:[
.....                 "$avg",
.....                 2
.....             ],
.....         },
.....         _id:0
.....     },
..... },
..... {
.....     $sort:{
.....         avgNetIncome:-1
.....     }
..... }
... ])
[
  { mes: 2, ingresosnetos: 278750 },
  { mes: 3, ingresosnetos: 167000 },
  { mes: 5, ingresosnetos: 162000 },
  { mes: 12, ingresosnetos: 276194.4 },
  { mes: 7, ingresosnetos: 231605 },
  { mes: 4, ingresosnetos: 215500 }
]
Atlas atlas-1arypp-shard-0 [primary] myFirstDatabase>
```


8º consulta:

```
Atlas atlas-1arypp-shard-0 [primary] myFirstDatabase> db.camiones.aggregate([
...   {
.....     $unwind:{
.....       path:"$item"
.....     }
.....   },
...   {
.....     $project:{
.....       _id:0,
.....       vendedor:"$cliente.nombre",
.....       dinerototal:{
.....         $multiply: ["$item.unidadesvendidas","$item.precioventa"] }
.....     },
.....   },
...   {
.....     $group: {
.....       _id:"$vendedor",
.....       dinerototal:{ $sum:"$dinerototal"}
.....     },
.....   },
...   {
.....     $sort: {
.....       dinerototal:-1
.....     }
.....   }
... ]))
[
  { _id: 'TIR Sevilla', dinerototal: 2761602 },
  { _id: 'Nanta', dinerototal: 2630000 },
  { _id: 'Heineken', dinerototal: 2132000 },
  { _id: 'Garcia', dinerototal: 1892000 },
  { _id: 'Garciden', dinerototal: 1651920 },
  { _id: 'Primafrío', dinerototal: 1403000 }
]
Atlas atlas-1arypp-shard-0 [primary] myFirstDatabase>
```

9º consulta:

9º consulta:

```
Atlas atlas-1arypp-shard-0 [primary] myFirstDatabase> db.camiones.aggregate([
... {
..... $lookup: {
.....   from:"ventas",
.....   localField:"id",
.....   foreignField:"id",
.....   as:"ventas"
..... }
..... },
... {
..... $project:{
.....   _id:0,
.....   vendedores:{$arrayElemAt:["$ventas",0]},
.....   dinero:{$multiply:["$unidadesvendidas","$precioventa"]},
.....   vendedor:"$vendedor"
..... }
..... },
... {
..... $group: {
.....   _id:"$vendedor",
.....   dinero: { $sum:"$dinero"}
..... }
..... },
... {
..... $sort: {
.....   dinero:-1
..... }
..... }
... ]))
[
  { _id: 'Autocamiones', dinero: 0 },
  { _id: 'Europages', dinero: 0 },
  { _id: 'Truckscout', dinero: 0 },
  { _id: 'Mascus', dinero: 0 },
  { _id: 'Autoline', dinero: 0 },
  { _id: 'Empresite', dinero: 0 }
]
Atlas atlas-1arypp-shard-0 [primary] myFirstDatabase>
```

10º consulta:

```
Atlas atlas-1arypp-shard-0 [primary] myFirstDatabase> db.camiones.aggregate([{$sample : {size : 2}}])
[
  {
    _id: 5,
    item: [
      {
        marca: 'scania',
        modelo: 'crewcab',
        preciofabricacion: 80000,
        precioventa: 120000,
        unidadesvendidas: 5,
        pais: 'suecia',
        extras: [ 'radio', 'camara', 'bluetooth' ],
        caracteristicas: {
          automatico: false,
          caballosvapor: [ 410, 510, 620, 680 ],
          peso: 4220,
          carburante: 'hvo'
        }
      }
    ],
    {
      marca: 'volvo',
```

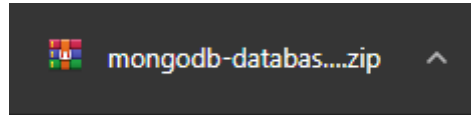
Mongo DataBaseTool

Si no tenemos Mongo Import nos lo tendremos que instalar desde este enlace:


Para usar estas utilidades hay que instalar las "database-tools", añadiendo el path en las variables del sistema.

<https://www.mongodb.com/try/download/database-tools>
















Se nos descargara un archivo winrar:



Una vez descargado lo extraemos y en la carpeta "bin" los archivos que hay los tenemos que pegar en la variable del sistema:

 bsondump.exe	19.158.478	5.841.001	Aplicación	01/02/2022 14:49	B6A3BDC0
 mongodump.exe	22.915.001	7.023.634	Aplicación	01/02/2022 14:49	155E0A8A
 mongoexport.exe	22.471.290	6.897.723	Aplicación	01/02/2022 14:50	D104A7D8
 mongofiles.exe	23.680.579	7.227.040	Aplicación	01/02/2022 14:50	A5F25CDD
 mongoimport.exe	22.822.627	7.003.663	Aplicación	01/02/2022 14:50	DEEF9E9F
 mongorestore.exe	23.382.439	7.178.595	Aplicación	01/02/2022 14:49	B734F017
 mongostat.exe	22.124.786	6.786.519	Aplicación	01/02/2022 14:50	4BD330D9
 mongotop.exe	21.728.005	6.663.386	Aplicación	01/02/2022 14:50	D307AA43

Y los pegamos en los archivos de programa de MongoDB

Este equipo > Disco local (C:) > Archivos de programa > MongoDB > Server > 5.0 > bin				
Nombre	Fecha de modificación	Tipo	Tamaño	
 bsondump.exe	01/02/2022 14:49	Aplicación	18.710 KB	
 InstallCompass.ps1	19/09/2021 19:29	Script de Window...	2 KB	
 mongo.exe	19/09/2021 20:59	Aplicación	21.673 KB	
 mongod.cfg	17/10/2021 16:59	Archivo de origen ...	1 KB	
 mongod.exe	19/09/2021 20:59	Aplicación	45.935 KB	
 mongod.pdb	19/09/2021 20:59	VisualStudio.pdb....	521.076 KB	
 mongodump.exe	01/02/2022 14:49	Aplicación	22.378 KB	
 mongoexport.exe	01/02/2022 14:50	Aplicación	21.945 KB	
 mongofiles.exe	01/02/2022 14:50	Aplicación	23.126 KB	
 mongoimport.exe	01/02/2022 14:50	Aplicación	22.288 KB	
 mongorestore.exe	01/02/2022 14:49	Aplicación	22.835 KB	
 mongos.exe	19/09/2021 20:19	Aplicación	29.096 KB	
 mongos.pdb	19/09/2021 20:19	VisualStudio.pdb....	305.484 KB	
 mongostat.exe	01/02/2022 14:50	Aplicación	21.607 KB	
 mongotop.exe	01/02/2022 14:50	Aplicación	21.219 KB	

Mongo Export / Import

`mongoexport --collection=events --db=reporting --out=events.json`

Usas ese comando y lo importas poniendo tus datos, puedes importar en json, csv o tsv