

1. Abrir el shell de MongoDB (mongo) y pararse en la db finanzas:

mongo

use finanzas

2. Práctica 2a - Realizar las siguientes operaciones:
  - a. Obtener los números de factura de los clientes que contengan una i (minúscula o mayúscula) pero que no sea Manoni.

```
> db.facturas.find(<<"cliente.apellido":{$regex:"i",$options:"i",$ne:"Manoni"}>>,<
nroFactura:1,_id:0>)
{ "nroFactura" : 1002 }
{ "nroFactura" : 1000 }
{ "nroFactura" : 1001 }
{ "nroFactura" : 1007 }
{ "nroFactura" : 1008 }
{ "nroFactura" : 1009 }
```

```
db.facturas.find({"cliente.apellido":{$regex:"I",$options:"i",$
ne:"Manoni"}},{nroFactura:1,_id:0})
```

- b. Obtener las facturas de clientes con apellido terminado en i, cuya región sea CABA y donde hayan comprado una TUERCA, sin importar su medida.

```
> db.facturas.find(<<"cliente.apellido":/i$/,"cliente.region":"CABA","item.produc
to":/^TUERCA/i>>)
{ "_id" : ObjectId<"55e4a6fbbfc68c676a041065">, "cliente" : { "apellido" : "Zava
si", "cuit" : 2038373771, "nombre" : "Martin", "region" : "CABA" }, "condPago" :
"CONTADO", "fechaEmision" : ISODate<"2014-02-20T00:00:00Z">, "fechaVencimiento"
: ISODate<"2014-02-20T00:00:00Z">, "item" : [ { "producto" : "TUERCA 2mm", "can
tidad" : 6, "precio" : 60 }, { "cantidad" : 12, "precio" : 134, "producto" : "CO
RREA 10mm" } ], "nroFactura" : 1002 }
{ "_id" : ObjectId<"55e4a6fbbfc68c676a04106c">, "cliente" : { "apellido" : "Zava
si", "cuit" : 2038373771, "nombre" : "Martin", "region" : "CABA" }, "condPago" :
"CONTADO", "fechaEmision" : ISODate<"2014-02-20T00:00:00Z">, "fechaVencimiento"
: ISODate<"2014-02-20T00:00:00Z">, "item" : [ { "producto" : "TUERCA 2mm", "can
tidad" : 6, "precio" : 60 }, { "cantidad" : 12, "precio" : 134, "producto" : "CO
RREA 10mm" } ], "nroFactura" : 1002 }
```

```
db.facturas.find({"cliente.apellido":/i$/,"cliente.region":"CAB
A","item.producto":/^TUERCA/i})
```

- c. Obtener las facturas donde se haya comprado una correa, sin importar la medida.

```
> db.facturas.find(<<"item.producto":/^CORREA/>>)
{ "_id" : ObjectId<"55e4a6fbbfc68c676a041065">, "cliente" : { "apellido" : "Zava
si", "cuit" : 2038373771, "nombre" : "Martin", "region" : "CABA" }, "condPago" :
"CONTADO", "fechaEmision" : ISODate<"2014-02-20T00:00:00Z">, "fechaVencimiento"
: ISODate<"2014-02-20T00:00:00Z">, "item" : [ { "producto" : "TUERCA 2mm", "can
tidad" : 6, "precio" : 60 }, { "cantidad" : 12, "precio" : 134, "producto" : "CO
RREA 10mm" } ], "nroFactura" : 1002 }
{ "_id" : ObjectId<"55e4a6fbbfc68c676a041064">, "cliente" : { "apellido" : "Zava
si", "cuit" : 2038373771, "nombre" : "Martin", "region" : "CABA" }, "condPago" :
"CONTADO", "fechaEmision" : ISODate<"2014-02-20T00:00:00Z">, "fechaVencimiento"
: ISODate<"2014-02-20T00:00:00Z">, "item" : [ { "producto" : "TUERCA 2mm", "can
tidad" : 6, "precio" : 60 }, { "cantidad" : 12, "precio" : 134, "producto" : "CO
RREA 10mm" } ], "nroFactura" : 1002 }
```

```
db.facturas.find({"item.producto":/^CORREA/})
```

- d. Obtener los datos de cliente de las facturas donde se haya comprado 12 o más de cualquier ítem.

```
> db.facturas.find(<<"item.cantidad":{$gte:12}>>,<cliente:1,_id:0>>)
{ "cliente" : { "apellido" : "Zavasi", "cuit" : 2038373771, "nombre" : "Martin",
"region" : "CABA" } }
{ "cliente" : { "apellido" : "Manoni", "cuit" : 2029889382, "nombre" : "Juan Man
uel", "region" : "NEA" } }
{ "cliente" : { "apellido" : "Zavasi", "cuit" : 2038373771, "nombre" : "Martin",
"region" : "CABA" } }
{ "cliente" : { "apellido" : "Manoni", "cuit" : 2029889382, "nombre" : "Juan Man
uel", "region" : "NEA" } }
{ "cliente" : { "apellido" : "Zavasi", "cuit" : 2038373771, "nombre" : "Martin",
```

```
db.facturas.find({"item.cantidad":{$gte:12}},{cliente:1,_i
d:0})
```

- e. Obtener las facturas donde no se hayan comprado "SET HERRAMIENTAS".

```
> db.facturas.find(<<"item.producto":{$ne:"SET HERRAMIENTAS"}>>)
{ "_id" : ObjectId<"55e4a6fbbfc68c676a041065">, "cliente" : { "apellido" : "Zava
si", "cuit" : 2038373771, "nombre" : "Martin", "region" : "CABA" }, "condPago" :
"CONTADO", "fechaEmision" : ISODate<"2014-02-20T00:00:00Z">, "fechaVencimiento"
: ISODate<"2014-02-20T00:00:00Z">, "item" : [ { "producto" : "TUERCA 2mm", "can
tidad" : 6, "precio" : 60 }, { "cantidad" : 12, "precio" : 134, "producto" : "CO
RREA 10mm" } ], "nroFactura" : 1002 }
{ "_id" : ObjectId<"55e4a6fabfc68c676a041063">, "cliente" : { "nombre" : "Marina
", "region" : "CENTRO", "apellido" : "Malinez", "cuit" : 2740488484 }, "condPago
" : "CONTADO", "fechaEmision" : ISODate<"2014-02-20T00:00:00Z">, "fechaVencimien
to" : ISODate<"2014-02-20T00:00:00Z">, "item" : [ { "cantidad" : 11, "precio" :
```

```
db.facturas.find({"item.producto":{$ne:"SET HERRAMIENTAS"}})
```

- f. Insertar una factura número 999 con usted como cliente, habiendo comprado un Destornillador.

```
> db.facturas.insert(<nroFactura:999,cliente:<apellido:"Boffino",nombre:"Patricio",region:"PBA">,item:[<producto:"DESTORNILLADOR",cantidad:1>]>)
WriteResult(< "nInserted" : 1 >)
```

```
db.facturas.insert({nroFactura:999,cliente:{apellido:"Boffino",
nombre:"Patricio",region:"PBA"},item:[{producto:"DESTORNILLADOR",
,cantidad:1}]})
```

- g. ¿De qué forma insertará el siguiente array de documentos para que, sin importar si alguno de ellos falla, el resto igualmente se inserte?

```
[{_id:1},{_id:1},{_id:2}]
```

```
> db.facturas.insert([<_id:1>,<_id:1>,<_id:2>],<ordered:false>)
BulkWriteResult(<
  "writeErrors" : [
    {
      "index" : 1,
      "code" : 11000,
      "errmsg" : "E11000 duplicate key error index: finanzas.facturas.$_id_ dup key: { : 1.0 }",
      "op" : {
        "_id" : 1
      }
    }
  ],
  "writeConcernErrors" : [ ],
  "nInserted" : 2,
  "nUpserted" : 0,
  "nMatched" : 0,
  "nModified" : 0,
  "nRemoved" : 0,
  "upserted" : [ ]
})
```

```
db.facturas.insert([{_id:1},{_id:1},{_id:2}])
```

- h. Eliminar todas las facturas de los clientes del CENTRO.

```
> db.facturas.remove(<"cliente.region":"CENTRO">)
WriteResult(< "nRemoved" : 4589 >)
```

```
db.facturas.remove({"cliente.region":"CENTRO"})
```

- i. Incrementar el número de factura de la 1501.

```
> db.facturas.update(<nroFactura:1501>,<$inc:<nroFactura:1>>)<br>WriteResult<< "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 >><br>>
```

```
db.facturas.update({nroFactura:1501},{ $inc:{nroFactura:1}})
```

- j. A la factura número 1500 cambiarle la condición de pago a “30 Ds FF”

```
> db.facturas.update(<nroFactura:1500>,<$set:<condPago:"30 Ds FF">>)<br>WriteResult<< "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 >><br>>
```

```
db.facturas.update({nroFactura:1500},{ $set:{condPago:"30 Ds<br>FF" }})
```

- k. A la factura número 1515 incrementar el campo “vistas” y decrementar el campo “contador”.

```
> db.facturas.update(<nroFactura:1515>,<$inc:<vistas:1,contador:-1>>)<br>WriteResult<< "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 >><br>>
```

```
db.facturas.update({nroFactura:1515},<br>{$inc:{vistas:1,contador:-1}})
```