

## Práctica 3 – NoSQL

Vamos a crear la siguiente colección con los pedidos de una tienda de moda:

```
db.andrajos.insert(
[{'_id': 'Oldo', 'País': 'España',
  pedidos:[{item:'cazadora',precio:200,enviada: true},
            {item:'pajarita',precio:40,enviada: true}]}],
{'_id': 'Ambros', 'País': 'Italia',
  pedidos:[{item:'pantalón chulo',precio:160,enviada: false},
            {item:'marianos',precio:25,enviada: true}]}],

{'_id': 'Isca', 'País': 'España',
  pedidos:[{item:'pañuelo seda',precio:400,enviada: false},
            {item:'falda',precio:60,enviada: false},
            {item:'calcetines',precio:6,enviada: false} ]}],
])
```

1) El usuario 'Isca' corrige su País, que ya no es España sino 'Andorra'.  
Copia en la solución la instrucción escribirías para hacer el cambio

**Solución:**

```
db.andrajos.update({'_id':'Isca'},{$set:{'País':'Andorra'}})
```

2) Llega un nuevo pedido del usuario Oldo. Se trata de una 'corbata con dibujos de ositos' y precio 120. Añadirla al principio de su array de pedidos (se puede suponer que \_id:Oldo ya existe), poniendo enviada=false.

**Solución**

```
db.andrajos.update({'_id':"Oldo"}, {$push:{ pedidos: {$each:[{item:"corbata con dibujos de ositos",
precio: 120, enviada: false}], $position:0}}})
```

3) Ya hemos enviado la corbata a Oldo. Cambiar el valor de enviada=true para la 'corbata con dibujos de ositos'. Nota: se sabe que ocupa la posición 0

**Solución**

```
db.andrajos.update({'_id':"Oldo"},{$set:{"pedidos.0.enviada":true}})
```

4) Añadir un nueva pareja clave valor de la forma {premium:true} para aquellos clientes con alguna compra sobre 100 euros (Nota: para todos los clientes con esta característica).

**Solución**

```
db.andrajos.update({'pedidos.precio':{'$gt':100}}, {'$set':{'premium:true'}},{multi:true})
```

5) La cuenta premium lleva incluido envío gratuito, pero queremos hacer esto solo para envíos fuera de España. Quitar la clave premium:true para aquellos clientes de fuera de España que disfruten de este tipo de cuenta.

**Solución**

```
db.andrajos.update({'País':{'$ne: "España"}}, {$unset:{premium:true}}, {multi:true})
```

6) Oldo se ha borrado de nuestra página web; escribe en la solución la instrucción para eliminarlo siempre y cuando no tenga ningún pedido con enviado a false.

**Solución:**

```
db.andrajos.remove({'$and':{'_id: "Oldo"},{'pedidos.enviada':{'$nin:[false]}}})
```

o

```
db.andrajos.remove({$and:[{_id: "Oldo"}, {"pedidos.enviada": {$ne: false}}]})
```

7) Suponer documentos de la colección friends:

```
{ _id : "Mike", interests : [ "chess", "botany" ] }
```

¿Cuál será el resultado de los siguientes modificaciones?

```
db.friends.update( { _id : "Mike" }, { $push : { interests : "skydiving" } } );  
db.friends.update( { _id : "Mike" }, { $pop : { interests : -1 } } );  
db.friends.update( { _id : "Mike" }, { $addToSet : { interests : "skydiving" } } );  
db.friends.update( { _id : "Mike" }, { $pushAll: { interests : [ "skydiving" , "skiing" ] } } );
```

### Solución

```
{ _id : "Mike", interests : [ "botany", "skydiving", "skydiving" , "skiing" ] }
```

(algunas soluciones detallan el resultado de cada update...también está bien).

8) Difícil. La colección pru tiene documentos con el aspecto

```
db.pru.insert({a:[1,2,3,4,5],_id:0})  
db.pru.insert({a:[1,2,9],_id:1})
```

Queremos escribir una consulta para encontrar documentos sean subconjuntos del conjunto [1,4,5,2,3,5,1,4,8]

En el ejemplo anterior:

```
{ "_id" : 0, "a" : [ 1, 2, 3, 4, 5 ] }
```

### Solución:

```
db.pru.find({a:{$not:{$elemMatch:{$nin:[1,4,5,2,3,5,1,4,8]}}}})  
{ "_id" : 0, "a" : [ 1, 2, 3, 4, 5 ] }
```

9) [2] Un nuevo usuario con nombre 'Oldo' se ha registrado en el sistema. En el 'País' ha escrito 'España'. Escribir una instrucción que:

- [1] Se asegure de que un usuario \_id:'Oldo' con 'País':'Italia' existe.

Es decir si ya estaba, solo le cambiará el país, si no existe lo insertará.

- [1] Si no existe, es decir si se inserta, le pondrá el campo 'pedidos' a []

Copia en la solución la consulta.

(Para probar conviene hacer la misma consulta con un \_id que no exista y ver si funciona en este caso).

### Solución

```
db.andrajos.update({'_id':'Oldo'},{'$set':{'País':'Italia'},'$setOnInsert':{'pedidos':[]}},{'upsert:true'})
```