CreateSample Documents

16 Aralık 2021 Perşembe 04:49

```
db.shoppingCart.insertMany([{index:NumberInt(1)},
{index:NumberInt(2)}... 5 ])

show collections
    └>shoppingCart.

db.getCollection('shoppingCart').find({})
```

```
db.collection.method(query, update, options)
```

$set

```
db.shoppingCart.update(
  {index:3},
  {
    $set:{
        cartId: NumberInt(325),
        customer:{
            name: "Mike Foster",
            email: "mfoster@test.com",
            age: NumberInt(22)
        }
        cart:[]
    }
  }
);

db.getCollection('shoppingCart').find({index:2})
```

$unset

```
db.getCollection('shoppingCart')
  .update(
    {index:1},
    {$set:{
        cartId:NumberInt(125),
        customer:{
            name:"Mike Foster",
            email:"mfoster@test.com",
            age:NumberInt(22)
        },
        cart:[]
    }
    },
    {}
  );
```

→ Mistake

```
db.getCollection('shoppingCart')
  .update(
    {index:1},
    {$unset:{
        cartId:1,
        cart:""
    }},
    {}
  )
```

```
db.getCollection('shoppingCart').find({index:1})
```

↳ cartId } alanların siler.
cart

index
customer:{
name
email
age
}

alanları kalır.

```
db.getCollection('shoppingCart')
  .update(
    {index:1},
    {$unset:{
```

```
    "customer.name": 1,
    "customer.age": 1
  }
);
```

⇒ sadece email alanı kalır.

customer.email

```
db.getCollection('shoppingCart')
  .update(
    {index: 1},
    {$unset: {
        customer: 1
    }}
  )
```

⇒ customer alanını siler. Sadece index kalır.

```
db.getCollection('shoppingCart')
  .update(
    {},
    {$unset: {
        customer: 1
    }},
    {}
  )
```

index:1'i güncelle.
index:2'yi güncellemez.

NOT: update metodu bulduğu ilk dökümanı güncelle. Diğerlerini güncellemez.

```
db.shoppingCart.update({}, {$set: {processed: false}})
```

↳ processed: false alanını ekler, güncellemiş olur.

```
{ _id: ... ,
  "index": 1,
  "processed": false
}
```

db.getCollection("shoppingCart").find({})

```
db.shoppingCart.update({index:10}, {$set: {processed: false}})
```

update() with {multi:true}

```
db.collection.update(query, update, {multi:true})
```

```
write Result {
  "nMatched" : 35,
  "nUpserted" : 0,
  "nModified" : 35
}
```

```
db.shoppingCart.update({}, {$set:{processed: false}}, {multi:true})
```

↳ `WriteResult({ "nMatched" : 5, "nUpserted" : 0, "nModified" : 4 })`

updateOne()

```
db.getCollection('shoppingCart')
  .updateOne(
    {cartId:325},
    { $set: {
        processed:true
    }},
    {}
  )
```

⇒ ack : true
matched: 1
modified : 1

2 dəfə çalıştırınca modified: 0 olacaq.

```
db.getCollection('shoppingCart')
  .find({cartId: 325})
```

* consistency için təkrar processed: false yaptık. (Tutor(11:16))

```
db.collection("shoppingCart")
  .updateOne(
    {cartId:325},
    { $set : { processed:false }},
    {}
  )
```

updateMany()

```
db.collection.updateMany(query, update, options)
```

↳ Yuxarıdakı cart (array) (boş dizi) eləcə.

```
db.getCollection('shoppingCart')
  .updateMany(
    {cart:{$exists: false}},
    { $set : {
        cart: []
    }},
    {}
  )
```

⟹

```
ack :true
matched : 4
modified :4
```

replaceOne()

```
db.collection.replaceOne( query, replacement, option)
```

```
db.shoppingCart.replaceOne(
  {index:1},
  {
    index:1,
    processed:false,
    cart:[]
  }
)
```

Note: _id değişmez.

```
db.getCollection("shoppingCart")
  .replaceOne(
    {"index":1},
    {"index":1,
     "processed":true,
     "cart":["item1","item2"]
    },
    {}
  )
```

```
db.getCollection("shoppingCart")
  .find({index:1})
```

↳

```
{ "_id": ObjectId(...),
  "index": 1.0,
```

```
    "processed" : true,
    "cart" : [
            "item1",
            "item2"
        ]
}
```

```
db.getCollection('shoppingCart')
  .replaceOne(
    {"index" : 1},
    {
      "index" : NumberInt(1),
      "processed" : true,
      "cart" : ["item1","item2"]
    },
    {}
  )
```

index ⇒ 1.0 → 1 (tamsay.)

Combine Multiple Update Operates

```
db.shoppingCart.update(
  {index:4},
  { $set:{
      cartId: NumberInt(435),
      "customer.name" : "Sonata Larsen",
      "customer.email" : "slarsen@test.com"
    },
    $unset:{
      neworder:1
    }
  }
)
db.getCollection('shoppingCart')
  .updateOne(
    {index:4},
    { $set :{
```

```
                    newOrder :true
                }
            },
            {}
        )

db.getCollection('shoppingCart')
    .updateOne(
        {index:4},
        {
            $set:{
                cartId: NumberInt(435),
                "customer.name":"Sanata Larsen",
                "customer.email": "slarsen@tett.com"
            },
            $unset: {
                neworder: 1
            }
        },
        {}
    )
```

$rename

```
db.shoppingCart.update(
    { cartId:{ $exists:true }},
    {
        $rename: {
            cartId:"orderId",
            anotherField: "anotherName"
        }
    },
    { multi:true }
);
```

```
update(
    :
    ;          =  updateMany()

{multi:true}
)
```

```
{multi:true}
)
```

```
db.getCollection('shoppingCart')
    .updateMany(
        {orderId:{$exists:true}},
        {
            $rename:{
                orderId:"cartId"
            }
        },
        {}
    )
```

`$currentDate`

```
db.getCollection('shoppingCart')
    .updateOne(
        {index:13},
        {
            $set:{
                updatedAt:new Date()
            }
        },
        {}
    )
```

```
db.getCollection('shoppingCart').find({index:13})
```

↳ `"updatedAt" : ISODate("2018-04-...")`

```
db.getCollection('shoppingCart')
    .updateMany(
        {updateAt:{$exists:false}},
        {
            $currentDate:{
                updatedAt:true
            }
        },
        {}
    )
```

```
db.getCollection('shoppingCart')
  .updateMany(
    {cartId:325},
    {
      $set:{
          cart:["item1"]
      },
      $currentDate:{
          updatedAt:true
      }
    },
    {}
  )
```

Array Update Operators

`$` `$push` `$pull` `$pullAll` `$pop` `$addToSet`

$push Operator

```
{$push:{<arrayfieldname>:<element>}}
```

```
db.shoppingCart.update(
  {cartId:561},
  {
    $push:{
      cart:"item1"
    }
  }
)
```

```
db.shoppingCart.update(
  {cartId:561},
  {
    $push:{
      cart:"item1"
    }
  }
)
```

```
db.shoppingCart.update(
  {cartId:561},
  {
    $push:{
      cart:{ $each:["item2","item3"]}
    }
  }
)
```

Not: Öyle bir alan ismesa adını, bir elemanlı bir dizi otomatik olarak oluşturur.

```
db.getCollection('shoppingCart')
  .updateOne(
    {index:1},
    {
      $set:{
        cartId:NumberInt(561)
      },
      $unset:{
        cart:1
      }
    },
    {}
  )
```

```
db.getCollection('shoppingCart')
  .updateOne(
    {cartId:561},
    {
      $push:{
        cart:"item1"
      }
    },
    {}
  )
```

```
db.getCollection('shoppingCart')
  .updateOne(
    {cartId:561},
    {
      $push:"item2"
    },
    {}
  )
```

```
db.getCollection('shoppingCart')
  .updateOne(
    {cartId:561},
    {
      $push:{
        cart:{$each:["item3","item4","item5"]}
      }
```

item1
item2
item3
item4
item5

```
            cart: {$each: ["item3", "item4", "item5"]}
        }
    },
    {}
)
```

item5

```
db.getCollection('shoppingCart')
    .updateOne(
        {cartId: 561},
        {
            $push: {
                cart: "item2"
            }
        },
        {}
    )
```

item1
item2
item3
item4
item5
item2

$addToSet → elemanı yoksa ekler, varsa eklemez.
Eklemeden önce var mı yok mu kontrol ediyor.

```
db.shoppingCart.update(
    {cartId:561},
    {
        $addToset: {
            cart: "item1"
        }
    },
)
```

```
db.shoppingCart.update(
    {cartId: 561},
    {
        $addToSet:
        { cart: {$each: ["item1", "item2"]}}
    }
)
```

* Array alanı yoksa, bir elemanlı bir diziyi oluşturur.

```
db.shoppingCart.updateOne(
    {cartId:561},
    {
        $addToSet: {
            cart: "item1"
        }
    }
)
```

→ Dizide zaten "item1" olduğundan, eklemez.

```
    }
  }
}
```

elemanı.

```
db.shoppingCart.updateOne(
  {cartId:561},
  {
    $unset:{
        cart:1
    }
  },
  {}
)
```

⇒ cart alanını sildik

```
db.shoppingCart.updateOne(
    {cartId:561},
    {
      $addToSet:{
          cart:"item1"
      }
    },
    {}
)
```

```
db.shoppingCart.updateOne(
    {cartId:561},
    {
      $addToSet:{
          cart:{$each:["item2","item3"]}
      }
    },
    {}
)
```

$pop

{$pop:{cart:-1}} → baştan siler

{$pop:{cart:1}} → sondan siler

-1 → baştan

1 → sondan

```
db.shoppingCart.updateOne(
  {cartId:561},
```

```
db.shoppingCart.updateOne(
  {cartId:561},
```

```
{cartId:561},
{
  $pop:{cart:1}
},
{}
)
```

```
db.shoppingCart.updateOne(
{cartId:561},
{
  $pop:{cart:1}
},
{}
)
```

```
db.shoppingCart.updateOne(
{cartId:561},
{
  $pop:{cart:-1}
},
{}
)
```

$pull

```
{$pull:{<arrayFieldName>:<element | condition>}}
```

```
db.shoppingCart.update(
{cartId:561},
{
  $pull:{
    cart:"item1",
    spentAmounts:{$gt:400}
  }
}
)
```

```
db.shoppingCart.updateOne(
{cartId:561},
{
  $pull:{
    cart:"item1",
    spentAmounts:{$gt:400}
  }
}
)
```

```
db.shoppingCart.updateOne(
{cartId:561},
{
  $addToSet:{
    spentAmounts:{$each:[NumberInt(400),
    NumberInt(700),NumberInt(800),
```

```
            NumberInt(200), NumberInt(500)] }
        }
    },
    {}
)

db.shoppingCart.updateOne(
    {cartId: 561},
    {
        $addToSet: {
            spentAmounts: {
                $each:
                    [NumberInt(400), NumberInt(700), NumberInt(800),
                     NumberInt(200), NumberInt(500)]
            }
        }
    },
    {}
)
```

```
db.shoppingCart.updateOne(
    {cartId: 561},
    {
        $pull: {
            spentAmounts: {$gt: 500}
        }
    },
    {}
)
```

```
db.shoppingCart.updateOne(
    {cartId: 561},
    {
        $pull: {
            spentAmounts: {$gt: 500}
        }
    },
    {}
)
```

```
db.shoppingCart.updateOne(
    {cartId: 561},
    {
        $pull: {
            spentAmounts: 200
        }
    },
    {}
)
```

$pullAll

```
db.shoppingcart.update(
    {cartId: 561},
    {
```

⟺

```
db.shoppingCart.update(
    {cartId: 561},
    {
```

```
{cartId: 561},
{
  $pullAll: {
      cart: ["item1","item2"]
  }
}
)
```

⟺

```
{cartId: 561},
{
  $pull: {
    cart: {$in: ["item1","item2"]}
  }
}
)
```

```
db.shoppingCart.updateOne(
  {cartId: 561},
  {
    $pullAll: {
      spentAmounts: [400,500]
    }
  }
)
```

```
db.shoppingCart.updateOne(
  {cartId: 561},
  {
    $push: {
      cart: {$each: ["item1","item2","item1","item4","item3"]}
    },
  {}
)
```

```
db.shoppingCart.updateOne(
  {cartId: 561},
  {
    $pullAll: {
      cart: ["item1","item2"]
    }
  }
)
```

⟹ item1 item2 item1 item4 item1 item3 ⟹ item3

Positional Operator $

```
{$set: {<arrayField.$>: <value>}}
{$set: {<arrayField.$.field>: <value>}}
```

```
db.shoppingCart.updateOne(
  {cartId: 325, cart: "item2"},
  {
```

```
{cartId : 325, cart : "item2" },
{
    $set : {
        "cart.$" : "updatedItem2"
    }
}
)
```

```
db.shoppingCart.updateOne(
    {cartId : 561, cart : "item3" },
    {
        $unset : {
            "cart.$" : 1
        }
    },
    {}
)
```

item1
updatedItem2
item3 → null
item4

```
db.shoppingCart.updateOne(
    {cartId : 561 },
    {
        $pull : {
            cart : null
        }
    },
    {}
)
```

updatedItem2
~~item3~~
item4

Positional Operator $ in Nested Docs

```
{cart : [
    {
        title : "TV",
        price : NumberInt(340),
        quantity : NumberInt(2)
    },
    {
        title : "Phone",
        price : NumberInt(150),
        quantity : NumberInt(1)
    }
]}
```

```
db.shoppingCart.updateOne(
    {
        "cart.title" : "Phone"
    },
    {
        $set : {
            "cart.$.quantity" : NumberInt(4)
        }
    }
)
```

```
db.shoppingCart.updateOne(
    {"cartId" : 456 },
    {
```

```
    {"cartId":456},
    {
      $push:{
        cart:{$each[{"title":"TV"},{"title":"Phone"}]}
      }
    },
    {}
)

db.shoppingCart.updateOne(
  {"cartId":456,"cart.title":"TV"},
  {
    $set:{
      "cart.$.price":NumberInt(340),
      "cart.$.quantity":NumberInt(2)
    }
  },
  {}
)

db.shoppingCart.updateOne(
  {"cartId":456,"cart.title":"Phone"},
  {
    $set:{
      "cart.$.price":NumberInt(150),
      "cart.$.quantity":NumberInt(1)
    }
  },
  {}
)

db.shoppingCart.updateOne(
  {
    cart:{$elemMatch:{title:"Phone",price:150}}
  },
  {
    $set:{
      "cart.$.quantity":NumberInt(2)
    }
  }
)

db.shoppingCart.updateOne(
  {
    cart:{$elemMatch:{title:"Phone",price:150}}
  },
  {
    $set:{
      "cart.$.quantity":NumberInt(2)
    }
```

```
    { $set : {
        "cart.$.quantity" : NumberInt(2)
      }
    }
)

db.shoppingCart.updateOne (
  { "cartId" :456,
    cart : {
        $elemMatch : {
            title :"Phone",
            price : 150
        }
    }
  },
  { $set : {
      "cart.$.quantity" : NumberInt(2)
    }
  },
  {}
)

db.shoppingCart.updateOne(
  { "cartId" : 456,
    cart : {
        $elemMatch : {
            title :"Phone",
            price : 150
        }
    }
  },
  { $set : {
      "cart.$.quantity" : NumberInt(2)
    }
  },
  {}
)
```

$inc

```
db.shoppingCart.update (
```

```
db.shoppingCart.update(
    {cartId:325},
    {
      $inc:{
          totalCartItems:NumberInt(1)
      }
    }
)

db.shoppingCart.update(
    {cartId:325},
    {
      $inc:{
          totalCartItems:NumberInt(1)
      }
    }
)

db.shoppingCart.updateOne(
    {
      "cartId":456,
      cart:{
            $elemMatch:{
                title:"phone",
                price:150
            }
      },
      {
        $inc:{
            "cart.$.quantity":NumberInt(5)
        }
      },
      {}
)
```