

# MongoTest 22BCE10375

Name Ayush Pandey

Registration 22BCE10375

```
db.sales.insertMany([
  { "_id" : 1, "item" : "Americanos", "price" : 5, "size": "Short", "quantity" : 22, "date" :
    ISODate("2022-01-15T08:00:00Z") },
  { "_id" : 2, "item" : "Cappuccino", "price" : 6, "size": "Short","quantity" : 12, "date" :
    ISODate("2022-01-16T09:00:00Z") },
  { "_id" : 3, "item" : "Lattes", "price" : 15, "size": "Grande","quantity" : 25, "date" :
    ISODate("2022-01-16T09:05:00Z") },
  { "_id" : 4, "item" : "Mochas", "price" : 25,"size": "Tall", "quantity" : 11, "date" : ISODate("2022-
    02-17T08:00:00Z") },
  { "_id" : 5, "item" : "Americanos", "price" : 10, "size": "Grande","quantity" : 12, "date" :
    ISODate("2022-02-18T21:06:00Z") },
  { "_id" : 6, "item" : "Cappuccino", "price" : 7, "size": "Tall","quantity" : 20, "date" :
    ISODate("2022-02-20T10:07:00Z") },
  { "_id" : 7, "item" : "Lattes", "price" : 25,"size": "Tall", "quantity" : 30, "date" : ISODate("2022-
    02-21T10:08:00Z") },
  { "_id" : 8, "item" : "Americanos", "price" : 10, "size": "Grande","quantity" : 21, "date" :
    ISODate("2022-02-22T14:09:00Z") },
  { "_id" : 9, "item" : "Cappuccino", "price" : 10, "size": "Grande","quantity" : 17, "date" :
    ISODate("2022-02-23T14:09:00Z") },
  { "_id" : 10, "item" : "Americanos", "price" : 8, "size": "Tall","quantity" : 15, "date" :
    ISODate("2022-02-25T14:09:00Z")}
]);
```

```
mongosh mongodb+srv://<u>c</u> X + v
Atlas atlas-x6taep-shard-0 [primary] test> use MongoClass
switched to db MongoClass
Atlas atlas-x6taep-shard-0 [primary] MongoClass> db.products.insertMany([
...   { "_id" : 1, "name" : "xPhone", "price" : 799, "releaseDate": ISODate("2011-05-14"), "spec" : { "ram" : 4, "screen" : 6.5, "cpu" : 2.66 }, "color": ["white", "black"], "storage": [64, 128, 256]},
...   { "_id" : 2, "name" : "xTablet", "price" : 899, "releaseDate": ISODate("2011-09-01"), "spec" : { "ram" : 16, "screen" : 9.5, "cpu" : 3.66 }, "color": ["white", "black", "purple"], "storage": [128, 256, 512]},
...   { "_id" : 3, "name" : "SmartTablet", "price" : 899, "releaseDate": ISODate("2015-01-14"), "spec" : { "ram" : 12, "screen" : 9.7, "cpu" : 3.66 }, "color": ["blue"], "storage": [16, 64, 128]},
...   { "_id" : 4, "name" : "SmartPad", "price" : 699, "releaseDate": ISODate("2020-05-14"), "spec" : { "ram" : 8, "screen" : 9.7, "cpu" : 1.66 }, "color": ["white", "orange", "gold", "gray"], "storage": [128, 256, 1024]},
...   { "_id" : 5, "name" : "SmartPhone", "price" : 599, "releaseDate": ISODate("2022-09-14"), "spec" : { "ram" : 4, "screen" : 9.7, "cpu" : 1.66 }, "color": ["white", "orange", "gold", "gray"], "storage": [128, 256]}
... ])
{
  acknowledged: true,
  insertedIds: { '0': 1, '1': 2, '2': 3, '3': 4, '4': 5 }
}
Atlas atlas-x6taep-shard-0 [primary] MongoClass> db.sales.find().pretty()
[
  {
    _id: 1,
    item: 'Americanos',
    price: 5,
    size: 'Short',
    quantity: 22,
    date: ISODate('2022-01-15T08:00:00.000Z')
  },
  {
    _id: 2,
    item: 'Cappuccino',
    price: 6,
    size: 'Short',
    quantity: 12,
    date: ISODate('2022-01-16T09:00:00.000Z')
  },
  {
    _id: 3,
    item: 'Lattes',
    price: 15,
    size: 'Tall',
    quantity: 11,
    date: ISODate('2022-02-17T08:00:00.000Z')
  }
]
```

1. How do you find all documents where the field "item" is "Mochas"?

```
Atlas atlas-x6taep-shard-0 [primary] MongoClass> db.sales.find({ "item": "Mochas" })
[
  {
    _id: 4,
    item: 'Mochas',
    price: 25,
    size: 'Tall',
    quantity: 11,
    date: ISODate('2022-02-17T08:00:00.000Z')
  }
]
Atlas atlas-x6taep-shard-0 [primary] MongoClass>
```

2. How do you find all items who are less than 15 of price?

```

Atlas atlas-x6taep-shard-0 [primary] MongoClient> db.sales.find({ "price": { "$lt": 15 } })
[
  {
    _id: 1,
    item: 'Americanos',
    price: 5,
    size: 'Short',
    quantity: 22,
    date: ISODate('2022-01-15T08:00:00.000Z')
  },
  {
    _id: 2,
    item: 'Cappuccino',
    price: 6,
    size: 'Short',
    quantity: 12,
    date: ISODate('2022-01-16T09:00:00.000Z')
  },
  {
    _id: 5,
    item: 'Americanos',
    price: 10,
    size: 'Grande',
    quantity: 12,
    date: ISODate('2022-02-18T21:06:00.000Z')
  },
  {
    _id: 6,
    item: 'Cappuccino',
    price: 7,
    size: 'Tall',
    quantity: 20,
    date: ISODate('2022-02-20T10:07:00.000Z')
  },
  {
    _id: 8,
    item: 'Americanos',
    price: 10,
    size: 'Grande',
  }
]

```

3. How do you find items who are either less than 10 or greater than 20?

```

Atlas atlas-x6taep-shard-0 [primary] MongoClient> db.sales.find({ $or: [ { "price": { "$lt": 10 } }, { "price": { "$gt": 20 } } ] })
[
  {
    _id: 1,
    item: 'Americanos',
    price: 5,
    size: 'Short',
    quantity: 22,
    date: ISODate('2022-01-15T08:00:00.000Z')
  },
  {
    _id: 2,
    item: 'Cappuccino',
    price: 6,
    size: 'Short',
    quantity: 12,
    date: ISODate('2022-01-16T09:00:00.000Z')
  },
  {
    _id: 4,
    item: 'Mochas',
    price: 25,
    size: 'Tall',
    quantity: 11,
    date: ISODate('2022-02-17T08:00:00.000Z')
  },
  {
    _id: 6,
    item: 'Cappuccino',
    price: 7,
    size: 'Tall',
    quantity: 20,
    date: ISODate('2022-02-20T10:07:00.000Z')
  },
  {
    _id: 7,
    item: 'Lattes',
    price: 25,
    size: 'Tall',
    quantity: 30,
  }
]

```

4. How do you find all documents where the quantity field exists?

```

Atlas atlas-x6taep-shard-0 [primary] MongoClass> db.sales.find({ "quantity": { "$exists": true } })
[
  {
    _id: 1,
    item: 'Americanos',
    price: 5,
    size: 'Short',
    quantity: 22,
    date: ISODate('2022-01-15T08:00:00.000Z')
  },
  {
    _id: 2,
    item: 'Cappuccino',
    price: 6,
    size: 'Short',
    quantity: 12,
    date: ISODate('2022-01-16T09:00:00.000Z')
  },
  {
    _id: 3,
    item: 'Lattes',
    price: 15,
    size: 'Grande',
    quantity: 25,
    date: ISODate('2022-01-16T09:05:00.000Z')
  },
  {
    _id: 4,
    item: 'Mochas',
    price: 25,
    size: 'Tall',
    quantity: 11,
    date: ISODate('2022-02-17T08:00:00.000Z')
  },
  {
    _id: 5,
    item: 'Americanos',
    price: 10,
    size: 'Grande'
  }
]

```

```

db.products.insertMany([
  { "_id" : 1, "name" : "xPhone", "price" : 799, "releaseDate": ISODate("2011-05-14"), "spec" : {
    "ram" : 4, "screen" : 6.5, "cpu" : 2.66 }, "color":["white","black"],"storage":[64,128,256]},
  { "_id" : 2, "name" : "xTablet", "price" : 899, "releaseDate": ISODate("2011-09-01") , "spec" : {
    "ram" : 16, "screen" : 9.5, "cpu" : 3.66 }, "color":["white","black","purple"],"storage":
    [128,256,512]},
  { "_id" : 3, "name" : "SmartTablet", "price" : 899, "releaseDate": ISODate("2015-01-14"),
    "spec" : { "ram" : 12, "screen" : 9.7, "cpu" : 3.66 }, "color":["blue"],"storage":[16,64,128]},
  { "_id" : 4, "name" : "SmartPad", "price" : 699, "releaseDate": ISODate("2020-05-14"), "spec" :
    { "ram" : 8, "screen" : 9.7, "cpu" : 1.66 }, "color":["white","orange","gold","gray"],"storage":
    [128,256,1024]},
  { "_id" : 5, "name" : "SmartPhone", "price" : 599, "releaseDate": ISODate("2022-09-14"),
    "spec" : { "ram" : 4, "screen" : 9.7, "cpu" : 1.66 }, "color":
    ["white","orange","gold","gray"],"storage":[128,256]}
])

```

5. How do you find all documents where the color array contains the value "white"?

```

Atlas atlas-x6taep-shard-0 [primary] MongoClass> db.collection.find({ "color": "white" }).pretty()

```

6. How do you update the ram of the user named "xTablet" to 24?

```

Atlas atlas-x6taep-shard-0 [primary] MongoClass> db.collection.updateOne(
...   { "user": "xTablet" },
...   { "$set": { "ram": 24 } }
... )
...
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}

```

7. How do you find all products and only return their 'spec' field?

```
Atlas atlas-x6taep-shard-0 [primary] MongoClass> db.products.find({}, { "spec": 1, "_id": 0 }).pretty()
[
  { spec: { ram: 4, screen: 6.5, cpu: 2.66 } },
  { spec: { ram: 16, screen: 9.5, cpu: 3.66 } },
  { spec: { ram: 12, screen: 9.7, cpu: 3.66 } },
  { spec: { ram: 8, screen: 9.7, cpu: 1.66 } },
  { spec: { ram: 4, screen: 9.7, cpu: 1.66 } }
]
Atlas atlas-x6taep-shard-0 [primary] MongoClass> |
```

8. How do you find all products and sort them by their price in descending order?

```
Atlas atlas-x6taep-shard-0 [primary] MongoClass> db.products.find().sort({ "price": -1 }).pretty()
[
  {
    _id: 2,
    name: 'xTablet',
    price: 899,
    releaseDate: ISODate('2011-09-01T00:00:00.000Z'),
    spec: { ram: 16, screen: 9.5, cpu: 3.66 },
    color: [ 'white', 'black', 'purple' ],
    storage: [ 128, 256, 512 ]
  },
  {
    _id: 3,
    name: 'SmartTablet',
    price: 899,
    releaseDate: ISODate('2015-01-14T00:00:00.000Z'),
    spec: { ram: 12, screen: 9.7, cpu: 3.66 },
    color: [ 'blue' ],
    storage: [ 16, 64, 128 ]
  },
  {
    _id: 1,
    name: 'xPhone',
    price: 799,
    releaseDate: ISODate('2011-05-14T00:00:00.000Z'),
    spec: { ram: 4, screen: 6.5, cpu: 2.66 },
    color: [ 'white', 'black' ],
    storage: [ 64, 128, 256 ]
  },
  {
    _id: 4,
    name: 'SmartPad',
    price: 699,
    releaseDate: ISODate('2020-05-14T00:00:00.000Z'),
    spec: { ram: 8, screen: 9.7, cpu: 1.66 },
    color: [ 'white', 'orange', 'gold', 'gray' ],
    storage: [ 128, 256, 1024 ]
  },
  {
    _id: 5,
    name: 'SmartPhone',
    price: 699,
    releaseDate: ISODate('2020-05-14T00:00:00.000Z'),
    spec: { ram: 8, screen: 9.7, cpu: 1.66 },
    color: [ 'white', 'orange', 'gold', 'gray' ],
    storage: [ 128, 256, 1024 ]
  }
]
```

9. How do you find the first 2 products, skipping the first one?

```

Atlas atlas-x6taep-shard-0 [primary] MongoClient> db.products.find().sort({ "price": -1 }).pretty()
[
  {
    _id: 2,
    name: 'xTablet',
    price: 899,
    releaseDate: ISODate('2011-09-01T00:00:00.000Z'),
    spec: { ram: 16, screen: 9.5, cpu: 3.66 },
    color: [ 'white', 'black', 'purple' ],
    storage: [ 128, 256, 512 ]
  },
  {
    _id: 3,
    name: 'SmartTablet',
    price: 899,
    releaseDate: ISODate('2015-01-14T00:00:00.000Z'),
    spec: { ram: 12, screen: 9.7, cpu: 3.66 },
    color: [ 'blue' ],
    storage: [ 16, 64, 128 ]
  },
  {
    _id: 1,
    name: 'xPhone',
    price: 799,
    releaseDate: ISODate('2011-05-14T00:00:00.000Z'),
    spec: { ram: 4, screen: 6.5, cpu: 2.66 },
    color: [ 'white', 'black' ],
    storage: [ 64, 128, 256 ]
  },
  {
    _id: 4,
    name: 'SmartPad',
    price: 699,
    releaseDate: ISODate('2020-05-14T00:00:00.000Z'),
    spec: { ram: 8, screen: 9.7, cpu: 1.66 },
    color: [ 'white', 'orange', 'gold', 'gray' ],
    storage: [ 128, 256, 1024 ]
  },
  {
    _id: 5,

```

10. How do you find all users whose name starts with the letter "S" and price should be greater of 700?

```

Atlas atlas-x6taep-shard-0 [primary] MongoClient> db.products.find(
...   "name": { "$regex": "^S", "$options": "i" },
...   "price": { "$gt": 700 }
... }).pretty()
[
  {
    _id: 3,
    name: 'SmartTablet',
    price: 899,
    releaseDate: ISODate('2015-01-14T00:00:00.000Z'),
    spec: { ram: 12, screen: 9.7, cpu: 3.66 },
    color: [ 'blue' ],
    storage: [ 16, 64, 128 ]
  }
]
Atlas atlas-x6taep-shard-0 [primary] MongoClient> |

```

11. How do you calculate the total price of all items?

```

Atlas atlas-x6taep-shard-0 [primary] MongoClient> db.sales.aggregate([
...   {
...     "$group": {
...       "_id": null,
...       "totalPrice": { "$sum": "$price" }
...     }
...   }
... ])
...
[ { _id: null, totalPrice: 121 } ]
Atlas atlas-x6taep-shard-0 [primary] MongoClient> |

```

12. How do you project only the size field for each items?

```
Atlas atlas-x6taep-shard-0 [primary] MongoClient> db.sales.find({}, { "size": 1, "_id": 0 }).pretty()
[
  { size: 'Grande' },
  { size: 'Grande' },
  { size: 'Tall' },
  { size: 'Tall' },
  { size: 'Short' },
  { size: 'Tall' },
  { size: 'Grande' },
  { size: 'Grande' },
  { size: 'Tall' },
  { size: 'Short' }
]
Atlas atlas-x6taep-shard-0 [primary] MongoClient> |
```

13. How do you find items who are Tall size & group them by item wise?

```
Atlas atlas-x6taep-shard-0 [primary] MongoClient> db.sales.aggregate([
...   { "$match": { "size": "Tall" } }, // Filter only 'Tall' size items
...   {
...     "$group": {
...       "_id": "$item", // Group by 'item'
...       "totalQuantity": { "$sum": "$quantity" } // Sum up quantities for each item
...     }
...   })
[
  { _id: 'Mochas', totalQuantity: 11 },
  { _id: 'Lattes', totalQuantity: 30 },
  { _id: 'Americanos', totalQuantity: 15 },
  { _id: 'Cappuccino', totalQuantity: 20 }
]
Atlas atlas-x6taep-shard-0 [primary] MongoClient> |
```

14. How do you find the second item when sorted by price in ascending order?

```
Atlas atlas-x6taep-shard-0 [primary] MongoClient> db.sales.find().sort({ "price": 1 }).skip(1).limit(1).pretty()
[
  {
    _id: 2,
    item: 'Cappuccino',
    price: 6,
    size: 'Short',
    quantity: 12,
    date: ISODate('2022-01-16T09:00:00.000Z')
  }
]
Atlas atlas-x6taep-shard-0 [primary] MongoClient> |
```

15. How do you group by items and calculate the total number of items and average price in a single aggregation?

```
Atlas atlas-x6taep-shard-0 [primary] MongoClass> db.sales.aggregate([
...   {
...     "$group": {
...       "_id": "$item",
...       "totalQuantity": { "$sum": "$quantity" },
...       "averagePrice": { "$avg": "$price" }
...     }
...   }
... ])
...
[
  { _id: 'Americanos', totalQuantity: 70, averagePrice: 8.25 },
  {
    _id: 'Cappuccino',
    totalQuantity: 49,
    averagePrice: 7.666666666666667
  },
  { _id: 'Mochas', totalQuantity: 11, averagePrice: 25 },
  { _id: 'Lattes', totalQuantity: 55, averagePrice: 20 }
]
Atlas atlas-x6taep-shard-0 [primary] MongoClass> |
```