```
db.createCollection("orders")
{ ok: 1 }
test> db.orders.insertMany( [
...    { _id: 0, name: "Pepperoni", size: "small", price: 19,
...      quantity: 10, date: ISODate( "2021-03-13T08:14:30Z" ) },
...    { _id: 1, name: "Pepperoni", size: "medium", price: 20,
...      quantity: 20, date : ISODate( "2021-03-13T09:13:24Z" ) },
...    { _id: 2, name: "Pepperoni", size: "large", price: 21,
...      quantity: 30, date : ISODate( "2021-03-17T09:22:12Z" ) },
...    { _id: 3, name: "Cheese", size: "small", price: 12,
...      quantity: 15, date : ISODate( "2021-03-13T11:21:39.736Z" ) },
...    { _id: 4, name: "Cheese", size: "medium", price: 13,
...      quantity:50, date : ISODate( "2022-01-12T21:23:13.331Z" ) },
...    { _id: 5, name: "Cheese", size: "large", price: 14,
...      quantity: 10, date : ISODate( "2022-01-12T05:08:13Z" ) },
...    { _id: 6, name: "Vegan", size: "small", price: 17,
...      quantity: 10, date : ISODate( "2021-01-13T05:08:13Z" ) },
...    { _id: 7, name: "Vegan", size: "medium", price: 18,
...      quantity: 10, date : ISODate( "2021-01-13T05:10:13Z" ) }
... ] )
{
  acknowledged: true,
  insertedIds: { '0': 0, '1': 1, '2': 2, '3': 3, '4': 4, '5': 5, '6': 6, '7': 7 }
}
test> db.orders.find()
[
  {
    _id: 0,
    name: 'Pepperoni',
    size: 'small',
    price: 19,
    quantity: 10,
    date: ISODate("2021-03-13T08:14:30.000Z")
  },
  {
    _id: 1,
    name: 'Pepperoni',
    size: 'medium',
    price: 20,
    quantity: 20,
    date: ISODate("2021-03-13T09:13:24.000Z")
  },
  {
    _id: 2,
  name: 'Pepperoni',
  size: 'large',
  price: 21,
  quantity: 30,
  date: ISODate("2021-03-17T09:22:12.000Z")
},
{
  _id: 3,
  name: 'Cheese',
  size: 'small',
  price: 12,
  quantity: 15,
  date: ISODate("2021-03-13T11:21:39.736Z")
},
{
  _id: 4,
  name: 'Cheese',
  size: 'medium',
  price: 13,
  quantity: 50,
  date: ISODate("2022-01-12T21:23:13.331Z")
},
{
  _id: 5,
  name: 'Cheese',
  size: 'large',
  price: 14,
  quantity: 10,
  date: ISODate("2022-01-12T05:08:13.000Z")
},
{
  _id: 6,
  name: 'Vegan',
  size: 'small',
  price: 17,
  quantity: 10,
  date: ISODate("2021-01-13T05:08:13.000Z")
},
{
  _id: 7,
  name: 'Vegan',
  size: 'medium',
  price: 18,
  quantity: 10
```

```
          date: ISODate("2021-01-13T05:10:13.000Z")
  }
]
test>
db.orders.aggregate([{$match:{size:"medium"}},{$group:{_id:"$name",totalQuantity:{$sum:"$qua
ntity"}}}])
[
  { _id: 'Pepperoni', totalQuantity: 20 },
  { _id: 'Vegan', totalQuantity: 10 },
  { _id: 'Cheese', totalQuantity: 50 }
]
test>
db.orders.aggregate([{$match:{size:"medium"}},{$group:{_id:"$name",totalCount:{$sum:1}}}])
[
  { _id: 'Pepperoni', totalCount: 1 },
  { _id: 'Vegan', totalCount: 1 },
  { _id: 'Cheese', totalCount: 1 }
]
test> db.orders.aggregate([{$group:{_id:"$name",pizzaCount:{$sum:1}}}])
[
  { _id: 'Pepperoni', pizzaCount: 3 },
  { _id: 'Vegan', pizzaCount: 2 },
  { _id: 'Cheese', pizzaCount: 3 }
]
test> db.orders.aggregate({ $group:{_id:"PIZZA",averagePrice:{$avg:"$price"}}})
[ { _id: 'PIZZA', averagePrice: 16.75 } ]
test> db.orders.aggregate({ $group:{_id:"PIZZAs",averagePrice:{$avg:"$price"}}})
[ { _id: 'PIZZAs', averagePrice: 16.75 } ]
test> db.orders.aggregate({ $group:{_id:"$size",averagePrice:{$avg:"$price"}}})
[
  { _id: 'large', averagePrice: 17.5 },
  { _id: 'small', averagePrice: 16 },
  { _id: 'medium', averagePrice: 17 }
]
test> db.orders.aggregate({
$group:{_id:"$size",averagePrice:{$avg:"$price"}}}).sort(averagePrice:1)
Uncaught:
SyntaxError: Unexpected token, expected "," (1:91)

> 1 | db.orders.aggregate({
$group:{_id:"$size",averagePrice:{$avg:"$price"}}}).sort(averagePrice:1)
    |                                                              ^
  2 |
test> db.orders.aggregate({
$group:{_id:"$size",averagePrice:{$avg:"$price"}}}).order(averagePrice:1)
Uncaught:
SyntaxError: Unexpected token, expected "," (1:92)

> 1 | db.orders.aggregate({
$group:{_id:"$size",averagePrice:{$avg:"$price"}}}).order(averagePrice:1)
    |                                                            ^
  2 |

test> db.orders.aggregate({
$group:{_id:"$size",averagePrice:{$avg:"$price"}}}).order({averagePrice:1})
TypeError: db.orders.agg ... e"}}}).order is not a function
test> db.orders.aggregate({
$group:{_id:"$size",averagePrice:{$avg:"$price"}}}).sort({averagePrice:1})
[
  { _id: 'small', averagePrice: 16 },
  { _id: 'medium', averagePrice: 17 },
  { _id: 'large', averagePrice: 17.5 }
]
test> db.orders.aggregate({
$group:{_id:"$size",averagePrice:{$avg:"$price"}}}).sort({averagePrice:-1})
[
  { _id: 'large', averagePrice: 17.5 },
  { _id: 'medium', averagePrice: 17 },
  { _id: 'small', averagePrice: 16 }
]
test> db.orders.aggregate({ $group:{_id:"$size",averagePrice:{$avg:"$price"}}})
[
  { _id: 'small', averagePrice: 16 },
  { _id: 'medium', averagePrice: 17 },
  { _id: 'large', averagePrice: 17.5 }
]
test>
```