

```
> _MONGOSH
< email_1
> db.locations.createIndex({
  lat:1,
  long:1},
                                { unique:true });
< lat_1_long_1
> db.locations.insertOne({
  address: "dev bbootcamp,USA",
  LAT:37.222555,
  long:-122.895412
})
< {
  acknowledged: true,
  insertedId: ObjectId('6841a53136c5ddf65e83dc48')
}
> db.locations.insertOne({
  address: "yes the place,USA",
  LAT:43.222555,
  long:-152.895412
})
< {
  acknowledged: true,
  insertedId: ObjectId('6841a56a36c5ddf65e83dc49')
}
> db.movies.createIndex({ title:1 ,year:2 })
< title_1_year_2
```

>_MONGOSH

```
{
  _id: ObjectId('6841a39c36c5ddf65e83dc44'),
  email: 'jane@test.com',
  name: 'Jane'
}
> db.users.createIndex({email:1},{unique:true});
< email_1
> db.users.createIndex({email:1},{unique:true});
< email_1
> db.users.drop()
< true
> db.users.insertMany([
  { email: "joh@test.com", name: "John" },
  { email: "jane@test.com", name: "Jane" },
  { email: "john@test.com", name: "johnny"}
]);
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('6841a45136c5ddf65e83dc45'),
    '1': ObjectId('6841a45136c5ddf65e83dc46'),
    '2': ObjectId('6841a45136c5ddf65e83dc47')
  }
}
> db.users.createIndex({email: 1},{unique:true})
< email_1
> db.locations.createIndex({
  lat:1,
  long:1},
```

>_MONGOSH

```
> db.sales.dropIndex({item:1})
< { nIndexesWas: 2, ok: 1 }
> db.users.insertMany([
  [ email: "joh@test.com ", name:"John"],
  [ email: "jane@test.com", name:"jane"],
]);
SyntaxError: Unexpected token, expected ",", (2:7)

[0m [90m 1 | [39m db[33m.[39musers[33m.[39minsertMany([
[31m[1m>[22m[39m[90m 2 | [39m [ email[33m:[39m [32m"joh@test.com "[39m[33m,[39m name[33m:[39m[32m"John"[39m
[90m | [39m [31m[1m^[22m[39m
[90m 3 | [39m [ email[33m:[39m [32m"jane@test.com"[39m[33m,[39m name[33m:[39m[32m"jane"[39m][33m,[39m
[90m 4 | [39m ]][33m;[39m
[90m 5 | [39m[0m
> db.users.insertMany([
  { email: "joh@test.com", name: "John" },
  { email: "jane@test.com", name: "Jane" }
]);
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('6841a39c36c5ddf65e83dc43'),
    '1': ObjectId('6841a39c36c5ddf65e83dc44')
  }
}
> db.users.find()
< {
  _id: ObjectId('6841a39c36c5ddf65e83dc43'),
  email: 'joh@test.com',
```

>_MONGOSH

```
{
  item: 'Cappuccino',
  price: 10,
  size: 'Grande',
  quantity: 17
}
{
  item: 'Lattes',
  price: 15,
  size: 'Grande',
  quantity: 25
}
> db.sak
< test.sak
> db.sales.getIndexes()
< [ { v: 2, key: { _id: 1 }, name: '_id_' } ]
> db.sales.createIndex({item:1})
< item_1
> db.sales.getIndexes()
< [
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { item: 1 }, name: 'item_1' }
]
```

>_MONGOSH

```
> db.sales.aggregate([
  {
    $setWindowFields: {
      partitionBy: "$item",          // Group by 'item'
      sortBy: { quantity: -1 },      // Sort each group by quantity descending
      output: {
        rank: { $rank: {} }         // Assign rank within each group
      }
    }
  },
  {
    $match: { rank: 2 }              // Filter only the second highest quantity per item
  },
  {
    $project: {
      _id: 0,
      item: 1,
      price: 1,
      size: 1,
      quantity: 1
    }
  }
]);
< {
  item: 'Americanos',
  price: 10,
  size: 'Grande',
  quantity: 21
}
```

>_MONGOSH

```
> db.sales.aggregate([
  {
    $group: {
      _id: null,
      maxQty: { $max: { $multiply: ["$price", "$quantity"] } },
    },
  },
  {
    $sort: { maxQty: -1 },
  },
  {
    $limit: 2,
  },
  {
    $skip: 1,
  }
]);
< {
  _id: null,
  maxQty: 750
}
> db.sales.aggregate([{$group: {_id: "$item", max_qt: {$sum: "$quantity"}}, {$sort: {max_qt: -1}}, {$limit: 2}]).skip(1);
< {
  _id: 'Lattes',
  max_qt: 55
}
```

>_MONGOSH

```
> db.sales.aggregate([
  {
    $group: {
      _id: null,
      maxQty: { $max: '$quantity'},
    },
  },
  {
    $project: {
      _id: 0,
    },
  },
]);
< {
  maxQty: 30
}
> db.sales.aggregate([
  {
    $group: {
      _id: null,
      maxQty: { $multiply: ["price", "quantity"] },
    },
  },
  {
    $project: {
      _id: 0,
    },
  },
]);
```

>_MONGOSH

```
> db.sales.drop()
< true
> 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")}
]);
< {
  acknowledged: true,
  insertedIds: {
    '0': 1,
    '1': 2,
    '2': 3,
    '3': 4,
    '4': 5,
    '5': 6,
    '6': 7,
    '7': 8,
    '8': 9,
    '9': 10
  }
}
```