

Product JSON: <https://github.com/rvsp/database/blob/master/mongodb/product.json>

1.Find all the information about each products

`db.products.find().pretty()`

```
{
> db.products.find().pretty()
{
  "_id" : ObjectId("65fef5afd27205534761cb20"),
  "id" : "1",
  "product_name" : "Intelligent Fresh Chips",
  "product_price" : 655,
  "product_material" : "Concrete",
  "product_color" : "mint green"
}
{
  "_id" : ObjectId("65fef5afd27205534761cb21"),
  "id" : "2",
  "product_name" : "Practical Fresh Sausages",
  "product_price" : 911,
  "product_material" : "Cotton",
  "product_color" : "indigo"
}
{
  "_id" : ObjectId("65fef5afd27205534761cb22"),
  "id" : "3",
  "product_name" : "Refined Steel Car",
  "product_price" : 690,
  "product_material" : "Rubber",
  "product_color" : "gold"
}
{
  "_id" : ObjectId("65fef5afd27205534761cb23"),
  "id" : "4",
```

2.Find the product price which are between 400 to 800

`db.products.find({product_price:{$gte:400, $lte:800}}).pretty()`

```
Command Prompt - mongo
> db.products.find({product_price:{$gte:400, $lte:800}}).pretty()

{"_id" : ObjectId("65fef5afd27205534761cb20"),
  "id" : "1",
  "product_name" : "Intelligent Fresh Chips",
  "product_price" : 655,
  "product_material" : "Concrete",
  "product_color" : "mint green"}

{"_id" : ObjectId("65fef5afd27205534761cb22"),
  "id" : "3",
  "product_name" : "Refined Steel Car",
  "product_price" : 690,
  "product_material" : "Rubber",
  "product_color" : "gold"}

{"_id" : ObjectId("65fef5afd27205534761cb23"),
  "id" : "4",
  "product_name" : "Gorgeous Plastic Pants",
  "product_price" : 492,
  "product_material" : "Soft",
  "product_color" : "plum"}

{"_id" : ObjectId("65fef5afd27205534761cb25"),
  "id" : "6",
  "product_name" : "Awesome Wooden Towels",
  "product_price" : 474,
```

3.Find the product price which are not between 400 to 600

`db.products.find({product_price:{$not:{$gte:400, $lte:800}}}).pretty()`

```
Command Prompt - mongo
> db.products.find({product_price:{$not:{$gte:400, $lte:800}}}).pretty()
{
  "_id" : ObjectId("65fef5afd27205534761cb21"),
  "id" : "2",
  "product_name" : "Practical Fresh Sausages",
  "product_price" : 911,
  "product_material" : "Cotton",
  "product_color" : "indigo"
}

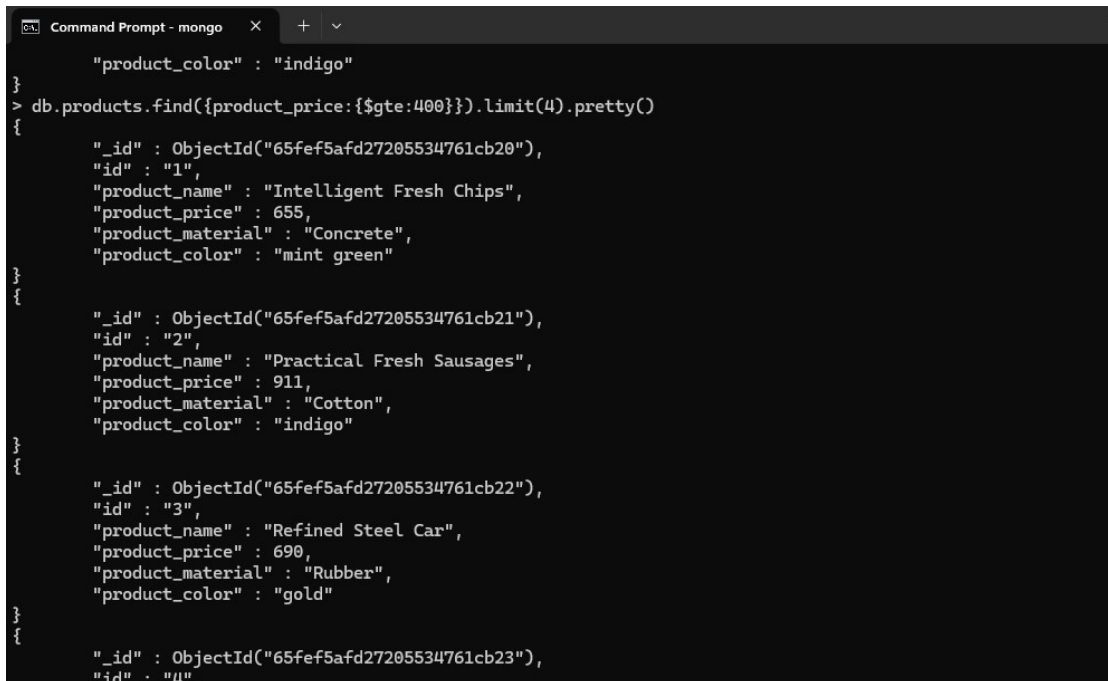
{
  "_id" : ObjectId("65fef5afd27205534761cb24"),
  "id" : "5",
  "product_name" : "Sleek Cotton Chair",
  "product_price" : 33,
  "product_material" : "Fresh",
  "product_color" : "black"
}

{
  "_id" : ObjectId("65fef5afd27205534761cb27"),
  "id" : "8",
  "product_name" : "Incredible Steel Hat",
  "product_price" : 78,
  "product_material" : "Rubber",
  "product_color" : "violet"
}

{
  "_id" : ObjectId("65fef5afd27205534761cb28"),
  "id" : "9",
  "product_name" : "Awesome Wooden Ball",
  "product_price" : 28,
```

4. List the four products which are greater than 500 in price

```
db.products.find({product_price:{$gte:400}}).limit(4).pretty()
```



```
Command Prompt - mongo X + v
    "product_color" : "indigo"
  }
> db.products.find({product_price:{$gte:400}}).limit(4).pretty()
{
  "_id" : ObjectId("65fef5afd27205534761cb20"),
  "id" : "1",
  "product_name" : "Intelligent Fresh Chips",
  "product_price" : 655,
  "product_material" : "Concrete",
  "product_color" : "mint green"
}

{
  "_id" : ObjectId("65fef5afd27205534761cb21"),
  "id" : "2",
  "product_name" : "Practical Fresh Sausages",
  "product_price" : 911,
  "product_material" : "Cotton",
  "product_color" : "indigo"
}

{
  "_id" : ObjectId("65fef5afd27205534761cb22"),
  "id" : "3",
  "product_name" : "Refined Steel Car",
  "product_price" : 690,
  "product_material" : "Rubber",
  "product_color" : "gold"
}

{
  "_id" : ObjectId("65fef5afd27205534761cb23"),
  "id" : "4",
```

5. Find the product name and product material of each product

```
db.products.find({}, { product_name: 1, product_material: 1 }).pretty()
```

```
Command Prompt - mongo X + v
uncaught exception: ReferenceError: product_price is not defined :
p(shell):1:19
> db.products.find({}, { product_name: 1, product_material: 1 }).pretty()

  "_id" : ObjectId("65fef5afd27205534761cb20"),
  "product_name" : "Intelligent Fresh Chips",
  "product_material" : "Concrete"

  "_id" : ObjectId("65fef5afd27205534761cb21"),
  "product_name" : "Practical Fresh Sausages",
  "product_material" : "Cotton"

  "_id" : ObjectId("65fef5afd27205534761cb22"),
  "product_name" : "Refined Steel Car",
  "product_material" : "Rubber"

  "_id" : ObjectId("65fef5afd27205534761cb23"),
  "product_name" : "Gorgeous Plastic Pants",
  "product_material" : "Soft"

  "_id" : ObjectId("65fef5afd27205534761cb24"),
  "product_name" : "Sleek Cotton Chair",
  "product_material" : "Fresh"

  "_id" : ObjectId("65fef5afd27205534761cb25"),
```

6.Find the product with a row id of 10

```
db.products.find({ id: "10" }).pretty()
```

```
Type "it" for more
> db.products.find({ id: "10" }).pretty()

  "_id" : ObjectId("65fef5afd27205534761cb29"),
  "id" : "10",
  "product_name" : "Generic Wooden Pizza",
  "product_price" : 84,
  "product_material" : "Frozen",
  "product_color" : "indigo"
```

7.Find only the product name and product material

```
db.products.find({}, { _id: 0, product_name: 1, product_material:
1 }).pretty()
```

```

Command Prompt - mongo
> db.products.find({}, { _id: 0, product_name: 1, product_material: 1 }).pretty()
{
  "product_name" : "Intelligent Fresh Chips",
  "product_material" : "Concrete"
}
{
  "product_name" : "Practical Fresh Sausages",
  "product_material" : "Cotton"
}
{
  "product_name" : "Refined Steel Car", "product_material" : "Rubber" }
{
  "product_name" : "Gorgeous Plastic Pants", "product_material" : "Soft" }
{
  "product_name" : "Sleek Cotton Chair", "product_material" : "Fresh" }
{
  "product_name" : "Awesome Wooden Towels", "product_material" : "Plastic" }
{
  "product_name" : "Practical Soft Shoes", "product_material" : "Rubber" }
{
  "product_name" : "Incredible Steel Hat", "product_material" : "Rubber" }
{
  "product_name" : "Awesome Wooden Ball", "product_material" : "Soft" }
{
  "product_name" : "Generic Wooden Pizza", "product_material" : "Frozen" }
{
  "product_name" : "Unbranded Wooden Cheese", "product_material" : "Soft" }
{
  "product_name" : "Unbranded Plastic Salad", "product_material" : "Wooden" }
}
{
  "product_name" : "Gorgeous Cotton Keyboard",
  "product_material" : "Concrete"
}
{
  "product_name" : "Incredible Steel Shirt", "product_material" : "Metal" }
{
  "product_name" : "Ergonomic Cotton Hat", "product_material" : "Rubber" }
{
  "product_name" : "Small Soft Chair", "product_material" : "Cotton" }
{
  "product_name" : "Incredible Metal Car", "product_material" : "Fresh" }
{
  "product_name" : "Licensed Plastic Bacon", "product_material" : "Steel" }
{
  "product_name" : "Intelligent Cotton Chips", "product_material" : "Soft" }
}

```

8. Find all products which contain the value of soft in product material

```
db.products.find({ product_material: "Soft" }).pretty()
```

```

> db.products.find({ product_material: "Soft" }).pretty()
{
  "_id" : ObjectId("65fef5afd27205534761cb23"),
  "id" : "4",
  "product_name" : "Gorgeous Plastic Pants",
  "product_price" : 492,
  "product_material" : "Soft",
  "product_color" : "plum"
}

{
  "_id" : ObjectId("65fef5afd27205534761cb28"),
  "id" : "9",
  "product_name" : "Awesome Wooden Ball",
  "product_price" : 28,
  "product_material" : "Soft",
  "product_color" : "azure"
}

{
  "_id" : ObjectId("65fef5afd27205534761cb2a"),
  "id" : "11",
  "product_name" : "Unbranded Wooden Cheese",
  "product_price" : 26,
  "product_material" : "Soft",
  "product_color" : "black"
}

{
  "_id" : ObjectId("65fef5afd27205534761cb32"),
  "id" : "19",

```

9. Find products which contain product color indigo and product price 492.00

```

db.products.find({$or:[{ product_color: "indigo"},
{product_price:'492.00'}]}).pretty()

```

```
Command Prompt - mongo X + v
    "product_color" : "indigo"
> db.products.find({$or:[{ product_color: "indigo"}, {product_price:'492.00'}]}).pretty()
{
  "_id" : ObjectId("65fef5afd27205534761cb21"),
  "id" : "2",
  "product_name" : "Practical Fresh Sausages",
  "product_price" : 911,
  "product_material" : "Cotton",
  "product_color" : "indigo",

  "_id" : ObjectId("65fef5afd27205534761cb29"),
  "id" : "10",
  "product_name" : "Generic Wooden Pizza",
  "product_price" : 84,
  "product_material" : "Frozen",
  "product_color" : "indigo",

  "_id" : ObjectId("65fef5afd27205534761cb30"),
  "id" : "17",
  "product_name" : "Incredible Metal Car",
  "product_price" : 36,
  "product_material" : "Fresh",
  "product_color" : "indigo",

  "_id" : ObjectId("65fef5afd27205534761cb38"),
```

10.Delete the products which product price value are same

```
db.products.aggregate([
  {
    $group: {
      _id: "$product_price",
      count: { $sum: 1 },
      duplicates: { $addToSet: "$_id" }
    }
  },
  {
    $match: {
      count: { $gt: 1 }
    }
  }
])
```

```

    }
  ]).forEach(function(doc) {

    doc.duplicates.shift();

    db.products.deleteMany({ _id: { $in: doc.duplicates } });

  });

```

```

> db.products.aggregate([
...   {
...     $group: {
...       _id: "$product_price",
...       count: { $sum: 1 },
...       duplicates: { $addToSet: "$_id" }
...     }
...   },
...   {
...     $match: {
...       count: { $gt: 1 }
...     }
...   }
... ])
...   ]).forEach(function(doc) {
...     doc.duplicates.shift();
...     db.products.deleteMany({ _id: { $in: doc.duplicates } });
...   });
> |

```

```

db.products.find({}, { product_price : 1 }).pretty();

```



```

}
> db.products.aggregate([
...   {
...     $group: {
...       _id: "$product_price",
...       count: { $sum: 1 },
...       duplicates: { $addToSet: "$_id" }
...     }
...   },
...   {
...     $match: {
...       count: { $gt: 1 }
...     }
...   }
... ])
... ]).forEach(function(doc) {
...   doc.duplicates.shift();
...   db.products.deleteMany({ _id: { $in: doc.duplicates } });
... });
> db.products.find({}, { product_price : 1}).pretty();
{ "_id" : ObjectId("65fef5afd27205534761cb20"), "product_price" : 655 }
{ "_id" : ObjectId("65fef5afd27205534761cb21"), "product_price" : 911 }
{ "_id" : ObjectId("65fef5afd27205534761cb22"), "product_price" : 690 }
{ "_id" : ObjectId("65fef5afd27205534761cb23"), "product_price" : 492 }
{ "_id" : ObjectId("65fef5afd27205534761cb24"), "product_price" : 33 }
{ "_id" : ObjectId("65fef5afd27205534761cb25"), "product_price" : 474 }
{ "_id" : ObjectId("65fef5afd27205534761cb26"), "product_price" : 500 }
{ "_id" : ObjectId("65fef5afd27205534761cb27"), "product_price" : 78 }
{ "_id" : ObjectId("65fef5afd27205534761cb28"), "product_price" : 28 }
{ "_id" : ObjectId("65fef5afd27205534761cb29"), "product_price" : 84 }

```