|  |  |
| --- | --- |
| A picture of a winding road and trees  MONGODB QUERIES DOCUMENTATION | MongoDB Queries Documentation  Author: [CHANDRU VIJAYAKUMAR]  Date: [03-03-2024]  ---  chandru V |

MongoDB Day 1

**1.Find all the information about each products**

Query :

db.product.find();

**Query Output:**

**Retrieve details of all products, including their ID, name, price, material, and color**

{

\_id: ObjectId('65e37384a9e2145934538821'),

id: '1',

product\_name: 'Intelligent Fresh Chips',

product\_price: 655,

product\_material: 'Concrete',

product\_color: 'mint green'

}

{

\_id: ObjectId('65e37384a9e2145934538822'),

id: '2',

product\_name: 'Practical Fresh Sausages',

product\_price: 911,

product\_material: 'Cotton',

product\_color: 'indigo'

}

{

\_id: ObjectId('65e37384a9e2145934538823'),

id: '3',

product\_name: 'Refined Steel Car',

product\_price: 690,

product\_material: 'Rubber',

product\_color: 'gold'

}

{....................}

{....................}

{

\_id: ObjectId('65e37384a9e2145934538839'),

id: '25',

product\_name: 'Licensed Steel Car',

product\_price: 20,

product\_material: 'Cotton',

product\_color: 'indigo'

}

...... 25 Products

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

Query : db.product.find({'product\_price': {$gte: 400, $lte: 800}});

**Query Output:**

**Fetch products with prices ranging from 400 to 800, including details such as ID, name, price, material, and color.**

{

\_id: ObjectId('65e37384a9e2145934538821'),

id: '1',

product\_name: 'Intelligent Fresh Chips',

product\_price: 655,

product\_material: 'Concrete',

product\_color: 'mint green'

}

{

\_id: ObjectId('65e37384a9e2145934538823'),

id: '3',

product\_name: 'Refined Steel Car',

product\_price: 690,

product\_material: 'Rubber',

product\_color: 'gold'

}

{

\_id: ObjectId('65e37384a9e2145934538824'),

id: '4',

product\_name: 'Gorgeous Plastic Pants',

product\_price: 492,

product\_material: 'Soft',

product\_color: 'plum'

}

{

\_id: ObjectId('65e37384a9e2145934538826'),

id: '6',

product\_name: 'Awesome Wooden Towels',

product\_price: 474,

product\_material: 'Plastic',

product\_color: 'orange'

}

{

\_id: ObjectId('65e37384a9e2145934538827'),

id: '7',

product\_name: 'Practical Soft Shoes',

product\_price: 500,

product\_material: 'Rubber',

product\_color: 'pink'

}

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

Query :

db.product.find({$or: [{ 'product\_price': { $lt: 400 } },{ 'product\_price': { $gt: 600 } }]});

**Query Output:**

**Retrieve products with prices outside the 400 to 600 range, displaying details like ID, name, price, material, and color.**

{

\_id: ObjectId('65e37384a9e2145934538821'),

id: '1',

product\_name: 'Intelligent Fresh Chips',

product\_price: 655,

product\_material: 'Concrete',

product\_color: 'mint green'

}

{

\_id: ObjectId('65e37384a9e2145934538822'),

id: '2',

product\_name: 'Practical Fresh Sausages',

product\_price: 911,

product\_material: 'Cotton',

product\_color: 'indigo'

}

{

\_id: ObjectId('65e37384a9e2145934538823'),

id: '3',

product\_name: 'Refined Steel Car',

product\_price: 690,

product\_material: 'Rubber',

product\_color: 'gold'

}

{---------}

{---------}

{---------}

{

\_id: ObjectId('65e37384a9e2145934538838'),

id: '24',

product\_name: 'Tasty Rubber Cheese',

product\_price: 47,

product\_material: 'Frozen',

product\_color: 'orchid'

}

{

\_id: ObjectId('65e37384a9e2145934538838'),

id: '24',

product\_name: 'Tasty Rubber Cheese',

product\_price: 47,

product\_material: 'Frozen',

product\_color: 'orchid'

}

**............. 22 Products**

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

Query : db.product.find({'product\_price':{$gt:500}}).limit(4);

**Query Output:**

**Fetch the details of the top four products with prices exceeding 500.**

{

\_id: ObjectId('65e37384a9e2145934538821'),

id: '1',

product\_name: 'Intelligent Fresh Chips',

product\_price: 655,

product\_material: 'Concrete',

product\_color: 'mint green'

}

{

\_id: ObjectId('65e37384a9e2145934538822'),

id: '2',

product\_name: 'Practical Fresh Sausages',

product\_price: 911,

product\_material: 'Cotton',

product\_color: 'indigo'

}

{

\_id: ObjectId('65e37384a9e2145934538823'),

id: '3',

product\_name: 'Refined Steel Car',

product\_price: 690,

product\_material: 'Rubber',

product\_color: 'gold'

}

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

**Query** :

db.product.find({}, {'\_id':0, 'product\_name': 1, 'product\_material': 1 });

**Query Output:**

**Retrieve only the product name and material for each product.**

{

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: 'Handcrafted Wooden Bacon',

product\_material: 'Concrete'

}

**............25 Products**

**6.Find the product with a row id of 10**

**Query** :

db.product.find().skip(9).limit(1);

**Query Output:**

**Retrieve the product details with an ID of 10.**

{

\_id: ObjectId('65e37384a9e214593453882a'),

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**

Query : db.product.find({}, {'\_id':0, 'product\_name': 1, 'product\_material': 1 });

**Query Output:**

**Fetch the product name and material details for all products.**

{

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: 'Handcrafted Wooden Bacon',

product\_material: 'Concrete'

}

**............25 Products**

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

Query : db.product.find({ 'product\_material': 'Soft' } });

**Query Output:**

**Retrieve products where the material contains the value "Soft".**

{

\_id: ObjectId('65e37384a9e2145934538824'),

id: '4',

product\_name: 'Gorgeous Plastic Pants',

product\_price: 492,

product\_material: 'Soft',

product\_color: 'plum'

}

{

\_id: ObjectId('65e37384a9e2145934538829'),

id: '9',

product\_name: 'Awesome Wooden Ball',

product\_price: 28,

product\_material: 'Soft',

product\_color: 'azure'

}

{

\_id: ObjectId('65e37384a9e214593453882b'),

id: '11',

product\_name: 'Unbranded Wooden Cheese',

product\_price: 26,

product\_material: 'Soft',

product\_color: 'black'

}

{

\_id: ObjectId('65e37384a9e2145934538833'),

id: '19',

product\_name: 'Intelligent Cotton Chips',

product\_price: 46,

product\_material: 'Soft',

product\_color: 'azure'

}

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

Query : db.product.find({ 'product\_color': 'indigo' , 'product\_price': 492.00 });

**Query Output:**

Fetch products with a color of "indigo" and a price of 492.00.

**10.Delete the products which product price value are same**

Query : let Prices = db.product.distinct("product\_price");

Prices.forEach(price => {

let product = db.product.find({ product\_price: price }).sort({ \_id: 1 }).skip(1);

product.forEach(doc => {

db.product.deleteOne({ \_id: doc.\_id });

});

});

**Query Output:**

Delete products with identical price values, keeping only one instance of each unique price.