# MongoDB Queries

## -MANIKANDAN R

**Database Name: MongoDB**

**Collection Name: QueriesTask**

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

**db.QueriesTask.find()**

**find() will return all documents present in the collection**

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

**db.QueriesTask.find({$and:[**

**{product\_price:{$gte:400}},**

**{product\_price:{$lte:800}}**

**]})**

**$and logical operator return the documents which satisfies**

**Both the specified condition**

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

**db.QueriesTask.find({$or:[**

**{product\_price:{$lte:400}},**

**{product\_price:{$gte:600}}**

**]})**

**Used $or operator to display docs having price either less than**

**400 or greater than 600.**

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

**db.QueriesTask.find({product\_price:{$gte:500}}).limit(4)**

**This returns 4 docs with product price greater than 500**

**Using $gte operator and limit function**

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

**db.QueriesTask.find(**

**{},product\_name:1,product\_material:1}**

**)**

**Used Projection to get the specified fields in the doc**

**By setting the value to 1**

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

**db.QueriesTask.find({id:"10"})**

**filtered the doc which has id 10**

1. **Find only the product name and product material**

**db.QueriesTask.find(**

**{},{product\_name:1,product\_material:1,\_id:0}**

**)**

**Projection by default returns the id object, so to avoid it**

**Set id to 0**

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

**db.QueriesTask.find({product\_material:"Soft"})**

**filtered the docs having product material as soft.**

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

**db.QueriesTask.find({$or:[**

**{product\_color:"indigo"},**

**{product\_price:492}**

**]})**

**Used $or operator to get the docs having product color as**

**Indigo and product price as 492.00**

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

**db.QueriesTask.aggregate([**

**{$group:{\_id:"$product\_price",count:{$sum:1}}},**

**{$match:{count:{$gt:1}}}**

**]).forEach(function(e){**

**db.QueriesTask.deleteMany({product\_price: e.\_id})**

**})**