

Lab 8 – Week 11

Paras Singh: 165-114-232

Tasks

1. Write a query to return *name* and *price* of each product in the *inventory* database.

```
inventory> db.products.find({}, {name: 1, price: 1, _id: 0})
[
  { name: 'AC3 Phone', price: 200 },
  { name: 'AC7 Phone', price: 320 },
  { name: 'AC3 Series Charger', price: 19 },
  { name: 'AC3 Case Green', price: 12 },
  { name: 'Phone Extended Warranty', price: 38 },
  { name: 'AC3 Case Black', price: 12.5 },
  { name: 'AC3 Case Red', price: 12 },
  { name: 'Phone Service Basic Plan' },
  { name: 'Phone Service Core Plan' },
  { name: 'Phone Service Family Plan' },
  { name: 'Cable TV Basic Service Package' }
]
inventory>
```

2. Write a query to return *name* and *price* for products of type *accessory* in the *inventory* database.

```
inventory> db.products.find({type: "accessory"}, {name: 1, price: 1, _id: 0})
[
  { name: 'AC3 Series Charger', price: 19 },
  { name: 'AC3 Case Green', price: 12 },
  { name: 'AC3 Case Black', price: 12.5 },
  { name: 'AC3 Case Red', price: 12 }
]
inventory>
```

3. Write a query to return *name* and *price* for products with price between \$12 and \$20 (Values 12 and 20 are included).

```
inventory> db.products.find({ price: { $gte: 12, $lte: 20 } }, { name: 1, price: 1, _id: 0 })
[
  { name: 'AC3 Series Charger', price: 19 },
  { name: 'AC3 Case Green', price: 12 },
  { name: 'AC3 Case Black', price: 12.5 },
  { name: 'AC3 Case Red', price: 12 }
]
inventory>
```

4. Write a query to return *id*, *name*, *price*, and *type* for products that are not of type *accessory*.

```
inventory> db.products.find({ type: { $ne: "accessory" } }, { _id: 1, name: 1, price: 1, type: 1 })
[
  { _id: 'ac3', name: 'AC3 Phone', type: 'phone', price: 200 },
  { _id: 'ac7', name: 'AC7 Phone', type: 'phone', price: 320 },
  {
    _id: ObjectId('507d95d5719dbef170f15bfb'),
    name: 'Phone Extended Warranty',
    type: 'warranty',
    price: 38
  },
  {
    _id: ObjectId('507d95d5719dbef170f15bfe'),
    name: 'Phone Service Basic Plan',
    type: 'service'
  },
  {
    _id: ObjectId('507d95d5719dbef170f15bff'),
    name: 'Phone Service Core Plan',
    type: 'service'
  },
  {
    _id: ObjectId('507d95d5719dbef170f15c00'),
    name: 'Phone Service Family Plan',
    type: 'service'
  },
  {
    _id: ObjectId('507d95d5719dbef170f15c01'),
    name: 'Cable TV Basic Service Package',
    type: 'tv'
  }
]
inventory>
```

5. Write a query to return *id*, *name*, *price*, and *type* for products with type *accessory* or *service*.

```
inventory> db.products.find({ type: { $in: ["accessory", "service"] } }, { _id: 1, name: 1, price: 1, type: 1 })
[
  {
    _id: ObjectId('507d95d5719dbef170f15bf9'),
    name: 'AC3 Series Charger',
    type: [ 'accessory', 'charger' ],
    price: 19
  },
  {
    _id: ObjectId('507d95d5719dbef170f15bfa'),
    name: 'AC3 Case Green',
    type: [ 'accessory', 'case' ],
    price: 12
  },
  {
    _id: ObjectId('507d95d5719dbef170f15bfc'),
    name: 'AC3 Case Black',
    type: [ 'accessory', 'case' ],
    price: 12.5
  },
  {
    _id: ObjectId('507d95d5719dbef170f15bfd'),
    name: 'AC3 Case Red',
    type: [ 'accessory', 'case' ],
    price: 12
  },
  {
    _id: ObjectId('507d95d5719dbef170f15bfe'),
    name: 'Phone Service Basic Plan',
    type: 'service'
  },
  {
    _id: ObjectId('507d95d5719dbef170f15bff'),
    name: 'Phone Service Core Plan',
    type: 'service'
  },
  {
    _id: ObjectId('507d95d5719dbef170f15c00'),
    name: 'Phone Service Family Plan',
    type: 'service'
  }
]
inventory>
```

6. Write a query to return *id*, *name*, *price*, and *type* for products that do have the *type* key.

```

inventory> db.products.find({ type: { $exists: true } }, { _id: 1, name: 1, price: 1, type: 1 })
[
  { _id: 'ac3', name: 'AC3 Phone', type: 'phone', price: 200 },
  { _id: 'ac7', name: 'AC7 Phone', type: 'phone', price: 320 },
  {
    _id: ObjectId('507d95d5719dbef170f15bf9'),
    name: 'AC3 Series Charger',
    type: [ 'accessory', 'charger' ],
    price: 19
  },
  {
    _id: ObjectId('507d95d5719dbef170f15bfa'),
    name: 'AC3 Case Green',
    type: [ 'accessory', 'case' ],
    price: 12
  },
  {
    _id: ObjectId('507d95d5719dbef170f15bfb'),
    name: 'Phone Extended Warranty',
    type: 'warranty',
    price: 38
  },
  {
    _id: ObjectId('507d95d5719dbef170f15bfc'),
    name: 'AC3 Case Black',
    type: [ 'accessory', 'case' ],
    price: 12.5
  },
  {
    _id: ObjectId('507d95d5719dbef170f15bfd'),
    name: 'AC3 Case Red',
    type: [ 'accessory', 'case' ],
    price: 12
  },
  {
    _id: ObjectId('507d95d5719dbef170f15bfe'),
    name: 'Phone Service Basic Plan',
    type: 'service'
  },
  {
    _id: ObjectId('507d95d5719dbef170f15bff'),
    name: 'Phone Service Core Plan',
    type: 'service'
  },
  {
    _id: ObjectId('507d95d5719dbef170f15c00'),
    name: 'Phone Service Family Plan',
    type: 'service'
  },
  {
    _id: ObjectId('507d95d5719dbef170f15c01'),
    name: 'Cable TV Basic Service Package',
    type: 'tv'
  }
]

```

7. Write a query to return *id*, *name*, *price*, and *type* for products that their type is both *accessory* and *case*.

```
inventory> db.products.find({ type: { $all: ["accessory", "case"] } }, { _id: 1, name: 1, price: 1, type: 1 })
[
  {
    _id: ObjectId('507d95d5719dbef170f15bfa'),
    name: 'AC3 Case Green',
    type: [ 'accessory', 'case' ],
    price: 12
  },
  {
    _id: ObjectId('507d95d5719dbef170f15bfc'),
    name: 'AC3 Case Black',
    type: [ 'accessory', 'case' ],
    price: 12.5
  },
  {
    _id: ObjectId('507d95d5719dbef170f15bfd'),
    name: 'AC3 Case Red',
    type: [ 'accessory', 'case' ],
    price: 12
  }
]
inventory>
```

Good luck.