

—: LAB-02:—

Q1 Creating a Collection by name Customers with

- ' Cust-id
- ' Acc-Bal
- ' Acc-Type

Soln

② Inserting 5 values into the tables:

```
db.Customers.insertMany([
  { Cust-id: 1, Acc-Bal: 1500, Acc-Type: 'Z' },
  { Cust-id: 2, Acc-Bal: 500, Acc-Type: 'Z' },
  { Cust-id: 3, Acc-Bal: 1800, Acc-Type: 'X' },
  { Cust-id: 4, Acc-Bal: 2500, Acc-Type: 'Z' },
  { Cust-id: 5, Acc-Bal: 1200, Acc-Type: 'Z' }
]);
```

O/P

```
{
  acknowledged: true,
}
```

③ db.Customers.find({ Acc-Bal: { \$gt: 1200 },
Acc-Type: 'Z' });

```
{
  - id: ObjectId("
    Cust-id: 1
    Acc-Bal: 1500,
    Acc-Type: 'Z'
  },
```


db.products.insertMany(C[


```
{  
  product_id: "P001",  
  name: "Laptop",  
  Category: "Electronics",  
  price: 999.99,  
  quantity: 50,  
  description: "High-end gaming laptop"  
}
```

```
{  
  product_id: "P002",  
  name: "Headphones",  
  Category: "Electronics",  
  price: 199.99,  
  quantity: 100,  
  description: "Noise-cancelling headphones"  
}  
}]
```

O/P

```
{  
  acknowledged: true  
}
```

```
db.createCollection("Users")  
→ {ok: 1}
```

```
db.Users.insertOne({
```

```
  username: "John-doe",  
  password: "hashed password 123",  
  email: "john.doe@example.com",  
})
```



```
phone-number: "123-456-7890"
```

```
}
)
```

O/P

```
{
```

```
  acknowledged: true
}
```

```
db.createCollection("carts")
```

```
{ok: 1}
```

```
db.carts.insertOne({
```

```
  user-id: ObjectId("User-id-here"),
```

```
  products: [
```

```
    {
```

```
      product-id: ObjectId("product-id-here"),
```

```
      quantity: 2,
```

```
      price-at-time: 999.99
```

```
    }
  ]
})
```

```
db.createCollection("orders")
```

```
{ok: 1}
```

```
db.orders.insertOne({
```

```
  user-id: ObjectId("User-id"),
```

```
  order-status: "Pending"
```

```
},
```

```
  products: [
```

```
    {
```



```
product_id: ObjectId("Id"),  
  quantity: 1,  
  price_at_time: 999.99  
},  
  total_price: 999.99,  
}
```

Retrieving All products

⇒

```
db.products.find({})
```

```
{  
  _id: ObjectId(" "),  
  product_id: 'P001',  
  name: 'Laptop',  
  category: 'Electronics',  
  price: 999.99,  
  description: 'High-end gaming laptop'  
},
```

```
{  
  _id: ObjectId(" "),  
  product_id: 'P002',  
  name: 'Headphones',  
  category: 'Electronics',  
  price: 199.99,  
  quantity: 100  
}
```

]

* Retrieve Products in a specific Category

```
db.products.find({category: "Electronics"})
[
  {
    _id: ObjectId("PP.PP"),
    product_id: 'P001',
    name: 'Laptop',
    category: 'Electronics',
    price: 999.99,
    description: 'High-end gaming laptop'
  },
```

```
{
  _id: ObjectId("PP.PP"),
  product_id: 'P002',
  name: 'Headphones',
  category: 'Electronics',
  price: 199.99,
  quantity: 100
}]
```

* Products with Quantity Greater Than 0

```
db.products.find({"quantity": { $gt: 0 }})
[
  {
    _id: ObjectId("PP.PP"),
    product_id: 'P001',
    name: 'Laptop',
    category: 'Electronics',
    price: 999.99,
  },
```



```
{
  id: ObjectId("..."),
  product_id: 'P002',
  name: 'Headphones',
  category: 'Electronics',
  price: 199.99,
  quantity: 100
}
```

★ Retrieve Products with Price less Than or Equal to \$100

§12

```
db.products.find({ "price": { $lte: 100 } })
```

None

★ Retrieve Products Added to a User's Cart

⇒

```
db.users.aggregate([
  { $match: { "_id": ObjectId("...") } },
  { $unwind: "$cart" },
  { $project: { "product_details": 1,
    "cart.quantity": 1 } }
])
```

★ Retrieve orders placed by a User

⇒

```
db.users.aggregate([
  { $match: { "_id": ObjectId("123abc...") } },
  { $unwind: "$orders" },
  { $project: { "order_details": 1,
    "order.quantity": 1 } }
])
```



```
{ $project: { "order-details": 1 } }
})
```

★ Retrieve total price of orders
→

```
db.orders.aggregate([
  { $match: { "id": ObjectId("123abc...") } },
  { $group: {
    _id: null,
    total-order-price: { $sum: "$order.total-price" }
  } }
])
```

★ Aggregation Queries:

01 db.products.aggregate([
 { \$group: { _id: "\$category", total-products: { \$sum: 1 } } }
])

02 db.products.aggregate([
 { \$group: { _id: "\$category", total-price: { \$sum: { \$multiply: ["\$price", "\$quantity"] } } }
])

03 db.products.aggregate([
 { \$group: { _id: null, average-price: { \$avg: "\$price" } } }
])

04.

```
db.products.find({"quantity": {"$lt": 10}})
```

05.

```
db.products.find({}).sort({"price": -1})
```

06.

```
db.orders.aggregate([
  {"$unwind": "$products"},
  {"$group": {
    "_id": "$user_id",
    "total-order-price": {"$sum": {"$multiply":
      ["$products.price", "$product.quantity"]}}
  }}
])
```

07.

```
db.orders.aggregate([
  {"$unwind": "$product"},
  {"$group": {
    "_id": "$user_id",
    "total-order-price": {"$sum": {"$multiply":
      ["$product.price", "$product.quantity"]}}
  }}
])
```

08.

```
db.orders.aggregate([
  {"$group": {"_id": null, "average-order-price":
    {"$avg": "$total-price"}}
})
```

88
11/12/20