

1) Create customers collection:

```
db.createCollection("customers");
```

```
db.customers.insertMany([
```

```
{cust_id: 1, accBal: 1000, type: "c"},
```

```
{cust_id: 1, accBal: 200, type: "c"},
```

```
{cust_id: 2, accBal: 1000, type: "c"},
```

```
{cust_id: 2, accBal: 100, type: "c"},
```

```
{cust_id: 2, accBal: 1000, type: "c"},
```

```
{cust_id: 3, accBal: 1000, type: "c"}]);
```

2)

```
db.customers.aggregate([
  { $match: { type: "c" } },
  { $group: { _id: "$cust_id", totaccBal: { $sum: "$accBal" } } },
  { $match: { totaccBal: { $gt: 1200 } } }
]);
```

```
[ { _id: 2, totaccBal: 1500 } ]
```

3)

```
db.customers.aggregate([
  { $group: { _id: "$cust_id", minaccBal: { $min: "$accBal" }, maxaccBal: { $max: "$accBal" } } },
  [ { _id: 1, minaccBal: 800, maxaccBal: 1000 },
    { _id: 2, minaccBal: 100, maxaccBal: 1000 },
    { _id: 3, minaccBal: 500, maxaccBal: 1000 } ]
]);
```


Create products and users collection.

```
db.createCollection("products");
```

```
db.products.insertMany([
```

```
{ p_id: "P01", name: "Iphone 16", category:
  "electronics", price: 299.99, quantity: 50 },
{ p_id: "P02", name: "HP Laptop", category:
  "electronics", price: 799.99, quantity: 30 },
{ p_id: "P03", name: "Beats airpods", category:
  "Accessories", price: 59.99, quantity: 150 },
{ p_id: "P04", name: "Gaming chair", category:
  "Furniture", price: 199.99, quantity: 20 },
{ p_id: "P05", name: "Bluetooth speaker",
  price: 59.99, category: "Accessories",
  quantity: 100 }
])
```

→ Retrieve All products.

```
> db.products.find();
```

→ Retrieve products in specific category

```
> db.products.find({category: "Electronics"});
```

```
{ p_id: 'P01', name: 'Iphone 16', category:
  'electronics', quantity: 50, price: 299.99
},
```

```
{ p_id: 'P02', name: 'HP Laptop', category:
  'electronics', price: 799.99, quantity: 30
}
```

```
[
```


→ Retrieve Products with quantity greater than 0

> db.products.find({'quantity': {'\$gt': 0}})

→ Retrieve Products Sorted by Price in Ascending Order

db.products.find().sort({'price': 1})

→ Retrieve products with price less than or equal to \$100

db.products.find({'price': {'\$lte': 100}})

[{'_id': 'P03', 'name': 'Beats Airpods', 'category': 'Accessories', 'price': 49.99, 'quantity': 150},

{ '_id': 'P05', 'name': 'Bluetooth Speaker', 'category': 'Accessories', 'price': 59.99, 'quantity': 100 }

]

db.createCollection("users")

> db.users.insertMany([

{ 'user_id': '123abc',
 'cart': [{ 'p_id': 'P01', 'quantity': 2 },
 { 'p_id': 'P03', 'quantity': 1 }],
 'orders': [{ 'order_id': '01',

products [\$ p id: 'P01', quantity: 1 \$,
\$ p id: 'P03', quantity: 2 \$],
total price: 399.99,
\$]]

\$ user id: '789ghi',
cart: [\$ p id: 'P02', quantity: 1 \$,
\$ p id: 'P04', quantity: 1 \$],
orders: [\$ order id: 'O2', products: [\$ p id: 'P02', quantity: 1 \$],
total price: 799.99,
\$]]]]

→ Retrieve Products Added to User's Cart

db.users.aggregate([\$ match: \$ user id: "789ghi",
\$ unwind: "\$cart", \$ lookup: \$ from: "products",
localField: "cart.product_id", foreignField: "product_id", as: "product_details"],
\$ project: \$ "products_details.name": 1,
"products_details.price": 1, "cart.quantity": 1
\$]] ;

→ Retrieve Orders placed by a User

db.users.find(\$ user id: "123abc", \$ "orders")

→ Retrieve Total Price of Orders placed by a User

db.users.aggregate([\$ match: \$ user id: "123abc",
\$ unwind: "\$orders",
\$ group: \$ _id: "\$user_id",
total order price: \$ sum: "\$orders.totalprice"
\$]])

→ Retrieve Total Price of Orders Placed by User.

db.products.aggregate([
 { \$group: { \$id: "\$category", total_products: { \$sum: 13333 } } }
])

S. Pradus

Additional Aggregation.

1) db.products.aggregate([
 { \$group: { \$id: "\$category", total_products: { \$sum: 13333 } } }
])

2) db.products.aggregate([
 { \$group: { \$id: "\$category", total_price: { \$sum: { \$multiply: ["\$price", "\$quantity"] } } } }
])

3) db.products.aggregate([
 { \$group: { \$id: null, avg_price: { \$avg: "\$price" } } }
])

4) db.products.find({ \$quantity: { \$lt: 10 } })

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

6) db.orders.aggregate([
 { \$summarize: { \$orders: "\$" } }
 { \$group: { \$id: "\$user_id", total_order_price: { \$sum: "\$orders" } } }
 { \$project: { "totalprice": "\$total_order_price" } }
])

7. db.users.aggregate([
 \$ unwind: '\$orders',
 \$ sort: '\$total_order_price',
 \$ limit: 1
]).

8. db.users.aggregate([
 \$ unwind: '\$orders',
 \$ group: { \$ _id: '\$user_id', total_order_price: \$ sum: '\$orders.total_price' },
 \$ group: { \$ _id: null, avg_total_price: \$ avg: '\$total_order_price' }
]).

SB