

```
Last login: Wed Jun 11 16:09:10 from 172.30.1.18
mini~ mongosh
Current Mongosh Log ID: 68495dabf16168df8fc4dc69
Connecting to:      mongodb://127.0.0.1:27017/?
directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.2
Using MongoDB:      8.0.10
Using Mongosh:       2.5.2
```

For mongosh info see: <https://www.mongodb.com/docs/mongodb-shell/>

```
-----
The server generated these startup warnings when booting
2025-06-10T10:57:40.591+09:00: Access control is not enabled for the
database. Read and write access to data and configuration is unrestricted
-----
```

```
test> use lab02
switched to db lab02
lab02> db
lab02
lab02> show dbs
admin      40.00 KiB
config     60.00 KiB
lab01      72.00 KiB
lab02      40.00 KiB
local      40.00 KiB
mydb       40.00 KiB
myemp      96.00 KiB
test       40.00 KiB
lab02> print("Q1 ) 모든 제품 중에서 가장 높은 가격을 찾아서 리턴하자.")
Q1 ) 모든 제품 중에서 가장 높은 가격을 찾아서 리턴하자.
```

```
lab02> show collections
Product
lab02> db.Product.aggregate([{$group: {_id: null, maxprice: {$max: $Price}}},
{$project: {_id: 0, max_price: 1}}]);
ReferenceError: $Price is not defined
lab02> db.Product.aggregate([{$group: {_id: null, maxprice: {$max: "$Price"}}},
{$project: {_id: 0, max_price: 1}}]);
[ { } ]
lab02> db.Product.findOne()
{
  _id: ObjectId('6849338e1098d5ed94ba6179'),
  Name: 'notebook',
  Price: 200,
  Category: 'material'
}
lab02> db.Product.aggregate([{$group: {_id: null, max_price:
{$max: "$Price"}}}, {$project: {_id: 0, max_price: 1}}]);
[ { max_price: 220 } ]
lab02> print("Q2) 가격이 100보다 큰 상품의 개수를 출력 하자.")
Q2) 가격이 100보다 큰 상품의 개수를 출력 하자.
```

```

lab02> db.Product.aggregate([{$match:{Price:{$gte:100}}},
{$count:"count"}]);
[ { count: 3 } ]
lab02> db.Product.aggregate([{$match:{Name:{$regex:/^S/i}}},{ $group:
{_id:null,total_price:{$sum:"$Price"}}},{ $project:
{_id:0,total_price:1}}]);
[ { total_price: 220 } ]
lab02> db.Product.find({},{_id:0});
[
  { Name: 'notebook', Price: 200, Category: 'material' },
  { Name: 'pencil', Price: 80, Category: 'material' },
  { Name: 'salad', Price: 220, Category: 'food' },
  { Name: 'others', Price: 20, Category: 'material' },
  { Name: 'bread', Price: 100, Category: 'food' }
]
lab02> print("Q4) Category의 material의 평균 가격을 출력하자.")
Q4) Category의 material의 평균 가격을 출력하자.

lab02> db.Product.aggregate([{$match:{Category:"material"}},{ $group:
{_id:null,avg_price:{$avg:"$Price"}}},{ $project:{_id:0,avg_price:1}}]);
[ { avg_price: 100 } ]
lab02> (200+80+20)/3
100

lab02> ( "Q5) material 있는 모든 제품의 총 가격을 계산하되 가격이 50보다 큰 제품만 포함한
다. 또한 제품 가격이 150보다 큰 경우 가격에 10% 할인을 포함한다." );
Q5) material 있는 모든 제품의 총 가격을 계산하되 가격이 50보다 큰 제품만 포함한다. 또한 제품
가격이 150보다 큰 경우 가격에 10% 할인을 포함한다.
lab02> print("> $match, $group $sum $cond( $gt $multiply) 사용")
> $match, $group $sum $cond( $gt $multiply) 사용

lab02> edit
Opening an editor...

lab02> db.Product.aggregate([ { $match: { Category: "material", Price: {
$gte: 50 } } }, { $group: { _id:0, total_price: { $sum: { $cond: { if: { $gt: ["$Price", 150] }, then: {
$multiply: ["$Price", 0.9] }, else: "$Price" } } } } }, { $project: { _id:
0, total_price: 1 } } ] );
[ { total_price: 260 } ]
lab02> db.Product.find({},{_id:0})
[
  { Name: 'notebook', Price: 200, Category: 'material' },
  { Name: 'pencil', Price: 80, Category: 'material' },
  { Name: 'salad', Price: 220, Category: 'food' },
  { Name: 'others', Price: 20, Category: 'material' },
  { Name: 'bread', Price: 100, Category: 'food' }
]
lab02> (80+(200*0.9))
260
lab02> exit
mini> ~

```

