

- Q1) 모든 제품 중에서 가장 높은 가격을 찾아서 리턴 하자.

```
mydb> db.Product.aggregate([ { $group: { _id: null, highestPrice: { $max: "$Price" } } } ] )
```

```
[ { _id: null, highestPrice: 220 } ]
```

```
mydb> db.Product.aggregate([
...   {
...     $group: {
...       _id: null,
...       highestPrice: { $max: "$Price" }
...     }
...   }
... ])
[ { _id: null, highestPrice: 220 } ]
```

- Q2) 가격이 100보다 큰 상품의 개수를 출력하자.

```
mydb> db.Product.aggregate([ { $group: { _id: null, count: { $sum: { $cond: [ { $gt:
["$Price", 100] }, 1, 0 ] } } } ] )
```

```
[ { _id: null, count: 2 } ]
```

```
mydb> db.Product.aggregate([
...   {
...     $group: {
...       _id: null,
...       count: {
...         $sum: {
...           $cond: [ { $gt: [ "$Price", 100 ] }, 1, 0 ]
...         }
...       }
...     }
...   }
... ])
[ { _id: null, count: 2 } ]
```

- Q3) name이 문자 S로 시작하는 제품의 총 가격을 계산하자.

```
mydb> db.Product.aggregate([ { $match: { Name: { $regex: /^S/i } } }, { $group: { _id:
null, totalPrice: { $sum: "$Price" } } } ] )
```

```
[ { _id: null, totalPrice: 220 } ]
```

```

mydb> db.Product.aggregate([
...   {
...     $match: { Name: { $regex: /^S/i } }
...   },
...   {
...     $group: {
...       _id: null,
...       totalPrice: { $sum: "$Price" }
...     }
...   }
... ])
[ { _id: null, totalPrice: 220 } ]

```

- Q4) Category의 material의 평균 가격을 출력하자.

```

mydb> db.Product.aggregate([ { $match: { Category: "material" } }, { $group: { _id:
"$Category", averagePrice: { $avg: "$Price" } } } ] )

```

```

[ { _id: 'material', averagePrice: 100 } ]

```

```

mydb> db.Product.aggregate([
...   {
...     $match: { Category: "material" }
...   },
...   {
...     $group: {
...       _id: "$Category",
...       averagePrice: { $avg: "$Price" }
...     }
...   }
... ])
[ { _id: 'material', averagePrice: 100 } ]

```

- Q5) material 있는 모든 제품의 총 가격을 계산하되 가격이 50보다 큰 제품만 포함한다. 또한 제품 가격이 150보다 큰 경우 총 가격에 10% 할인을 포함한다.

```
mydb> db.Product.aggregate([ { $match: { Category: "material", Price: { $gt: 50 } } },  
{ $group: { _id: null, totalPrice: { $sum: { $cond: [ { $gt: ["$Price", 150] }, { $multiply:  
["$Price", 0.9] }, /* Apply 10% discount*/ "$Price" ] } } } ] )
```

```
[ { _id: null, totalPrice: 260 } ]
```

```
mydb> db.Product.aggregate([  
...   {  
...     $match: { Category: "material", Price: { $gt: 50 } }  
...   },  
...   {  
...     $group: {  
...       _id: null,  
...       totalPrice: {  
...         $sum: {  
...           $cond: [  
...             { $gt: ["$Price", 150] },  
...             { $multiply: ["$Price", 0.9] }, // Apply 10% discount  
...             "$Price"  
...           ]  
...         }  
...       }  
...     }  
...   ]  
... ] )  
[ { _id: null, totalPrice: 260 } ]
```