Counting Number of Row

```
db.employees.aggregate([

        {$count: 'toatal'}

])
```

Sorting, Skipping & Selecting

```
db.employees.aggregate([

{$sort:{salary:-1}}

])
```

```
db.employees.aggregate([

{$sort: {_id:-1}},
{$limit:2}

])
```

```
db.employees.aggregate([

{$sort:{_id:-1}},
{$skip:2},
{$limit:2}

])

//Same code
db.employees.find({}).sort({"_id":-1}).skip(2).limit(2)
```

Find data using `$match` condition

```
db.employees.aggregate([

 {$match: {salary:{$lt:90000}}}

])
```

Using `$match` with multiple condition

```
db.employees.aggregate([

{$match:{salary:{$lt:100000}}},
{$match:{city:{$nin:["Dhaka"]}}}

])
```

```
db.employees.aggregate([

{$match:{salary:{$lte:100000}}},
{$match:{city:"Dhaka"}}

])

//Same Code
db.employees.aggregate([

{$match:{$and:[
                {salary:{$lte:100000}},
                {city:"Dhaka"}
        ]}}
])
```

Select alike data

```
db.employees.aggregate([
        {$match:{designation:/Scientist/}}
])

//Same code
db.employees.find({designation:/Eng/})
```

Projection the data by Column

```
db.employees.aggregate([

        {$project:{_id:0, name:1}}

])
```

Show data by Group

Show only unique data

```
db.employees.aggregate([

        {$group:{_id:"$city"}}

        //In here, _id is not that you think. This is the property of $group

])

//Same code
db.employees.distinct('city')
```

show sum by grouping

```
db.employees.aggregate([

        {$group:{_id:"$designation", total:{$sum:"$salary"}}}

])
```

show average by grouping

```
db.employees.aggregate([

        {$group:{_id:"$designation", avg:{$avg:"$salary"}}}

])
```

show max value by grouping

```
db.employees.aggregate([

        {$group:{_id:"$designation", max:{$max:"$salary"}}}

])
```

show min value by grouping

```
db.employees.aggregate([

        {$group:{_id:"$designation", min:{$min:"$salary"}}}

])
```

combine group

```
db.employees.aggregate([
{$group:
         {
                  _id:{city: "$city"},
                  avg:{$avg:"$salary"},
                  sum:{$sum:"$salary"},
                  max:{$max:"$salary"},
                  min:{$min:"$salary"}
        }}
])
```

```
db.employees.aggregate([
{$group:
        {
                _id:{designation: "$designation", city: "$city"},
                avg:{$avg:"$salary"},
                sum:{$sum:"$salary"},
                max:{$max:"$salary"},
                min:{$min:"$salary"}
        }}
])
```

Show Max, Min & Avg value of a column

- Max value

```
db.employees.aggregate([

        {$group:{_id:0, max:{$max:"$salary"}}},

])
```

- Max value

```
db.employees.aggregate([

        {$group:{_id:0, min:{$min:"$salary"}}},

])
```

- Avg value

```
db.employees.aggregate([

        {$group:{_id:0, avg:{$avg:"$salary"}}},

])
```

Add a new column & Set value with condition

- Add a column and the values of this column depend on the value of the price

```
db.products.aggregate([

{
        $addFields:{
                something:{
                        $gte:[{$toDouble:"$price"} ,12000]
                }
        }
},

{$match:{"something" : false}}

])
```

- Add a column and the values of this column depend on the value of the price and city.
- if both condition is true then it will be true.

```
db.employees.aggregate([

{
    $addFields: {
        something: {
            $and: [
                { $gt: ["$salary", 80000] },
                { $eq: ["$city", "Dhaka"] }
            ]
        }
    }
}
,

{$match:{"something" : true}}

])
```

```
db.employees.aggregate([
    {
        $addFields: {
            value: {
                $switch: {
                    branches: [
                        {
                            case: { $and: [
                                                                { $gt:
["$salary", 80000] },
                                                                { $eq:
["$city", "Dhaka"] }
                                                            ]
                                                        },
                            then: 30
                        },
                        {
                            case: { $gt: ["$salary", 60000] },
                            then: 20
                        },
                        {
                            case: { $eq: ["$city", "Mumbai"] },
                            then: 10
                        }
                    ],

                    default: 0
                        // Default value if no conditions match
                }
            }
        }
    }
]);
```