

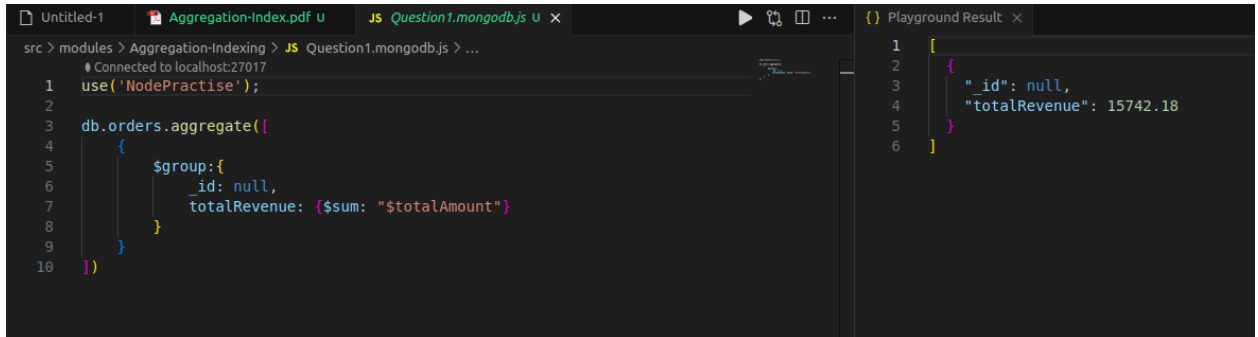
# Aggregation and Indexing Assignment

Sample Json-

```
{
  "_id": ObjectId,
  "orderId": String,
  "customerName": String,
  "orderDate": Date,
  "status": String, // e.g., "Pending", "Shipped", "Delivered"
  "items": [
    { "productName": String, "quantity": Number, "price": Number }
  ],
  "totalAmount": Number
}
```

Aggregation Queries Results( Also implemented on Mongo Compass )

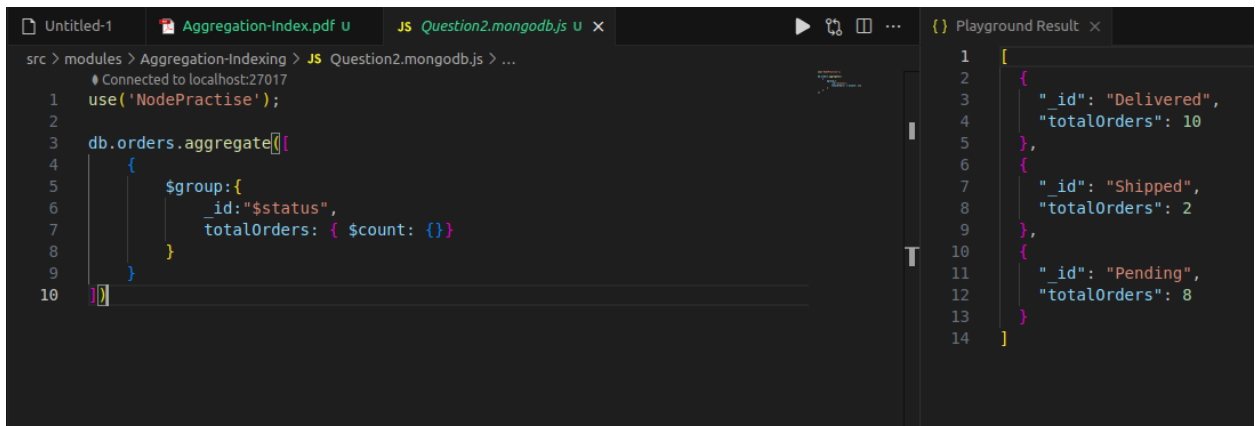
1.



```
src > modules > Aggregation-Indexing > JS Question1.mongodbs.js > ...
# Connected to localhost:27017
1 use('NodePractise');
2
3 db.orders.aggregate([
4   {
5     $group:{
6       _id: null,
7       totalRevenue: {$sum: "$totalAmount"}
8     }
9   }
10 ])

1 [
2   {
3     "_id": null,
4     "totalRevenue": 15742.18
5   }
6 ]
```

2



```
src > modules > Aggregation-Indexing > JS Question2.mongodbs.js > ...
# Connected to localhost:27017
1 use('NodePractise');
2
3 db.orders.aggregate([
4   {
5     $group:{
6       _id:"$status",
7       totalOrders: { $count: {} }
8     }
9   }
10 ])

1 [
2   {
3     "_id": "Delivered",
4     "totalOrders": 10
5   },
6   {
7     "_id": "Shipped",
8     "totalOrders": 2
9   },
10  {
11    "_id": "Pending",
12    "totalOrders": 8
13  }
14 ]
```

3

```

src > modules > Aggregation-Indexing > JS Question3.mongodb.js > ...
# Connected to localhost:27017
1 use('NodePractise');
2
3 db.orders.aggregate([
4   {
5     $group: {
6       _id: "$customerName",
7       totalSpent: { $sum: "$totalAmount" }
8     }
9   },
10  {
11    $sort: { totalSpent: -1 }
12  },
13  {
14    $limit: 3
15  }
16 ],
17 )

```

```

1 [
2   {
3     "_id": "Antonia Hessel",
4     "totalSpent": 1608.88
5   },
6   {
7     "_id": "Edith Lesch",
8     "totalSpent": 1400.66
9   },
10  {
11    "_id": "Johanna Gulgowski",
12    "totalSpent": 1307.64
13  }
14 ]

```

4

```

src > modules > Aggregation-Indexing > JS Question4.mongodb.js > ...
# Connected to localhost:27017
1 use('NodePractise');
2
3 db.orders.aggregate([
4   {
5     $group: {
6       _id: "$customerName",
7       averageTotal: { $avg: "$totalAmount" }
8     }
9   },
10  {
11    $sort: { averageTotal: -1 }
12  },
13  {
14    $limit: 10
15  }
16 ],
17 )

```

```

1 [
2   {
3     "_id": "Elmer Koelpin",
4     "averageTotal": 508.13
5   },
6   {
7     "_id": "Horace Smith",
8     "averageTotal": 559.06
9   },
10  {
11    "_id": "Terence Gutkowski",
12    "averageTotal": 1182.65
13  },
14  {
15    "_id": "Antonia Hessel",
16    "averageTotal": 1608.88
17  },
18  {
19    "_id": "Angelica Cole",
20    "averageTotal": 1101.02
21  },
22  {
23    "_id": "Jenny Glover",
24    "averageTotal": 287.82
25  },
26  {
27    "_id": "Alicia Connelly",
28    "averageTotal": 997.4
29  },
30  {
31    "_id": "Edith Lesch",
32    "averageTotal": 1400.66
33  },
34  {
35    "_id": "Elisa Ritchie",
36    "averageTotal": 1275.84
37  },
38  {
39    "_id": "Johanna Gulgowski",
40    "averageTotal": 1307.64
41  }
42 ]

```

5

```

src > modules > Aggregation-Indexing > JS Question5.mongodb.js > ...
# Connected to localhost:27017
1 use('NodePractise');
2
3 db.orders.aggregate([
4   {
5     $unwind: {
6       path: "$items",
7     }
8   },
9   {
10    $group: {
11      _id: "$items.productName",
12      quantity: {$sum: "$items.quantity"}
13    }
14  },
15  {
16    $match: {
17      quantity: {$gt: 10}
18    }
19  }
20 ])
21

```

```

1 [
2   {
3     "_id": "Bespoke Rubber Tuna",
4     "quantity": 16
5   }
6 ]

```

6

```

src > modules > Aggregation-Indexing > JS Question6.mongodb.js > ...
# Connected to localhost:27017
1 use('NodePractise');
2
3 db.orders.aggregate([
4   {
5     $match: {
6       orderDate: {
7         $gte: new Date(new Date().setMonth(new Date().getMonth() - 6))
8       }
9     }
10  },
11  {
12    $group: {
13      _id: {
14        year: { $year: "$orderDate" },
15        month: { $month: "$orderDate" }
16      },
17      monthlyRevenue: { $sum: "$totalAmount" }
18    }
19  },
20  {
21    $sort: {
22      "_id.year": -1,
23      "_id.month": -1
24    }
25  }
26 ])
27

```

```

1 [
2   {
3     "_id": {
4       "year": 2025,
5       "month": 7
6     },
7     "monthlyRevenue": 6085.1
8   },
9   {
10    "_id": {
11      "year": 2025,
12      "month": 6
13    },
14    "monthlyRevenue": 3957.4900000000002
15  },
16  {
17    "_id": {
18      "year": 2025,
19      "month": 5
20    },
21    "monthlyRevenue": 1400.66
22  },
23  {
24    "_id": {
25      "year": 2025,
26      "month": 4
27    },
28    "monthlyRevenue": 2074.94
29  },
30  {
31    "_id": {
32      "year": 2025,
33      "month": 3
34    },
35    "monthlyRevenue": 1960.74
36  },
37  {
38    "_id": {
39      "year": 2025,
40      "month": 2
41    },
42    "monthlyRevenue": 263.25
43  }
44 ]

```

7

```

src > modules > Aggregation-Indexing > JS Question7.mongodb.js > $match > totalOrders > Sgt
1 use('NodePractise');
2
3 db.orders.aggregate([
4   {
5     $group: {
6       _id: "$customerName",
7       totalOrders: {$sum: 1}
8     },
9   },
10  {
11    $match: {
12      totalOrders: {$gt: 0}
13    },
14  },
15  {
16    $project: {
17      _id: 0,
18      customerName: "$_id",
19      totalOrders: 1
20    }
21  }
22 ])

```

```

1 {
2   "totalOrders": 1,
3   "customerName": "Edith Lesch"
4 },
5 {
6   "totalOrders": 1,
7   "customerName": "Ramon Durgan"
8 },
9 {
10  "totalOrders": 1,
11  "customerName": "Raymond Ratke"
12 },
13 {
14  "totalOrders": 1,
15  "customerName": "Louise Hansen"
16 },
17 {
18  "totalOrders": 1,
19  "customerName": "Gilbert Yundt"
20 },
21 {
22  "totalOrders": 1,
23  "customerName": "Mr. Abraham Kerluke"
24 },
25 {
26  "totalOrders": 1,
27  "customerName": "Elmer Koelpin"
28 },
29 {
30  "totalOrders": 1,
31  "customerName": "Horace Smith"
32 },
33 {
34  "totalOrders": 1,
35  "customerName": "Raquel Turcotte"
36 },
37 }

```

8

```

src > modules > Aggregation-Indexing > JS Question8.mongodb.js > ...
1 use('NodePractise')
2
3 db.orders.aggregate([
4   {
5     $unwind: {
6       path: "$items",
7     },
8   },
9   {
10    $group: {
11      _id: "$items.productName",
12    },
13  },
14  {
15    $project: {
16      _id: 0,
17      productName: "$_id",
18    }
19  }
20 ])

```

```

1 {
2   "productName": "Elegant Cotton Chair"
3 },
4 {
5   "productName": "Modern Steel Shoes"
6 },
7 {
8   "productName": "Soft Granite Soap"
9 },
10 {
11  "productName": "Rustic Bamboo Chair"
12 },
13 {
14  "productName": "Tasty Plastic Keyboard"
15 },
16 {
17  "productName": "Intelligent Cotton Fish"
18 },
19 {
20  "productName": "Fresh Aluminum Fish"
21 },
22 {
23  "productName": "Awesome Aluminum Chair"
24 },
25 {
26  "productName": "Recycled Metal Salad"
27 },
28 {
29  "productName": "Tasty Metal Cheese"
30 },
31 {
32  "productName": "Refined Rubber Car"
33 },
34 {
35  "productName": "Ergonomic Granite Mouse"
36 },
37 {
38  "productName": "Incredible Marble Ball"
39 },
40 {
41  "productName": "Elegant Metal Ball"
42 },
43 }

```

9

```

src > modules > Aggregation-Indexing > JS Question9.mongodb.js > ...
# Connected to localhost:27017
use('NodePractise')

1
2
3 db.orders.aggregate([
4   {
5     $match: {
6       status: "Delivered"
7     }
8   },
9   {
10    $group: {
11      _id: null,
12      revenue: {$sum: "$totalAmount"}
13    }
14  },
15  {
16    $project: {
17      _id: 0,
18      totalRevenue: "$revenue"
19    }
20  }
21 ])

```

Playground Result

```

1 {
2   "totalRevenue": 8375.74
3 }
4
5

```

10

```

src > modules > Aggregation-Indexing > JS Question10.mongodb.js > ...
# Connected to localhost:27017
use('NodePractise')

1
2
3 db.orders.aggregate([
4   {
5     $unwind: {
6       path: "$items",
7     }
8   },
9   {
10    $group: {
11      _id: "$items.productName",
12      totalQuantity: {$sum: "$items.quantity"},
13      totalRevenue: {$sum: {$multiply: ["$items.price", "$items.quantity"]}}
14    }
15  }
16 ])

```

Playground Result

```

1 {
2   "_id": "Soft Ceramic Cheese",
3   "totalQuantity": 2,
4   "totalRevenue": 75.8
5 },
6 {
7   "_id": "Elegant Cotton Chair",
8   "totalQuantity": 2,
9   "totalRevenue": 193.38
10 },
11 {
12   "_id": "Modern Steel Shoes",
13   "totalQuantity": 1,
14   "totalRevenue": 72.55
15 },
16 {
17   "_id": "Soft Granite Soap",
18   "totalQuantity": 1,
19   "totalRevenue": 22.92
20 },
21 {
22   "_id": "Modern Bronze Chicken",
23   "totalQuantity": 3,
24   "totalRevenue": 131.67000000000002
25 },
26 {
27   "_id": "Rustic Bamboo Chair",
28   "totalQuantity": 1,
29   "totalRevenue": 41.39
30 },
31 {
32   "_id": "Fresh Aluminum Fish",
33   "totalQuantity": 9,
34   "totalRevenue": 149.85
35 },
36 {
37   "_id": "Awesome Aluminum Chair",
38   "totalQuantity": 6,
39   "totalRevenue": 519.54
40 },
41 {
42   "_id": "Recycled Metal Salad",
43   "totalQuantity": 1,
44   "totalRevenue": 1.0
45 }

```

Index Queries:

1

```

src > modules > Aggregation-Indexing > JS Indexing1.mongodb.js
  Connected to localhost:27017
1 use("NodePractise")
2 db.orders.getIndexes();

```

```

1 {
2   {
3     "v": 2,
4     "key": {
5       "_id": 1
6     },
7     "name": "_id_"
8   },
9   {
10    "v": 2,
11    "key": {
12      "_fts": "text",
13      "_ftsx": 1
14    },
15    "name": "items.productName_text",
16    "weights": {
17      "items.productName": 1
18    },
19    "default language": "english",
20    "language override": "language",
21    "textIndexVersion": 3
22  },
23  {
24    "v": 2,
25    "key": {
26      "status": 1,
27      "orderDate": -1
28    },
29    "name": "status_1_orderDate_-1"
30  }
31 }

```

2

```

src > modules > Aggregation-Indexing > JS Indexing2.mongodb.js > ...
  Connected to localhost:27017
1 use("NodePractise")
2 db.orders.createIndex({ customerName: 1 })
3 db.orders.find({ customerName: "John Doe" }).explain("executionStats")
4

```

```

1 {
2   "explainVersion": "1",
3   "queryPlanner": {
4     "namespace": "NodePractise.orders",
5     "parsedQuery": {
6       "customerName": {
7         "$eq": "John Doe"
8       }
9     },
10    "indexFilterSet": false,
11    "queryHash": "17B49A18",
12    "planCacheShapeHash": "17B49A18",
13    "planCacheKey": "0557E263",
14    "optimizationTimeMillis": 0,
15    "maxIndexedOrSolutionsReached": false,
16    "maxIndexedAndSolutionsReached": false,
17    "maxScansToExplodeReached": false,
18    "prunedSimilarIndexes": false,
19    "winningPlan": {
20      "isCached": false,
21      "stage": "FETCH",
22      "inputStage": {
23        "stage": "IXSCAN",
24        "keyPattern": {
25          "customerName": 1
26        },
27        "indexName": "customerName_1",
28        "isMultiKey": false,
29        "multiKeyPaths": {
30          "customerName": []
31        },
32        "isUnique": false,
33        "isSparse": false,
34        "isPartial": false,
35        "indexVersion": 2,
36        "direction": "forward",
37        "indexBounds": {
38          "customerName": [
39            ["John Doe", "John Doe"]
40          ]
41        }
42      }
43    }
44  }

```

3

```
src > modules > Aggregation-Indexing > JS Indexing3.mongodb.js > ...
# Connected to localhost:27017
1 use('NodePractise');
2
3
4
5 db.orders.createIndex({ status: 1, orderDate: -1 })
6
7 db.orders.find({
8   status: "Delivered",
9   orderDate: { $gte: ISODate("2025-01-01") }
10 }).explain("executionStats")
11
```

```
1 {
2   "explainVersion": "1",
3   "queryPlanner": {
4     "namespace": "NodePractise.orders",
5     "parsedQuery": {
6       "$and": [
7         {
8           "status": {
9             "$eq": "Delivered"
10          }
11        },
12        {
13          "orderDate": {
14            "$gte": {
15              "$date": "2025-01-01T00:00:00Z"
16            }
17          }
18        }
19      ]
20    },
21    "indexFilterSet": false,
22    "queryHash": "888CD4EF",
23    "planCacheShapeHash": "888CD4EF",
24    "planCacheKey": "A8F87C76",
25    "optimizationTimeMillis": 0,
26    "maxIndexedOrSolutionsReached": false,
27    "maxIndexedAndSolutionsReached": false,
28    "maxScansToExplodeReached": false,
29    "prunedSimilarIndexes": false,
30    "winningPlan": {
31      "isCached": false,
32      "stage": "FETCH",
33      "inputStage": {
34        "stage": "IXSCAN",
35        "keyPattern": {
```