

JavaScript

```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("mydb");
  var query = { address: "Park Lane 38" };
  dbo.collection("customers").find(query).toArray(function(err, result) {
    if (err) throw err;
    console.log(result);
    db.close();
  });
});
```

JavaScript with MongoDB Aggregation Pipeline

Collection

```
db.orders.aggregate( [  
  $match stage → { $match: { status: "A" } },  
  $group stage → { $group: { _id: "$cust_id", total: { $sum: "$amount" } } }  
] )
```

{ cust_id: "A123", amount: 500, status: "A" }
{ cust_id: "A123", amount: 250, status: "A" }
{ cust_id: "B212", amount: 200, status: "A" }
{ cust_id: "A123", amount: 300, status: "D" }

orders

\$match

{ cust_id: "A123", amount: 500, status: "A" }
{ cust_id: "A123", amount: 250, status: "A" }
{ cust_id: "B212", amount: 200, status: "A" }

\$group

Results	
	<pre>{ _id: "A123", total: 750 }</pre>
	<pre>{ _id: "B212", total: 200 }</pre>

JavaScript with Map Reduce

```
db.orders.mapReduce(  
    mapFunction1,  
    reduceFunction1,  
    { out: "map_reduce_example" }  
)
```

Driver or Connector

- JDBC, ODBC or Custom
- Query by SQL
- Hadoop Integration