Project 4 Group 3 CS585/DS 503: Big Data Management Spring 2019

Group Members: Wenhan Ji, Omkar Kulkarni

Work Distribution and Methodology:

Wenhan - 50%

1. Planning queries
2. Writing Queries
3. Writing report

Omkar - 50%

1. Panning queries
2. Writing Queries
3. Writing report

Relative Contributions: Collaborated equally on all the solutions, discussed the solution and submitted the ones which we deemed as best ones.

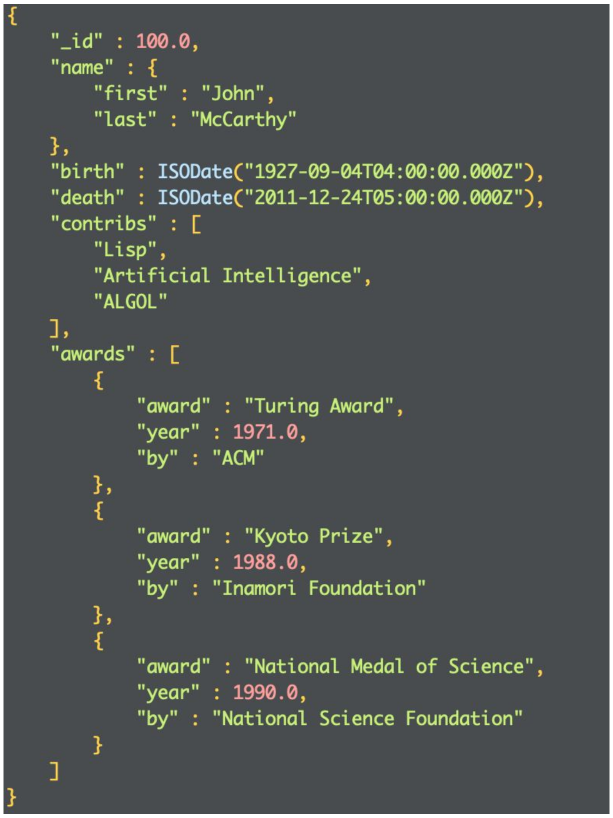
Solution:
We did the queries on **Robo 3T,** which is open source compiler for MongoDB.

For Creating the collection, we use createcollection() with collection name in it. Then used insertMany() to insert multiple documents in it.

Problem 1: Unstructured Data Modeling and Database Updating.

1) Write an operation(s) that changes the _id of “John McCarthy” to value 100.

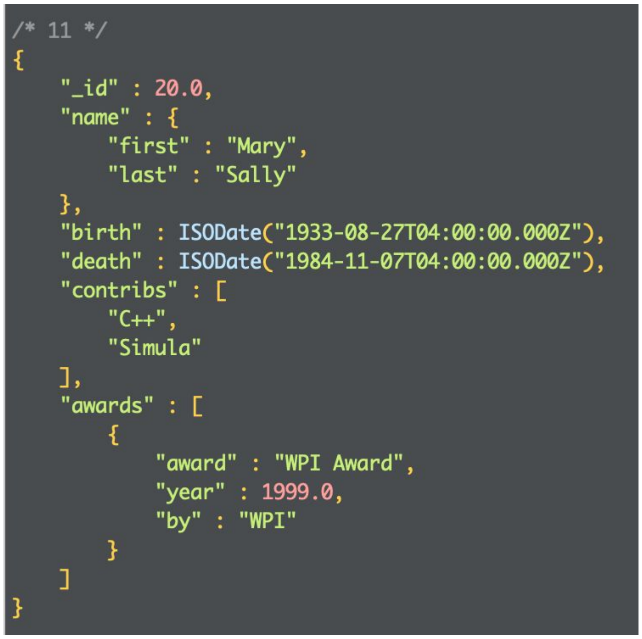
The previous ID of “John McCarthy” was `ObjectId("51df07b094c6acd67e492f41")`
We changed it to 100 by first finding the document with “John Mccarthy” in it using findone() and then the output will be:



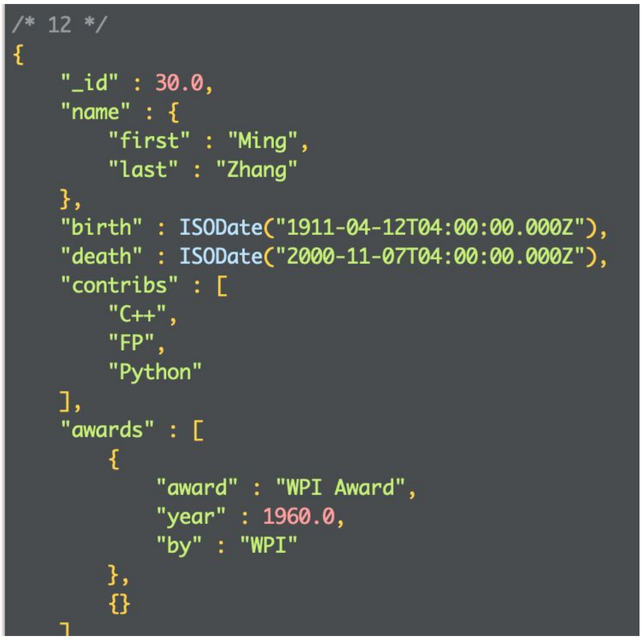
```
{
    "_id" : 100.0,
    "name" : {
        "first" : "John",
        "last" : "McCarthy"
    },
    "birth" : ISODate("1927-09-04T04:00:00.000Z"),
    "death" : ISODate("2011-12-24T05:00:00.000Z"),
    "contribs" : [
        "Lisp",
        "Artificial Intelligence",
        "ALGOL"
    ],
    "awards" : [
        {
            "award" : "Turing Award",
            "year" : 1971.0,
            "by" : "ACM"
        },
        {
            "award" : "Kyoto Prize",
            "year" : 1988.0,
            "by" : "Inamori Foundation"
        },
        {
            "award" : "National Medal of Science",
            "year" : 1990.0,
            "by" : "National Science Foundation"
        }
    ]
}
```

2) Write an operation(s) that inserts the following new records into the collection:

After using insertMany() we get,



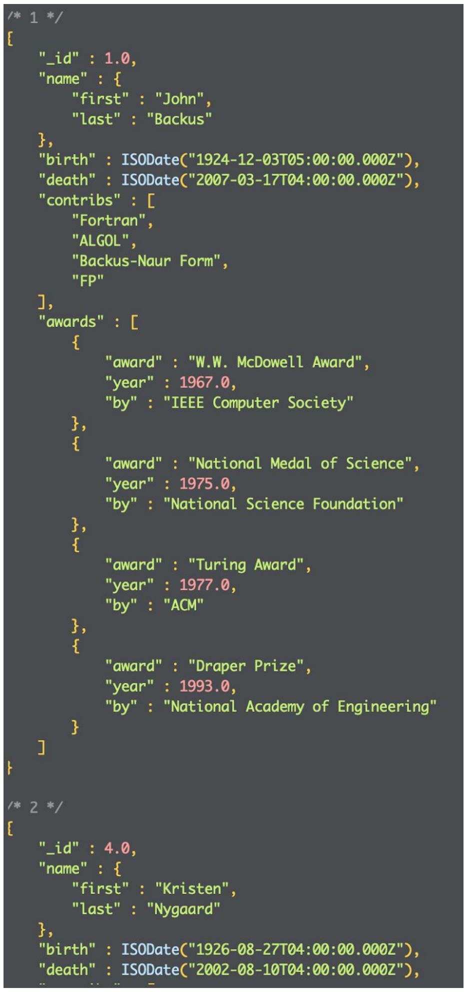
```
/* 11 */
{
    "_id" : 20.0,
    "name" : {
        "first" : "Mary",
        "last" : "Sally"
    },
    "birth" : ISODate("1933-08-27T04:00:00.000Z"),
    "death" : ISODate("1984-11-07T04:00:00.000Z"),
    "contribs" : [
        "C++",
        "Simula"
    ],
    "awards" : [
        {
            "award" : "WPI Award",
            "year" : 1999.0,
            "by" : "WPI"
        }
    ]
}
```



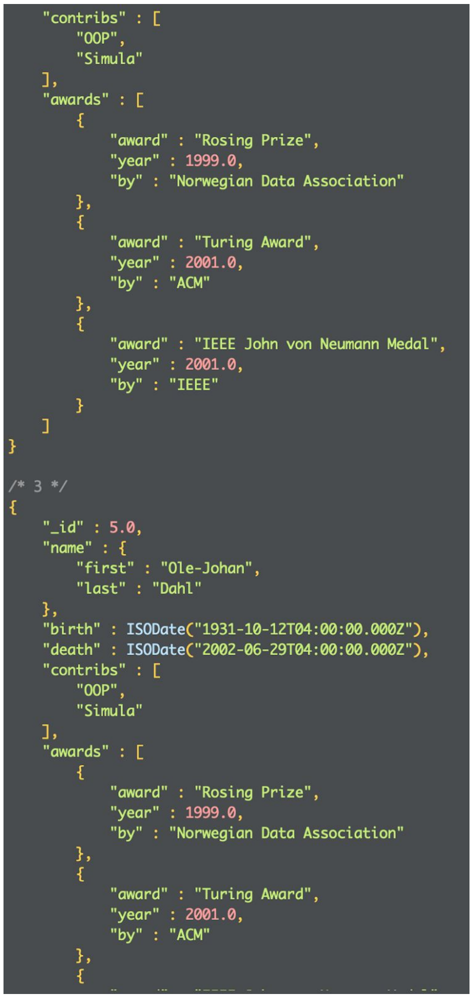
```
/* 12 */
{
    "_id" : 30.0,
    "name" : {
        "first" : "Ming",
        "last" : "Zhang"
    },
    "birth" : ISODate("1911-04-12T04:00:00.000Z"),
    "death" : ISODate("2000-11-07T04:00:00.000Z"),
    "contribs" : [
        "C++",
        "FP",
        "Python"
    ],
    "awards" : [
        {
            "award" : "WPI Award",
            "year" : 1960.0,
            "by" : "WPI"
        },
        {}
    ]
```

3) Report all documents of people who got a “Turing Award” after 1960.

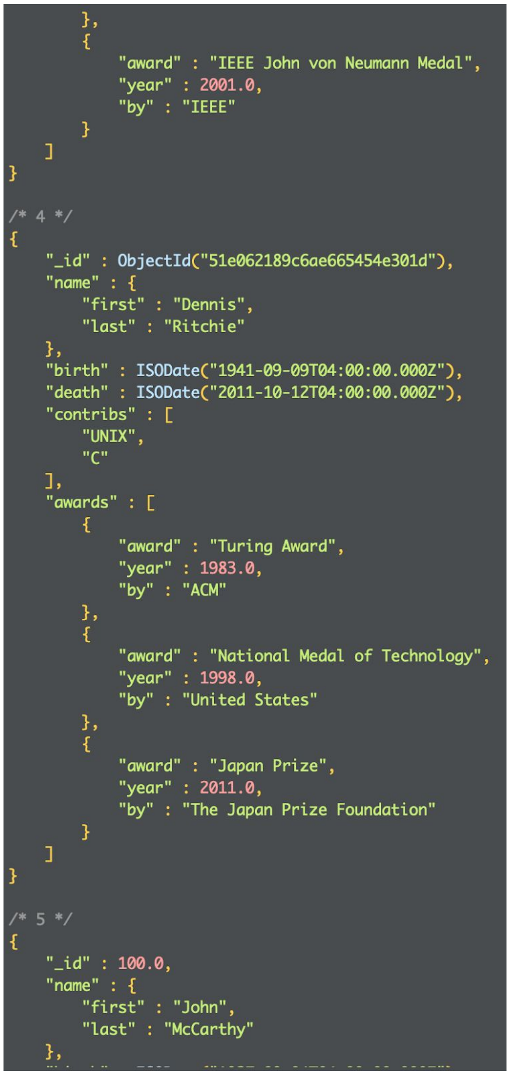
For this we just used find() with $and for joining two searched with $gt 1960 and award = turing award.



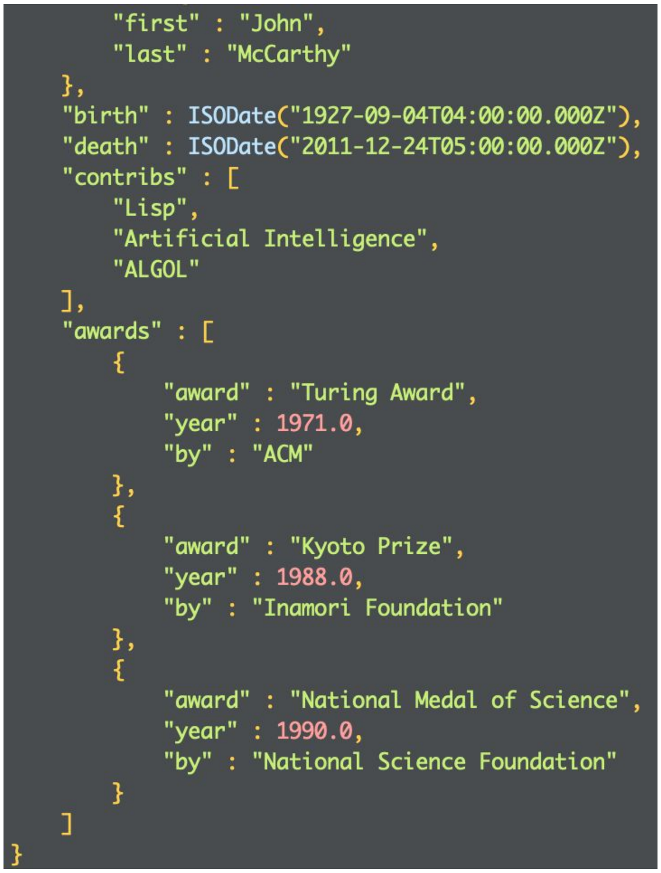
```
/* 1 */
{
    "_id" : 1.0,
    "name" : {
        "first" : "John",
        "last" : "Backus"
    },
    "birth" : ISODate("1924-12-03T05:00:00.000Z"),
    "death" : ISODate("2007-03-17T04:00:00.000Z"),
    "contribs" : [
        "Fortran",
        "ALGOL",
        "Backus-Naur Form",
        "FP"
    ],
    "awards" : [
        {
            "award" : "W.W. McDowell Award",
            "year" : 1967.0,
            "by" : "IEEE Computer Society"
        },
        {
            "award" : "National Medal of Science",
            "year" : 1975.0,
            "by" : "National Science Foundation"
        },
        {
            "award" : "Turing Award",
            "year" : 1977.0,
            "by" : "ACM"
        },
        {
            "award" : "Draper Prize",
            "year" : 1993.0,
            "by" : "National Academy of Engineering"
        }
    ]
}

/* 2 */
{
    "_id" : 4.0,
    "name" : {
        "first" : "Kristen",
        "last" : "Nygaard"
    },
    "birth" : ISODate("1926-08-27T04:00:00.000Z"),
    "death" : ISODate("2002-08-10T04:00:00.000Z"),
```



```
    "contribs" : [
        "OOP",
        "Simula"
    ],
    "awards" : [
        {
            "award" : "Rosing Prize",
            "year" : 1999.0,
            "by" : "Norwegian Data Association"
        },
        {
            "award" : "Turing Award",
            "year" : 2001.0,
            "by" : "ACM"
        },
        {
            "award" : "IEEE John von Neumann Medal",
            "year" : 2001.0,
            "by" : "IEEE"
        }
    ]
}

/* 3 */
{
    "_id" : 5.0,
    "name" : {
        "first" : "Ole-Johan",
        "last" : "Dahl"
    },
    "birth" : ISODate("1931-10-12T04:00:00.000Z"),
    "death" : ISODate("2002-06-29T04:00:00.000Z"),
    "contribs" : [
        "OOP",
        "Simula"
    ],
    "awards" : [
        {
            "award" : "Rosing Prize",
            "year" : 1999.0,
            "by" : "Norwegian Data Association"
        },
        {
            "award" : "Turing Award",
            "year" : 2001.0,
            "by" : "ACM"
        },
        {
```



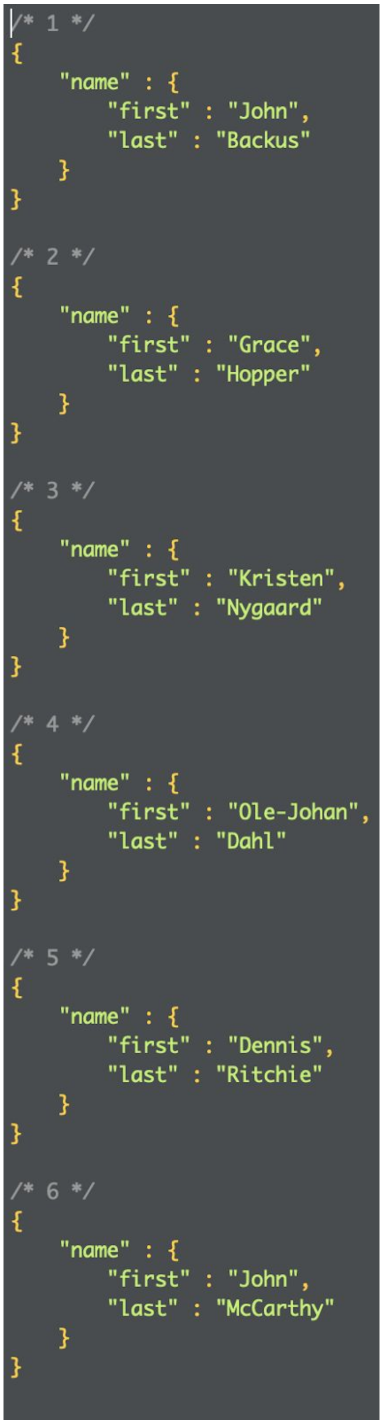
```
        },
        {
            "award" : "IEEE John von Neumann Medal",
            "year" : 2001.0,
            "by" : "IEEE"
        }
    ]
}

/* 4 */
{
    "_id" : ObjectId("51e062189c6ae665454e301d"),
    "name" : {
        "first" : "Dennis",
        "last" : "Ritchie"
    },
    "birth" : ISODate("1941-09-09T04:00:00.000Z"),
    "death" : ISODate("2011-10-12T04:00:00.000Z"),
    "contribs" : [
        "UNIX",
        "C"
    ],
    "awards" : [
        {
            "award" : "Turing Award",
            "year" : 1983.0,
            "by" : "ACM"
        },
        {
            "award" : "National Medal of Technology",
            "year" : 1998.0,
            "by" : "United States"
        },
        {
            "award" : "Japan Prize",
            "year" : 2011.0,
            "by" : "The Japan Prize Foundation"
        }
    ]
}

/* 5 */
{
    "_id" : 100.0,
    "name" : {
        "first" : "John",
        "last" : "McCarthy"
    },
```



```
        "first" : "John",
        "last" : "McCarthy"
    },
    "birth" : ISODate("1927-09-04T04:00:00.000Z"),
    "death" : ISODate("2011-12-24T05:00:00.000Z"),
    "contribs" : [
        "Lisp",
        "Artificial Intelligence",
        "ALGOL"
    ],
    "awards" : [
        {
            "award" : "Turing Award",
            "year" : 1971.0,
            "by" : "ACM"
        },
        {
            "award" : "Kyoto Prize",
            "year" : 1988.0,
            "by" : "Inamori Foundation"
        },
        {
            "award" : "National Medal of Science",
            "year" : 1990.0,
            "by" : "National Science Foundation"
        }
    ]
}
```

4) Report all people who got more than 2 awards.

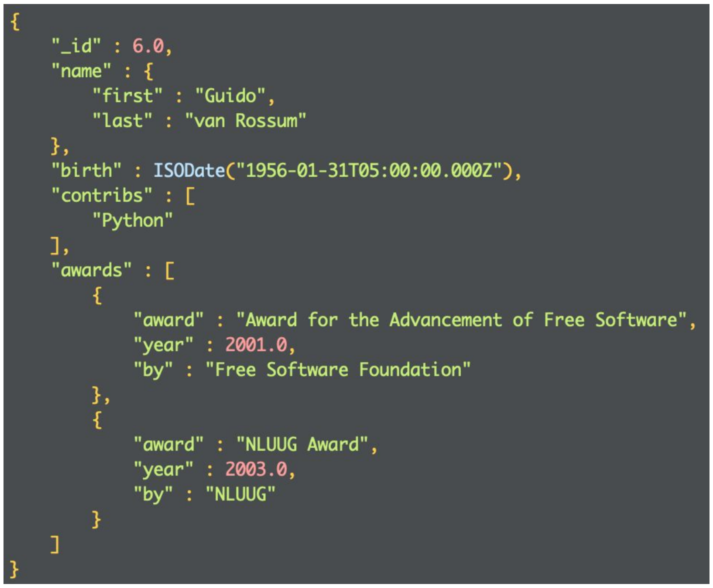
We used constraint of awards.2 and exists = true.



```
/* 1 */
{
    "name" : {
        "first" : "John",
        "last" : "Backus"
    }
}

/* 2 */
{
    "name" : {
        "first" : "Grace",
        "last" : "Hopper"
    }
}

/* 3 */
{
    "name" : {
        "first" : "Kristen",
        "last" : "Nygaard"
    }
}

/* 4 */
{
    "name" : {
        "first" : "Ole-Johan",
        "last" : "Dahl"
    }
}

/* 5 */
{
    "name" : {
        "first" : "Dennis",
        "last" : "Ritchie"
    }
}

/* 6 */
{
    "name" : {
        "first" : "John",
        "last" : "McCarthy"
    }
}
```

5) Update the document of “Guido van Rossum” to add “Python” to the contribution list.

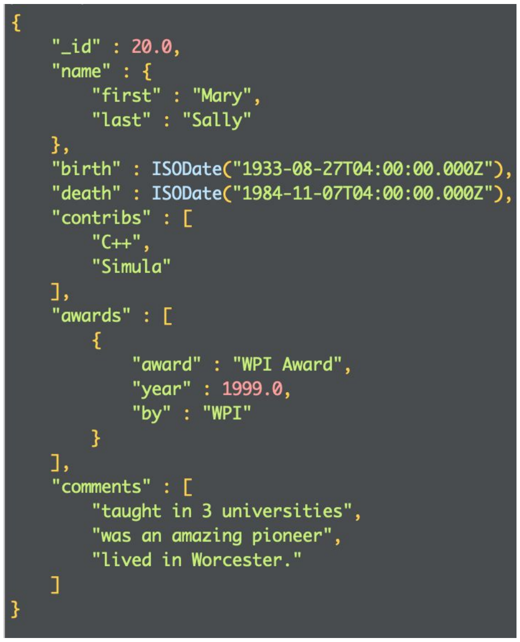
We used update() to update the field with addToSet() function.



```
{
    "_id" : 6.0,
    "name" : {
        "first" : "Guido",
        "last" : "van Rossum"
    },
    "birth" : ISODate("1956-01-31T05:00:00.000Z"),
    "contribs" : [
        "Python"
    ],
    "awards" : [
        {
            "award" : "Award for the Advancement of Free Software",
            "year" : 2001.0,
            "by" : "Free Software Foundation"
        },
        {
            "award" : "NLUUG Award",
            "year" : 2003.0,
            "by" : "NLUUG"
        }
    ]
}
```

6) Insert a new field called “comments” of type array into document of “Mary Sally” storing the comments: “taught in 3 universities”, “was an amazing pioneer”, “lived in Worcester.”

We used update() with $set to add new field called comments and then inserted three new comments into it.



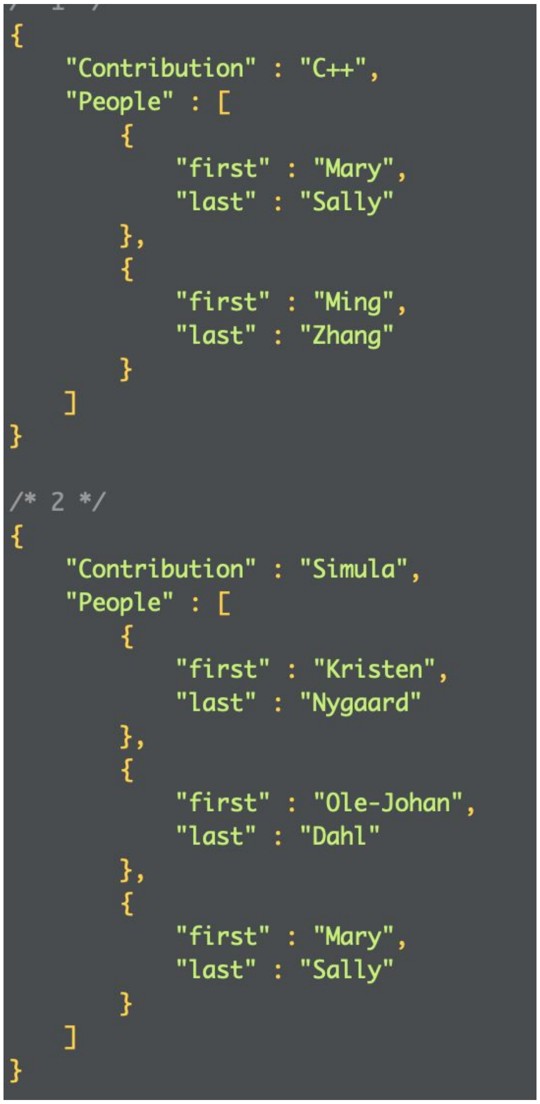
```
{
    "_id" : 20.0,
    "name" : {
        "first" : "Mary",
        "last" : "Sally"
    },
    "birth" : ISODate("1933-08-27T04:00:00.000Z"),
    "death" : ISODate("1984-11-07T04:00:00.000Z"),
    "contribs" : [
        "C++",
        "Simula"
    ],
    "awards" : [
        {
            "award" : "WPI Award",
            "year" : 1999.0,
            "by" : "WPI"
        }
    ],
    "comments" : [
        "taught in 3 universities",
        "was an amazing pioneer",
        "lived in Worcester."
    ]
}
```

7) For each contribution by “Mary Sally”, say contribution “C”, list the people’s first and last names who have the same contribution “C”. For example, since “Mary Sally” has two contributions in “C++” and “Simula”, the output for her should be similar to:

{Contribution: “C++”,

People: [{first: “Mary”, last: “Sally”}, {first: “Ming”, last: “Zhang”}]}, { Contribution: “Simula”, ...}

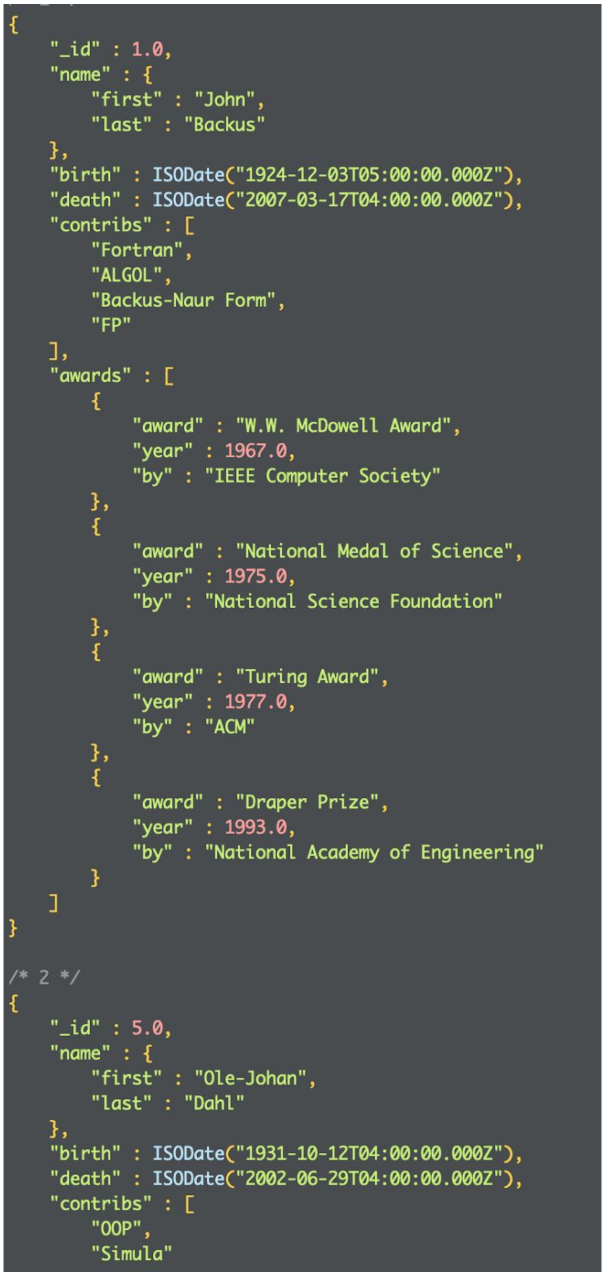
We first created a variable called contribsMary which stores all the contributions of mary, then iterated through the list of that to get array of Contribution and people. Next we created variable called docs and pushed the name and id of people who got the same contributions as mary.



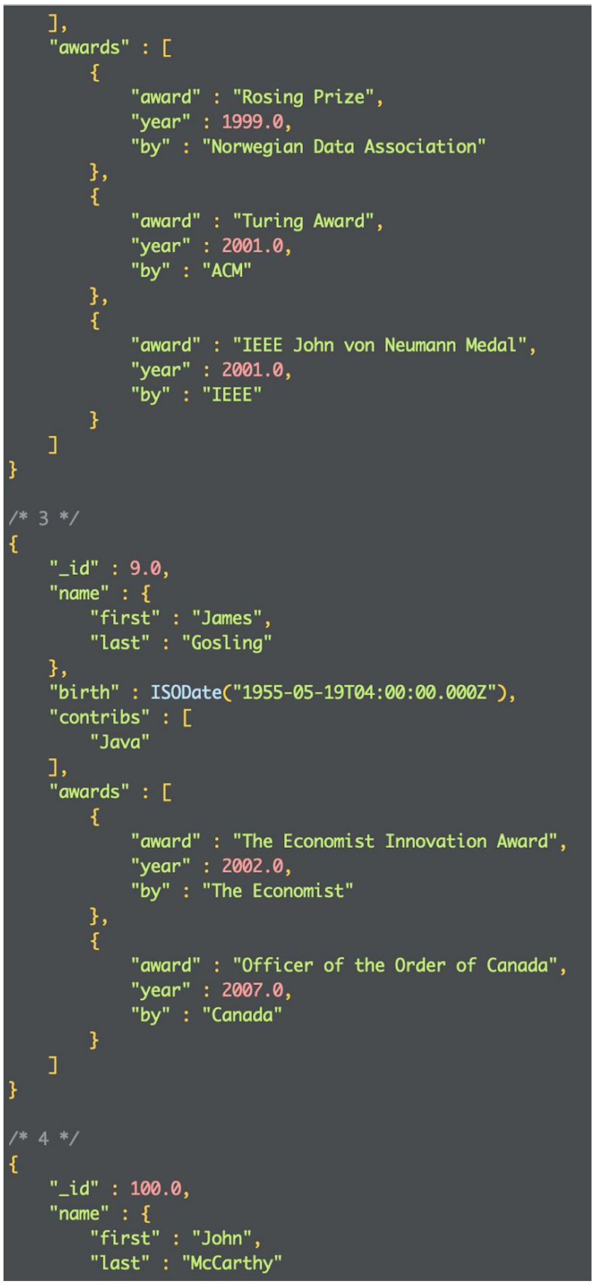
```
/* 1 */
{
    "Contribution" : "C++",
    "People" : [
        {
            "first" : "Mary",
            "last" : "Sally"
        },
        {
            "first" : "Ming",
            "last" : "Zhang"
        }
    ]
}

/* 2 */
{
    "Contribution" : "Simula",
    "People" : [
        {
            "first" : "Kristen",
            "last" : "Nygaard"
        },
        {
            "first" : "Ole-Johan",
            "last" : "Dahl"
        },
        {
            "first" : "Mary",
            "last" : "Sally"
        }
    ]
}
```

8) Report all documents where the first name matches the regular expression “Jo*”, where “*” means any number of characters. Report the documents sorted by the last name.

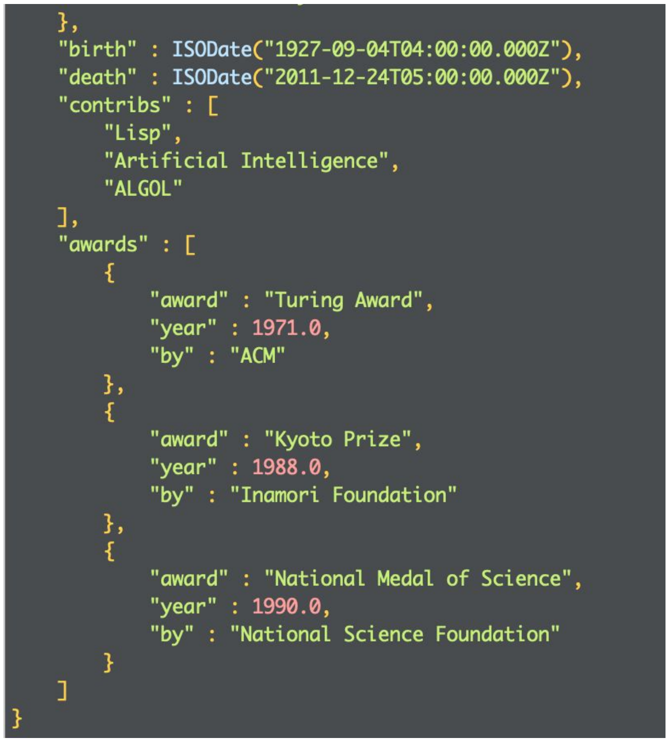
We used Regex functionality $regex: "Jo*" and sorted by last name.



```
{
    "_id" : 1.0,
    "name" : {
        "first" : "John",
        "last" : "Backus"
    },
    "birth" : ISODate("1924-12-03T05:00:00.000Z"),
    "death" : ISODate("2007-03-17T04:00:00.000Z"),
    "contribs" : [
        "Fortran",
        "ALGOL",
        "Backus-Naur Form",
        "FP"
    ],
    "awards" : [
        {
            "award" : "W.W. McDowell Award",
            "year" : 1967.0,
            "by" : "IEEE Computer Society"
        },
        {
            "award" : "National Medal of Science",
            "year" : 1975.0,
            "by" : "National Science Foundation"
        },
        {
            "award" : "Turing Award",
            "year" : 1977.0,
            "by" : "ACM"
        },
        {
            "award" : "Draper Prize",
            "year" : 1993.0,
            "by" : "National Academy of Engineering"
        }
    ]
}

/* 2 */
{
    "_id" : 5.0,
    "name" : {
        "first" : "Ole-Johan",
        "last" : "Dahl"
    },
    "birth" : ISODate("1931-10-12T04:00:00.000Z"),
    "death" : ISODate("2002-06-29T04:00:00.000Z"),
    "contribs" : [
        "OOP",
        "Simula"
```



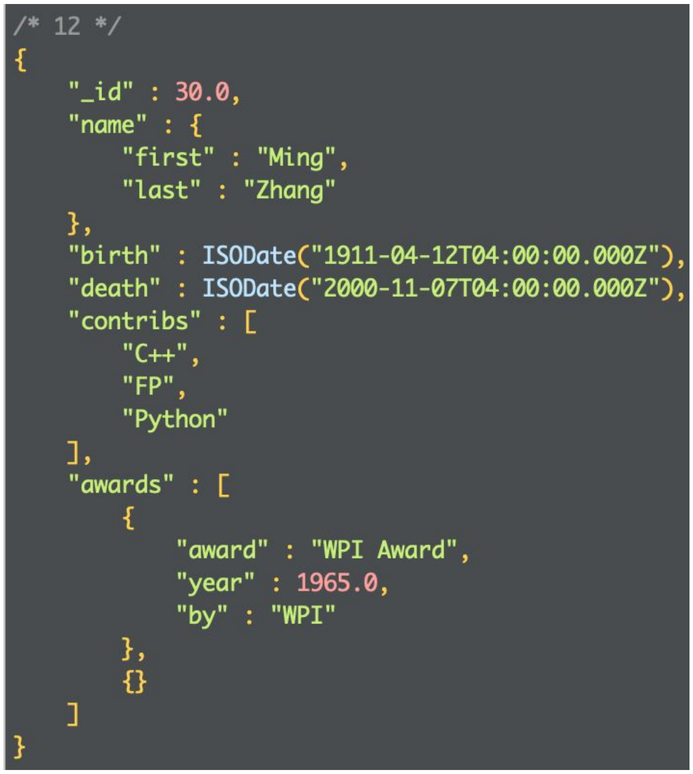
```
    ],
    "awards" : [
        {
            "award" : "Rosing Prize",
            "year" : 1999.0,
            "by" : "Norwegian Data Association"
        },
        {
            "award" : "Turing Award",
            "year" : 2001.0,
            "by" : "ACM"
        },
        {
            "award" : "IEEE John von Neumann Medal",
            "year" : 2001.0,
            "by" : "IEEE"
        }
    ]
}

/* 3 */
{
    "_id" : 9.0,
    "name" : {
        "first" : "James",
        "last" : "Gosling"
    },
    "birth" : ISODate("1955-05-19T04:00:00.000Z"),
    "contribs" : [
        "Java"
    ],
    "awards" : [
        {
            "award" : "The Economist Innovation Award",
            "year" : 2002.0,
            "by" : "The Economist"
        },
        {
            "award" : "Officer of the Order of Canada",
            "year" : 2007.0,
            "by" : "Canada"
        }
    ]
}

/* 4 */
{
    "_id" : 100.0,
    "name" : {
        "first" : "John",
        "last" : "McCarthy"
```



```
    },
    "birth" : ISODate("1927-09-04T04:00:00.000Z"),
    "death" : ISODate("2011-12-24T05:00:00.000Z"),
    "contribs" : [
        "Lisp",
        "Artificial Intelligence",
        "ALGOL"
    ],
    "awards" : [
        {
            "award" : "Turing Award",
            "year" : 1971.0,
            "by" : "ACM"
        },
        {
            "award" : "Kyoto Prize",
            "year" : 1988.0,
            "by" : "Inamori Foundation"
        },
        {
            "award" : "National Medal of Science",
            "year" : 1990.0,
            "by" : "National Science Foundation"
        }
    ]
}
```

9) Update the award of document _id =30, which is given by WPI, and set the year to 1965.

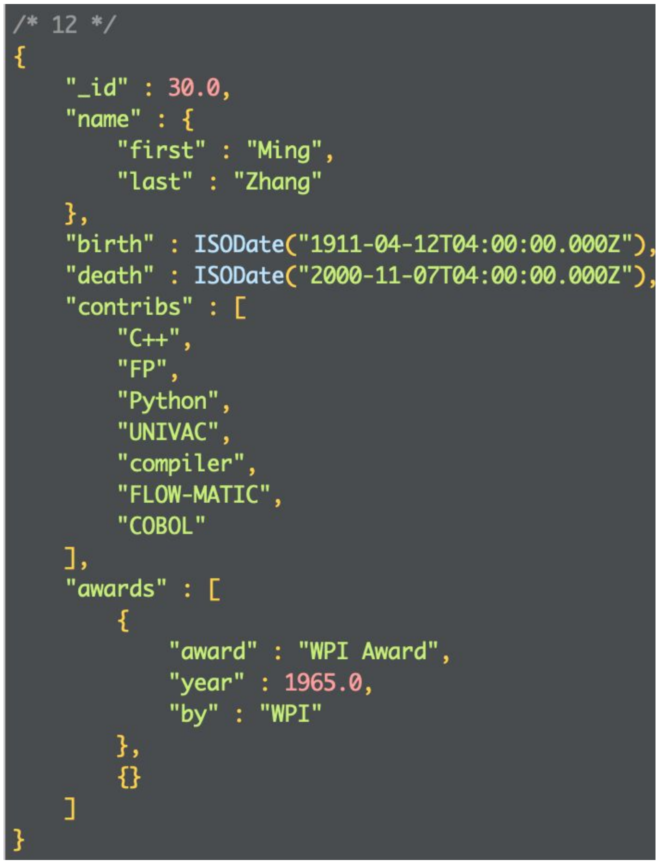
We used the update function with $set to get the results. It’s just simple culmination of queries.



```
/* 12 */
{
    "_id" : 30.0,
    "name" : {
        "first" : "Ming",
        "last" : "Zhang"
    },
    "birth" : ISODate("1911-04-12T04:00:00.000Z"),
    "death" : ISODate("2000-11-07T04:00:00.000Z"),
    "contribs" : [
        "C++",
        "FP",
        "Python"
    ],
    "awards" : [
        {
            "award" : "WPI Award",
            "year" : 1965.0,
            "by" : "WPI"
        },
        {}
    ]
}
```

10) Add (copy) all the contributions of document _id = 3 to that of document _id = 30.

We created a new variable and copied the contributions of id = 3 into it. Then used update() to push the contributions to id = 30



```
/* 12 */
{
    "_id" : 30.0,
    "name" : {
        "first" : "Ming",
        "last" : "Zhang"
    },
    "birth" : ISODate("1911-04-12T04:00:00.000Z"),
    "death" : ISODate("2000-11-07T04:00:00.000Z"),
    "contribs" : [
        "C++",
        "FP",
        "Python",
        "UNIVAC",
        "compiler",
        "FLOW-MATIC",
        "COBOL"
    ],
    "awards" : [
        {
            "award" : "WPI Award",
            "year" : 1965.0,
            "by" : "WPI"
        },
        {}
    ]
}
```