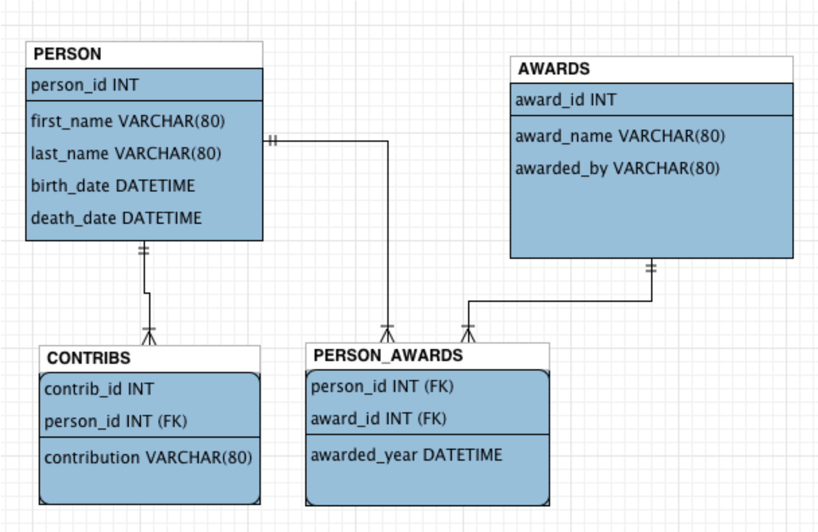
Practice Quiz #3

Due Oct 24 at 11:59pm **Points** 30 **Questions** 11 **Available** until Oct 24 at 11:59pm **Time Limit** None
Allowed Attempts Unlimited

Instructions

SQL Data Model:



Formatting MongoDB Queries for Quiz Submission (Example)

```
db.bios.find( { type: 'food', $or: [ { qty: { $gt: 100 } }, { price: { $lt: 9.95 } } ]} )
```

Should be key sorted as:

```
db.bios.find( { $or: [ { price: { $lt: 9.95 }}, {qty: { $gt: 100 } } ], type: 'food' } )
```

Output formatted by www.dirtymarkup.com (http://www.dirtymarkup.com) :

```
db.bios.find({
  $or: [{
    price: {
      $lt: 9.95
    }
  }, {
    qty: {
      $gt: 100
    }
  }],
  type: 'food'
})
```

Take the Quiz Again

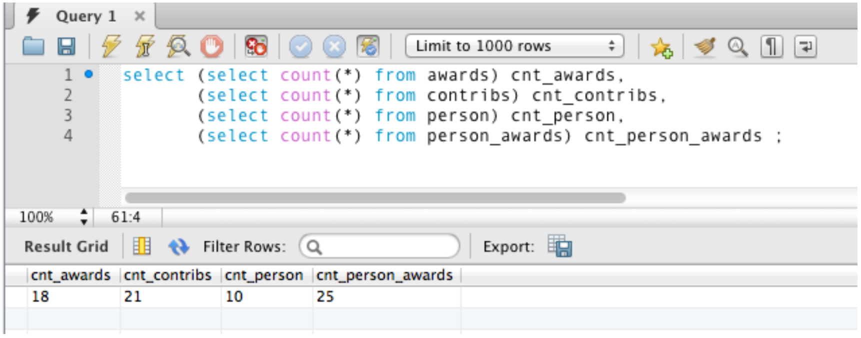
Attempt History

	Attempt	Time	Score
KEPT	Attempt 4	5 minutes	30 out of 30
LATEST	Attempt 4	5 minutes	30 out of 30
	Attempt 3	4 minutes	28 out of 30
	Attempt 2	8 minutes	28 out of 30
	Attempt 1	8628 minutes	0 out of 30

Score for this attempt: **30** out of 30
Submitted Oct 23 at 8:27pm
This attempt took 5 minutes.

Quiz Setup:

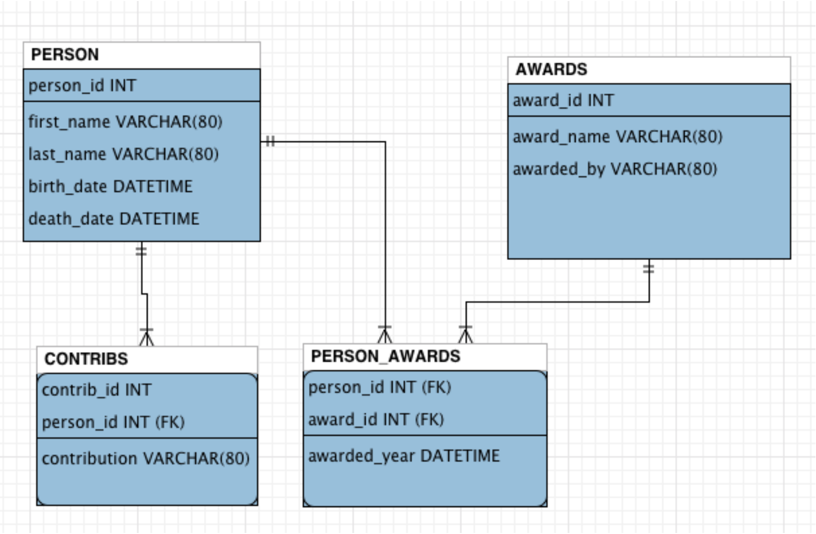
- Before taking the quiz, make sure you have:
  - MySQL Locally Installed or Accessed via Cloud (i.e. Cloud9)
  - MongoDB Locally Installed or Accessed via MongoLabs
  - MySQL Workbench Installed (Optional)
  - Robomongo Installed (Required as Output from this Tools is used as Quiz Answers)
- Tools we will using during the Quiz (visit the site and get familiar with them)
- Dirty Markup will be used to Format MongoDB Queries before submission. Quiz will check the answers for exact match, so a correct answer with extra whitespace would result in a zero for the answer.
  - http://www.dirtymarkup.com (http://www.dirtymarkup.com)
- MongoDB Queries make sue of Property Lists and Arrays. These sets are interpreted unordered but the Quiz expects an exact match! Thus, we will follow the convention for sorting property keys within a set:
  - "_id" gets special treatment. If present, this property should always be first in the set
  - The rest should be sorted left-to-right (or top-to-bottom) in Alphanumeric Sort Order. The URL below can be use to check if you are nor sure. (*Also see an example of formatting an answer below*).
  - http://www.textfixer.com/tools/alphabetize-text-words.php (http://www.textfixer.com/tools/alphabetize-text-words.php)
- Database Schema Setup
  - Install MySQL Schema. Files: bios-schema.sql and bios-data.sql
  - Insert MongoDB Documents (bios collection). File: bios-bulk.js
  - Run the following Queries to confirm proper Database Setup:
    - db.bios.find() in MongoDB/Robomongo should return 10 documents
    - The following SQL Query should return the appropriate counts as shown.



```
select (select count(*) from awards) cnt_awards,
       (select count(*) from contribs) cnt_contribs,
       (select count(*) from person) cnt_person,
       (select count(*) from person_awards) cnt_person_awards ;
```

cnt_awards	cnt_contribs	cnt_person	cnt_person_awards
18	21	10	25

SQL Data Model:



Formatting MongoDB Queries for Quiz Submission:

As noted, we will follow the formatting rules:

We will follow the convention for sorting property keys within a set. Note, sets can be nested and sorting applies recursively to each set.

- "_id" gets special treatment. If present, this property should always be first in the set
- The rest should be sorted left-to-right (or top-to-bottom) in Alphanumeric Sort Order. The URL below can be use to check if you are nor sure. (*Also see an example of formatting an answer below*).
- http://www.textfixer.com/tools/alphabetize-text-words.php (http://www.textfixer.com/tools/alphabetize-text-words.php)

For example:

db.bios.find({ **type**: 'food', **$or**: [{ **qty**: { $gt: 100 } }, { **price**: { $lt: 9.95 } }]})

Should be rewritten as:

db.bios.find({ **$or**: [{ **price**: { $lt: 9.95 }}, {**qty**: { $gt: 100 } }], **type**: 'food' })

Note: $or is before type. price is before qty. $gt and $lt are single member sets so stay as is.

Then run the resorted query in www.dirtymarkup.com for canonical formatting.

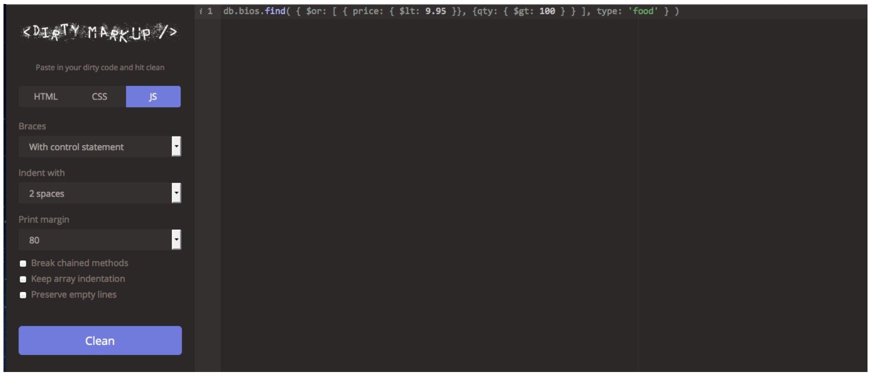
Output is: (Submit this as the answer -- Copy-and-paste into Canvas)

```
db.bios.find({
  $or: [{
    price: {
      $lt: 9.95
    }
  }, {
    qty: {
      $gt: 100
    }
  }],
  type: 'food'
})
```

See Screenshots below on Format Settings.

- *Braces with Control Statement*
- *Indent with 2 spaces*
- *Print Margin 80*

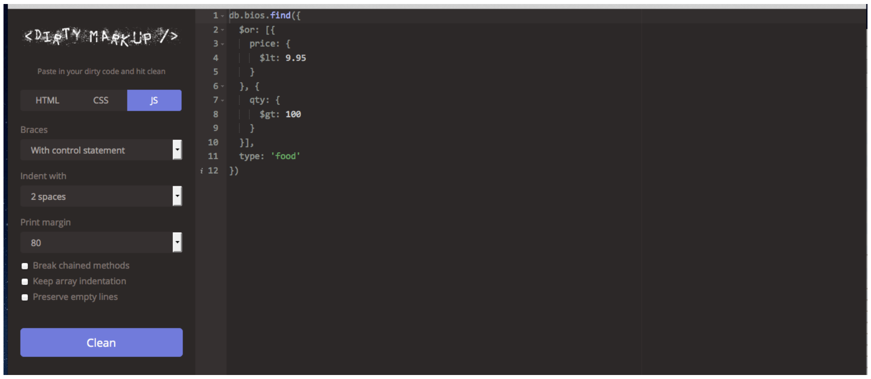
Before Formatting:



Braces: With control statement
Indent with: 2 spaces
Print margin: 80
■ Break chained methods
■ Keep array indentation
■ Preserve empty lines

```
db.bios.find( { $or: [ { price: { $lt: 9.95 }}, {qty: { $gt: 100 } } ], type: 'food' } )
```

After Formatting:



Braces: With control statement
Indent with: 2 spaces
Print margin: 80
■ Break chained methods
■ Keep array indentation
■ Preserve empty lines

```
db.bios.find({
  $or: [{
    price: {
      $lt: 9.95
    }
  }, {
    qty: {
      $gt: 100
    }
  }],
  type: 'food'
})
```

Question 1

2 / 2 pts

Count of Records/Documents

Translate the following SQL Query into MongoDB:

select count(*) from person

The MongoDB Output Should be:

10

MongoDB Query Is:

Correct!

```
db.bios.count()
```

Correct Answers

db.bios.count()

db.bios.find().count()

CLOUD

Question 2

2 / 2 pts

Find Bios with Birth Date before 1950

Translate the following SQL Query into MongoDB:

```
select first_name, last_name, birth_date
from person
where birth_date < date('1950-01-01')
```

The MongoDB Output Should be:

```
/* 0 */
{
    "_id" : ObjectId("543efb5060ac6af54acbe765"),
    "name" : {
        "first" : "John",
        "last" : "Backus"
    }
}

/* 1 */
{
    "_id" : ObjectId("543efb5260ac6af54acbe766"),
    "name" : {
        "first" : "John",
        "last" : "McCarthy"
    }
}

/* 2 */
{
    "_id" : ObjectId("543efb5360ac6af54acbe767"),
    "name" : {
        "first" : "Grace",
        "last" : "Hopper"
    }
}

/* 3 */
{
    "_id" : ObjectId("543efb5460ac6af54acbe768"),
    "name" : {
        "first" : "Kristen",
        "last" : "Nygaard"
    }
}

/* 4 */
{
    "_id" : ObjectId("543efb5460ac6af54acbe769"),
    "name" : {
        "first" : "Ole-Johan",
        "last" : "Dahl"
    }
}

/* 5 */
```

```
{
    "_id" : ObjectId("543efb5560ac6af54acbe76b"),
    "name" : {
        "first" : "Dennis",
        "last" : "Ritchie"
    }
}
```

MongoDB Query Is:

Correct!

```
db.bios.find({   birth: {     $lt: ISODate("1950-01-01")   } }, {   name: 1 })
```

Correct Answers

```
db.bios.find({ birth: { $lt: ISODate("1950-01-01") } }, { name: 1 })
CLOUD
```

Question 3

2 / 2 pts

Get a Unique Listing of all the Awards (in DB/Collection) granted

Translate the following SQL Query into MongoDB:

```
select distinct(a.award_name)
from person_awards pa, awards a
where pa.award_id = a.award_id
```

The MongoDB Output Should be:

```
/* 0 */
{
    "0" : "Draper Prize",
    "1" : "National Medal of Science",
    "2" : "Turing Award",
    "3" : "W. W. McDowell Award",
    "4" : "Kyoto Prize",
    "5" : "Computer Sciences Man of the Year",
    "6" : "Distinguished Fellow",
    "7" : "National Medal of Technology",
    "8" : "IEEE John von Neumann Medal",
    "9" : "Rosing Prize",
    "10" : "Award for the Advancement of Free Software",
    "11" : "NLUUG Award",
    "12" : "Japan Prize",
    "13" : "Officer of the Order of Canada",
    "14" : "The Economist Innovation Award"
}
```

MongoDB Query Is:

Correct!

```
db.bios.distinct("awards.award")
```

Correct Answers

```
db.bios.distinct("awards.award")
db.bios.distinct( 'awards.award' )
CLOUD
```

Question 4

2 / 2 pts

Get a Sorted Listing of all the First Names (ascending order)

Translate the following SQL Query into MongoDB:

```
select first_name
```

```
from person
order by 1
```

The MongoDB Output Should be:

```
/* 0 */
{
    "_id" : ObjectId("543efb5560ac6af54acbe76b"),
    "name" : {
        "first" : "Dennis"
    }
}

/* 1 */
{
    "_id" : ObjectId("543efb5360ac6af54acbe767"),
    "name" : {
        "first" : "Grace"
    }
}

/* 2 */
{
    "_id" : ObjectId("543efb5460ac6af54acbe76a"),
    "name" : {
        "first" : "Guido"
    }
}

/* 3 */
{
    "_id" : ObjectId("543efb5560ac6af54acbe76d"),
    "name" : {
        "first" : "James"
    }
}

/* 4 */
{
    "_id" : ObjectId("543efb5060ac6af54acbe765"),
    "name" : {
        "first" : "John"
    }
}

/* 5 */
{
    "_id" : ObjectId("543efb5260ac6af54acbe766"),
    "name" : {
        "first" : "John"
    }
}

/* 6 */
{
    "_id" : ObjectId("543efb5460ac6af54acbe768"),
    "name" : {
        "first" : "Kristen"
    }
}

/* 7 */
{
    "_id" : ObjectId("543efb5660ac6af54acbe76e"),
    "name" : {
        "first" : "Martin"
    }
}

/* 8 */
{
    "_id" : ObjectId("543efb5460ac6af54acbe769"),
    "name" : {
        "first" : "Ole-Johan"
    }
}

/* 9 */
{
    "_id" : ObjectId("543efb5560ac6af54acbe76c"),
    "name" : {
        "first" : "Yukihiro"
    }
}
```

MongoDB Query Is:

Correct!

db.bios.find({}, {   'name.first': 1 }).sort({   'name.first': 1 })

Correct Answers

db.bios.find({}, { 'name.first': 1 }).sort({ 'name.first': 1 })

db.bios.find({}, { "name.first": 1 }).sort({ "name.first": 1 })

CLOUD

Question 5

2 / 2 pts

Get a Sorted Listing of all the First Names (descending order)

Translate the following SQL Query into MongoDB:

```
select first_name
from person
order by 1 desc
```

The MongoDB Output Should be:

```
/* 0 */
{
    "_id" : ObjectId("543efb5560ac6af54acbe76c"),
    "name" : {
        "first" : "Yukihiro"
    }
}

/* 1 */
{
    "_id" : ObjectId("543efb5460ac6af54acbe769"),
    "name" : {
        "first" : "Ole-Johan"
    }
}

/* 2 */
{
    "_id" : ObjectId("543efb5660ac6af54acbe76e"),
    "name" : {
        "first" : "Martin"
    }
}

/* 3 */
{
    "_id" : ObjectId("543efb5460ac6af54acbe768"),
    "name" : {
        "first" : "Kristen"
    }
}

/* 4 */
{
    "_id" : ObjectId("543efb5060ac6af54acbe765"),
    "name" : {
        "first" : "John"
    }
}

/* 5 */
{
    "_id" : ObjectId("543efb5260ac6af54acbe766"),
    "name" : {
        "first" : "John"
    }
}

/* 6 */
{
    "_id" : ObjectId("543efb5560ac6af54acbe76d"),
    "name" : {
        "first" : "James"
    }
}

/* 7 */
{
    "_id" : ObjectId("543efb5460ac6af54acbe76a"),
    "name" : {
        "first" : "Guido"
    }
}

/* 8 */
{
    "_id" : ObjectId("543efb5360ac6af54acbe767"),
    "name" : {
        "first" : "Grace"
    }
}

/* 9 */
{
    "_id" : ObjectId("543efb5560ac6af54acbe76b"),
    "name" : {
        "first" : "Dennis"
    }
}
```

MongoDB Query Is:

Correct!

```
db.bios.find({}, {   'name.first': 1 }).sort({   'name.first': -1 })
```

Correct Answers

```
db.bios.find({}, { 'name.first': 1 }).sort({ 'name.first': -1 })
db.bios.find({}, { "name.first": 1 }).sort({ "name.first": -1 })
CLOUD
```

Question 6

2 / 2 pts

Count the number of BIOS that do not yet have an award

Translate the following SQL Query into MongoDB:

```
select count(*) from person p
where not exists
     (select 1 from person_awards
       where person_id = p.person_id)
```

The MongoDB Output Should be:

```
1
```

MongoDB Query Is:

Correct!

```
db.bios.count({   awards: {     $exists: false   } })
```

Correct Answers

```
db.bios.find({ awards: { $exists: false } }).count()
db.bios.count({ awards: { $exists: false } })
CLOUD
```

Question 7

2 / 2 pts

Display the System ID (Primary Key) for the BIO matched in the previous Query

Translate the following SQL Query into MongoDB:

```
select p.person_id from person p
where not exists
   (select 1 from person_awards
    where person_id = p.person_id)
```

The MongoDB Output Should be:

```
/* 0 */
{
   "_id" : ObjectId("543efb5660ac6af54acbe76e")
}
```

MongoDB Query Is:

Correct!

```
db.bios.find({   awards: {     $exists: false   } }, {   _id: true })
```

Correct Answers

```
db.bios.find({ awards: { $exists: false } }, { _id: true })
db.bios.find({ awards: { $exists: false } }, { "_id": true })
CLOUD
```

Question 8

2 / 2 pts

Display names (first and last) from BIOS with 1 Contribution AND 2 Awards

Translate the following SQL Query into MongoDB:

```
select p.first_name, p.last_name
from person p
where (select count(*) from contribs c where c.person_id = p.person_id) = 1
and (select count(*) from person_awards pa where pa.person_id = p.person_id) = 2
```

The MongoDB Output Should be:

```
/* 0 */
{
   "_id" : ObjectId("543efb5460ac6af54acbe76a"),
   "name" : {
      "first" : "Guido",
      "last" : "van Rossum"
   },
   "contribs" : [
      "Python"
   ],
   "awards" : [
      {
         "award" : "Award for the Advancement of Free Software",
         "year" : 2001,
         "by" : "Free Software Foundation"
      },
      {
            "award" : "NLUUG Award",
            "year" : 2003,
            "by" : "NLUUG"
        }
    ]
}

/* 1 */
{
    "_id" : ObjectId("543efb5560ac6af54acbe76d"),
    "name" : {
        "first" : "James",
        "last" : "Gosling"
    },
    "contribs" : [
        "Java"
    ],
    "awards" : [
        {
            "award" : "The Economist Innovation Award",
            "year" : 2002,
            "by" : "The Economist"
        },
        {
            "award" : "Officer of the Order of Canada",
            "year" : 2007,
            "by" : "Canada"
        }
    ]
}
```

MongoDB Query Is:

db.bios.find (

```
    { awards: { $siz ,
    { 'awards': 1, 'co
)
```

Answer 1:

Correct! { awards: { $size: 2 }, contribs: { $size: 1 } }

Correct Answer CLOUD

Answer 2:

Correct! { 'awards': 1, 'contribs': 1, 'name.first': 1, 'name.last': 1 }

Correct Answer CLOUD

Question 9

2 / 2 pts

Display names (first and last) from BIOS with 1 Contributions OR 2 Awards

Translate the following SQL Query into MongoDB:

```
select p.first_name, p.last_name
from person p
where (select count(*) from contribs c where c.person_id = p.person_id) = 1
or (select count(*) from person_awards pa where pa.person_id = p.person_id) = 2
```

The MongoDB Output Should be:

```
/* 0 */
{
    "_id" : ObjectId("543efb5460ac6af54acbe76a"),
    "name" : {
        "first" : "Guido",
        "last" : "van Rossum"
    },
    "contribs" : [
        "Python"
    ],
    "awards" : [
        {
            "award" : "Award for the Advancement of Free Software",
            "year" : 2001,
            "by" : "Free Software Foundation"
        },
        {
            "award" : "NLUUG Award",
            "year" : 2003,
            "by" : "NLUUG"
        }
    ]
}

/* 1 */
{
    "_id" : ObjectId("543efb5560ac6af54acbe76c"),
    "name" : {
        "first" : "Yukihiro",
        "last" : "Matsumoto"
    },
    "contribs" : [
        "Ruby"
    ],
    "awards" : [
        {
            "award" : "Award for the Advancement of Free Software",
            "year" : "2011",
            "by" : "Free Software Foundation"
        }
    ]
}

/* 2 */
{
    "_id" : ObjectId("543efb5560ac6af54acbe76d"),
    "name" : {
        "first" : "James",
        "last" : "Gosling"
    },
    "contribs" : [
        "Java"
    ],
    "awards" : [
        {
            "award" : "The Economist Innovation Award",
            "year" : 2002,
            "by" : "The Economist"
        },
        {
            "award" : "Officer of the Order of Canada",
            "year" : 2007,
            "by" : "Canada"
        }
    ]
}

/* 3 */
{
    "_id" : ObjectId("543efb5660ac6af54acbe76e"),
    "name" : {
        "first" : "Martin",
        "last" : "Odersky"
    },
    "contribs" : [
        "Scala"
    ]
```

}

MongoDB Query Is :

Note: change to formatting for this answer as follows:

- *Braces with Control Statement*
- *Indent with Tabs*
- *Print Margin 80*

db.bios.find (

{ $or: [{ ,

{ 'awards': 1,

)

Answer 1:

Correct! { $or: [{ awards: { $size: 2 } }, { contribs: { $size: 1 } }] }

Correct Answer CLOUD

Answer 2:

Correct! { 'awards': 1, 'contribs': 1, 'name.first': 1, 'name.last': 1 }

Correct Answer CLOUD

Question 10

2 / 2 pts

List all the Awards for a BIO

Translate the following SQL Query into MongoDB:

```
select p.first_name, p.last_name, a.award_name
from awards a, person_awards pa, person p
where a.award_id = pa.award_id
and p.person_id = pa.person_id
and p.person_id = 1
```

The MongoDB Output Should be:

```
/* 0 */
{
    "_id" : ObjectId("543efb5060ac6af54acbe765"),
    "name" : {
        "first" : "John",
        "last" : "Backus"
    },
    "awards" : [
        {
            "award" : "W. W. McDowell Award",
            "year" : 1967,
            "by" : "IEEE Computer Society"
        },
        {
            "award" : "National Medal of Science",
            "year" : 1975,
            "by" : "National Science Foundation"
        },
        {
            "award" : "Turing Award",
            "year" : 1977,
            "by" : "ACM"
        },
        {
            "award" : "Draper Prize",
            "year" : 1993,
            "by" : "National Academy of Engineering"
        }
    ]
}
```

MongoDB Query Is:

Correct!

`db.bios.find({   _id: ObjectId('543efb5060ac6af54acbe765') }, {   awards: 1,   name: 1 })`

Correct Answers

db.bios.find({ _id: ObjectId('543efb5060ac6af54acbe765') }, { awards: 1, name: 1 })

CLOUD

Question 11

10 / 10 pts

Display a list of Awards (names) and a count of how many of them where granted for those granted to two or more recipients

SQL Query:

```
select a.award_name as award, count(*) as granted
from person_awards pa, awards a
where pa.award_id = a.award_id
group by a.award_name
having count(*) >= 2
```

MongoDB Query:

For this question, change to formatting for this answer as follows:

- *Braces with Control Statement*
- *Indent with Tabs*
- *Print Margin 80*

```
db.bios.aggregate( [
                        /* unwind stage */   {     $unwind: "$aw ,
                        /* group stage */    {     $group: {   ,
                        /* project stage */  {     $project: {   ,
                        /* sort stage */     {     $sort: {   ,
                        /* match stage */    {     $match: {
                    ] )
```

MongoDB Output:

```
/* 0 */
{
    "result" : [
        {
            "award" : "Award for the Advancement of Free Software",
            "granted" : 2,
            "granted_gte_2" : true
        },
        {
            "award" : "IEEE John von Neumann Medal",
            "granted" : 2,
            "granted_gte_2" : true
        },
        {
            "award" : "National Medal of Science",
            "granted" : 2,
            "granted_gte_2" : true
        },
        {
            "award" : "National Medal of Technology",
            "granted" : 2,
            "granted_gte_2" : true
        },
        {
            "award" : "Rosing Prize",
            "granted" : 2,
            "granted_gte_2" : true
        },
        {
            "award" : "Turing Award",
            "granted" : 5,
            "granted_gte_2" : true
        },
        {
            "award" : "W. W. McDowell Award",
            "granted" : 2,
            "granted_gte_2" : true
        }
    ],
    "ok" : 1
}
```

Answer 1:

Correct! { $group: { _id: "$awards.award", count: { $sum: 1 } } }

Correct Answer { $group: { _id: '$awards.award', count: { $sum: 1 } } }

Correct Answer CLOUD

Answer 2:

Correct! { $project: { _id: 0, award: "$_id", granted: "$count", granted_gte_2: { $gte: ["$count", 2] } } }

Correct Answer { $project: { _id: 0, award: '$_id', granted: '$count', granted_gte_2: { $gte: ['$count', 2] } } }

Correct Answer CLOUD

Answer 3:

Correct! { $sort: { award: 1 } }

Correct Answer { $sort: { award: 1 } }

Correct Answer CLOUD

Answer 4:

Correct! { $match: { granted_gte_2: true } }

Correct Answer { $match: { granted_gte_2: 1 } }

Correct Answer CLOUD

Quiz Score: **30** out of 30