به نام خدا

تمرین 4 پایگاه داده

آقای دکتر امینی

سارا آذرنوش

98170668

(1

Π(c-name)(σ(b-name=’valiasr’)(Deposit)) − Π(c-name)(σ(b-name=’valiasr’)(Borrow))

Customer ÷ Π(c-city, c-street)(σ(c-name=’reza’)(Customer))

Π(c-name, b-name)(Borrow) ÷ Π(b-name)(Branch)

Π(c-name, c-city, b-name)(Customer ▷◁ Borrow) ÷ Π(b-name)(Customer ▷◁ c-city != b-city Branch) ∩Π(c-name, c-city, b-name)(Customer ▷◁ (Deposit)) ÷ Π(b-name)(σ(b-city=’shiraz’)(Branch))

Π(b-name, c-name)(Borrow) ÷ Π(c-name)(Customer ÷ Π(c-city)(σ(c-name=’sara’)(Customer))) −Π(b-name)(Branch ▷◁ (Customer ÷ Π(c-city)(σ(c-name=’ali’)(Customer))))

(2

rangvar c ranges over customer

rangvar b ranges over branch

rangvar d ranges over deposit

rangvar bo ranges over borrow

c.c-name where exists d (d.c-name = c.c-name and d.balance < 1000 and d.b-name = ‘melli’)

c.c-name where forall b (exists bo (bo.b-name = b.b-name and bo.c-name = c.c-name))

c.c-name where exists b (b.b-city = ’tehran’ and exists d (d.c-name = c.c-name and d.b-name = c.b-name))

c.c-name where forall bo(bo.c-name != c.c-name or exists b (b.b-city = ’isfahan’ and b.b-name = bo.b-name))

(3

(1

a → b, a → c, a → d, b → c, b → d, d → b, (a, b) → c

(2

a → b, b → c, b → d, d → b

(4

R = (A,B,C,D)

R’ = (A,B,C) R3=(C,D)

R1 =(A,B) R2=(B,C)

5)

الف)4

R = (A,B,C,D,E,F,G)

R’ = (B,C,G,E,F) R4=(B,D,A)

R1 =(B,C,E) R2=(E,F) R3=(F,G)

ب)5

R = (A,B,C,D,E,F,G)

R’ = (B,C,G,E,F) R4=(B,D,A)

R’ =(B,C,E) R2=(E,F) R3=(F,G)

R1 =(C,E) R5 =(B,C)

بخش دوم:

1)

با استفاده از لینک گفته شده سرور و تولز را دانلود و نصب میکنیم

2)

```
C:\Users\sa>mongoimport --db hw4 --collection links --file C:\Users\sa\Desktop\db4\hw4\links.json
2022-01-20T22:22:40.780+0330    connected to: mongodb://localhost/
2022-01-20T22:22:40.803+0330    10 document(s) imported successfully. 0 document(s) failed to import.
```

3)

```
C:\Users\sa>mongo
MongoDB shell version v5.0.5
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("7011e0d1-0aa4-4946-8596-78147268 9e2f") }
MongoDB server version: 5.0.5
================
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility.The "mongo" shell has been deprecated and will be removed in
an upcoming release.
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
================
---
The server generated these startup warnings when booting:
        2022-01-20T21:02:56.436+03:30: Access control is not enabled for the database. Read and write access to data and co
nfiguration is unrestricted
---
---
        Enable MongoDB's free cloud-based monitoring service, which will then receive and display
        metrics about your deployment (disk utilization, CPU, operation statistics, etc).

        The monitoring data will be available on a MongoDB website with a unique URL accessible to you
        and anyone you share the URL with. MongoDB may use this information to make product
        improvements and to suggest MongoDB products and deployment options to you.

        To enable free monitoring, run the following command: db.enableFreeMonitoring()
        To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
> use hw4
switched to db hw4
> db.links.find().count();
10
>
```

(4

متن کامل در انتهای PDF پیوست شده است

"_id" : STRING

"author" :

"name" : STRING

"email" : STRING

"twitter" : STRING

"title" : STRING

" body " : STRING

"url" : STRING

"date" : DATE

"starred" : INTEGER

"ratings" : LIST OF INTEGERS

"comments"

"user" : STRING

"upVotes" : INTEGER

"downVotes" : INTEGER

"text" : STRING

"replies" :

"user" : STRING

"upVotes" : INTEGER

"downVotes" : INTEGER

"text" : STRING

"tags" : LIST OF STRING

"draft" : BOOLEAN

"published" : BOOLEAN

(5

links								
id	authorName ----------	title	body	url	date	starred	draft	published

دراینجا اسم ها به نظر یکتا هستند اگر نباشند میتوان یک id درنظر گرفت و به جدول نویسنده نیز افزود و آن کلید اولیه جدول باشد

author		
name	email	twitter

comments					
id	user	upVotes	downVotes	text	linksId ----------

replies					
id	commentsId ----------	user	upVotes	downVotes	text

ratings	
linksId ----------	rating

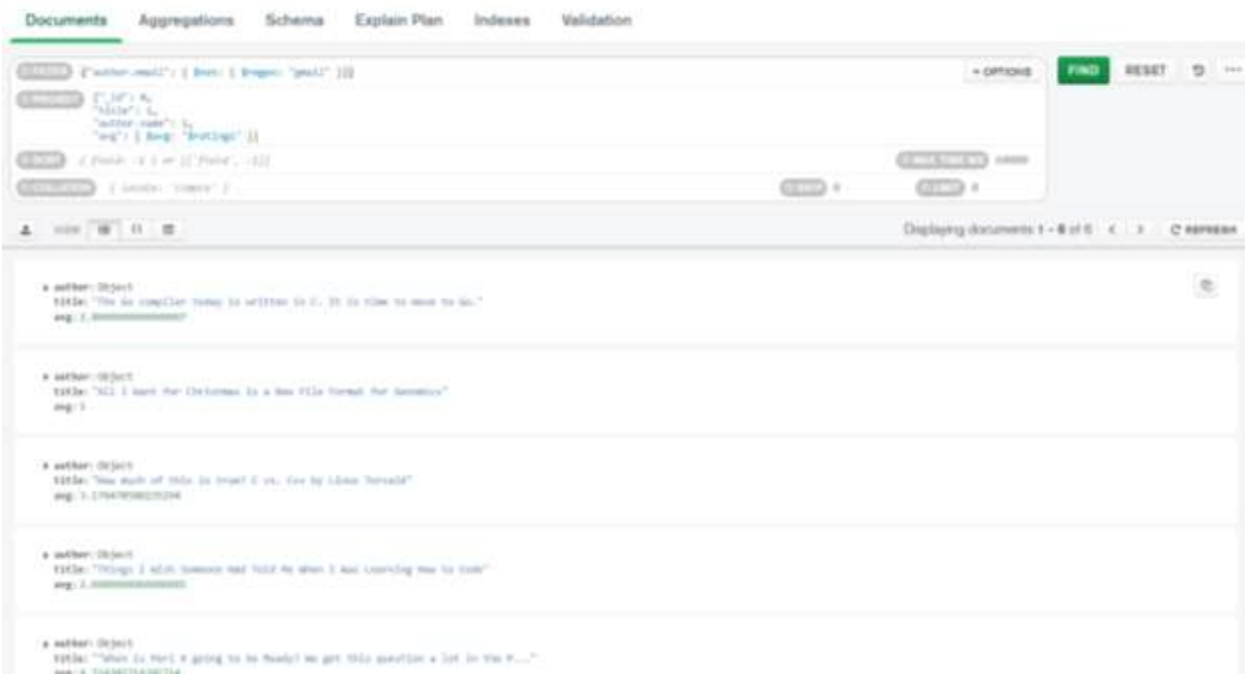
tags	
linksId ----------	tag

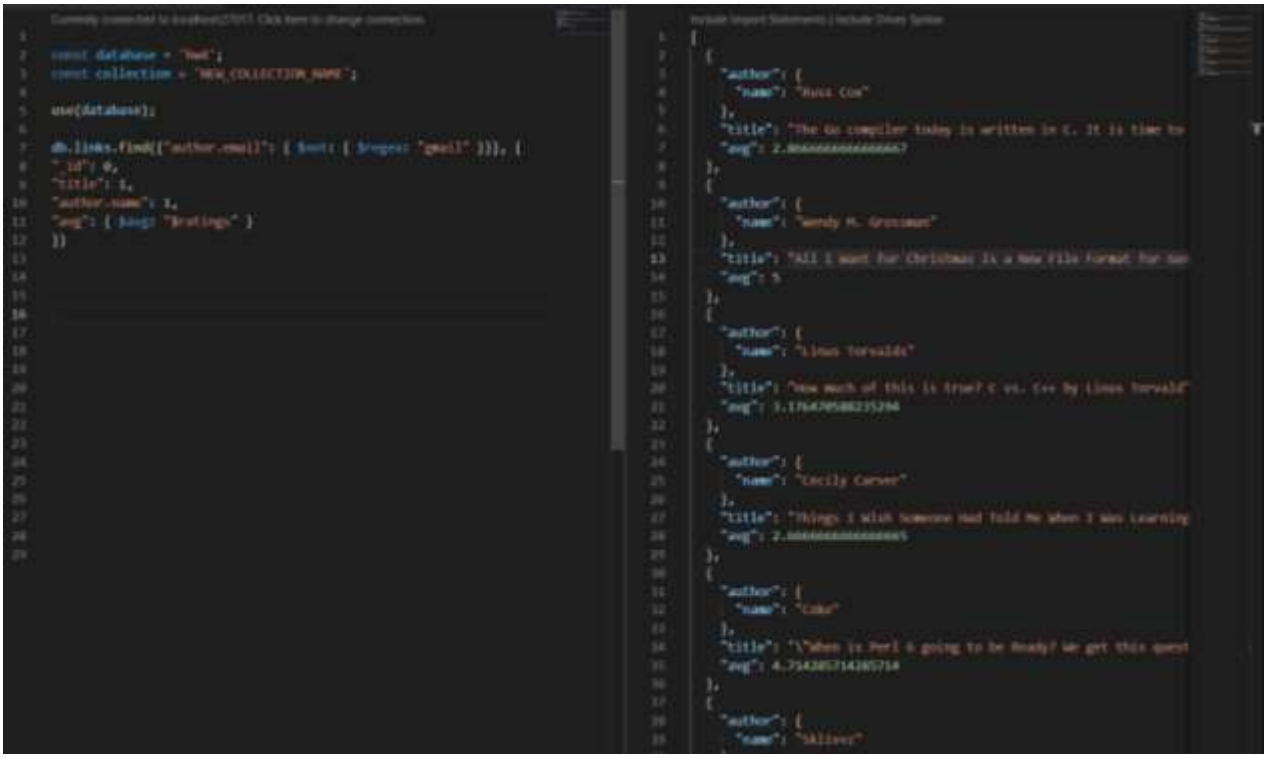
غیر رابطه ای بهتر ایت زیرا داده ها حجم بسیاری دارند و حجم بسیاری نیز مقدار null دارند.

6)

در دو محیط visual studio و mongodb انجام دادم.

الف)





```
db.links.find({"author.email": { $not: { $regex: "gmail" }}}, {
"_id": 0,
"title": 1,
"author.name": 1,
```

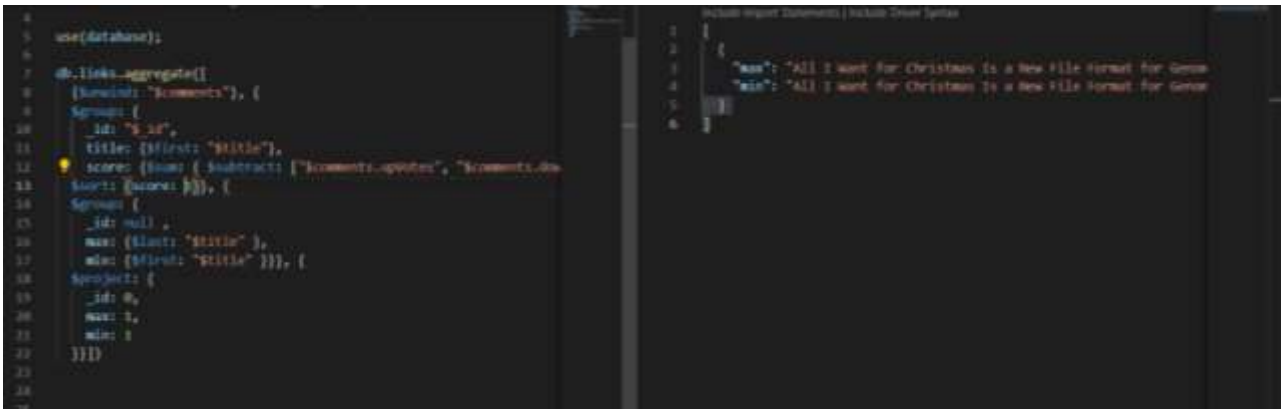
```
"avg": { $avg: "$ratings" }
})
```

خروجی


```
[
  {
    "author": {
      "name": "Russ Cox"
    },
    "title": "The Go compiler today is written in C. It is time to
move to Go.",
    "avg": 2.8666666666666667
  },
  {
    "author": {
      "name": "Wendy M. Grossman"
    },
    "title": "All I Want for Christmas Is a New File Format for
Genomics",
    "avg": 5
  },
  {
    "author": {
      "name": "Linus Torvalds"
    },
    "title": "How much of this is true? C vs. C++ by Linus Torvald",
    "avg": 3.176470588235294
  },
  {
    "author": {
      "name": "Cecily Carver"
    },
    "title": "Things I Wish Someone Had Told Me When I Was Learning
How to Code",
    "avg": 2.6666666666666665
  },
  {
    "author": {
      "name": "Coke"
    },
    "title": "\"When is Perl 6 going to be Ready? We get this question
a lot in the Perl 6 community, and the answer is never...\"",
```

```
    "avg": 4.714285714285714
  },
  {
    "author": {
      "name": "Sklivvz"
    },
    "title": "Has Stack Overflow saved billions of dollars in
programmer productivity?",
    "avg": 3.0833333333333335
  }
]
```

ب)

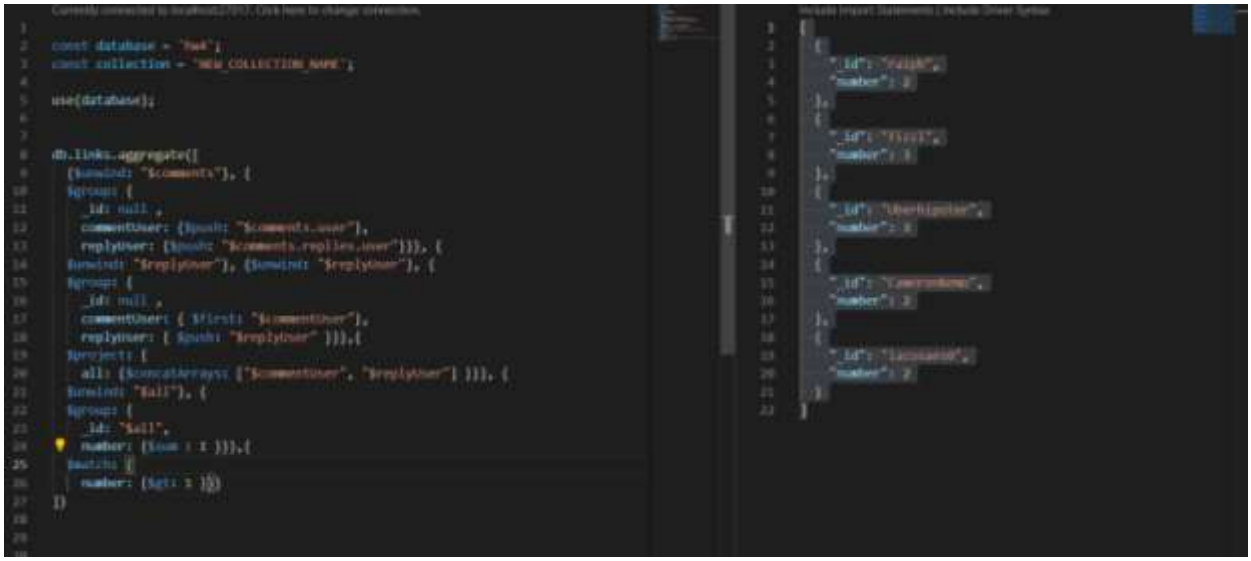


```
db.links.aggregate([
  {$unwind: "$comments"}, {
  $group: {
    _id: "$_id",
    title: {$first: "$title"},
    score: {$sum: { $subtract: ["$comments.upVotes",
"$comments.downVotes"] }}}}, {
  $sort: {score: 1}}, {
  $group: {
    _id: null ,
    max: {$last: "$title" },
    min: {$first: "$title" }}}, {
  $project: {
    _id: 0,
    max: 1,
    min: 1
  }}])
```

خروجی

```
[
  {
    "max": "All I Want for Christmas Is a New File Format for
Genomics",
    "min": "All I Want for Christmas Is a New File Format for
Genomics"
  }
]
```

ج)



```
db.links.aggregate([
  {$unwind: "$comments"}, {
  $group: {
    _id: null ,
    commentUser: {$push: "$comments.user"},
    replyUser: {$push: "$comments.replies.user"}}}, {
  $unwind: "$replyUser"}, {$unwind: "$replyUser"}, {
  $group: {
    _id: null ,
    commentUser: { $first: "$commentUser"},
    replyUser: { $push: "$replyUser" }}},{
  $project: {
    all: {$concatArrays: ["$commentUser", "$replyUser"] }}}, {
  $unwind: "$all"}, {
  $group: {
```

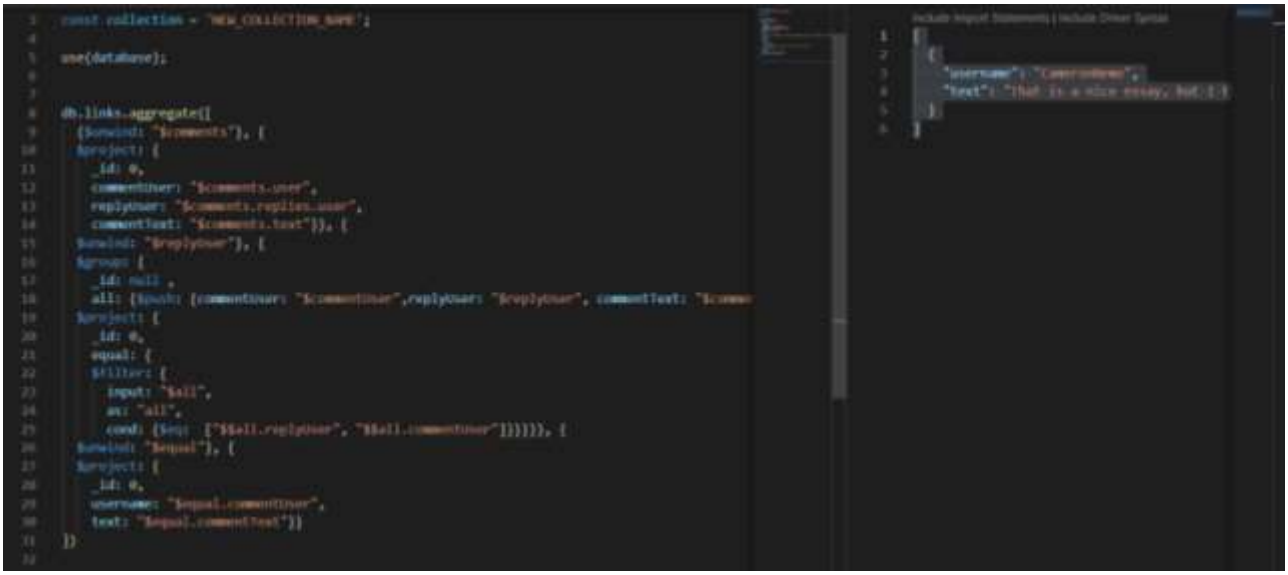
```
    _id: "$all",
    commentNum: {$sum : 1 }}},{
  $match: {
    commentNum: {$gt: 1 }}},{
  $group: {
    _id: null,
    moreComment: {$push: "$$ROOT"},
    userNum: {$sum: 1 }}},{
  $project: {
    _id: 0,
    moreComment: 1,
    userNum:1
    }}
])
```

خروجی


```
[
[
  {
    "moreComment": [
      {
        "_id": "fizzl",
        "commentNum": 3
      },
      {
        "_id": "raiph",
        "commentNum": 2
      },
      {
        "_id": "Uberhipster",
        "commentNum": 3
      },
      {
        "_id": "CameronNemo",
        "commentNum": 2
      },
      {
        "_id": "lacosaes0",
        "commentNum": 2
      }
    ],
    "userNum": 5
```

```
  }
]
```

د)

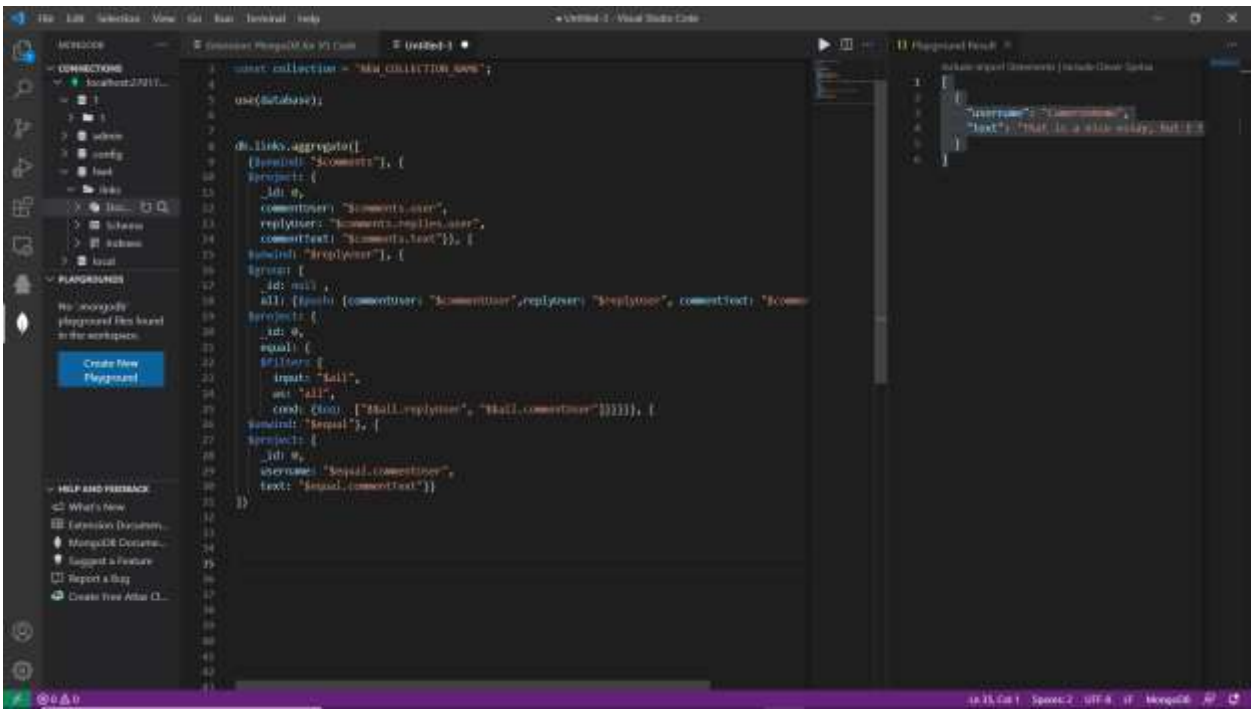


```
db.links.aggregate([
  {$unwind: "$comments"}, {
  $project: {
    _id: 0,
    commentUser: "$comments.user",
    replyUser: "$comments.replies.user",
    commentText: "$comments.text"}}, {
  $unwind: "$replyUser"}, {
  $group: {
    _id: null ,
    all: {$push: {commentUser: "$commentUser",replyUser: "$replyUser",
commentText: "$commentText" }}}}, {
  $project: {
    _id: 0,
    equal: {
    $filter: {
      input: "$all",
      as: "all",
      cond: {$eq:  ["$$all.replyUser", "$$all.commentUser"]}}}}}, {
  $unwind: "$equal"}, {
  $project: {
    _id: 0,
    username: "$equal.commentUser",
    text: "$equal.commentText"}}
```

```
])
```

خروجی

```
[
  {
    "username": "CameronNemo",
    "text": "That is a nice essay, but I think they stress eliminating
other languages too much. Sure, Go may be flexible with the current
compilers, but LLVM is also very flexible, in different ways.
Additionally, I have heard great things about GNU's Gold linker, which
I think can be used from both GCC and LLVM. Complaining about parts of
the compiler being written in C and/or C++ is just inane, and over-
zealous."
  }
]
```



(7

```
db.links.aggregate([
  {$unwind: "$comments"}, {
  $project: {
    _id: 0,
    commentUser: "$comments.user",
    commentUp: "$comments.upVotes",
    commentDown: "$comments.downVotes",
```

```
    }}, {
  $project: {
    _id: 0,
    commentUser: "$commentUser",
    commentUp: "$commentUp",
    commentDown: "$commentDown",
    di: { $divide: ["$commentDown",2] }
    }}, {
  $group: {
    _id: null ,
    all: {$push: {commentUser: "$commentUser", commentUp:
"$commentUp", di: "$di"}}}},{
  $project: {
    _id: 0,
    equal: {
    $filter: {
      input: "$all",
      as: "all",
      cond: {$and: [{ $gt: [ "$$all.commentUp", 20 ] }, { $gt:
["$$all.di", 4]}]}}}}}, {
  $project: {
    _id: 0,
    username: "$equal.commentUser"}}
])
```

متن کامل بخش دوم بخش 4

{

"_id" : ObjectId("52b3833bd3e98582d2bfb627"),

"author" : {

"name" : "Gabe Rudy",

"email" : "gabe@gmail.com",

"twitter" : "@gabeinformatics"

},

"title" : "All I Want for Christmas Is a New File Format for Genomics",

"body" : "Tis the season of quiet, productive hours. I’ve been spending a lot of mine thinking about file formats. Actually I’ve been spending mine implementing a new one, but more on that later.\n\nFile formats are amazingly important in big data science. In genomics, it

is hard not to be awed by how successful the BAM file format is.\n\nHeng Li thinking up another algorithm...\nHeng Li thinking up another algorithm…\nI thought one of the most tweetable moments at ASHG 2013 was when Jeffrey Reid from BCM Human Genome Sequencing Center (HGSC) talked about how they offloaded to the cloud (via DNAnexus) 2.4 million hours of compute time to perform the alignment and variant calling on ~4k genomes and ~12k exomes.",

"url" : "http://blog.goldenhelix.com/?p=2032",

"date" : ISODate("2013-12-16T13:23:00Z"),

"starred" : 9,

"ratings" : [

1,

5,

4,

3,

2,

1,

5,

2,

4,

2,

3,

4,

1,

3

],

"comments" : [

{

"user" : "rp21a",

"upVotes" : 2,

```
            "downVotes" : 0,
            "text" : "Very impressive work. I looked at your website to see what kind of analysis you guys do, but I can't tell. What are the hard algorithmic problems you guys face? And is there anything you can only dream of being able to do?",
            "replies" : [
                {
                    "user" : "tinyNP",
                    "upVotes" : 0,
                    "downVotes" : 0,
                    "text" : "http://en.wikipedia.org/wiki/Sequence_alignment"
                }
            ]
        },
        {
            "user" : "rick2g",
            "upVotes" : 3,
            "downVotes" : 1,
            "text" : "I still get told by virologists that fasta is all that's needed."
        },
        {
            "user" : "holocauster-ride",
            "upVotes" : 0,
            "downVotes" : 5,
            "text" : "Have you tried XML?"
        }
    ],
    "tags" : [
        "Biology",
```

```
            "Computer"
        ],
        "draft" : false,
        "published" : true
}
{
        "_id" : ObjectId("52b3833bd3e98582d2bfb628"),
        "author" : {
            "name" : "Graydon Hoare",
            "email" : "graydon@gmail.com"
        },
        "title" : "Why Rust ditched pure functions",
        "body" : "On 30/04/2013 10:08 AM, Max Cantor wrote:\n\n> I know this will be an unpopular opinion, but pure functions would be a\n> massive win for Rust, especially if the eventual goal is high\n> performance, highly parallelizable (browser rendering engines..)\n> development.\n\nCareful. It's important to understand that \"purity\" seems like it has a \nsimple definition but in languages with mutable memory, state, time and \nIO, it gets hard to be exact.\n\nThings you can throw at a SIMD unit or GPU and get parallel kernels out \nof them will almost certainly be a different version of \"pure\" than \nthings you can evaluate at compile time. Or, as in our previous attempts \nat defining it, things that won't break existing borrows or existing \ntypestates. Each of these is a static approximation of the \nset-of-all-things-a-function-might-do. Since our functions can generally \ndo quite a lot, the set of possible subsets you might mean by \"pure\" is \ncorrespondingly much larger.\n\n> The typestate system did seme very complex but isn't there a middle\n> ground via annotations perhaps?  A subset of primitives and core\n> functions can be annotated as pure and then any function calling only\n> pure functions can itself be annotated as pure.\n\nThis gets difficult fast. You wind up dividing your functions up into \ngroups and then getting annoyed that one that's \"mostly almost pure\" or \n\"essentially pure, for my purposes\" that you wanted to call actually \nisn't (someone forgot to mark it as such, or some sub-function, or some \ntrait-implied function) and then you can't. Or is pure using one way of \nthinking about purity, but not another. Or is pure except for the bit \nwhere it calls unsafe but promises it's going to maintain purity, just \nknows better than you (oops, that can't be done at compile time, nor on \na GPU, etc.)\n\nC++ has multiple concepts for this, each a not-entirely-obvious subset \nof the others, each affecting the others, and causing quite a lot of \nwork just to get things to
```

compile, much less reuse code.\n\nThey have const methods (don't mutate the object, unless you lie and \noverride it) and constexpr (can be evaluated at compile time), and \nmacros (can only operate on tokens), and template evaluation (can only \noperate on certain non-type args), and the openCL __kernel extension for \nGPU-applicable functions:",

"url" : "http://thread.gmane.org/gmane.comp.lang.rust.devel/3674/focus=3855",

"date" : ISODate("2013-04-30T13:23:00Z"),

"starred" : 105,

"ratings" : [

3,

5,

3,

2,

4,

1,

3,

3,

3,

2,

3

],

"comments" : [

{

"user" : "tr0lltherapy",

"upVotes" : 18,

"downVotes" : 2,

"text" : "they've made the right decision here. if people want to experiment with purity, there are already mature tools available. rust wants to focus on performance and be a tool a lot of people will use. this decision helps",

"replies" : [

```
                {
                    "user" : "thedeemon",
                    "upVotes" : 10,
                    "downVotes" : 0,
                    "text" : "On the other hand purity often allows more aggressive optimizations, like not calling a function twice with same arguments if compiler knows it's pure. This is what D compiler does in some simple cases, for example."
                },
                {
                    "user" : "mcandre",
                    "upVotes" : 0,
                    "downVotes" : 5,
                    "text" : "Performance? There are already a slew of performant languages. Assembler, C, C++, Go. What does Rust actually offer that's new and useful in this category, other than using my favorite abbreviation for the named function keyword, fn?"
                },
                {
                    "user" : "lacosaes0",
                    "upVotes" : 30,
                    "downVotes" : 6,
                    "text" : "Particular emphasis on memory safety."
                }
            ]
        },
        {
            "user" : "hypster",
            "upVotes" : 30,
            "downVotes" : 2,
```

```
            "text" : "tl;dr everybody was type-fu fighting",
            "replies" : [
                {
                    "user" : "homoiconic",
                    "upVotes" : 15,
                    "downVotes" : 0,
                    "text" : "Here comes the Big Boss, Hu! Simon Peyton-Jones."
                }
            ]
        }
    ],
    "tags" : [
        "Rust",
        "Computer",
        "Programming"
    ],
    "draft" : true,
    "published" : true
}
{
    "_id" : ObjectId("52b3833bd3e98582d2bfb629"),
    "author" : {
        "name" : "Russ Cox",
        "email" : "russ@google.com",
        "twitter" : "@russ"
    },
    "title" : "The Go compiler today is written in C. It is time to move to Go.",
```

"body" : "Abstract\nThe Go compiler today is written in C. It is time to move to Go.\nBackground\nThe “gc” Go toolchain is derived from the Plan 9 compiler toolchain. The assemblers, C\ncompilers, and linkers are adopted essentially unchanged, and the Go compilers (in cmd/gc,\ncmd/5g, cmd/6g, and cmd/8g) are new C programs that fit into the toolchain.\nWriting the compiler in C had some important advantages over using Go at the start of the\nproject, most prominent among them the fact that, at first, Go did not exist and so could not be\nused to write a compiler, and the fact that, once Go did exist, it often changed in significant,\nbackwards-incompatible ways. Using C instead of Go avoided both the initial and ongoing\nbootstrapping problems. Today, however, Go does exist, and its definition is stable as of Go 1,\nso the problems of bootstrapping are greatly reduced.\nAs the bootstrapping problems have receded, other engineering concerns have arisen that\nmake Go much more attractive than C for the compiler implementation. The concerns include:",

"url" : "https://docs.google.com/document/d/1P3BLR31VA8cvLJLfMibSuTdwTuF7WWLux71CYD0eeD8/preview?sle=true&pli=1",

"date" : ISODate("2013-12-01T13:23:00Z"),

"starred" : 256,

"ratings" : [

3,

3,

1,

2,

2,

3,

3,

5,

4,

5,

4,

2,

1,

```
        2,
        3
    ],
    "comments" : [
        {
            "user" : "CameronNemo",
            "upVotes" : 55,
            "downVotes" : 5,
            "text" : "That is a nice essay, but I think they stress eliminating other languages too much. Sure, Go may be flexible with the current compilers, but LLVM is also very flexible, in different ways. Additionally, I have heard great things about GNU's Gold linker, which I think can be used from both GCC and LLVM. Complaining about parts of the compiler being written in C and/or C++ is just inane, and over-zealous.",
            "replies" : [
                {
                    "user" : "YEPHENAS",
                    "upVotes" : 32,
                    "downVotes" : 2,
                    "text" : "The author of the gold linker, Ian Taylor, is in the Go team."
                },
                {
                    "user" : "mlu777",
                    "upVotes" : 3,
                    "downVotes" : 1,
                    "text" : "Go-ld..."
                },
                {
                    "user" : "aport",
```

```
                "upVotes" : 0,
                "downVotes" : 0,
                "text" : "http://i.imgur.com/sg017lt.gif"
            },
            {
                "user" : "zsaleeba",
                "upVotes" : 7,
                "downVotes" : 0,
                "text" : "Also gold is used in gccgo with the --with-ld command line option. So it's not like they're ignoring these tools at all."
            },
            {
                "user" : "CameronNemo",
                "upVotes" : 5,
                "downVotes" : 1,
                "text" : "EXACTLY! These projects are together on their goals and members, they should be together on their technology."
            },
            {
                "user" : "idiot_with_a_gun",
                "upVotes" : 2,
                "downVotes" : 2,
                "text" : "If writing Go in Go makes it easier to get compiler devs and makes it a better compiler, great! If it doesn't, don't do it!"
            }
        ]
    },
    {
```

```
            "user" : "Jurily",
            "upVotes" : 22,
            "downVotes" : 4,
            "text" : "For comparison, once upon a time Rust had a move keyword, classes, pure functions, green threads, garbage collection and a whole lot of other stuff that sounded good on paper but didn't work out in practice. Now it has move by default, type classes, owned pointers (GC, per-task refcounting and atomic refcounting available as libraries), a 1:1 threading backend is in the works after the M:N runtime was already completely rewritten once (and may or may not be removed in the future) and the verdict for purity was kill it with fire. And since rustc uses LLVM, those improvements came for free over the last couple of years.",
            "replies" : [
                {
                    "user" : "ahy1",
                    "upVotes" : 6,
                    "downVotes" : 3,
                    "text" : "I am just curious. Do you know the reason why pure functions didn't work out in practice?"
                },
                {
                    "user" : "kibwen",
                    "upVotes" : 7,
                    "downVotes" : 3,
                    "text" : "See http://thread.gmane.org/gmane.comp.lang.rust.devel/3674/focus=3855 for an explanation by Graydon Hoare on why purity just didn't work out for Rust."
                }
            ]
        }
    ],
    "tags" : [
```

```
            "Go",
            "Computer",
            "Programming"
        ],
        "draft" : false,
        "published" : true
}
{
        "_id" : ObjectId("52b3833bd3e98582d2bfb62a"),
        "author" : {
            "name" : "ssg",
            "email" : "ssg@gmail.com",
            "twitter" : "@ssg"
        },
        "title" : "Hello. I'm a compiler.",
        "body" : "I just scanned thousands of lines of code while you were reading this sentence. I browsed through millions of possibilities of optimizing a single line of yours using hundreds of different optimization techniques based on a vast amount of academic research that you would spend years getting at. I won't feel any embarrassment, not even a slight ick, when I convert a three-line loop to thousands of instructions just to make it faster. I have no shame to go to great lengths of optimization or to do the dirtiest tricks. And if you don't want me to, maybe for a day or two, I'll behave and do it the way you like. I can transform the methods I'm using whenever you want, without even changing a single line of your code. I can even show you how your code would look in assembly, on different processor architectures and different operating systems and in different assembly conventions if you'd like. Yes, all in seconds. Because, you know, I can; and you know, you can't.",
        "url" : "http://stackoverflow.com/questions/2684364/why-arent-programs-written-in-assembly-more-often/2685541#2685541",
        "date" : ISODate("2013-02-19T13:23:00Z"),
        "starred" : 1939,
        "ratings" : [
```

```
            5,
            4,
            2,
            3,
            5,
            2,
            3,
            4,
            2,
            3,
            1,
            3,
            3
        ],
        "comments" : [
            {
                "user" : "zip117",
                "upVotes" : 150,
                "downVotes" : 21,
                "text" : "Unless you're Kazushige Goto. See: GotoBLAS, now maintained as
OpenBLAS.",
                "replies" : [
                    {
                        "user" : "JohannWolfgangGoatse",
                        "upVotes" : 150,
                        "downVotes" : 11,
                        "text" : "Isn't that the guy who is considered harmful nowadays?"
```

```
                    },
                    {
                        "user" : "yerfatma",
                        "upVotes" : 10,
                        "downVotes" : 0,
                        "text" : "Thought that was Hans Reiser.\nDeprecated in favor of mysqli_real_escape_string, soon to be replaced by mysqli_real_escape_string_no_really_I_mean_it_give_me_back_my_lunch."
                    },
                    {
                        "user" : "ThunderWalrus",
                        "upVotes" : 22,
                        "downVotes" : 1,
                        "text" : "I was talking to a guy who was trying to reverse engineer the techniques to apply them to another problem. Apparently, Goto uses register/instruction scheduling techniques that are hard to coerce a C compiler to use without a lot of trickery."
                    }
                ]
            },
            {
                "user" : "ocharis",
                "upVotes" : 489,
                "downVotes" : 12,
                "text" : "\"I love you, mr. compiler. Now please stop caring so much about types.\" has 39 votes.",
                "replies" : [
                    {
                        "user" : "naoprakov",
                        "upVotes" : 12,
```

```
                    "downVotes" : 2,
                    "text" : "If the compiler didn't worry about types, I'm pretty sure I would have blown up my house by now."
                }
            ]
        },
        {
            "user" : "monkeycalculator",
            "upVotes" : 55,
            "downVotes" : 2,
            "text" : "While I was reading the entry its vote count kept increasing. I guess there's a lot of readers from here and potentially other places. I didn't know they broadcast increments in real-time. Cool!",
            "replies" : [
                {
                    "user" : "sebf",
                    "upVotes" : 22,
                    "downVotes" : 5,
                    "text" : "I didn't know it either. I'm impressed and jealous."
                },
                {
                    "user" : "achshar",
                    "upVotes" : 4,
                    "downVotes" : 0,
                    "text" : "It's first spot on HN ATM. Major traffic is coming from there i suppose."
                },
                {
```

```
                    "user" : "sevenletters",
                    "upVotes" : 22,
                    "downVotes" : 2,
                    "text" : "It is not line count that makes the C++ compilers slow, it is how the features of the language are specified (templates, overloading, virtual methods to name three). Look at any C++ compiler's RAM usage - that is not because of line count, that is because of features that require extensive changes in the AST post initial parsing based on later input."
                }
            ]
        }
    ],
    "tags" : [
        "Compilers",
        "Computer"
    ],
    "draft" : true,
    "published" : false
}
{
    "_id" : ObjectId("52b3833bd3e98582d2bfb62e"),
    "author" : {
        "name" : "Wendy M. Grossman",
        "twitter" : "@wendyg"
    },
    "title" : "All I Want for Christmas Is a New File Format for Genomics",
    "body" : "R.I.P. John McCarthy, father of AI, inventor of Lisp, suddenly at home last night. Pls RT.",
    "url" : "https://twitter.com/wendyg/status/128554733714669568",
```

```
"date" : ISODate("2011-10-24T13:23:00Z"),
"starred" : 209,
"ratings" : [
        5,
        5,
        5,
        5,
        5
],
"comments" : [
        {
                "user" : "benfitzg",
                "upVotes" : 1900,
                "downVotes" : 300,
                "text" : ")))))))))))))))))))))))))))))))))))))))))))))))))))))))))))",
                "replies" : [
                        {
                                "user" : "jmc",
                                "upVotes" : 10,
                                "downVotes" : 0,
                                "text" : ":-("
                        },
                        {
                                "user" : "code-affinity",
                                "upVotes" : 1022,
                                "downVotes" : 9,
```

```
                    "text" : "I think this was an appropriate comment. Anyone who codes in Lisp knows what it means to type a long series of close parens -- the more parens, the bigger the work being brought to a close."
                },
                {
                    "user" : "code",
                    "upVotes" : 60,
                    "downVotes" : 1,
                    "text" : "Unfortunately, in real life, we seldom get the opportunity to close all of those parens. Life just ends with a whole bunch of unmatched open parens."
                },
                {
                    "user" : "joedev_net",
                    "upVotes" : 500,
                    "downVotes" : 200,
                    "text" : "You win the award for most depressing coding analogy! Congrats!"
                }
            ]
        },
        {
            "user" : "lawpoop",
            "upVotes" : 17,
            "downVotes" : 2,
            "text" : "Life never ends; it always CONTINUEs;",
            "replies" : [ ]
        }
    ],
```

```
        "tags" : [
                "Lisp",
                "Programming",
                "AI"
        ],
        "draft" : false,
        "published" : false
}
{
        "_id" : ObjectId("52b3833bd3e98582d2bfb62c"),
        "author" : {
                "name" : "Linus Torvalds",
                "email" : "torvalds@linux-foundation.org"
        },
        "title" : "How much of this is true? C vs. C++ by Linus Torvald",
        "body" : "On Wed, 5 Sep 2007, Dmitry Kakurin wrote:\n> \n> When I first looked at Git source code two things struck me as odd:\n> 1. Pure C as opposed to C++. No idea why. Please don't talk about portability,\n> it's BS.\n\n*YOU* are full of bullshit.\n\nC++ is a horrible language. It's made more horrible by the fact that a lot \nof substandard programmers use it, to the point where it's much much \neasier to generate total and utter crap with it. Quite frankly, even if \nthe choice of C were to do *nothing* but keep the C++ programmers out, \nthat in itself would be a huge reason to use C.\n\nIn other words: the choice of C is the only sane choice. I know Miles \nBader jokingly said \"to piss you off\", but it's actually true. I've come \nto the conclusion that any programmer that would prefer the project to be \nin C++ over C is likely a programmer that I really *would* prefer to piss \noff, so that he doesn't come and screw up any project I'm involved with.",
        "url" : "http://article.gmane.org/gmane.comp.version-control.git/57918/",
        "date" : ISODate("2007-09-06T13:23:00Z"),
        "starred" : 900,
        "ratings" : [
```

```
            5,
            4,
            2,
            5,
            1,
            4,
            1,
            2,
            4,
            3,
            2,
            5,
            4,
            1,
            4,
            5,
            2
        ],
        "comments" : [
            {
                "user" : "ENOENT",
                "upVotes" : 500,
                "downVotes" : 145,
                "text" : "You need to put it in context. There were times when lkml had occasional invasions of C++ trolls like above. People that had no idea how kernel worked, what massive effort was put to make things reliable and fast while still keeping clean, readable code base. Imagine a mailing list where participants discuss best page cache invalidation strategy, power aware scheduling schemes, lockless dcache walks, etc. and then suddenly someone pops in and
```

tells you the biggest problem you have is lack of RAII - without having any clue what your real current problems are and how things really work.",

"replies" : [

{

"user" : "teambob",

"upVotes" : 7,

"downVotes" : 2,

"text" : "Git maintenance was handed over in 2005, so why is this person even asking about it in the kernel mailing list?"

},

{

"user" : "torh",

"upVotes" : 7,

"downVotes" : 4,

"text" : "It's not from the kernel mailing list, it's from the git mailinglist which makes the fact that the question is from 2007 irrelevant."

},

{

"user" : "nomad",

"upVotes" : 9,

"downVotes" : 2,

"text" : "Linus makes some good points, but this post is steeped in hyperbole. It's one (very respected) programmer's opinion on the matter, but if you're susceptible to arguments from authority, there are equally respected programmers who would argue for the utility of (at least certain parts of) C++. I agree that for system-level programming, C is still a very strong choice. However, it's not prima facie obvious that a version control system is a system-level application."

}

]

},

```
{
    "user" : "DEADBEEF",
    "upVotes" : 14,
    "downVotes" : 2,
    "text" : "I've actually hear him claim the opposite. Particularly since he only rates himself as a 7 on a 10 scale for C++ programming. I heard him state in presentation that he only utilizes certain aspects of C++. Remember C++ is standardized through an ISO committee. Last time I checked not all C++ compilers implement all aspects of C++ language... looks as Microsoft.",
    "replies" : [
        {
            "user" : "el_muchsto",
            "upVotes" : 3,
            "downVotes" : 0,
            "text" : "Actually, he has said just the opposite."
        },
        {
            "user" : "xandoid",
            "upVotes" : 99,
            "downVotes" : 7,
            "text" : "RAII is very powerful and I think Linus underestimates its usefulness. BUT it has to be said that it was the introduction of exceptions that made RAII so absolutely indispensible in C++. Adding exceptions to a non garbage collected language is a horrible mistake in my view. That this mistake has helped popularize RAII is a fortunate accident, but you can't blame C advocates for not appreciating a C++ feature that solves a problem that was massively exacerbated by another C++ feature that C doesn't have."
        },
        {
            "user" : "tejp",
            "upVotes" : 26,
```

```
                "downVotes" : 0,
                "text" : "Resource management on error handling is an absolute pain in C, without real good solutions. If your functions calls return error codes that you want to pass on, you have lots of points where you can exit your function and where you need to cleanup the function's resources (each error check after each function call). RAII would help."
            },
            {
                "user" : "goat",
                "upVotes" : 23,
                "downVotes" : 3,
                "text" : "I once worked on a codebase where everything was like this. Every error was logged every level going up, so we got stack traces that included file lines and numbers in our error logs. Every function had an int status, a cleanup: label, and only ever returned once, after the cleanup: label."
            },
            {
                "user" : "Decker",
                "upVotes" : 75,
                "downVotes" : 2,
                "text" : "Basically, in this case you do what a modern compiler can do automatically."
            }
        ]
    },
    {
        "user" : "GUIpsp",
        "upVotes" : 55,
        "downVotes" : 4,
        "text" : "goto",
```

```
            "replies" : [
                {
                    "user" : "Zenith",
                    "upVotes" : 26,
                    "downVotes" : 1,
                    "text" : "We have the technology..."
                },
                {
                    "user" : "iprobablydisagree",
                    "upVotes" : 6,
                    "downVotes" : 1,
                    "text" : "i'll take goto over longjmp any day"
                },
                {
                    "user" : "mbrx",
                    "upVotes" : 8,
                    "downVotes" : 3,
                    "text" : "That is a good comment (for the discussion) that I belive people not used to C probably might not realize why it is relevant to the discussion. Goto is not the catch-all evil and it is surprisingly easy to make clean resource allocation cleanup on error handling using it."
                }
            ]
        }
    ],
    "tags" : [
        "C",
        "C++",
```

```
            "Programming"
        ],
        "draft" : false,
        "published" : true
}
{
        "_id" : ObjectId("52b3833bd3e98582d2bfb62f"),
        "author" : {
            "name" : "Cecily Carver"
        },
        "title" : "Things I Wish Someone Had Told Me When I Was Learning How to Code",
        "body" : "Before you learn to code, think about what you want to code\nKnowing how to code is mostly about building things, and the path is a lot clearer when you have a sense of the end goal. If your goal is “learn to code,” without a clear idea of the kinds of programs you will write and how they will make your life better, you will probably find it a frustrating exercise.\n\nI’m a little ashamed to admit that part of my motivation for studying computer science was that I wanted to prove I was smart, and I wanted to be able to get Smart Person jobs. I also liked thinking about math and theory (this book blew my mind at an impressionable age) and the program was a good fit. It wasn’t enough to sustain me for long, though, until I found ways to connect technology to the things I really loved, like music and literature.\n\nSo, what do you want to code? Websites? Games? iPhone apps? A startup that makes you rich? Interactive art? Do you want to be able to impress your boss or automate a tedious task so you can spend more time looking at otter pictures? Perhaps you simply want to be more employable, add a buzzword to your resume, or fulfill the requirements of your educational program. All of these are worthy goals. Make sure you know which one is yours, and study accordingly.",
        "url" : "https://medium.com/learning-to-code/565fc9dcb329",
        "date" : ISODate("2013-11-25T13:23:00Z"),
        "starred" : 6,
        "ratings" : [
            1,
            2,
```

```
            2,
            2,
            5,
            2,
            4,
            3,
            3
        ],
        "comments" : [
            {
                "user" : "superking2",
                "upVotes" : 300,
                "downVotes" : 10,
                "text" : "\"It will never work the first time\" is, in my opinion, the single most valuable piece of advice from this list that all coders need to know. If I could have understood this I would have started coding 13 years ago (instead of 2).",
                "replies" : [
                    {
                        "user" : "fizzl",
                        "upVotes" : 181,
                        "downVotes" : 0,
                        "text" : "A. Don't overengineer."
                    },
                    {
                        "user" : "fizzl",
                        "upVotes" : 15,
                        "downVotes" : 0,
                        "text" : "B. Always design properly."
```

```
                    },
                    {
                        "user" : "fizzl",
                        "upVotes" : 15,
                        "downVotes" : 5,
                        "text" : "Surprisingly, my best project is one where I wrote the design from technical spec, which was written by my manager from customer requirements. It was executed by an experienced lead developer with two junior devs."
                    }
                ]
            }
        ],
        "tags" : [
            "Programming"
        ],
        "draft" : true,
        "published" : false
}
{
        "_id" : ObjectId("52b3833bd3e98582d2bfb62d"),
        "author" : {
            "name" : "Coke",
            "email" : "coke@perl.org"
        },
        "title" : "\"When is Perl 6 going to be Ready? We get this question a lot in the Perl 6 community, and the answer is never...\"",
        "body" : "When is Perl 6 going to be Ready? We get this question a lot in the Perl 6 community, and the answer is never\nas simple as we or the inquirers would like.\n\nOne part of the answer involves the specification; When we have an implementation that passes all of
```

the tests marked “perl 6.0”, that will be a Perl 6.\n\nMany people think of the specs as the Synopses, but Patrick Michaud makes a good point that the specification is really more about the tests.\n\nThus it was recognized early on (in Synopsis 1) that acceptance tests provide a far more objective measure of specification conformance than an English description. There are likely things that need to be “spec” that cannot be fully captured by testing… but I still believe that the test suite should be paramount.\n\nEvery language feature must have corresponding spec tests. Trying to find a test? Tests are broken up first by Synopsis (which themselves follow the numbering scheme of the Camel chapters), with multiple directories broken out by a group of features, then individual tests. For example, Synopsis 4 (S04) is about Blocks and Statements, including phasers. So to find the tests for the BEGIN phaser, you’ll want S04-phasers/begin.t in the roast suite.\n\nWe call the specification tests roast to follow in the tradition of “smoke test”, and also because TimToady can’t resist a punny backronym: “Repository Of All Spec Tests’.\n\nEach of these files tries to thoroughly test something from the Synopsis including a lot of edge cases that aren’t necessarily mentioned in the prose. This goes to Patrick’s point about the tests being the more canonical answer about what the spec is.\n\nAre we there yet?\nJust a little further… ",

"url" : "http://perl6advent.wordpress.com/2013/12/13/day-13-roasting-rakudo-star/",

"date" : ISODate("2013-12-13T13:23:00Z"),

"starred" : 19,

"ratings" : [

5,

4,

5,

5,

4,

5,

5

],

"comments" : [

{

"user" : "skullgnome",

"upVotes" : 4,

```
"downVotes" : 0,

"text" : "So when will it be a proper substitute for Perl 5? We've been waiting for a decade now, and Perl 5 just keeps getting better while retaining backward compatibility and only reaching for the blue sky one step at a time.",

"replies" : [

    {

        "user" : "riph",

        "upVotes" : 0,

        "downVotes" : 2,

        "text" : "Is a lisp \"a proper substitute for P5\"? An ML?"

    },

    {

        "user" : "skulgnome",

        "upVotes" : 3,

        "downVotes" : 1,

        "text" : "Neither of them are Perly enough: Lisp has too little syntax, and ML's syntax is mostly unhelpful. Both have the additional failure of being awful for knocking out a quick hack in."

    },

    {

        "user" : "everyname",

        "upVotes" : 6,

        "downVotes" : 5,

        "text" : "...as simple as we or the inquirers would like."

    },

    {

        "user" : "lacosaes0",

        "upVotes" : 7,
```

```
                    "downVotes" : 1,
                    "text" : "The real question is: who gives a shit about Perl 6?"
                }
            ]
        },
        {
            "user" : "educated_poo",
            "upVotes" : 7,
            "downVotes" : 2,
            "text" : "Clearly the people spending all of their time writing advent articles about how amazing it's going to be when it finally advents (but none of their time actually adventing it).",
            "replies" : [
                {
                    "user" : "xiong",
                    "upVotes" : 3,
                    "downVotes" : 0,
                    "text" : "I feel like they may be subconsciously procrastinating because the release of perl 6 will put to test all the claims they've been making for the last decade. Right now they can say \"oh, no one uses it because it's not finished\", but when it releases and still no one uses it, what are they going to do?"
                },
                {
                    "user" : "raiph",
                    "upVotes" : 0,
                    "downVotes" : 0,
                    "text" : "To all: educated_poo posted this reddit. Why do you think he did that?"
                }
```

```
            ]
        },
        {
            "user" : "raiph",
            "upVotes" : 0,
            "downVotes" : 1,
            "text" : "There's a small community of folk who do, most of whom hang out on the freenode IRC channel #perl6.",
            "replies" : [ ]
        }
    ],
    "tags" : [
        "Perl",
        "Computer",
        "Programming"
    ],
    "draft" : false,
    "published" : true
}
{
    "_id" : ObjectId("52b3833bd3e98582d2bfb630"),
    "author" : {
        "name" : "Sklivvz"
    },
    "title" : "Has Stack Overflow saved billions of dollars in programmer productivity?",
    "body" : "John Carmack is a reknown developer and CTO. His Twitter account has over 100,000 followers.\n\nIn occasion of Stack Exchange's 5th anniversary, he quipped on Twitter:\n\n[... Stack Overflow] has probably added billions of dollars of value to the world in
```

increased programmer productivity.\nThe tweet was widely read and retweeted over 100 times.\n\nIs the claim supported by evidence?\n\nWhat have I tried?\n\nI've made a quick calculation to see if he was completely off the mark, but the numbers seem to add up to a layman like me.\n\nOur visit counter recently overflowed Int32: 2,147,483,647\nthe above is a reference to my own tweet but I can also provide supporting evidence: the site is 5 years old and currently does around 6m visits/day\nAverage pay for a programmer in the UK is 45,000£/annum (source) or 22.5£/h. Let's assume a low pay of 20US$/h. I assume a hour is saved every time an answer is provided.\nEach visit will potentially save some money, but only 77% of visits land on answered questions\nThere is a cost relative to answering a question, however answers are useful to hundreds of people.\nThere are many other factors I am not calculating: for example, does answering a question make you more knowledgeable? Does finding an answer make you less knowledgeable than you would be otherwise? I am ignoring these as this is a ballpark estimate, certainly not a valid answer to my own question.",

"url" : "http://skeptics.stackexchange.com/questions/18539/has-stack-overflow-saved-billions-of-dollars-in-programmer-productivity",

"date" : ISODate("2013-11-30T13:23:00Z"),

"starred" : 55,

"ratings" : [

3,

5,

4,

1,

2,

3,

2,

4,

2,

5,

5,

1

],

```
"comments" : [
    {
        "user" : "hobbified",
        "upVotes" : 1100,
        "downVotes" : 90,
        "text" : "They've improved the world immensely by killing off expert sexchange, and that's all that really matters.",
        "replies" : [
            {
                "user" : "hex",
                "upVotes" : 400,
                "downVotes" : 100,
                "text" : "EDIT: w3fools.com provides a bit of context here. Generally the problem is that the information on w3schools is not kept up to date and can be wrong, or show an outdated way of doing things. A preferrable resource is the MDN"
            },
            {
                "user" : "Uberhipster",
                "upVotes" : 809,
                "downVotes" : 40,
                "text" : "The information on there is sometimes inaccurate but it is not cast in stone. You can email them and point out the omissions/discrepancies and they will amend them as they did in lieu of w3fools. It's kind of nitpicky to be that anal about several examples of things on w3schools that, although correct and functional, aren't considered 'best practice'."
            },
            {
                "user" : "Uberhipster",
                "upVotes" : 250,
```

```
                "downVotes" : 30,
                "text" : "stackoverflow is perfect if you know what questions to ask. But if you don't even know where to begin w3schools provides a reference of legible examples for beginners to approach a problem without having to overflow the stack (ithankyou) with questions like \"Hai how do I html?\""
            }
        ]
    },
    {
        "user" : "Jasper",
        "upVotes" : 200,
        "downVotes" : 85,
        "text" : "Also in w3schools you can always see links to the other topics in the top bar, and other elements(or whatever is being docced) to the left, whereas with MDN you dont, you have to go back some pages. If it showed up on a search result that decreases the chance the user will use it more extensively.",
        "replies" : [
            {
                "user" : "Uberhipster",
                "upVotes" : 90,
                "downVotes" : 20,
                "text" : "In addition, MDN reference only covers HTML5, CSS3, JS and SVG/WebGL. W3Schools reference covers (warts and all) those as well as HTML, CSS, PHP, ASP.NET, SQL, XSLT, RSS, JSON, JQuery and a number of other things for which there is a demand to have references for."
            }
        ]
    },
    {
        "user" : "morantz",
```

```
            "upVotes" : 25,
            "downVotes" : 0,
            "text" : "I find the MDN interface a bit terrible though.",
            "replies" : [ ]
        },
        {
            "user" : "bman4",
            "upVotes" : 4,
            "downVotes" : 0,
            "text" : "I think MDN is a better resource for developers that are already familiar with programming concepts. I've recently been teaching high schoolers how to do web development, coming from no programming background, and though my initial reaction was to point them to MDN to look things up after comparing the two on different topics I found justifying that quite difficult. Simply put, MDN has way too many details that honestly aren't vital to learn from, they will distract and confuse more then help.",
            "replies" : [
                {
                    "user" : "Modevs",
                    "upVotes" : 4,
                    "downVotes" : 1,
                    "text" : "Despite all the pedantic angst, for every coder whining about w3cs' few and generally minor inaccuracies there are thousands of people Googling \"html tutorial\", finding w3cs' articles and learning the fundamentals."
                }
            ]
        }
    ],
    "tags" : [
        "Programming",
```

```
            "StackOverflow"
        ],
        "draft" : false,
        "published" : true
}
{
        "_id" : ObjectId("52b3833bd3e98582d2bfb62b"),
        "author" : {
            "name" : "Peter Mortensen",
            "email" : "peter@gmail.com"
        },
        "title" : "How can I pass the string “Null” through WSDL (SOAP) from ActionScript 3 to a ColdFusion web service without receiving a “missing parameter error”?",
        "body" : "We have an employee whose last name is Null. He kills our employee lookup application when his last name is used as the search term (which happens to be quite often now). The error received (thanks Fiddler!) is\n\n  <soapenv:Fault>\n <faultcode>soapenv:Server.userException</faultcode>\n <faultstring>coldfusion.xml.rpc.CFCInvocationException: [coldfusion.runtime.MissingArgumentException : The SEARCHSTRING parameter to the getFacultyNames function is required but was not passed in.]</faultstring>\nCute, huh?\n\nThe parameter's type is string.\n\nI am using:\n\nWSDL (SOAP).\nFlex 3.5\nActionScript 3\nColdFusion 8\nNote that the error DOES NOT occur when calling the webservice as an object from a ColdFusion page.",
        "url" : "http://stackoverflow.com/questions/4456438/how-can-i-pass-the-string-null-through-wsdl-soap-from-actionscript-3-to-a-co",
        "date" : ISODate("2012-09-12T13:23:00Z"),
        "starred" : 19,
        "ratings" : [
            1,
            2,
            2,
```

```
        2
    ],
    "comments" : [
        {
            "user" : "pvc",
            "upVotes" : 550,
            "downVotes" : 111,
            "text" : "9 years ago I worked at Wells Fargo and we had the same thing. A loan had to be moved through manually because of a customer with the last name of Null. That was back when we coded our own Object/XML translators. I had an argument with a developer about proper handling of null values, and he ignored this possibility. When it finally happened he owed me a drink.",
            "replies" : [
                {
                    "user" : "user1231231323",
                    "upVotes" : 177,
                    "downVotes" : 12,
                    "text" : "From experience if you need to have a serious discussion about a scenario that someone claims won't happen for more than five minutes then you need prepare for that scenario cause its going to happen."
                },
                {
                    "user" : "MrVon",
                    "upVotes" : 12,
                    "downVotes" : 4,
                    "text" : "I do support for a living, so I'm not really a programmer, but I remember spending far too long trying to explain to a client that \"one in a million chance of data loss is not an acceptable risk in a system that does several million transactions a day.\""
                },
                {
```

```
                    "user" : "slqrm",
                    "upVotes" : 12,
                    "downVotes" : 4,
                    "text" : "I've had that exact conversation with a project manager who was complaining I was taking too long: \"Working 99.999% of the time isn't acceptable when you're building 100,000 a day!\""
                }
            ]
        },
        {
            "user" : "newstine",
            "upVotes" : 54,
            "downVotes" : 12,
            "text" : "What if it's the same one person that everyone keeps running into with the name Null, who is some sort of programming terrorist?",
            "replies" : [
                {
                    "user" : "elephantry",
                    "upVotes" : 12,
                    "downVotes" : 5,
                    "text" : "Doesn't appear to be all that uncommon http://www.ancestry.com/name-origin?surname=null\nShouldn't that just become \"The, The\"?"
                },
                {
                    "user" : "Muezza",
                    "upVotes" : 44,
                    "downVotes" : 3,
```

```
                    "text" : "So you're saying they're organized...\nI'm sure there are some out there by some overzealous developers who misunderstood what normalizing a mysql database meant."
                },
                {
                    "user" : "DoctorCube",
                    "upVotes" : 600,
                    "downVotes" : 0,
                    "text" : "And thus my new legal name is True NaN Null."
                },
                {
                    "user" : "P-01S",
                    "upVotes" : 110,
                    "downVotes" : 0,
                    "text" : "John O\"DROP TABLE LASTNAME\nIt's Irish."
                }
            ]
        }
    ],
    "tags" : [
        "Null",
        "Computer",
        "Bug",
        "ASP"
    ],
    "draft" : false,
    "published" : true
}
```