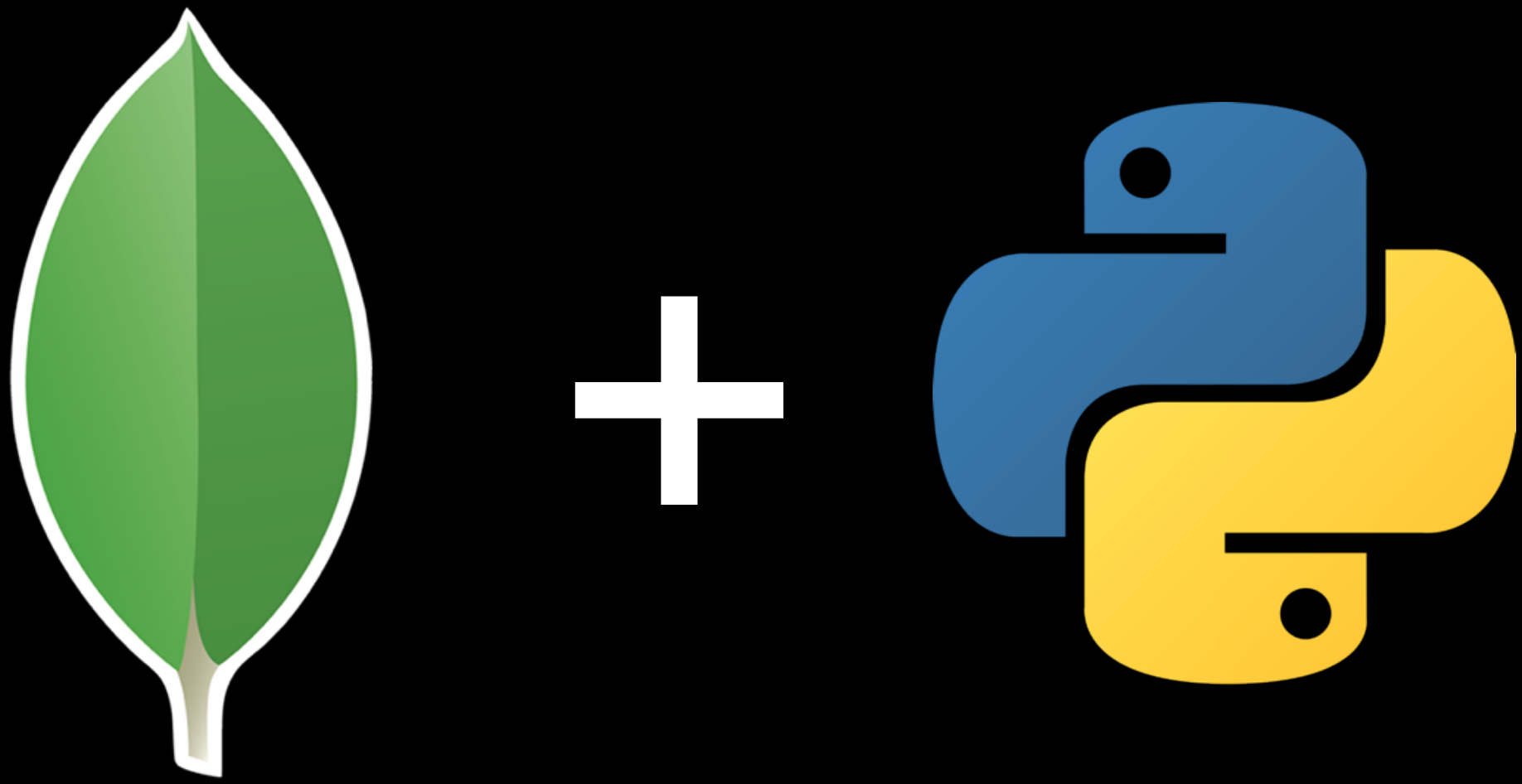
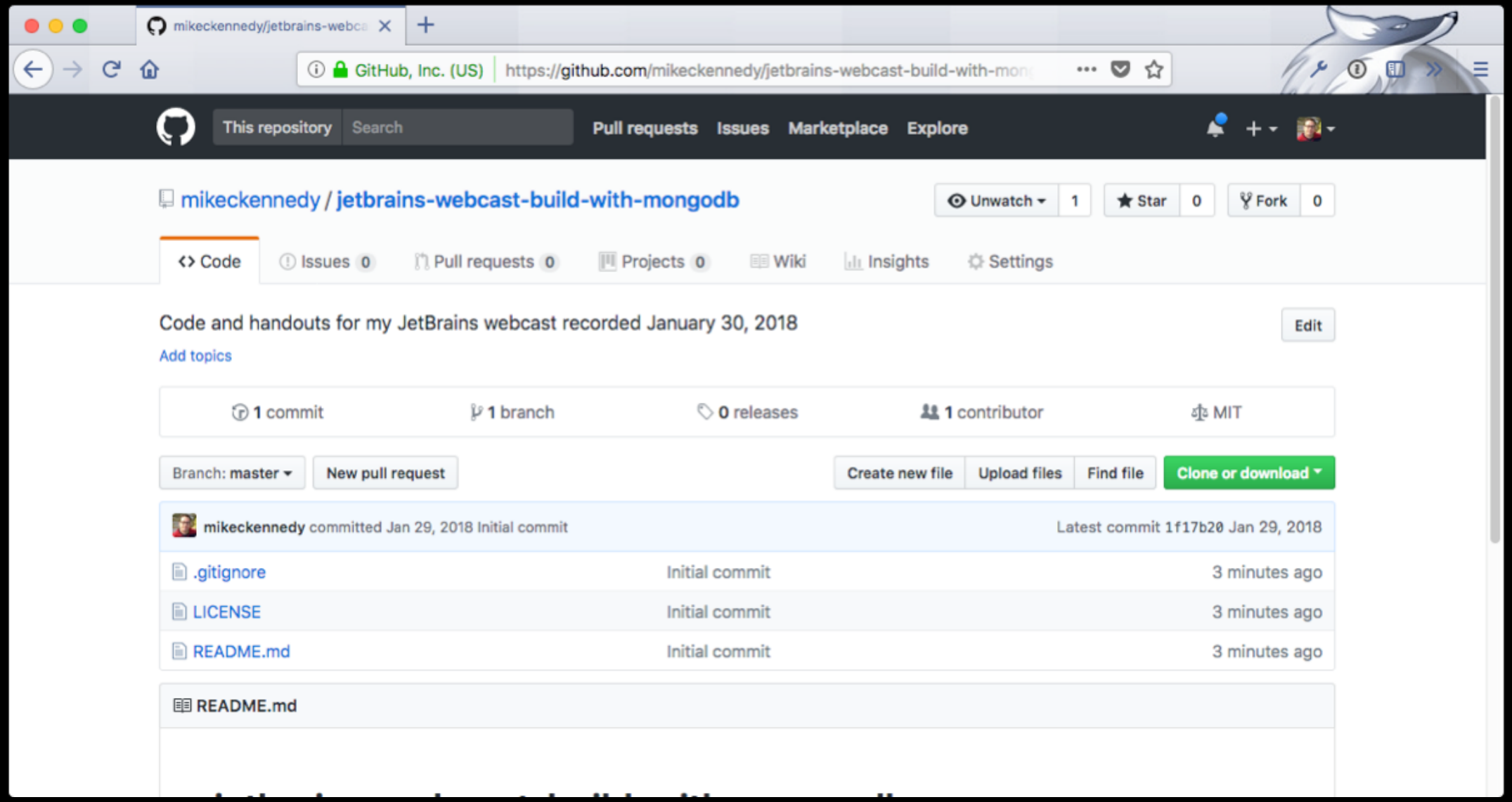
MongoDB + Python with PyCharm



Topics

- Tools
- Why MongoDB?
- Building something fun!
- Concepts
- Questions

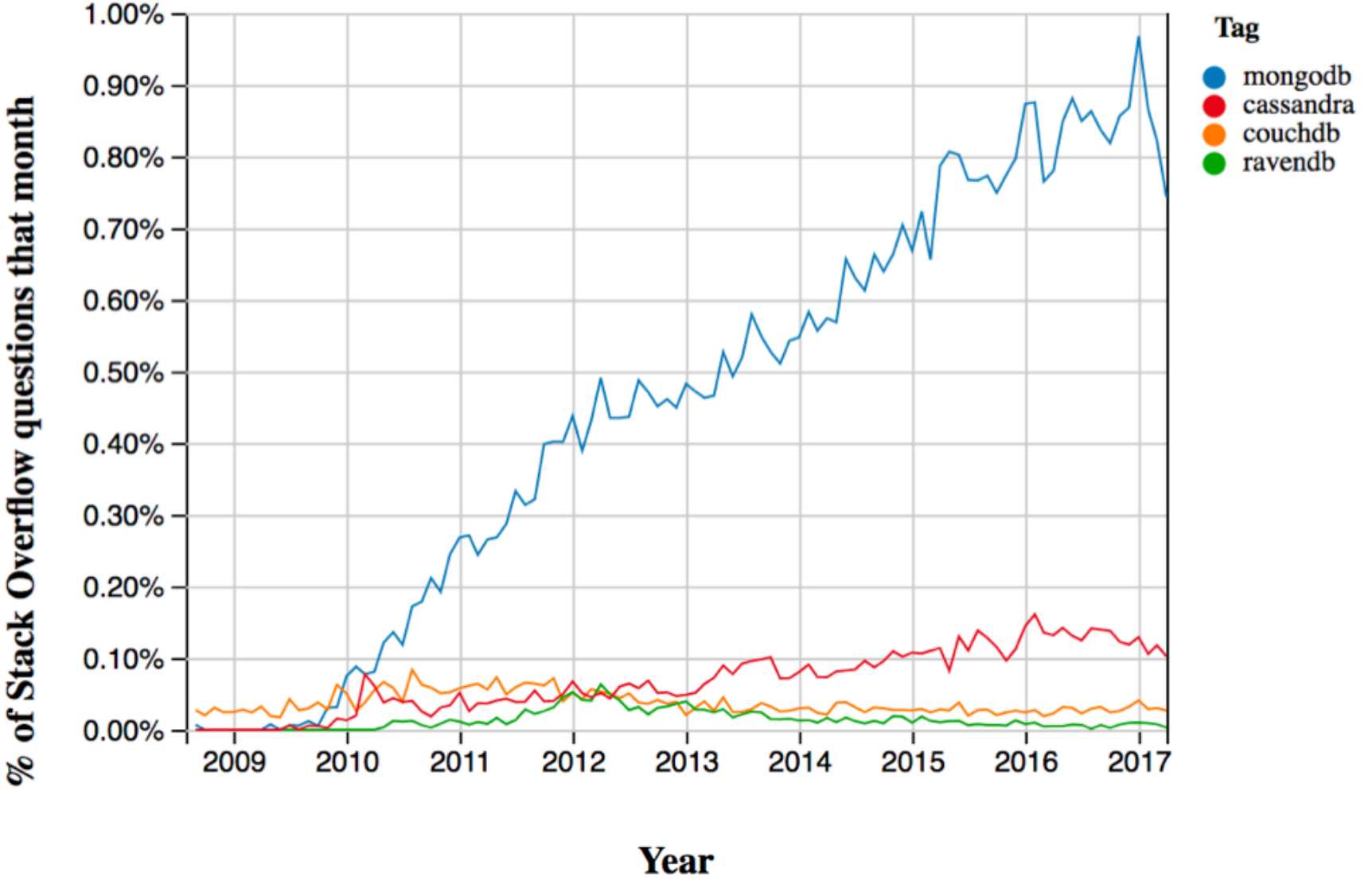
Take the code with you



github.com/mikeckennedy/jetbrains-webcast-build-with-mongodb

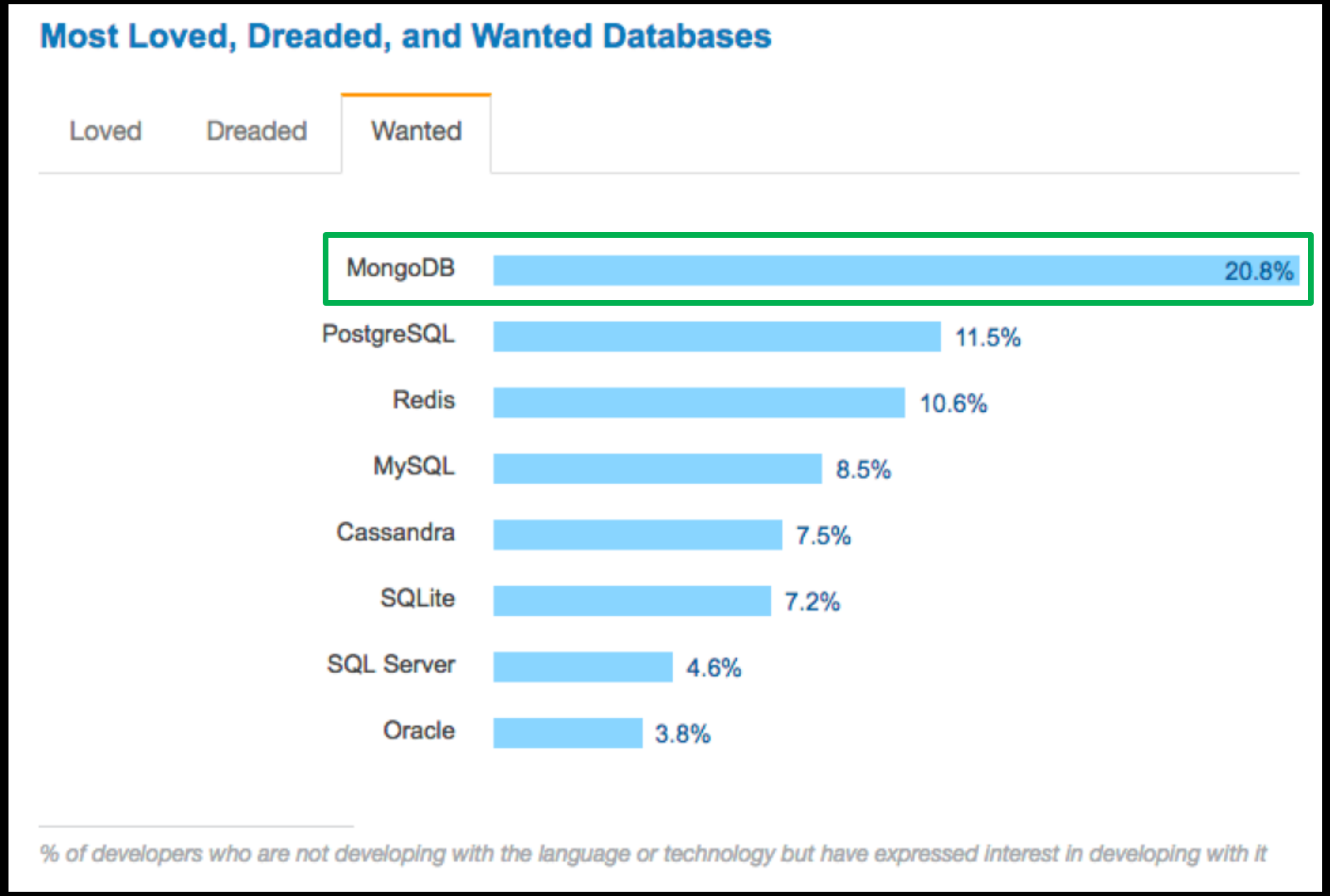
Tools

MongoDB is popular



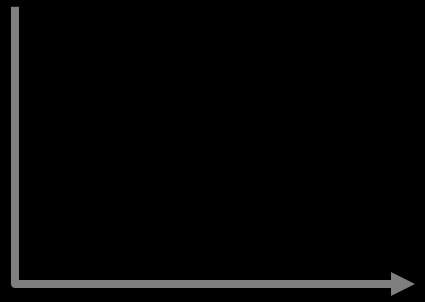
https://insights.stackoverflow.com/trends?tags=mongodb%2Ccouchdb%2Cravendb%2Ccassandra

MongoDB is wanted



How do document databases work?

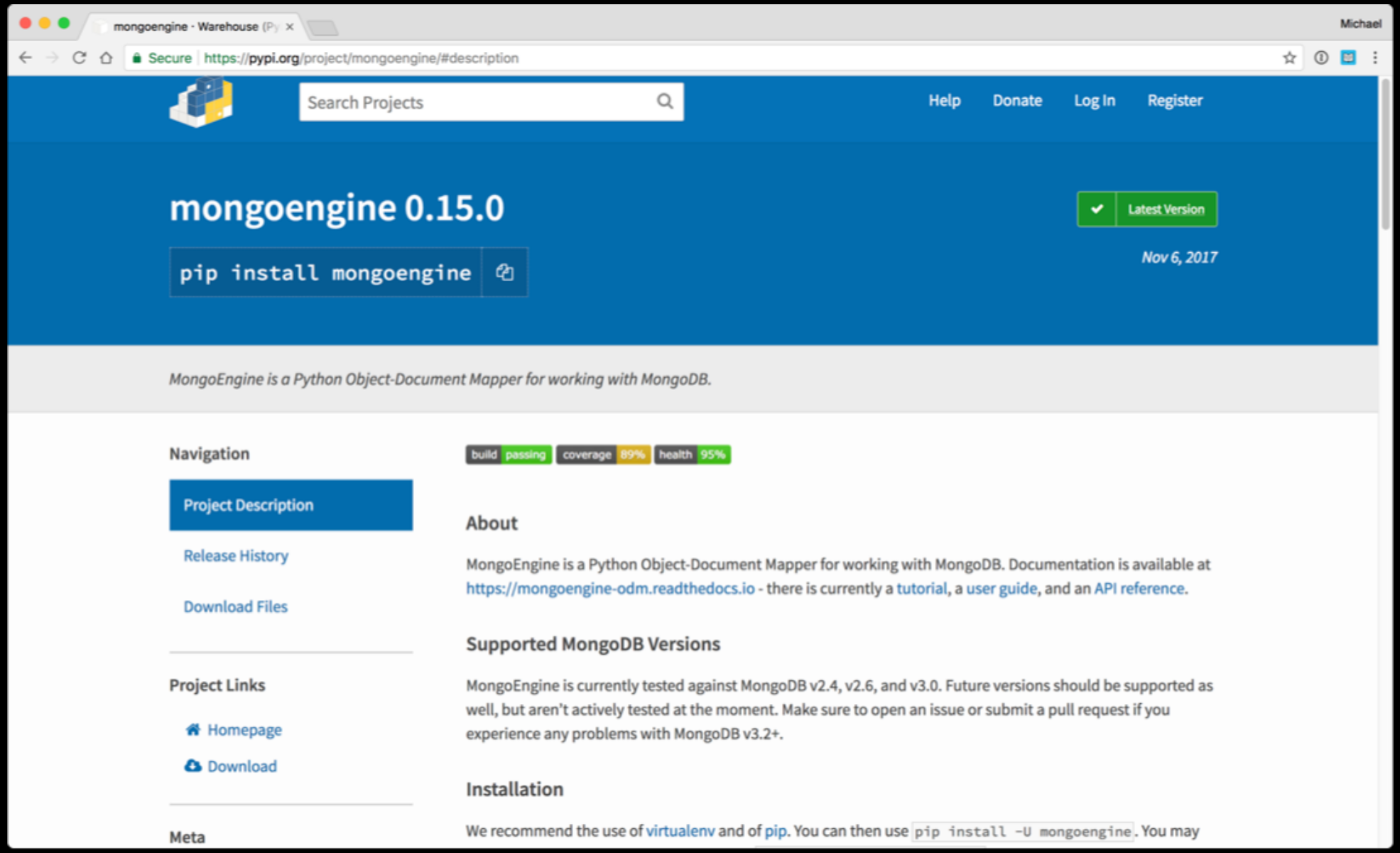
Consider lectures to be a pre-computed join.



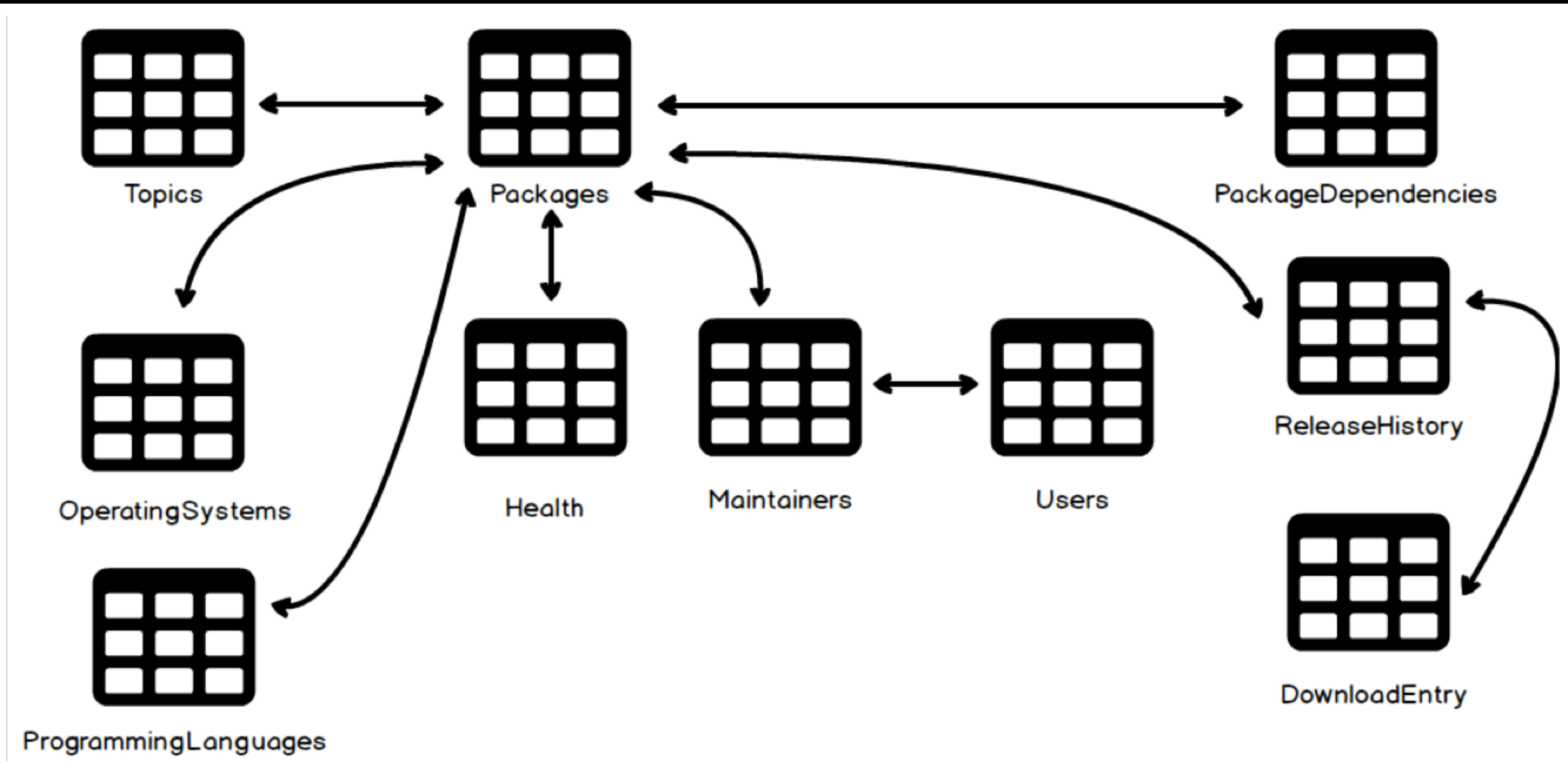
Question: Will I be able to find lecture 10106 directly?

```
// Chapter 1001 from Python Jumpstart by Building 10 Apps
{
    "_id" : 1001,
    "title" : "Welcome to the course",
    "course_id" : 1,
    "duration_in_sec" : 1707,
    "lectures" : [
        {
            "id" : 10101,
            "title" : "Welcome and thanks for coming",
            "video_url" : "https://s3.amazonaws.com/.../00-00-welcome.mp4",
            "duration_in_sec" : 380
        },
        // ...
        {
            "id" : 10106,
            "title" : "Linux: Installing Python and PyCharm",
            "video_url" : "https://s3.amazonaws.com/.../00-05-linux-setup.mp4,
            "duration_in_sec" : 307
        }
    ]
}
```

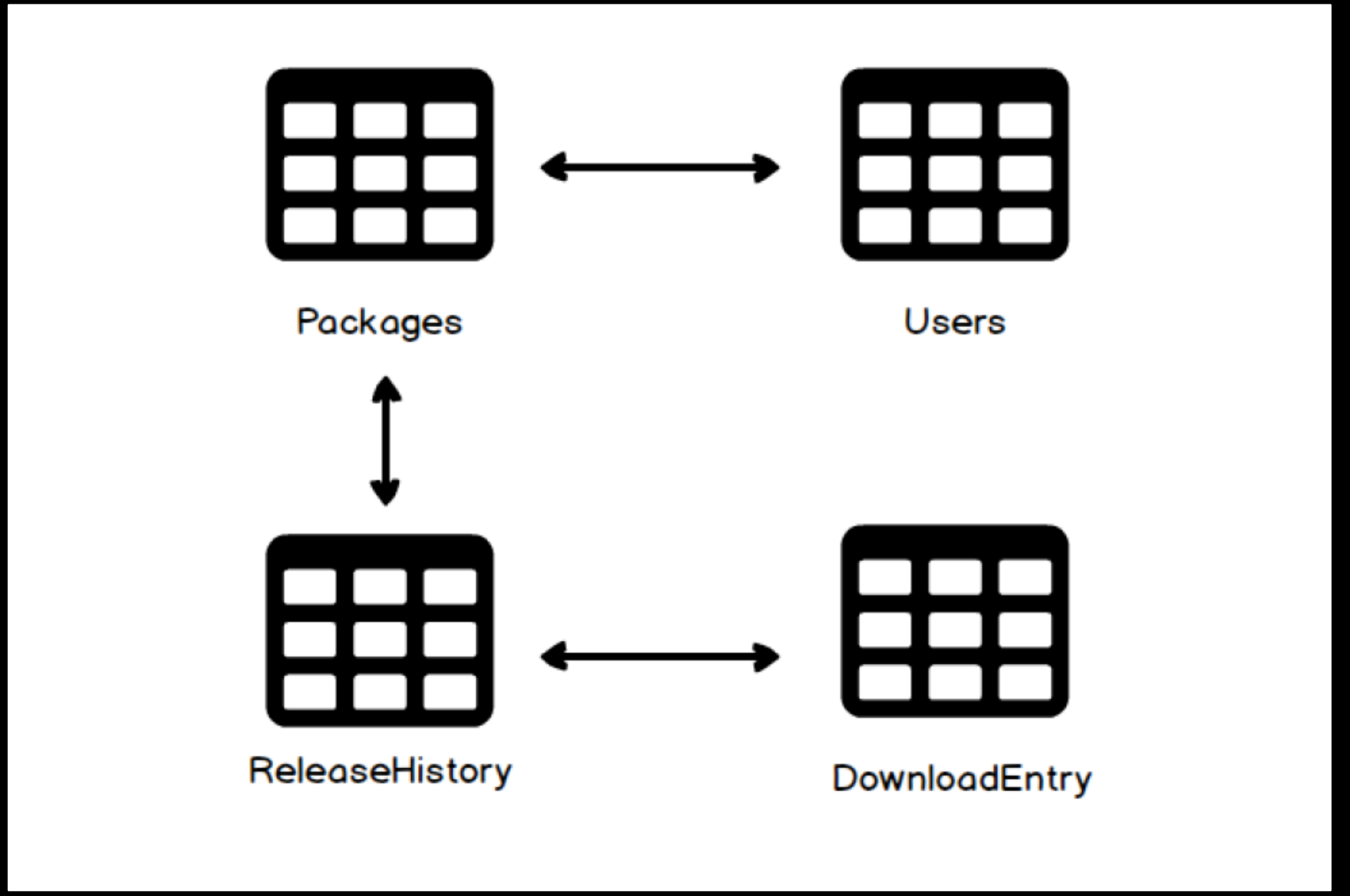
Our demo application



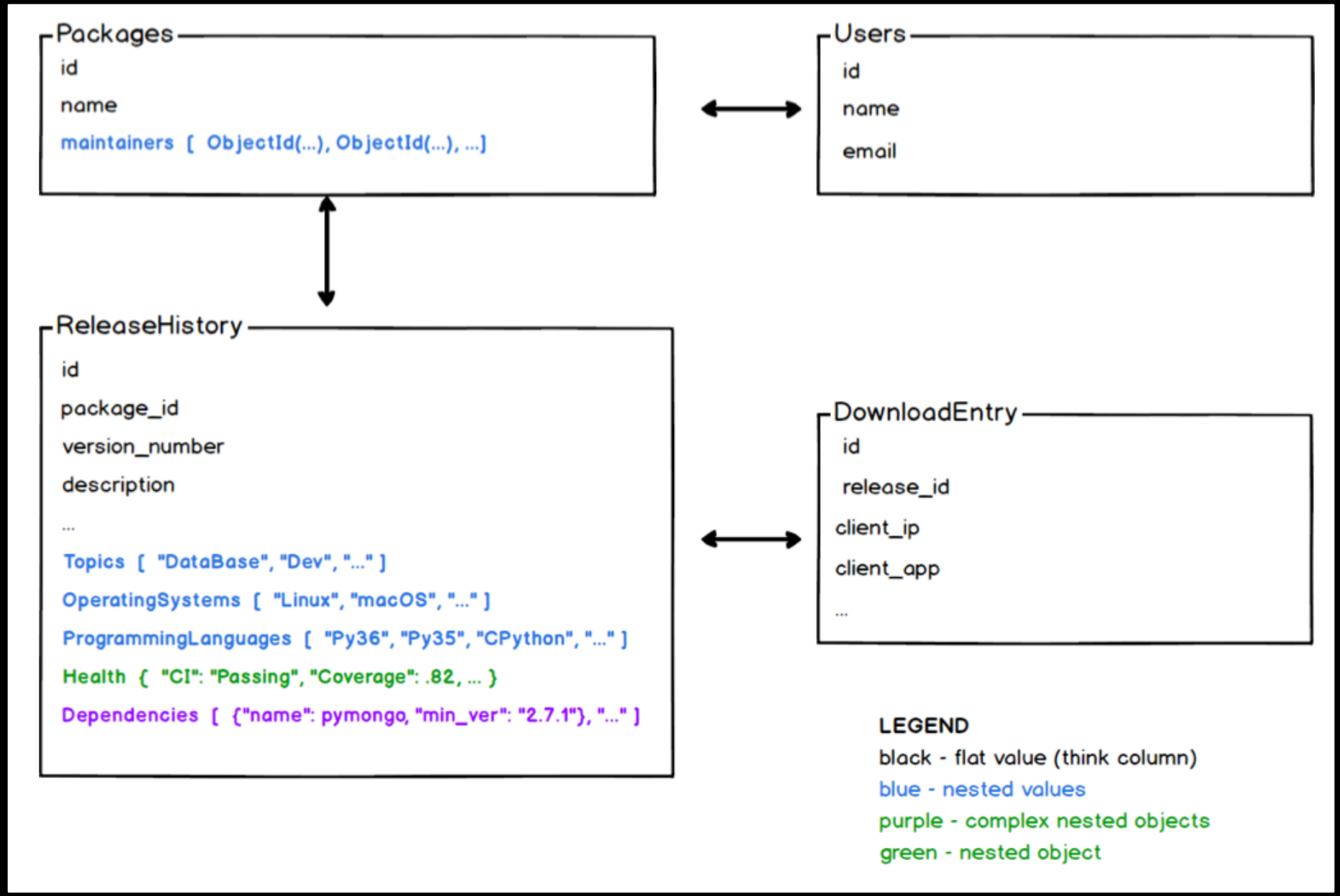
Relational model



Document DB model

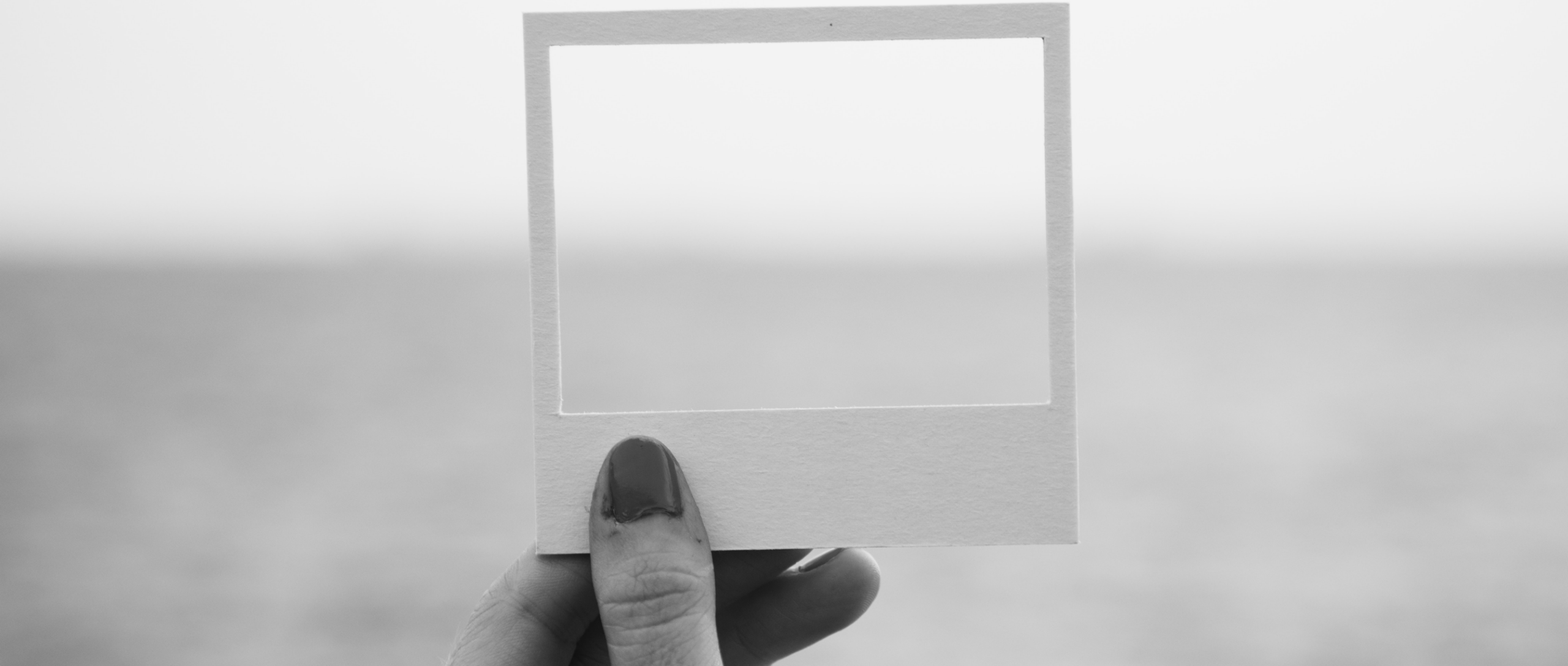


Document DB model (detailed)



Demo time

Concepts



The primary question becomes

To embed or not to embed?

1. Is the embedded data wanted 80% of the time?
2. How often do you want the embedded data without the containing document?
3. Is the embedded data a bounded set?
4. Is that bound small?
5. How varied are your queries?
6. Is this an integration DB or an application DB?

Registering connections

```
import mongoengine

alias_core = 'core'
db = 'snakebnb'

mongoengine.register_connection(alias=alias_core, name=db)
```

Creating basic classes

```
import mongoengine

class Snake(mongoengine.Document):
    registered_date = mongoengine.DateTimeField()
    length = mongoengine.FloatField()
    name = mongoengine.StringField()
    species = mongoengine.StringField()
    is_venomous = mongoengine.BooleanField()
```

Nested data (fields and lists)

```
import mongoengine

class Cage(mongoengine.Document):
    name = mongoengine.StringField(requi
    price = mongoengine.FloatField(requi
    square_meters = mongoengine.FloatFie

    bookings = mongoengine.EmbeddedDocum
```

```
{
    "_id" : ObjectId("59d3951c3ae740cffe16427d"),
    "registered_date" : ISODate("2017-10-03"),
    "name" : "Slither's first cage",
    "price" : 29.95,
    "square_meters" : 1.5,
    "is_carpeted" : true,
    "has_toys" : false,
    "allow_dangerous_snakes" : false,
    "bookings" : [
        {
            "check_in_date" : ISODate("2018-02-19"),
            "check_out_date" : ISODate("2018-02-28"),
            "rating" : 0
        },
        {
            "check_in_date" : ISODate("2018-03-01"),
            "check_out_date" : ISODate("2018-03-30"),
            "rating" : 0
        }
    ]
}
```

Querying (direct match)

```
def find_owner_by_email(email: str) -> Owner:

    owner = Owner.objects().filter(email=email).first()
    return owner
```

Filter on 1 or more fields. email must match passed email value.

Call first to execute the query and return 1 owner or None

Querying (subdocuments)

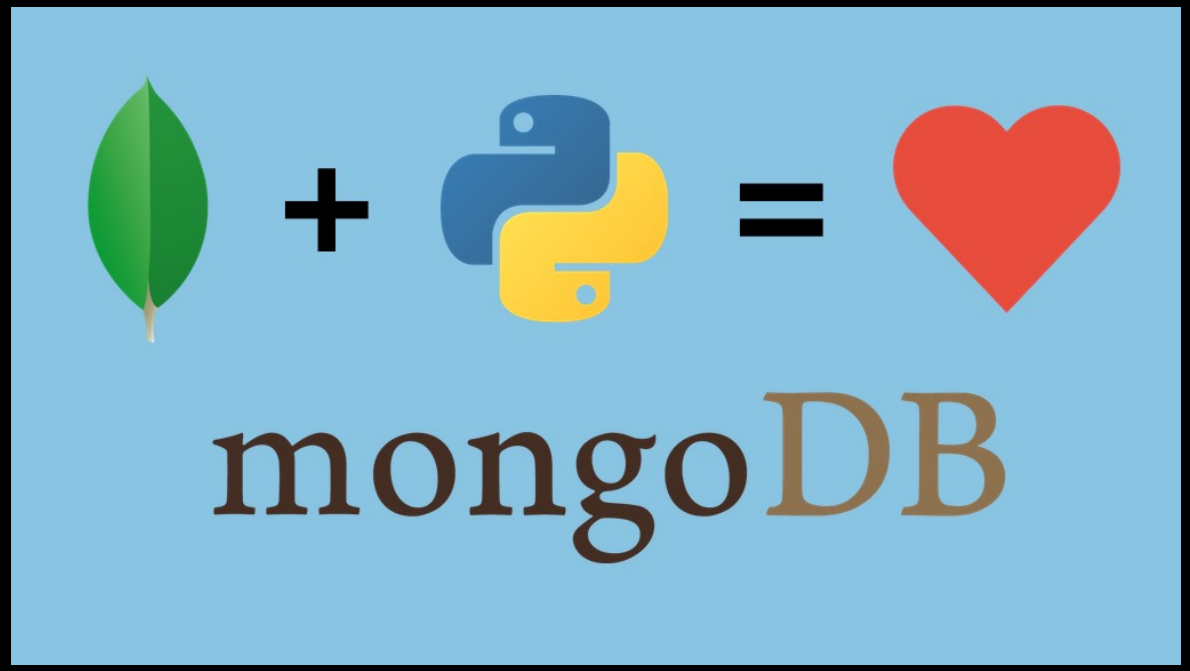
```
def get_bookings_for_user(user_id: ObjectId) -> List[Booking]:
    owner = Owner.objects(id=user_id).first()
    booked_cages = Cage \
        .objects(bookings__guest_snake_id__in=owner.snake_ids) \
        .all()

    return list(booked_cages)
```

Use __ to separate / navigate levels in subdocuments

Best practice: Execute the query before returning results.

Want more?



MongoDB for Developers with Python, 7 hrs

MongoDB Quickstart with Python, 2.5 hrs

training.talkpython.fm