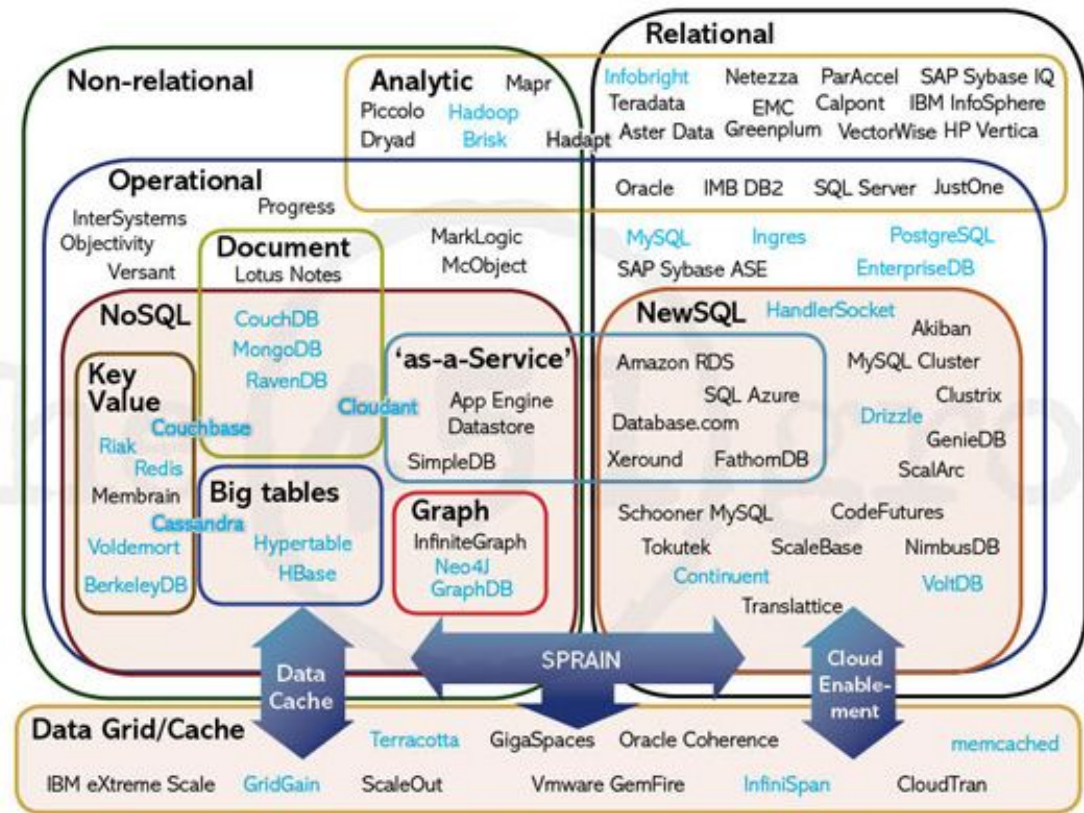


# Lesson web 03

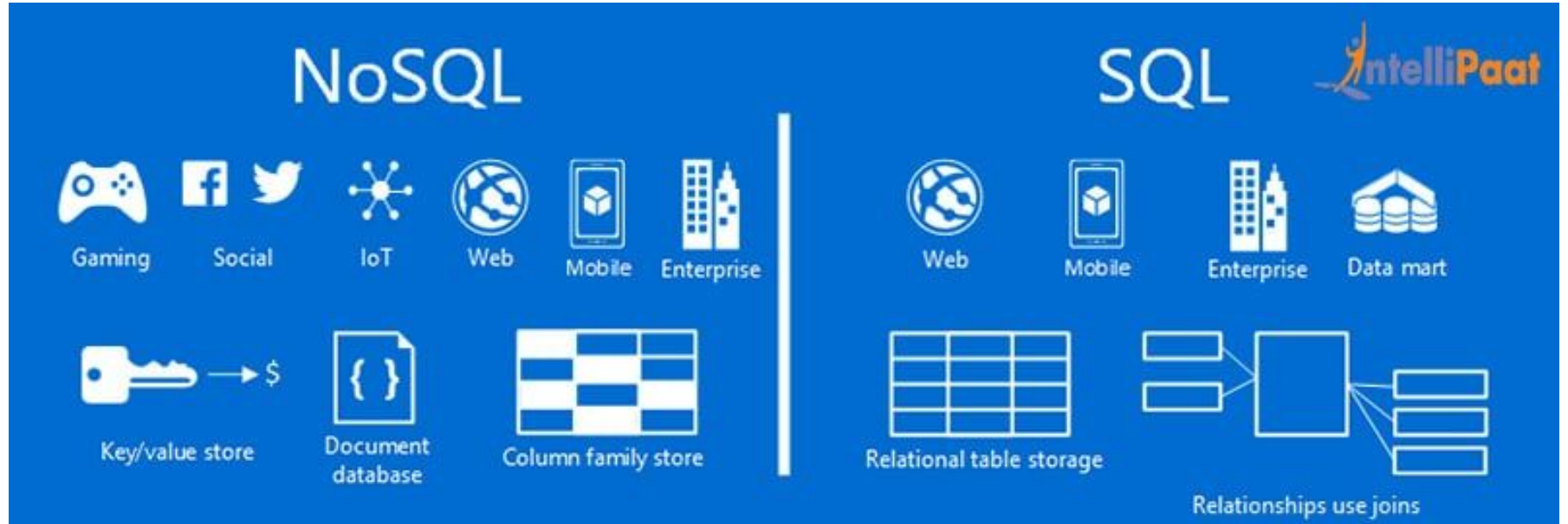
# Database

- Dữ liệu là quan trọng trong tất cả các ứng dụng vì vậy cần được bảo vệ an toàn
- Dữ liệu không chỉ có text, ảnh, file mà còn ở dạng có cấu trúc, liên kết với nhau
- Cần truy vấn theo nhiều điều kiện, xử lý theo nhiều cách

# Database



# Database



# Database

Document 1

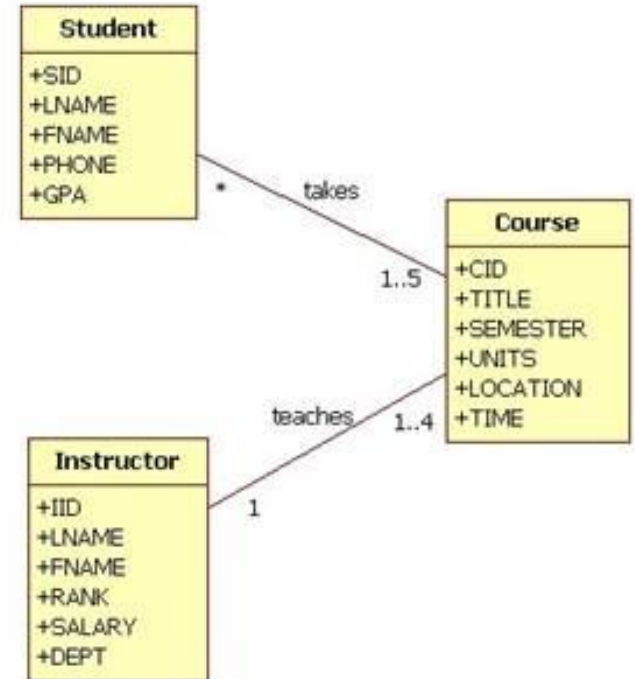
```
{
  "id": "1",
  "name": "John Smith",
  "isActive": true,
  "dob": "1964-30-08"
}
```

Document 2

```
{
  "id": "2",
  "fullName": "Sarah Jones",
  "isActive": false,
  "dob": "2002-02-18"
}
```

Document 3

```
{
  "id": "3",
  "fullName": {
    "first": "Adam",
    "last": "Stark"
  },
  "isActive": true,
  "dob": "2015-04-19"
}
```



# Database

Ưu nhược điểm của NoSql và Sql





# Document - Json

```
{  
  name: "sue",  
  age: 26,  
  status: "A",  
  groups: [ "news", "sports" ]  
}
```



← field: value

← field: value

← field: value

← field: value



# Document - Hosting

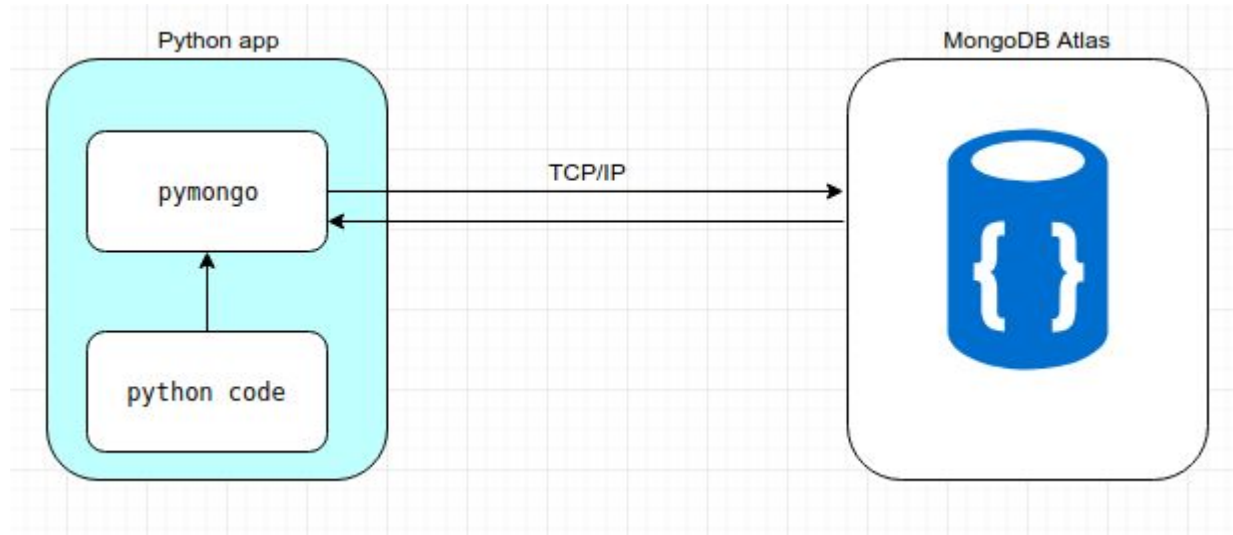
## MongoDB Atlas

Move faster with an automated cloud MongoDB service built for agile teams who'd rather spend their time building apps than managing databases. Available on AWS, Azure, and GCP.

[Start free](#)

Already have an account? [Log in here](#) →

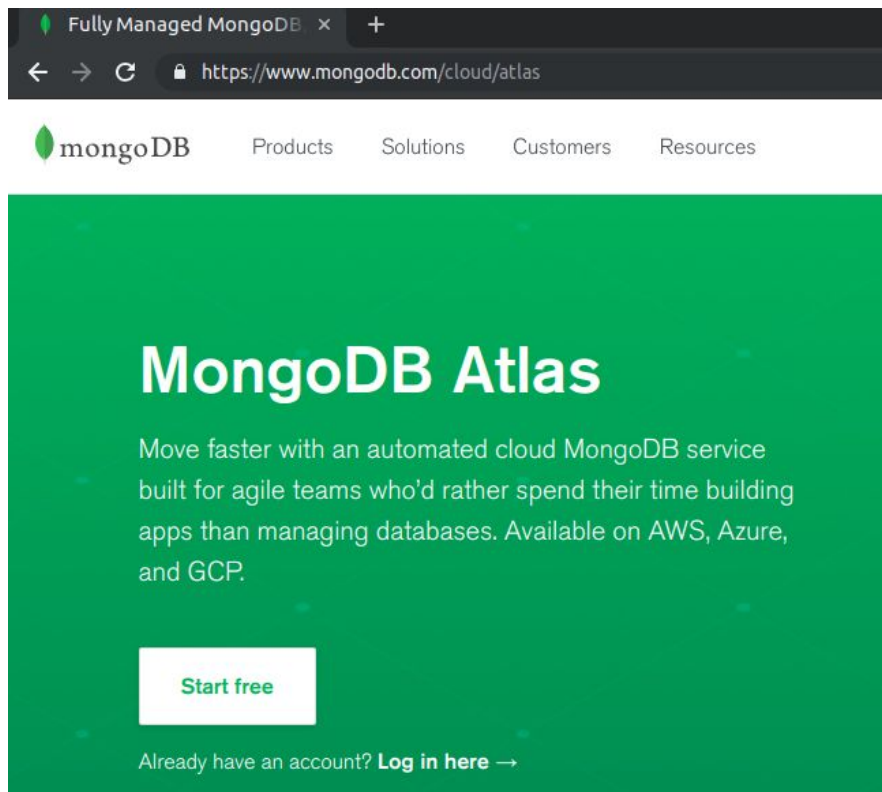
# Document - Hosting



# Mongodb atlas

- Đăng ký tài khoản tại

<https://www.mongodb.com/cloud/atlas>

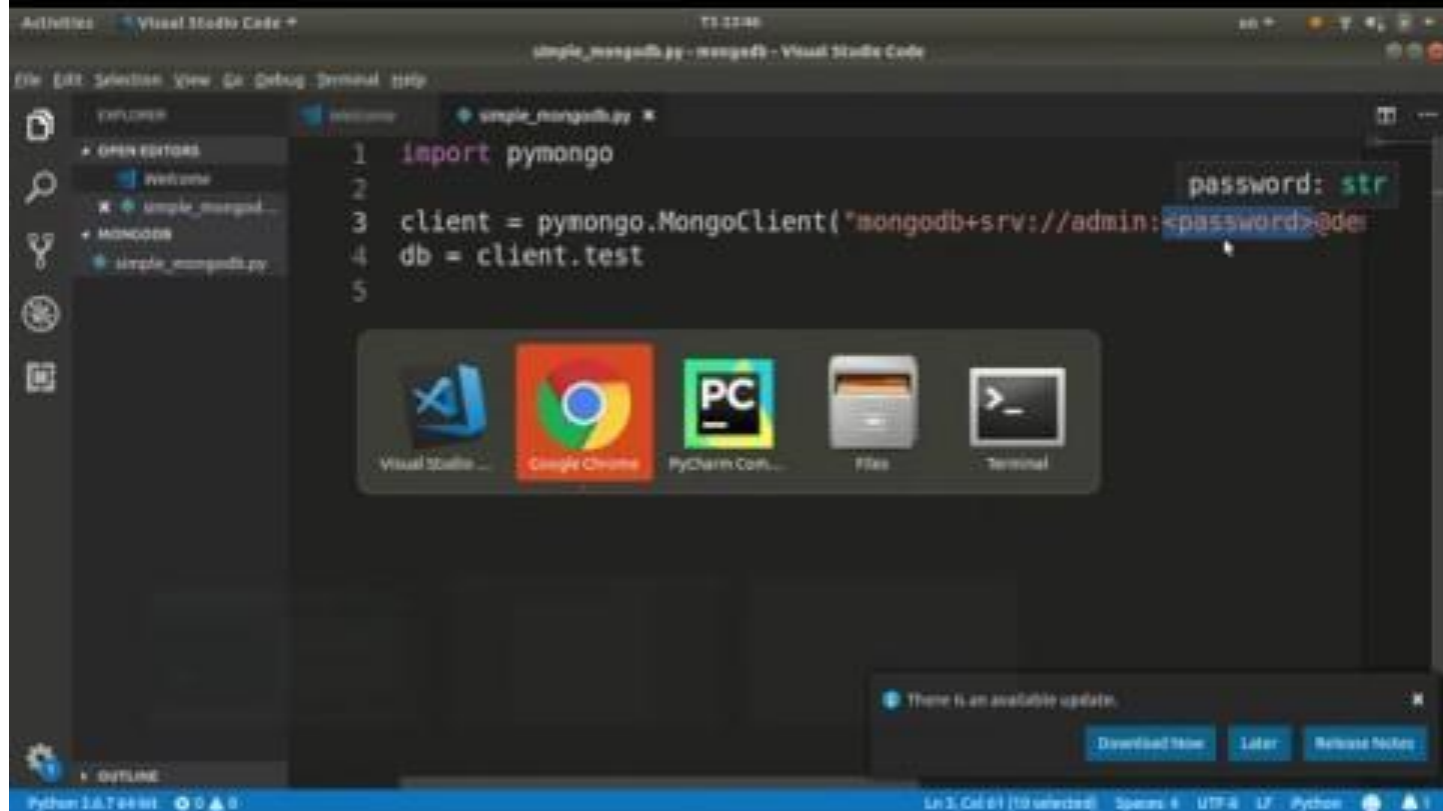


# Mongodb atlas

- Cài đặt các gói:

Python -m pip install pymongo dnspython

- Đăng nhập <https://cloud.mongodb.com> và làm theo video



# Mongodb CRUD

# Init

```
import pymongo

# Kết nối đến database mongodb
client = pymongo.MongoClient(

    "mongodb+srv://admin:bjm6zmHvxGhmECAY@demo-cluster-mvmw2.mongodb.net/test
    ?retryWrites=true")

# tạo db test 1
db = client.c4e
```

# Create - insert\_one

```
# Thêm 1 user vào collection users (collection sẽ tự tạo nếu chưa có)
db.users.insert_one({'name': 'Hoa', 'age': 30})
db.users.insert_one({'name': 'Lan', 'age': 25})

# ngoài ra còn có các lệnh: insert_many, insert
```



# Read - find\_one and find

```
# Liệt kê tất cả các user đang có
print(list(db.users.find({})))

# Tìm kiếm các user có tên là "Hoa"
print(list(db.users.find({"name": "Hoa"})))

# Tìm kiếm 1 user có tên là "Hoa"
print(db.users.find_one({"name": "Hoa"}))
```

# Update - update\_one

```
# Cập nhật tuổi các users có tên là "Lan" thành 40
db.users.update_one({"name":"Lan"}, {"$set":{"age":40}})

# ngoài ra còn có các lệnh: update, update_many, replace_one
```

# Delete - delete\_one and delete\_many

```
# Xóa người có tên Lan
db.users.delete_one({"name": "Lan"})
# ngoài ra còn có các lệnh: delete_many
```

# Restructure

Khi chương trình bắt đầu phức tạp hơn, khi đó cần nghĩ đến việc chia ra các hàm.

user\_db.py

```
import pymongo

client = pymongo.MongoClient(
    "mongodb+srv://admin:bjm6zmHvxGhmECAY@demo-cluster-mvmw2.mongodb.net/test?retryWrites=true")

db = client.c4e


def add_user(name, age):
    """
    Thêm một user mới
    :param name: Tên
    :param age: Tuổi
    """
    db.users.insert_one({'name': name, 'age': age})
```

sample.py

```
from users_db import get_all
```

```
add_user('hoa', 30)
```

Sử dụng mongodb để lưu trữ dữ liệu cho web food

# Thêm chức cập nhật food

- Tạo form mới
- Hiển thị thông tin gốc trước khi sửa
- Cập nhật database



# Thêm chức năng xóa food

Xóa theo ObjectId với route mới