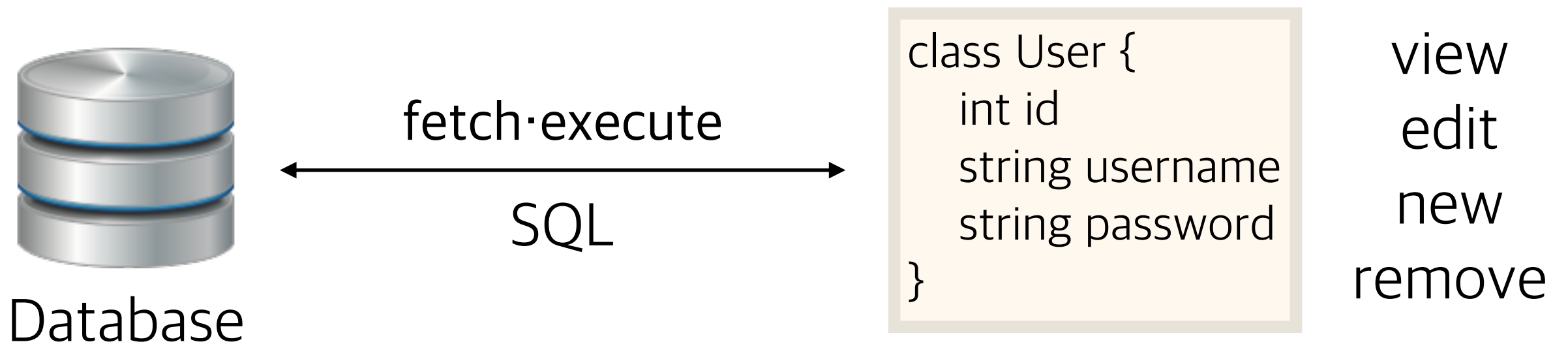


Flask 3일

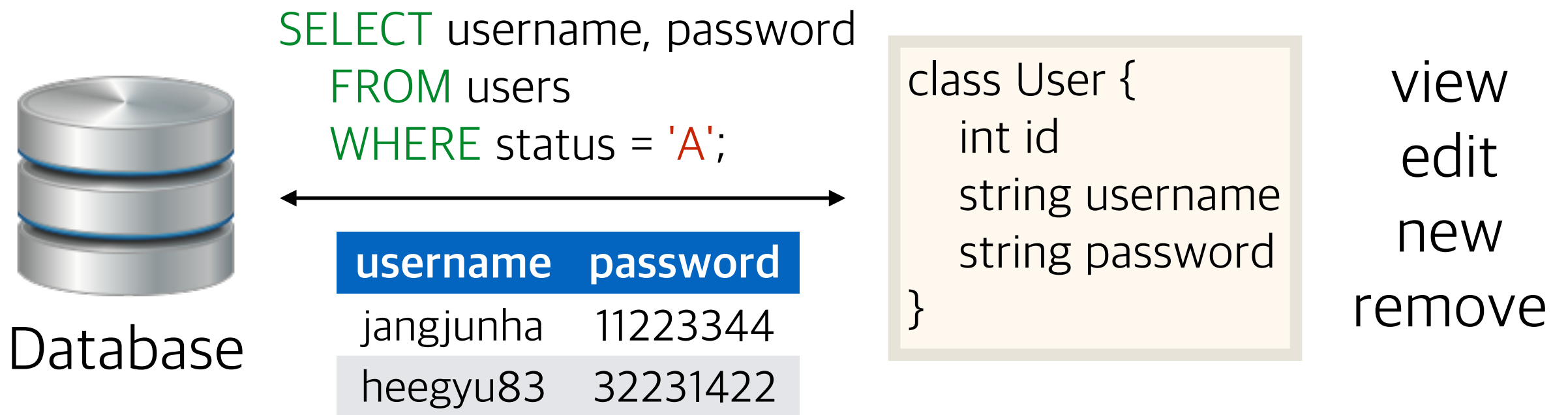
ORM (Feat. SQLAlchemy)

jangjunha<jangjunha113@gmail.com>

(거의) 항상 하는 일



(거의) 항상 하는 일



ORM

- Object-relational mapper
 - JAVA Hibernate
 - Python SQLAlchemy, Django ORM



“SQLAlchemy is the **Python SQL toolkit** and **Object Relational Mapper** that gives application developers the full power and flexibility of SQL.”

<http://www.sqlalchemy.org/>

Quickstart



<http://flask-sqlalchemy.pocoo.org/2.1/quickstart/>

**덧글 기능을
ORM을 사용해서 다시 만들어보자**

Flask-SQLAlchemy 설치

```
> pip install Flask-SQLAlchemy
```


Setup

```
from flask_sqlalchemy import SQLAlchemy

app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///yourdb.db'
db = SQLAlchemy(app)
```

Model 만들기

```
class Message(db.Model):  
    id = db.Column(db.Integer, primary_key=True)  
    title = db.Column(db.Text, nullable=False)  
    content = db.Column(db.Text, nullable=False)  
  
    def __init__(self, title, content):  
        self.title = title  
        self.content = content
```

DB 생성

```
> python
```

```
>>> from application import db
```

```
>>> db.create_all()
```

덧글 생성

```
message = Message(writer=request.form['writer'],  
                   content=request.form['content'])  
db.session.add(message)  
db.session.commit()
```

댓글 조회

```
comments = Message.query.order_by('-id').all()
```

실습

- 회원가입 기능을 추가해보자
- 필수:
 - username (아이디)
 - password (비밀번호)

<개인정보 침해 사고 관련 공지 드립니다>

...

비밀번호는 암호화되어있어 안전합니다.

...

비밀번호 암호화

- ▶ werkzeug.security.generate_password_hash
- ▶ werkzeug.security.check_password_hash

```
from werkzeug.security import generate_password_hash, \
    check_password_hash
```

```
hashed_password = generate_password_hash(password)
```

```
check_password_hash(hashed_password, password)
```


비밀번호 암호화

- 단방향 암호화

- Sunrin

—> 4204dc8d9919cf09c150f12f1446f286c4171db03a6ea841165d2918e254994a

비밀번호 암호화

김종수, D2 블로그 <안전한 비밀번호 저장>

<http://d2.naver.com/helloworld/318732>

SQL Injection

```
username = request.form['username']
```

```
sql = "SELECT * FROM user WHERE username='%s'" % username  
db.execute(sql)
```

```
username = request.form['username']
```

```
sql = "SELECT * FROM user WHERE username='?'"  
db.execute(sql, (username))
```

Transaction



thinkpanama, <ATM Machines in Panama>. <https://www.flickr.com/photos/23065375@N05/2235529194>.

다음 이 시간에

- Cookie / Session
- 로그인
- 내가 쓴 글 보기 / 애가 쓴 글 보기