

Mysql&파이썬 연동

18.07.15 임재근

세부 내용

- ▶ Pymysql을 사용한 마이스쿼-파이썬 연동
- ▶ 하나의 데이터베이스의 모든 테이블을 가져오는 함수(project-pre/func/load_all_tables.py) 사용법 및 주의사항

Pymysql

- ▶ Pymysql이란? Python에서 mysql을 사용하기 위한 패키지!(2.x,3.x 모두 사용가능)
- ▶ 크게 세가지 명령어로 나눌 수 있음:
 - ▶ 연결 및 설정 관련 명령어(connect, cursor 등)
 - ▶ 실행 관련 명령어(execute등)
 - ▶ 결과 적용 관련 명령어(fetch, commit등)

연결 및 설정 관련 명령어

- ▶ Connect : 연결할 데이터베이스의 주소, 포트 및 연결 권한등을 통해 데이터베이스와 연결을 도와주는 함수;
- ▶ Cursor : 커서 생성 함수(유저가 데이터베이스를 조작할수 있게 도와주는 객체라 생각하면 편함), 한 순간에 하나만 존재 가능;
- ▶ Close : 연결 혹은 커서를 닫는 함수, 클로즈 후엔 사용 x.
 - ▶ Ex) cur.close() , conn.close()

```
conn = pymysql.connect(  
    user=id,  
    passwd=pw,  
    host=ip,  
    charset='utf8')
```

```
cur = conn.cursor()
```

집행 관련 명령어

- ▶ Execute(): 중요한 함수, 커서로 지정된 변량의 자식 함수로, 커서 변량 뒤에 . 해서 사용. 괄호 안에는 집행할 sql언어를 string형식으로 넣어주면 됨;

```
cur.execute("use " + dblist[int(num)])  
cur.execute("show tables;")
```

- ▶ Executemany(): 이름 그대로, 여러개의 sql 문장을 집행하는 함수. 같은 문장을 안에 변량만 다르게 여러번 집행하고 싶을때 사용.

```
effect_row = cursor.executemany("insert into tb7(user,pass,licnese)values(%s,%s,%s)", [("u3","u3pass","11113"),("u4","u4pass","22224")])
```

결과 적용 관련 명령어

- ▶ 결과 받아오는 함수

- ▶ fetchall(): 모든 결과 반환, for 문을 통해 한 행씩 받아보기 가능;
- ▶ fetchone()(혹은 fetchmany(받아올 행 수)): 하나의 행(혹은 괄호안에 적은 행수 만큼의) 결과 반환;

```
for i in cur.fetchall():  
    tblist.append(' '.join(i))
```

```
row_1 = cursor.fetchone()
```


결과 적용 관련 명령어

- ▶ 결과 적용 함수

- ▶ Execute함수에서 결과를 받아보는 sql구문(예)select* from tbls)이 아닌, 테이블을 변화시키는 구문(예) insert into ~~)시, 이 함수를 사용해야 한다!
- ▶ commit(): 이거는 connect의 자식 함수로, conn.commit()으로 사용!

```
conn.commit()
```

함수 소개

- ▶ 함수 소개 : load_all_tables.py
 - ▶ 사용법 : `result = call_all('root', 'pyku2018', '154.8.160.186');`
 - ▶ 기능1 : `result` 에 딕셔너리 형식으로 모든 테이블 내용 반환(key는 테이블 이름, 내용은 리스트 형식의 테이블의 각 열);
 - ▶ 기능2: 함수 시작부분에서 `save`를 1로 지정시(함수 돌리면 나옴), 각각의 테이블을 각각 다른 이름의 txt형식으로 저장.

주의사항!

- ▶ Pymysql로 데이터를 불러오면, tuple 형식(리스트와 비슷하지만 고칠 수 없음)으로 반환됨, 이를 for 문으로 각 원소를 다시 받아서 저장하거나, ''.join() 함수로 str형식으로 바꿔줘야 우리가 평소 사용하던대로 사용 가능.
- ▶ 날짜는 datetime.date 형식으로 반환됨, 이걸 j=j.strftime("%Y-%m-%d-%H")을 통해 str로 변환 가능
- ▶ Null값은 None으로 반환되서, Missing value는 바로 None을 처리하는 식으로 처리 가능

현 상황 및 계획

- ▶ 데이터 베이스 구축:
 - ▶ 기본 틀 구축 완료;
 - ▶ 클라우드 서버 인터넷 속도 업그레이드? 필요할까요?
- ▶ 돌고래반 계획:
 - ▶ 현 데이터 베이스에 대한 유보수 작업;
 - ▶ 그 외, 각자 원하는 방향으로 나눠서 스터디 할 계획.