

Oracle 연동

Python DB API

- ❖ Python에서 데이터베이스를 액세스하기 위해서 Python DB API를 사용
- ❖ Python DB API는 데이터베이스를 액세스하는 표준 API로서 여러 DB 액세스 모듈은 최소한의 API 인터페이스 표준을 따름
- ❖ 표준 API는 크게 데이터베이스를 연결하고 SQL 문을 실행하고 연결을 닫는 등의 기본적인 DB 작업과 관련된 기능들을 정의
- ❖ 데이터베이스를 액세스하는 다른 방식으로 ORM (Object Relational Mapping)이 있는데 Django ORM, PonyORM, peewee 라이브러리 등이 ORM 방식을 사용한 데이터 액세스를 제공
- ❖ Python에서 지원하는 데이터베이스는 매우 다양하기 때문에 Python에서 각 데이터베이스를 사용하기 위해서는 각각의 DB에 상응하는 별도의 DB 모듈을 설치해서 사용(sqlite3는 Python 2.5 이상에서 기본 내장)
- ❖ 수많은 DB 모듈들이 있지만 이들이 거의 모두 Python DB API 표준을 따르고 있으므로 동일한 API를 사용해 데이터베이스를 사용
- ❖ python에서는 거의 모든 데이터베이스를 사용할 수 있는데 MySQL, PostgreSQL, MSSQL, Sqlite, Oracle, Sybase, Informix 등과 같은 DB 들을 모두 지원

Oracle 연동

❖ 오라클은 cx_Oracle 라이브러리를 이용해서 사용

❖ 모듈 설치: `pip install cx_Oracle`

❖ 연결 만들기

모듈을 불러옵니다.

```
import cx_Oracle
```

데이터베이스에 연결합니다.

```
변수1 = cx_Oracle.makedsn("IP주소 ", " 포트번호 ", "sid ")
```

```
변수2 = cx_Oracle.connect(user="계정", password= " 비밀번호", dsn=변수1)
```

Oracle 연동

모듈을 불러옵니다.

```
import cx_Oracle
```

```
import sys
```

```
try:
```

```
    # 데이터베이스에 연결합니다.
```

```
    dsnStr = cx_Oracle.makedsn("211.183.7.61", "1521", "xe")
```

```
    con = cx_Oracle.connect(user="scott", password="tiger", dsn=dsnStr)
```

```
    print(con)
```

```
except:
```

```
    print('exception:', sys.exc_info())
```

```
finally:
```

```
    con.close()
```

Oracle 연동

❖데이터베이스 접속이 되지 않는 경우 아래와 같은 예외 발생

Traceback (most recent call last):

```
File "C:\Users\Administrator\python\test\__main__.py", line 12, in <module>  
exception: (<class 'pymysql.err.OperationalError'>, OperationalError(2003, "Can't connect  
to MySQL server on '211.183.2.253' ([WinError 10060] 연결된 구성원으로부터 응답이 없  
어 연결하지 못했거나, 호스트로부터 응답이 없어 연결이 끊어졌습니다)", <traceback  
object at 0x00C839E0>)
```

```
if con != None:
```

```
NameError: name 'con' is not defined
```

Oracle 연동

❖ 파이썬에서 DML 수행

1. 연결 객체의 `cursor()` 메소드를 호출해서 sql 실행 객체를 가져옴
 2. `execute()`(실행 할 sql문장, 파라미터 튜플)
 3. 연결 객체의 `commit()` 을 호출하면 작업 내용이 반영되고 `rollback()`을 호출하면 작업 취소
- ❖ 완성된 sql 인 경우 파라미터 생략 가능
 - ❖ Sql 문장에 :번호 형태로 파라미터를 설정한 후 파라미터들을 tuple로 만들어서 대입 가능

Oracle 연동

❖ 데이터 삽입

```
import cx_Oracle
try:
    # 데이터베이스에 연결합니다.
    dsnStr = cx_Oracle.makedsn("211.183.7.253", "1521", "xe")
    con = cx_Oracle.connect(user="scott", password="tiger", dsn=dsnStr)

    cursor = con.cursor()
    #cursor.execute("insert into dept(deptno, dname, loc) values(50, '총무', '목포')")
    cursor.execute('insert into dept values(:1, :2, :3)', (60, '영업', '서울'))

    con.commit()
    print("삽입 성공")
except Exception as e:
    print('exception:', e)
finally:
    con.close()
```

Oracle 연동

❖ 데이터 수정

```
import cx_Oracle
try:
    # 데이터베이스에 연결합니다.
    dsnStr = cx_Oracle.makedsn("211.183.7.253", "1521", "xe")
    con = cx_Oracle.connect(user="scott", password="tiger", dsn=dsnStr)

    cursor = con.cursor()
    cursor.execute('update dept set dname = :1 where deptno = :2', ('회계', 60))

    con.commit()
    print("수정 성공")
except Exception as e:
    print('exception:', e)
finally:
    con.close()
```


Oracle 연동

❖ 데이터 삭제

```
import cx_Oracle
try:
    # 데이터베이스에 연결합니다.
    dsnStr = cx_Oracle.makedsn("211.183.7.253", "1521", "xe")
    con = cx_Oracle.connect(user="scott", password="tiger", dsn=dsnStr)

    cursor = con.cursor()
    cursor.execute('delete from dept where deptno = :1', (60,))

    con.commit()
    print("삭제 성공")
except Exception as e:
    print('exception:', e)
finally:
    con.close()
```

Oracle 연동

❖데이터 검색

1. 연결 객체의 cursor() 메소드를 호출해서 sql 실행 객체를 생성
2. execute(실행 할 sql문장)
3. cursor 객체를 가지고 fetchall 메소드를 호출하면 튜플들의 튜플로 결과가 리턴되며 fetchone 메소드를 호출하면 첫번째 데이터 1개만 튜플로 리턴

Oracle 연동

❖ 데이터 1개 검색

```
import cx_Oracle
try:
    # 데이터베이스에 연결합니다.
    dsnStr = cx_Oracle.makedsn("211.183.7.253", "1521", "xe")
    con = cx_Oracle.connect(user="scott", password="tiger", dsn=dsnStr)

    cursor = con.cursor()

    cursor.execute("select * from dept")
    data = cursor.fetchone()
    for imsi in data:
        print(imsi)
except Exception as e:
    print('exception:', e)
finally:
    con.close()
```

Oracle 연동

❖ 데이터 여러 개 검색

```
import cx_Oracle
try:
    # 데이터베이스에 연결합니다.
    dsnStr = cx_Oracle.makedsn("211.183.7.253", "1521", "xe")
    con = cx_Oracle.connect(user="scott", password="tiger", dsn=dsnStr)

    cursor = con.cursor()

    cursor.execute("select * from dept")
    data = cursor.fetchall()
    for imsi in data:
        print(imsi)
except Exception as e:
    print('exception:', e)
finally:
    con.close()
```