오라클연동

- 연결 함수를 공유해서 사용하면 안됨(충돌 발생), 각자 connection을 가지고 있어야 함

1. DB(오라클) 연동 함수 : 전화 번호

```
import os

def addr_insert(prm_name, prm_tel) :

    conn= cx_Oracle.connect("ai","0000", "localhost:1521/XE")
    sql ="INSERT INTO ADDR VALUES(ADDR_SEQ.NEXTVAL, :1, :2)"
    cur= conn.cursor()
    cur.execute(sql, [prm_name, prm_tel])

    conn.commit()
    cur.close()
    conn.close()

def addr_update(prm_name, prm_tel, prm_seq) :
    conn = cx_Oracle.connect("ai", "0000", "localhost:1521/XE")
    sql = "update addr set name=:1, tel=:2 where seq=:3"
    cur = conn.cursor()
    cur.execute(sql, [prm_name, prm_tel, prm_seq])
    conn.commit()
    cur.close()
    conn.close()

def addr_delete(prm_seq):
    conn= cx_Oracle.connect("ai","0000", "localhost:1521/XE")
    sql ="delete from addr where seq = :1"
    cur= conn.cursor()
    cur.execute(sql, [prm_seq])
    conn.commit()
    cur.close()
    conn.close()

def addr_select():
    conn = cx_Oracle.connect("ai", "0000", "localhost:1521/XE")
    sql = "select * from addr"
    cur = conn.cursor()
    cur.execute(sql)
    for row in cur:
        print( list(row)  )
    cur.close()
    conn.close()
```

```

def run():
    while(1):
        menu_print()
        cmd = input("명령어입력 : ")

         if cmd=="1":
             prm_name=input("이름 : ")
             prm_tel = input("전화번호 : ")
             addr_insert(prm_name, prm_tel)
             print("입력되었습니다.")
             addr_select()

         elif cmd=="2":
             addr_select()

         elif cmd =="3":
             addr_select()
             prm_seq= input("삭제 번호 : ")
             addr_delete(prm_seq)
             print("삭제되었습니다.")
             addr_select()

         elif cmd == "4":
             prm_seq = input("수정할 번호 : ")
             prm_name = input("이름 수정 : ")
             prm_tel = input("전화번호 수정 : ")
             addr_update(prm_name, prm_tel, prm_seq)
             print("수정되었습니다.")
             addr_select()

         elif cmd == "5":
             print("종료되었습니다.")
             break


 if __name__ == "__main__":
     print("직접돌려보기")
     run()
```

2. DB(오라클)로 파일보내기 : 파일저장

```
#기본틀
import cx_Oracle

fr = open(file="C:\\AI\\pythonProject\\PycharmProjects\\pythonProject\\venv\\test\\address.txt", mode ='r', encoding='utf-8')

txt_list = fr.readlines()
#print(txt_list)

data_list=[]
```

```
dict_key1=""
dict_key2=""

for i, txt in enumerate(txt_list):
    #print(i,txt)
    tlist = txt.strip().split("\t")
    print(tlist)
    if i==0:
        dict_key1=tlist[0]
        dict_key2=tlist[1]
    else:
        data_dic={dict_key1:tlist[0], dict_key2:tlist[1]}
        data_list.append(data_dic)


conn = cx_Oracle.connect("ai", "0000", "localhost:1521/XE")
sql= "insert into addr(seq,name,tel) values(addr_seq.nextval, :name, :tel)"
cur=conn.cursor()
cur.excutemany(sql, data_list)
conn.commit()
cur.close()
conn.close()
```

```
import os

import cx_Oracle
from os.path import isfile, join

def addr_insert(prm_name, prm_tel) :

    conn= cx_Oracle.connect("ai","0000", "localhost:1521/XE")
    sql ="INSERT INTO ADDR VALUES(ADDR_SEQ.NEXTVAL, :1, :2)"
    cur= conn.cursor()
    cur.execute(sql, [prm_name, prm_tel])

    conn.commit()
    cur.close()
    conn.close()

def addr_update(prm_name, prm_tel, prm_seq) :
    conn = cx_Oracle.connect("ai", "0000", "localhost:1521/XE")
    sql = "update addr set name=:1, tel=:2 where seq=:3"
    cur = conn.cursor()
    cur.execute(sql, [prm_name, prm_tel, prm_seq])
    conn.commit()
    cur.close()
    conn.close()

def addr_delete(prm_seq):
    conn= cx_Oracle.connect("ai","0000", "localhost:1521/XE")
    sql ="delete from addr where seq = :1"
    cur= conn.cursor()
    cur.execute(sql, [prm_seq])
    conn.commit()
    cur.close()
    conn.close()

def addr_select():
    conn = cx_Oracle.connect("ai", "0000", "localhost:1521/XE")
    sql = "select * from addr"
    cur = conn.cursor()
    cur.execute(sql)
    for row in cur:
        print( list(row)  )
    cur.close()
    conn.close()

def menu_print():
    print("------------------------------------------------")
    print("1.입력(insert)", end="\t")  # -----
    print("2.전체(select)", end="\t")  # -----
    print("3.수정(update)", end="\t")  # -----
    print("4.삭제(delete)", end="\t")  # -----
    print("5.파일저장")             # -----
    print("6.종료")  # -----
    print("------------------------------------------------")

def file_db_insert(fname) :
    fr = open(file=fname, encoding='UTF-8', mode='r')
    txt_list = fr.readlines()

    data_list = []
    dict_key1 = ""
    dict_key2 = ""
    for i, txt in enumerate(txt_list):
        tlist = txt.strip().split("\t")
        if len(tlist) != 2:
            print("__\t__ 형식의 포맷이 아닙니다")
            break

        if i == 0: #첫 줄(리스트[0])
            dict_key1 = tlist[0] #NAME
            dict_key2 = tlist[1] #TEL
        else: # 첫 줄 제외한 나머지 줄(리스트[1]~)
            data_dict = {dict_key1: tlist[0], dict_key2: tlist[1]}
                               #NAME 값(홍길동,...)    #TEL 값(010,...)
            data_list.append(data_dict)


    # datas = [  {"name":"나이름1", "tel":"111"} ,
    #            {"name":"나이름2", "tel":"222"} ,
    #            {"name":"나이름3", "tel":"333"}
    #         ]

    conn = cx_Oracle.connect("ai", "0000", "localhost:1521/XE")
    sql = "insert into addr(seq, name, tel) values(addr_seq.nextval, :name, :tel)"
    cur = conn.cursor()
    cur.executemany(sql, data_list)
    conn.commit()
    cur.close()
    conn.close()
```

```
def run() :
    while (True):
        menu_print()  # -----------------------------------------
        cmd = input("명령어입력:")
        if cmd == "1":
            nm = input("이름:")
            tel = input("전화번호:")
            addr_insert(nm,tel)  # -----------------------------------------
        elif cmd == "2":
            addr_select()
        elif cmd == "3":
            nm = input("이름:")
            tel = input("전화번호:")
            seq = input("번호:")
            addr_update(nm,tel, seq)
        elif cmd == "4":
            seq = input("번호:")
            addr_delete(seq)  # -----------------------------------------
        elif cmd == "5":
            fname = input("업로드 할 파일명 :")
            if len(fname) == 0:
                fname = "lec06_주소록.txt"
            fullname = join(os.getcwd() , fname)
            if isfile(fullname) :
                file_db_insert(fname)
            else :
                print("대상파일이 없습니다 확인 후 시도하세요")
        elif cmd == "6":
            break

if __name__ == "__main__" :
    print("직접돌려보기")
    run()
```

3. 키오스크 만들기

- 테이블 구조

```
<메뉴리스트>


KIO_GOODS(상품)
GOOD_SEQ    NUMBER      PRIMARY KEY    ---> KIO_GOODS_SEQ
GOOD_NAME   VC2(50)  상품명
GOOD_IMG    VC2(100) 이미지경로  /a/bb.png
GOOD_PRICE  NUMBER 상품가격     22000
GOOD_DESC   VC2(400) 상세설명
REG_DATE        DATE DEFAULT SYSDATE
```

```
def goods_list () : 상품목록출력

KIO_CART
CART_SEQ    NUMBER        PRIMARY KEY  ---> KIO_CART_SEQ
TEL                      NUMBER  : 휴대폰뒷번호
GOOD_SEQ        VC2(50)     상품번호
GOOD_PRICE      NUMBER   상품가격
ORDER_AMOUNT    NUMBER  주문수량
ORDER_PRICE     NUMBER  주문가격
-------------------
PAY_GUBUN           CHAR(1) 결제구분(1:현금 2:카드 3:포인트)
REG_DATE            DATE DEFAULT SYSDATE


def cart_add()  [ 1 햄버거 2 5000   4545
                      2 콜라 1 1500 ]    4545
                                  6500
def orders():
```

- 오라클

```
#오라클
    CART_SEQ NUMBER PRIMARY KEY,
    TEL varchar2(4),
    GOOD_SEQ     NUMBER default 0,
    GOOD_PRICE   NUMBER default 0,
    ORDER_AMOUNT NUMBER default 0,
    ORDER_PRICE  NUMBER default 0,
    PAY_GUBUN CHAR(1 BYTE) default '1',
    REG_DATE DATE default sysdate
   );

   CREATE TABLE KIO_GOODS (
    GOOD_SEQ   NUMBER PRIMARY KEY,
    GOOD_NAME  VARCHAR2(50),
    GOOD_IMG   VARCHAR2(100),
    GOOD_PRICE NUMBER default 0,
    GOOD_DESC  VARCHAR2(400),
    REG_DATE   DATE default sysdate
 );

 create sequence KIO_CART_SEQ  start with 1 increment by 1 nocache;
 create sequence KIO_GOODS_SEQ start with 1 increment by 1 nocache;

 insert into
KIO_CART(CART_SEQ,TEL,GOOD_SEQ,GOOD_PRICE,ORDER_AMOUNT,ORDER_PRICE,PAY_GUBUN,REG_DATE)
 values(KIO_CART_SEQ.nextval, '0505', 1, 1000, 2, 2000, '1', sysdate);

 select * from KIO_CART;

 insert into
KIO_GOODS(GOOD_SEQ,GOOD_NAME,GOOD_IMG,GOOD_PRICE,GOOD_DESC,REG_DATE)
 values (KIO_GOODS_SEQ.nextval, '도미노피자', './img/domino1.jpg', 1000, '이건 설명', sysdate);
 select * from KIO_GOODS;


# join문
 select *
 from KIO_ORDER o, KIO_CART c, KIO_GOODS g
 where c.tel = o.tel
 and c.good_seq = g.good_seq;
```

- python

```
import cx_Oracle

# --------------------------------------
# 상품목록
# --------------------------------------
def goods_list () :
    conn = cx_Oracle.connect("ai", "0000", "localhost:1521/XE")
    sql = "select * from KIO_GOODS"
    cur = conn.cursor()
    cur.execute(sql)
    for row in cur:
        print(list(row))
    cur.close()
    conn.close()

#--------------------------------------
# 카트담기
# insert into KIO_CART(CART_SEQ,TEL,GOOD_SEQ,GOOD_PRICE,ORDER_AMOUNT,REG_DATE)
#  values(KIO_CART_SEQ.nextval, '0505', 1, 1000, 2, 2000, sysdate);
#--------------------------------------
def cart_add(TEL,GOOD_SEQ,GOOD_PRICE,ORDER_AMOUNT)  :
    conn = cx_Oracle.connect("ai", "0000", "localhost:1521/XE")
    sql = '''insert
             into KIO_CART(CART_SEQ,TEL,GOOD_SEQ,GOOD_PRICE,ORDER_AMOUNT,REG_DATE)
             values(KIO_CART_SEQ.nextval, :1, :2, :3, :4, sysdate)'''
    cur = conn.cursor()
    cur.execute(sql, [TEL,GOOD_SEQ,GOOD_PRICE,ORDER_AMOUNT])
    conn.commit()
    cur.close()
    conn.close()

# --------------------------------------
# 상품주문
# --------------------------------------
def orders(TEL,ORDER_PRICE,PAY_GUBUN):
    conn = cx_Oracle.connect("ai", "0000", "localhost:1521/XE")
    sql = '''insert
             into KIO_ORDER(ORDER_SEQ,TEL,ORDER_PRICE,PAY_GUBUN,REG_DATE)
             values(KIO_ORDER_SEQ.nextval, :1, :2, :3, sysdate)'''
    cur = conn.cursor()
    cur.execute(sql, [TEL,ORDER_PRICE,PAY_GUBUN])
    conn.commit()
    cur.close()
    conn.close()
```