ai데이터융합\_모의평가1\_답안지

성명 : 정지용

작성일 : 2023년 06월 13일

<문제-1>

–하단의 내용은 자유롭게 변경해도 됩니다.

CREATE TABLE IF NOT EXISTS stud (

studID CHAR(8),

name VARCHAR(20),

jumin1 CHAR(6),

jumin2 CHAR(7),

addr1 VARCHAR(20),

addr2 VARCHAR(20)

);

INSERT INTO stud VALUES

('20180101', '홍길동', '200229', '3234567', '서울시', '서대1구'),

('20180102', '이순신', '720229', '1234567', '수원시', '서대2구'),

('20180103', '강감찬', '781212', '1328399', '강릉시', '서대3구'),

('20190101', '이성계', '790307', '2262722', '부산시', '서대4구'),

('20190102', '을지문덕', '780618', '2258211', '대구시', '서대5구'),

('20190103', '연개소문', '771009', '2215327', '울산시', '서대6구'),

('20190104', '안중근', '780209', '1573216', '이천시', '서대7구'),

('20200101', '김구', '770515', '2024213', '동두천시', '서대8구'),

('20200102', '정도전', '800613', '1078166', '대전시', '서대9구'),

('20200103', '이황', '720711', '2024219', '세종시', '서대10구');

CREATE TABLE IF NOT EXISTS course (

courseID CHAR(2),

CourseName VARCHAR(10)

);

INSERT INTO course VALUES ('01', '국어');

INSERT INTO course VALUES ('02', '영어');

INSERT INTO course VALUES ('03', '수학');

CREATE TABLE IF NOT EXISTS score (

studID CHAR(8),

CourseID VARCHAR(10),

score INT(03)

);

DELETE FROM score;

INSERT INTO score (studID)

SELECT studID

FROM stud;

INSERT INTO score (CourseID)

SELECT courseID

FROM course;

import sqlite3

print(sqlite3.sqlite\_version\_info)

try:

    conn = sqlite3.connect("./data/sqlite\_db.db")

    cursor = conn.cursor()

    sql = 'create table if not exists stud(studID text(8), name text(20), jumin1 text(6), jumin2 text(7), addr1 text(20), addr2 text(20))'

    cursor.execute(sql)

    cursor.execute("insert into stud values('20180101', '홍길동', '200229', '3234567', '서울시', '서대1구')")

    cursor.execute("insert into stud values('20180102', '이순신', '720229', '1234567',  '수원시', '서대2구')")

    cursor.execute("insert into stud values('20180103', '강감찬', '781212', '1328399', '강릉시', '서대3구')")

    cursor.execute("insert into stud values('20190101', '이성계', '790307', '2262722', '부산시', '서대4구')")

    cursor.execute("insert into stud values('20190102', '을지문덕', '780618', '2258211', '대구시', '서대5구')")

    cursor.execute("insert into stud values('20190103', '연개소문', '771009', '2215327', '울산시', '서대6구')")

    cursor.execute("insert into stud values('20190104', '안중근', '780209', '1573216', '이천시', '서대7구')")

    cursor.execute("insert into stud values('20200101', '김구', '770515', '2024213', '동두천시', '서대8구')")

    cursor.execute("insert into stud values('20200102', '정도전', '800613', '1078166', '대전시', '서대9구')")

    cursor.execute("insert into stud values('20200103', '이황', '720711', '2024219', '세종시', '서대10구')")

    conn.commit()

    cursor.execute("select \* from stud")

    rows = cursor.fetchall()

    for row in rows :

        print(row)

    # # print('이름 \t전화번호   \t주소')

    # for row in rows:

    #     print(row[0], '\t', row[1], '\t', row[2])

except Exception as e :

    print('db 연동 실패 : ', e)

    conn.rollback()

finally:

    cursor.close()

    conn.close()

try:

    conn = sqlite3.connect("./data/sqlite\_db.db")

    cursor = conn.cursor()

    sql = 'create table if not exists course(courseID text(02), CourseName text(10))'

    cursor.execute(sql)

    cursor.execute("insert into course values('01', '국어')")

    cursor.execute("insert into course values('02', '영어')")

    cursor.execute("insert into course values('03', '수학')")

    conn.commit()

    cursor.execute("select \* from course")

    rows = cursor.fetchall()

    for row in rows :

        print(row)

    # # print('이름 \t전화번호   \t주소')

    # for row in rows:

    #     print(row[0], '\t', row[1], '\t', row[2])

except Exception as e :

    print('db 연동 실패 : ', e)

    conn.rollback()

finally:

    cursor.close()

    conn.close()

try:

    conn = sqlite3.connect("./data/sqlite\_db.db")

    cursor = conn.cursor()

    sql = 'create table if not exists score(studID text(08), CourseID text(02), score integer(03))'

    cursor.execute(sql)

    cursor.execute("insert into score values('01', '국어')")

    cursor.execute("insert into course values('02', '영어')")

    cursor.execute("insert into course values('03', '수학')")

    conn.commit()

    cursor.execute("select \* from score")

    rows = cursor.fetchall()

    for row in rows :

        print(row)

    # # print('이름 \t전화번호   \t주소')

    # for row in rows:

    #     print(row[0], '\t', row[1], '\t', row[2])

except Exception as e :

    print('db 연동 실패 : ', e)

    conn.rollback()

finally:

    cursor.close()

    conn.close()

<문제-2>

–하단의 내용은 자유롭게 변경해도 됩니다.

import os

import sys

import pymysql # MySQL데이터베이스를 사용하기 위한 라이브러리를 등록함

config = {

'host' : '127.0.0.1',

'user' : 'user1',

'passwd' : '0000',

'database' : 'test\_db',

'port' : 3306,

'charset' : 'utf8',

'use\_unicode' : True

}

class GoodsRead :

    def \_\_init\_\_(self,read\_sel): # 생성자 : read\_sel : 코드/상품명/all ,

        self.read\_sel = read\_sel

        if read\_sel == '학번' :  #  read\_sql : "select \* from goods where code =" / "select \* from goods where name ="

            self.read\_sql = "select \* from stud where studID ="

        elif read\_sel == '학생이름' :

            self.read\_sql = "select \* from stud where name ="

        else :

            self.read\_sql = "select \* from stud"

    def goodsModify(self) :

        try :

            conn = pymysql.connect(\*\*config)    # 딕셔너리 config를 인수로 사용하여 conn 객체를 만듬.

            cursor = conn.cursor()   # conn 객체로부터 cursor() 메소드를 호출하여 cursor 참조변수를 만듬.

            print("<<<학번을 입력하세요>>>")

            in\_studID = int(input('수정할 학번 입력 : '))

            sql = f"select \* from stud where studID = {in\_studID}"

            cursor.execute(sql) # sql문 실행

            rows = cursor.fetchall()

            if rows :

                print('<<학번 조회결과입니다>>>')

                sql = f"select \* from stud where studID = {in\_studID}"

                print(rows)

                # os.system("pause")

                # print(f'{in\_code}, ')

                YN = input("수정하시겠습니까?(y/n) :")

                if YN == 'y' or YN == 'Y':

                    in\_name = input("학생이름을 입력하세요 : ")

                    in\_jumin1 = input("주민번호 앞자리를 입력하세요 : ")

                    in\_jumin2 = input("주민번호 뒷자리를 입력하세요 : ")

                    sql = f"update stud set name = '{in\_name}', jumin1 = {in\_jumin1}, jumin2 = {in\_jumin2} where studID = {in\_studID}"

                    cursor.execute(sql)

                    conn.commit()

                    print('수정을 완료했습니다.')

                    print('<<<학생 수정 결과입니다>>>')

                    sql = f"select \* from stud where studID = {in\_studID}"

                    cursor.execute(sql)

                    rows = cursor.fetchall()

                    print(rows)

                elif YN == 'n' or YN == 'N':

                    print('수정을 거절하셨습니다.')

                else:

                    print("y나 n을 입력하세요.")

            else :

                print('수정할 코드가 없습니다.')

                os.system("pause")

        except Exception as e :

            print('db 연동 실패 : ', e)

            conn.rollback() # 실행 취소

        finally:

            cursor.close()

            conn.close()

    def goodsDelete(self) :

        try :

            conn = pymysql.connect(\*\*config)    # 딕셔너리 config를 인수로 사용하여 conn 객체를 만듬.

            cursor = conn.cursor()   # conn 객체로부터 cursor() 메소드를 호출하여 cursor 참조변수를 만듬.

            print("<<<삭제할 학번을 입력하세요>>>")

            in\_studID = int(input('삭제할 코드 입력 : '))

            sql = f"select \* from stud where studID = {in\_studID}"

            cursor.execute(sql) # sql문 실행

            rows = cursor.fetchall()

            if rows :

                # 레코드 1개 출력 : index 이용

                print('삭제 성공했습니다.')

                os.system("pause")

                sql = f"delete from stud where studID = {in\_studID}"

                cursor.execute(sql) # sql문 실행

                conn.commit()

            else :

                print('삭제 실패했습니다.')

                os.system("pause")

        except Exception as e :

            print('db 연동 실패 : ', e)

            conn.rollback() # 실행 취소

        finally:

            cursor.close()

            conn.close()

    def goodsReadOne(self) :

        try :

            # (1) db 연동 객체

            conn = pymysql.connect(\*\*config)    # 딕셔너리 config를 인수로 사용하여 conn 객체를 만듬.

            # sql 실행 객체

            cursor = conn.cursor()     # conn 객체로부터 cursor() 메소드를 호출하여 cursor 참조변수를 만듬.

            # (3) 단일 레코드 조회

            rows = []

            os.system('cls')

            if self.read\_sel == '학번' :

                print("<<<학생 개별 조회({})입니다>>>".format(self.read\_sel))

                in\_studID = int(input("조회할"+self.read\_sel+"를 입력하세요 : "))

                sql = self.read\_sql + f"'{in\_studID}'"

            elif self.read\_sel == '학생이름' :

                print("<<<학생 개별 조회({})입니다>>>".format(self.read\_sel))

                in\_studID = input("조회할"+self.read\_sel+"를 입력하세요 : ")

                sql = self.read\_sql + f"'{in\_studID}'"

            else :

                #print("<<<상품 목록 조회결과입니다>>>")

                in\_studID = ''

                sql = self.read\_sql

            cursor.execute(sql)

            rows = cursor.fetchall()

            if self.read\_sel == '학번' or self.read\_sel == '학생이름' :

                if len(rows) > 0 :

                    print("===학생테이블 조회2({})===".format(self.read\_sel))

                    for row in rows :

                        print("조회결과는 코드:{}, 품명:{}, 수량:{}, 단가:{}  입니다.".format(int(row[0]),row[1],int(row[2]),int(row[3])))

                else:

                    print("조회결과 입력한 {}에 맞는 상품이 없습니다".format(self.read\_sel))

            else: # self.read\_sel == 'all'일때

                if len(rows) > 0 :

                    print("<<<학생 목록 조회결과입니다>>>")

                    for row in rows :

                        print("{} {} {} {} ".format(int(row[0]),row[1],int(row[2]),int(row[3])))

                else:

                    print("조회결과 없습니다")

        except Exception as e :

            print('db 연동 실패 : ', e)

            conn.rollback() # 실행 취소

        finally:

            cursor.close()

            conn.close()

# def tableCreate() :

#     try :

#         print("----->")

#         conn = pymysql.connect(\*\*config)

#         cursor = conn.cursor()

#         sql = """create table stud(

#             studID int primary key,

#             name varchar(30) not null,

#             su int default 0,

#             dan int default 0

#             )"""

#         cursor.execute(sql)

#         conn.commit()

#     except Exception as e :

#         print("오류 : ",e)

#         conn.rollback()

#     finally :

#         conn.close()

#         cursor.close()

def goodsCreate() :

    try :

        # (1) db 연동 객체

        conn = pymysql.connect(\*\*config)    # 딕셔너리 config를 인수로 사용하여 conn 객체를 만듬.

        # sql 실행 객체

        cursor = conn.cursor()       # conn 객체로부터 cursor() 메소드를 호출하여 cursor 참조변수를 만듬.

        os.system('cls')

        print("<<<학생 등록입니다>>>")

        in\_studID = int(input("학번을 입력하세요 : "))  #

        if in\_studID > 0 :

            #print("------>>>>")

            sql = f"select \* from stud where studID= {in\_studID}"

            cursor.execute(sql)

            rows = cursor.fetchall()  # rows = [ (선풍기, 25, 14) ]

            if len(rows) > 0 :

                print("이미 존재하고 있습니다. 다른 코드를 입력하세요")

            else :

                in\_name = input("학생이름을 입력하세요 : ")

                in\_jumin1 = input("주민번호 앞자리를 입력하세요 : ")

                in\_jumin2 = input("주민번호 뒷자리를 입력하세요 : ")

                sql = f"insert into stud(studID,name) values({in\_studID},'{in\_name}', {in\_jumin1}, {in\_jumin2})"

                cursor.execute(sql)

                rows = cursor.fetchall()

                conn.commit()

                print("학생등록을 성공했습니다.")

                print()

        else :

            print("학생 등록을 위해 학번을 입력해 주세요")

    except Exception as e :

        print('오류 : ', e)

        conn.rollback() # 실행 취소

    finally:

        cursor.close()

        conn.close()

def goodsReadAll() :

    try :

        # (1) db 연동 객체

        conn = pymysql.connect(\*\*config)    # 딕셔너리 config를 인수로 사용하여 conn 객체를 만듬.

        # sql 실행 객체

        cursor = conn.cursor()     # conn 객체로부터 cursor() 메소드를 호출하여 cursor 참조변수를 만듬.

        # (2) 모든 레코드 조회

        os.system('cls')

        print("<<<학생 목록 조회입니다>>>")

        cursor.execute("select \* from stud")

        rows = cursor.fetchall()

        print("===학생 테이블 조회1===")

        print("(studID, name)")

        for row in rows :

            print(row)

    except Exception as e :

        print('db 연동 실패 : ', e)

        conn.rollback() # 실행 취소

    finally:

        cursor.close()

        conn.close()

class filter :

    def filter\_stud(self, studID):

        flag1\_txt, flag2\_txt = '', ''

        result = match('[0-9]{8}', studID)

        if len(studID[0:]) > 8 :

            flag1 = False

            flag1\_txt = '학번 길이오류/'

        else:

            flag1 = True

    #

        if 1900 <= int(studID[0:4]) <= 2023 :

            flag2 = True

        else:

            flag2 = False

            flag2\_txt = '학번 입학년도 오류/'

    #

        if 1 <= int(studID[4:6]) <= 99 :

            flag2 = True

        else:

            flag2 = False

            flag2\_txt = '학번 입학월 오류/'

        if 1 <= int(studID[6:]) <= 99 :

            flag2 = True

        else:

            flag2 = False

            flag2\_txt = '학번 입학날짜 오류/'

    #

        # if flag2 == True and year\_st == '평년':

        #     if month[int(studID[2:4])-1] < int(studID[4:6]):

        #         flag4 = False

        #         flag4\_txt = '주민번호 앞자리 월 오류/'

        #     else:

        #         flag4 = True

        # if flag2 == True and year\_st == '윤년':

        #     if month[int(studID[2:4])-1] < int(studID[4:6]):

        #         flag4 = False

        #         flag4\_txt = '주민번호 앞자리 월 오류/'

        #     else:

        #         flag4 = True

    #

    # 함수의 역할을 끝내고 리턴할 값을 만듬

        return\_txt = ''

        if result and flag1 and flag2 :

            return studID+'-->','정상적인 학번'

        else :

            return\_txt += flag1\_txt + flag2\_txt

            return studID+'-->', '비정상적인 학번 : '+ return\_txt

def jumin\_chk(jumin) :

    #

    flag1\_txt, flag2\_txt, flag3\_txt = '', '', ''

    result = match('[0-9]{6}-[1-4][0-9]{6}', jumin)

    if len(jumin[0:]) > 14 :

        flag1 = False

        flag1\_txt = '주민번호 길이오류/'

    else:

        flag1 = True

    #

    if 1 <= int(jumin[2:4]) <= 12 :

        flag2 = True

    else:

        flag2 = False

        flag2\_txt = '주민번호 앞자리 월 오류/'

    #

    if len(jumin[7:]) > 7 :

        flag3 = False

        flag3\_txt = '주민번호 뒷자리 길이오류/'

    else:

        flag3 = True

    # 함수의 역할을 끝내고 리턴할 값을 만듬

    return\_txt = ''

    if result and flag1 and flag2 and flag3 :

        return jumin+'-->' + '정상적인 주민번호'

    else :

        return\_txt += flag1\_txt + flag2\_txt + flag3\_txt

        return jumin+'-->' + '비정상적인 주민번호 : '+ return\_txt

if \_\_name\_\_ == "\_\_main\_\_" :

    #tableCreate()

    while True:

        os.system('cls')

        print("---학생관리---")

        print("학생    등록 : 1 ")

        print("학생    조회 : 2 ")

        print("학반별  조회 : 3 ")

        print("학생이름별조회 : 4 ")

        print("학생    수정 : 5 ")

        print("학생    삭제 : 6 ")

        print("학생관리종료 : 9 ")

        sel = int(input("작업을 선택하세요 : "))

        if sel == 1 :

            goodsCreate()

            os.system("pause")

        elif sel == 2 :

            goodsReadAll()

            os.system("pause")

        elif sel == 3 :

            r1 = GoodsRead('학번')

            r1.goodsReadOne()

            os.system("pause")

        elif sel == 4 :

            r2 = GoodsRead('학생이름')

            r2.goodsReadOne()

            os.system("pause")

        elif sel == 5 :

            r3 = GoodsRead('all')

            r3.goodsModify()

            os.system("pause")

        elif sel == 6 :

            #print("상품삭제기능은 준비중입니다. ")

            r4 = GoodsRead('all')

            r4.goodsReadOne()

            r4.goodsDelete()

            r4.goodsReadOne()

            os.system("pause")

        elif sel == 9 :

            print("상품관리를 종료합니다. ")

            os.system("pause")

            os.system('cls')

            sys.exit(0)

        else :

            print("잘못 선택했습니다. ")

            os.system("pause")

<문제-3>

–하단의 내용은 자유롭게 변경해도 됩니다.

import os

import sys

import pymysql # MySQL데이터베이스를 사용하기 위한 라이브러리를 등록함

config = {

'host' : '127.0.0.1',

'user' : 'user1',

'passwd' : '0000',

'database' : 'test\_db',

'port' : 3306,

'charset' : 'utf8',

'use\_unicode' : True

}

class GoodsRead :

    def \_\_init\_\_(self,read\_sel): # 생성자 : read\_sel : 코드/상품명/all ,

        self.read\_sel = read\_sel

        if read\_sel == '과목코드' :  #  read\_sql : "select \* from goods where code =" / "select \* from goods where name ="

            self.read\_sql = "select \* from course where courseID ="

        elif read\_sel == '과목명' :

            self.read\_sql = "select \* from course where CourseName ="

        else :

            self.read\_sql = "select \* from course"

    def goodsModify(self) :

        try :

            conn = pymysql.connect(\*\*config)    # 딕셔너리 config를 인수로 사용하여 conn 객체를 만듬.

            cursor = conn.cursor()   # conn 객체로부터 cursor() 메소드를 호출하여 cursor 참조변수를 만듬.

            print("<<<과목코드를 입력하세요>>>")

            in\_courseID = int(input('수정할 과목코드 입력 : '))

            sql = f"select \* from course where courseID = {in\_courseID}"

            cursor.execute(sql) # sql문 실행

            rows = cursor.fetchall()

            if rows :

                print('<<과목코드 조회결과입니다>>>')

                sql = f"select \* from course where courseID = {in\_courseID}"

                print(rows)

                # os.system("pause")

                # print(f'{in\_code}, ')

                YN = input("수정하시겠습니까?(y/n) :")

                if YN == 'y' or YN == 'Y':

                    in\_CourseName = input("과목명을 입력하세요 : ")

                    sql = f"update course set CourseName = '{in\_CourseName}' where courseID = {in\_courseID}"

                    cursor.execute(sql)

                    conn.commit()

                    print('수정을 완료했습니다.')

                    print('<<<학생 수정 결과입니다>>>')

                    sql = f"select \* from course where courseID = {in\_courseID}"

                    cursor.execute(sql)

                    rows = cursor.fetchall()

                    print(rows)

                elif YN == 'n' or YN == 'N':

                    print('수정을 거절하셨습니다.')

                else:

                    print("y나 n을 입력하세요.")

            else :

                print('수정할 과목코드가 없습니다.')

                os.system("pause")

        except Exception as e :

            print('db 연동 실패 : ', e)

            conn.rollback() # 실행 취소

        finally:

            cursor.close()

            conn.close()

    def goodsDelete(self) :

        try :

            conn = pymysql.connect(\*\*config)    # 딕셔너리 config를 인수로 사용하여 conn 객체를 만듬.

            cursor = conn.cursor()   # conn 객체로부터 cursor() 메소드를 호출하여 cursor 참조변수를 만듬.

            print("<<<삭제할 과목코드 입력하세요>>>")

            in\_courseID = int(input('삭제할 코드 입력 : '))

            sql = f"select \* from course where courseID = {in\_courseID}"

            cursor.execute(sql) # sql문 실행

            rows = cursor.fetchall()

            if rows :

                # 레코드 1개 출력 : index 이용

                print('삭제 성공했습니다.')

                os.system("pause")

                sql = f"delete from course where courseID = {in\_courseID}"

                cursor.execute(sql) # sql문 실행

                conn.commit()

            else :

                print('삭제 실패했습니다.')

                os.system("pause")

        except Exception as e :

            print('db 연동 실패 : ', e)

            conn.rollback() # 실행 취소

        finally:

            cursor.close()

            conn.close()

    def goodsReadOne(self) :

        try :

            # (1) db 연동 객체

            conn = pymysql.connect(\*\*config)    # 딕셔너리 config를 인수로 사용하여 conn 객체를 만듬.

            # sql 실행 객체

            cursor = conn.cursor()     # conn 객체로부터 cursor() 메소드를 호출하여 cursor 참조변수를 만듬.

            # (3) 단일 레코드 조회

            rows = []

            os.system('cls')

            if self.read\_sel == '학번' :

                print("<<<과목 개별 조회({})입니다>>>".format(self.read\_sel))

                in\_coursedID = int(input("조회할"+self.read\_sel+"를 입력하세요 : "))

                sql = self.read\_sql + f"'{in\_courseID}'"

            elif self.read\_sel == '학생이름' :

                print("<<<과목 개별 조회({})입니다>>>".format(self.read\_sel))

                in\_courseID = input("조회할"+self.read\_sel+"를 입력하세요 : ")

                sql = self.read\_sql + f"'{in\_courseID}'"

            else :

                #print("<<<상품 목록 조회결과입니다>>>")

                in\_courseID = ''

                sql = self.read\_sql

            cursor.execute(sql)

            rows = cursor.fetchall()

            if self.read\_sel == '과목코드' or self.read\_sel == '과목명' :

                if len(rows) > 0 :

                    print("===과목테이블 조회2({})===".format(self.read\_sel))

                    for row in rows :

                        print("조회결과는 과목코드:{}, 과목명:{}  입니다.".format(int(row[0]),row[1]))

                else:

                    print("조회결과 입력한 {}에 맞는 상품이 없습니다".format(self.read\_sel))

            else: # self.read\_sel == 'all'일때

                if len(rows) > 0 :

                    print("<<<과목명 목록 조회결과입니다>>>")

                    for row in rows :

                        print("{} {}".format(int(row[0]),row[1]))

                else:

                    print("조회결과 없습니다")

        except Exception as e :

            print('db 연동 실패 : ', e)

            conn.rollback() # 실행 취소

        finally:

            cursor.close()

            conn.close()

def goodsCreate() :

    try :

        # (1) db 연동 객체

        conn = pymysql.connect(\*\*config)    # 딕셔너리 config를 인수로 사용하여 conn 객체를 만듬.

        # sql 실행 객체

        cursor = conn.cursor()       # conn 객체로부터 cursor() 메소드를 호출하여 cursor 참조변수를 만듬.

        os.system('cls')

        print("<<<과목 등록입니다>>>")

        in\_courseID = int(input("과목코드을 입력하세요 : "))  #

        if in\_courseID > 0 :

            #print("------>>>>")

            sql = f"select \* from course where courseID= {in\_courseID}"

            cursor.execute(sql)

            rows = cursor.fetchall()  # rows = [ (선풍기, 25, 14) ]

            if len(rows) > 0 :

                print("이미 존재하고 있습니다. 다른 코드를 입력하세요")

            else :

                in\_CourseName = input("과목명을 입력하세요 : ")

                sql = f"insert into course(courseID, CourseName) values({in\_courseID},'{in\_CourseName}')"

                cursor.execute(sql)

                rows = cursor.fetchall()

                conn.commit()

                print("과목등록을 성공했습니다.")

                print()

        else :

            print("과목 등록을 위해 과목코드를 입력해 주세요")

    except Exception as e :

        print('오류 : ', e)

        conn.rollback() # 실행 취소

    finally:

        cursor.close()

        conn.close()

def goodsReadAll() :

    try :

        # (1) db 연동 객체

        conn = pymysql.connect(\*\*config)    # 딕셔너리 config를 인수로 사용하여 conn 객체를 만듬.

        # sql 실행 객체

        cursor = conn.cursor()     # conn 객체로부터 cursor() 메소드를 호출하여 cursor 참조변수를 만듬.

        # (2) 모든 레코드 조회

        os.system('cls')

        print("<<<과목 목록 조회입니다>>>")

        cursor.execute("select \* from course")

        rows = cursor.fetchall()

        print("===과목 테이블 조회1===")

        print("(courseID, CourseName)")

        for row in rows :

            print(row)

    except Exception as e :

        print('db 연동 실패 : ', e)

        conn.rollback() # 실행 취소

    finally:

        cursor.close()

        conn.close()

if \_\_name\_\_ == "\_\_main\_\_" :

    #tableCreate()

    while True:

        os.system('cls')

        print("---과목관리---")

        print("과목    등록 : 1 ")

        print("과목    조회 : 2 ")

        print("과목코드 조회 : 3 ")

        print("과목명  조회 : 4 ")

        print("과목    수정 : 5 ")

        print("과목    삭제 : 6 ")

        print("과목관리종료 : 9 ")

        sel = int(input("작업을 선택하세요 : "))

        if sel == 1 :

            goodsCreate()

            os.system("pause")

        elif sel == 2 :

            goodsReadAll()

            os.system("pause")

        elif sel == 3 :

            r1 = GoodsRead('학번')

            r1.goodsReadOne()

            os.system("pause")

        elif sel == 4 :

            r2 = GoodsRead('학생이름')

            r2.goodsReadOne()

            os.system("pause")

        elif sel == 5 :

            r3 = GoodsRead('all')

            r3.goodsModify()

            os.system("pause")

        elif sel == 6 :

            #print("상품삭제기능은 준비중입니다. ")

            r4 = GoodsRead('all')

            r4.goodsReadOne()

            r4.goodsDelete()

            r4.goodsReadOne()

            os.system("pause")

        elif sel == 9 :

            print("과목관리를 종료합니다. ")

            os.system("pause")

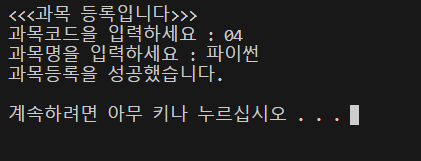
            os.system('cls')

            sys.exit(0)

        else :

            print("잘못 선택했습니다. ")

            os.system("pause")



<문제-4>

–하단의 내용은 자유롭게 변경해도 됩니다.