

```

import pymysql # 导入库

conn = pymysql.connect(
    host='localhost',
    user='python',
    password='python',
    database='testDB'
) # 建立连接
cursor = conn.cursor() # 获取控制器

def initialize():
    mysql = "select * from Question_types;"
    cursor.execute(mysql)
    result = cursor.fetchall()
    for i in result:
        print("drop table type" + str(i[0]) + ";")
        cursor.execute("drop table type" + str(i[0]) + ";")
    # cursor.execute("drop table 'question*")

    try:
        cursor.execute('drop table Users')
    except pymysql.err.Error:
        pass
    print(1)
    try:
        cursor.execute('drop table Question_types')
    except pymysql.err.Error:
        pass
    print(2)

    return True

def set_up():
    try:
        # set up the table Users
        mysql = '''create table if not exists Users(
            id int(4) not null primary key auto_increment,
            name varchar(64) not null,
            is_teacher int(4) not null default '0');''' # 创造表单users, 包含id值, 姓名, 是否为老师
        cursor.execute(mysql) # 执行
        mysql = "insert into Users (name,is_teacher) values ('teacher',1);" # 插入第一个老师
        cursor.execute(mysql) # 执行
        conn.commit() # 刷新数据库

        # set up the table Question_types
        mysql = '''create table if not exists Question_types(
            id int(4) not null primary key auto_increment,
            name varchar(64) not null
        );
        ''' # id为编号,name为名称
        cursor.execute(mysql) # 执行
    except pymysql.err.Error:
        return False

def new_question_type(name: str) -> bool: # 新建问题类型
    try:
        mysql = "select * from Question_types where name='" + name + "';"
        cursor.execute(mysql)

```

```

    result = cursor.fetchall() # 获取所有和正在创的题目名字一样的题目
    if len(result) != 0: # 如果有重名的
        return False # 不能进行操作

    mysql = "insert into Question_types (name) value ('" + name + "');"
    cursor.execute(mysql)

    mysql = 'select * from Question_types;'
    cursor.execute(mysql)
    result = cursor.fetchall() # 找到现在所有的题目类型
    thisid = result[0][0]
    if len(result):
        thisid = result[len(result) - 1][0] # 现在正在创建的题目编号为原有题目数量+

    mysql = "create table type" + str(thisid) + '''(
        id int(4) not null primary key auto_increment,
        name varchar(64) not null,
        is_multi_choice int(4) not null default '0'
    );
    ''' # 表格存储这个类型的题库下所有题目的名字, id, 以及问题是选择题还是填空题, 0
    # 为填空 (默认), 1为选择
    cursor.execute(mysql) # 创建该题目类型的专用表单
    conn.commit() # 插入Question_type并刷新数据库

    return True # 操作成功
except pymysql.err.Error:
    return False

def del_question_type(name='', id=-1) -> bool: # 从数据库删除一个题目类型
    try:
        if name == '' and id == -1:
            return False # 如果即没有指定名称也没有指定id, 就无法进行操作
        if id == -1: # 如果每给定id (给定了名称)
            return del_question_type_by_name(name) # 通过名称删除
        else:
            return del_question_type_by_id(id) # 反之
    except pymysql.err.Error:
        return False

def del_question_type_by_name(name) -> bool: # 通过名称删除
    try:
        mysql = "select * from Question_types where name='" + name + "';"
        cursor.execute(mysql)
        result = cursor.fetchall() # 查找是否有名字是这个的
        if len(result) == 0:
            return False # 如果根本没有这个名称的题目类型, 操作失败
        thisid = result[0][0] # 如果有, 那么要删除的就是id为是这个名字的题类的id, id在表单中每
        # 一行的第一列
        return del_question_type_by_id(thisid) # 通过id删除
    except pymysql.err.Error:
        return False

def del_question_type_by_id(id): #通过id删除
    try:
        mysql = "delete from Question_types where id='" + str(id) + "';"
        cursor.execute(mysql)
        mysql = "drop table type" + str(id) + ";";
        cursor.execute(mysql) # 删除

```

```

        return True
    except pymysql.err.Error:
        return False

def get_all_question_types() -> bool:          #获取所有的问题类型
    mysql = "select * from Question_types;"
    try:
        cursor.execute(mysql)
        result = cursor.fetchall()
    except pymysql.err.Error:
        print("<--error: please set-up first-->")
        return False
    if len(result) == 1:
        print("<-- ! no available question types yet ! -->")
        return True
    for i in range(len(result)):
        print(i + 1, ": ", result[i][1])
    return True

def new_question(name: str) -> bool:            #新建问题
    question_type = input("enter the question_type>")    #输入问题类型
    try:
        mysql = "select * from Question_types where name='" + name + "';"
        cursor.execute(mysql)          #查找所有可用的问题类型
    except pymysql.err.Error as e:
        if "doesn't exist" in str(e):
            print("<--error: please set-up first-->")
            return False                #如果没有question-types表的话，操作失败，提示需要先setup
    result = cursor.fetchall()
    if not len(result):
        print("<--no such question type, the available question types are:-->")
        get_all_question_types()
        return False                    #如果没有这个问题类型的话，也操作失败
    question_type_id = result[0][0]
    print("<--question_type_found-->")
    is_multi_choice = input("is this question a multiple choice? (Y or N)")
    if is_multi_choice == "Y" or is_multi_choice == "y":
        is_multi_choice = 1
    elif is_multi_choice == "N" or is_multi_choice == "n":
        is_multi_choice = 0
    else:
        print("<--error:invalid value for (Y or N) by " + is_multi_choice)
        return False                    #询问是否为选择题
    try:
        mysql = "insert into " + result + "(name,is_multi_choice) values('" + name +
        "','" + str(is_multi_choice) + "';"
        cursor.execute(mysql)
        mysql = "creat table "
    except pymysql.err.Error:
        return False
    # TODO 完善创建问题的表单的程序并测试

    return True

new_question("question type number 1")

```