



祢恹

码龄3年

暂无认证

19

原创

1183

积分



TA的主页

私信

关注

Q

最新文章

- Ubuntu 20.04初搭k8s 1.18.3集群
- ubuntu安装mysql
- whistle 在Android7.0以上手机的证书安装
- 采用nodejs实现grpc的简单demo
- 推荐一款远程控制App---Core shell

归档

2020			
7月 1篇	6月 1篇	3月 1篇	
2019			
11月 1篇	9月 1篇	8月 1篇	7月 1篇
2018			

热门文章

Python 中 (&, |) 和 (and, or) 之间的

python 借助pymysql操作MySQL及占位符问题

原创 祢恹 2019-04-02 14:00:51 4665 收藏 3

版权

文章目录

- pycharm操作MySQL
- SQL语句编写 (占位符的应用)

pycharm操作MySQL

在python3中, 主要借助pymysql进行MySQL操作, 简单记录下基本的操作步骤:
操作流程一般分为3步:

1. 建立数据库连接;
2. 执行操作 (查询、插入、更新、删除等)
3. 关闭连接

这里直接贴代码了, 用函数的形式进行表述了:

```
import pymysql

# 数据库连接
def connect():
    conn = pymysql.connect(host='localhost',
                           port=3306,
                           user='root',
                           password='root',
                           database='njust',
                           charset='utf8')

    # 获取操作游标
    cursor = conn.cursor()
    return {"conn": conn, "cursor": cursor}
```

在执行操作时, 借助游标方法: cursor.execute() 执行SQL操作。

```
# 1、查询操作并打印结果
def select_sql(table):
    connection = connect()
    conn, cursor = connection['conn'], connection['cursor']
    sql = "select * from %s" % table
    try:
```

区别 171106

pandas 对数据的读取和存储 (学习笔记)

18257

Python应用 pyecharts画中国地图 13496

雷达图的绘制 (Python, matplotlib)

11274

Pycharm中运行程序在Python console中执行, 不是直接Run 5464

最新评论



采用nodejs实现grpc的简单...

wonderful418 : 这是同步还是异步?

Pycharm中运行程序在Pyth...

AndyJane97 : 感谢

Python 中 (&, |...

fi_before : 学了C/C++后表示看了体验极差

Python 中 (&, |...

SimZhou : 位运算和逻辑运算, 所有编程语言都有的东西。

```

cursor.execute(sql)
results = cursor.fetchall()
print(results)
except Exception as e:
    raise e
finally:
    cursor.close()
    conn.close()

```

插入操作

```

def insert_sql(persons_values):
    connection = connect()
    conn, cursor = connection['conn'], connection['cursor']

    keys = ", ".join(persons_values.keys())
    qmark = ", ".join(["%s"] * len(persons_values))
    sql_insert = "insert into persons(%s) values (%s)" % (keys, qmark)
    print(sql_insert)
    try:
        cursor.execute(sql_insert, list(persons_values.values()))
        conn.commit()
        print("插入成功")
    except Exception as e:
        print(e)
        conn.rollback()
        print("插入失败")
    finally:
        cursor.close()
        conn.close()

```

利用字典进行插入

```

def insert_sql2(message):
    connection = connect()
    conn, cursor = connection['conn'], connection['cursor']

    sql_insert = "insert into persons(ID, LastName, FirstName) " \
        "values (%(ID)s, %(LastName)s, %(FirstName)s)"

    try:
        cursor.execute(sql_insert, message)
        conn.commit()
        print("插入成功")
    except Exception as e:
        print(e)
        conn.rollback()
        print("插入失败")
    finally:
        cursor.close()

```

目录



```

conn.close()

# 更新数据库
def update_sql():
    connection = connect()
    conn, cursor = connection['conn'], connection['cursor']

    sql_update = "update persons set birthday=%s where ID=%s"
    try:
        cursor.execute(sql_update, ('2001/7/5', 3))
        conn.commit()
        print('更新成功')
    except Exception as e:
        print('更新失败', e)
        conn.rollback()
    finally:
        cursor.close()
        conn.close()
    pass

# 删除操作
def delete_sql(lastname):
    connection = connect()
    conn, cursor = connection['conn'], connection['cursor']

    sql_delete = "delete from persons where LastName=%s"
    try:
        cursor.execute(sql_delete, lastname)
        conn.commit()
        print('删除成功')
    except Exception as e:
        print('删除失败', e)
        conn.rollback()
    finally:
        cursor.close()
        conn.close()
    pass

```

SQL语句编写（占位符的应用）

执行的关键还是在SQL语句的编写。这类有几种方式：

- 完整的SQL语句，直接调用

```

sql_select = "select * from tablename"
cursor.execute(sql_select)

```

- 利用占位符传递参数。这里要注意，无论整数、字符串，占位符都为 %s，且不需加引号，

- 在sql语句中借助占位符，组成完整SQL

```
tabel = 'persons'
sql = "select * from %s" % table
cursor.execute(sql)
```

- 参数替代

```
tabel = 'persons'
sql = "select * from %s"
cursor.execute(sql, tabel)

# 参数多于1个时，execute()传入参数应为list或者tuple类型
sql_update = "update persons set birthday=%s where ID=%s"
cursor.execute(sql_update, ('2001/7/5', 3))
```

- 字典类型传递变量，这里要保证占位符的keys要包含在传递的字典keys中

```
# 这里的占位符%s修改为%(字典keyname)s
sql_insert = "insert into persons(ID, LastName, FirstName) " \
            "values (%(ID)s, %(LastName)s, %(FirstName)s)"
message = {
    "ID": 7,
    "LastName": "Jone",
    "FirstName": "Bob",
}
cursor.execute(sql_insert, message)
```

参考来源：

[python3操作MySQL数据库](#)

[Python连接MySQL数据库执行sql语句时的参数问题](#)

👍 点赞 2 💬 评论 1 ➦ 分享 ★ 收藏 3 📱 手机看 💰 打赏 ... 关注

获取Linux下Ftp目录树并逐步绑定到treeview

06-08

在linux下抓取目录树，双击后获取该节点子节点（逐步生成）。另外有两个类，一个是windows下的（一次性获取目录树），一个是linux下的（足部获取...

Python中pymysql通过字典插入数据

tzyyyl的博

客

3223

引子：今天参加面试被大佬要求写一段数据库接口实操代码，首先在写字板上开头：definsert(table,d... [https://w.....](#)



mysql 的 占位符使用 以及一次执行多行sql语句	づ開始懂了的博	客	2万+
占位符的使用： drop PROCEDURE if EXISTS test12;-- 使用drop 来删除存储过程或者表create PROCEDURE test1... s			
python SQL语句 占位符	学	习python	816
今天在处理数据库中的数据是出现了一个错误，导致数据表格里数据一直... cursor.execute(sql update, (json.d.....			
pymysql中插入数据时format的坑	花样多的很的博	客	5752
入坑今天向mysql插入数据时，无意间使用到了format，发现了个小坑坑先看一段代码：我... -----+ Field ...			
Python 占位符格式化	***** 图标 *****	客	1万+
一、常用占位符的含义s：获取传入对象的 str 方法的返回值，并将其格式化到指定位置r：获取传入对象的 r... 0			
mysql的占位符问题	qq_35107993的博	客	1190
<if test=' field != "" and field != null '>order by \${field}<if test=' order != "" and order != null' >\${order}</if>... ...			
python写入MySQL数据库	m0_43445668的博	客	1万+
这次给大家带来的是将python爬取的数据写入数据库将爬取得数据写入数据库的步骤：连接数据库；创建表将...			
mysql的动态sql实现变量表名及占位符的使用	づ開始懂了的博	客	1万+
MySQL 在存储过程中是不支持直接使用变量名作为表名或者是列名的，而在实际的应用中确实会用到变量名或... ..			
python SQL语句变量拼接	fang156239305的博	客	508
tablename = ['t acq data 20180901','t acq data 20180902']curtable=tablename[0]sql = ... ults = cursor.fetchall()			
©2020 CSDN 皮肤主题: 终极编程指南 设计师: CSDN官方博客 返回首页			