

---

# **Mandb**

***Release 0.1.4***

**May 08, 2017**



---

## Contents

---

<b>1</b>	<b>Release history</b>	<b>5</b>
1.1	Version 0.1.4, May 08 2017 . . . . .	5
1.2	Version 0.1, Mar 31 2017 . . . . .	5
<b>2</b>	<b>Indices and tables</b>	<b>7</b>
	<b>Python Module Index</b>	<b>9</b>



Mandb is a lightweight wrapper around MySQLdb and sqlite3.

This lib is inspired by torndb and DBUtils. It supports DBUtils to manage your exists connection. If you has any good ideas, please contact me <kehr.china@gmail.com>

**exception** `mandb.MandbException`

Base exception for mandb

**class** `mandb.Row`

A dict that allows for object-like property access syntax.

**class** `mandb.Database` (*connection=None, \*\*kwargs*)

This class provide a series of base database operations. It can manage your connection, if you already has one.

Example:

```
import MySQLdb
from mandb import Database
from DBUtils.PooledDB import PooledDB

pdb = PooledDB(MySQLdb, host='localhost', port=3306, db='test_db',
               user='root', passwd='passwd', mincached=5, charset='utf8')
db = Database(pdb.connection())
db.query('SELECT ...')
db.insert('INSERT INTO ...')
db.update('UPDATE ...')
db.delete('DELETE ...')
...
```

Otherwise, please use *MySQLDatabase* or *SqliteDatabase* to create a new connection.

**Args:**

**connection** Specify an exists database connection.

**kwargs** Connection parameters.

**connect()**

Get this database connection

**close()**

Closes this database connection

**is\_closed()**

Return if connnection is closed

**iter** (*sql, \*args, \*\*kwargs*)

Returns an iterator for the given query and parameters.

**query** (*sql, \*args, \*\*kwargs*)

Returns a row list for the given query and parameters.

**get** (*sql, \*args, \*\*kwargs*)

Returns the (singular) row returned by the given query.

If the query has no results, returns None. If it has more than one result, raises an exception.

**execute** (*sql, \*args, \*\*kwargs*)

Executes the given sql, returning the lastrowid.

**rollback()**

Rolls backs the current transaction

**execute\_lastrowid** (*sql*, \**args*, \*\**kwargs*)

Executes the given sql, returning the lastrowid.

**execute\_rowcount** (*sql*, \**args*, \*\**kwargs*)

Executes the given query, returning the rowcount.

**executemany** (*sql*, *args*)

Executes the given query against all the given param sequences.

**executemany\_lastrowid** (*sql*, *args*)

Executes the given query against all the given param sequences.

**executemany\_rowcount** (*sql*, *args*)

Executes the given query against all the given param sequences.

**update** (*sql*, \**args*, \*\**kwargs*)

Executes the given query, returning the rowcount.

**delete** (*sql*, \**args*, \*\**kwargs*)

Executes the given query, returning the rowcount.

**updatemany** (*sql*, *args*)

Executes the given query against all the given param sequences.

**insert** (*sql*, \**args*, \*\**kwargs*)

Executes the given sql, returning the lastrowid.

**insertmany** (*sql*, *args*)

Executes the given query against all the given param sequences.

**class** mandb.**SqliteDatabase** (*db*, \**args*, \*\**kwargs*)

Subclass of *Database*, wrapper for Sqlite3

usage:

```
from mandb import SqliteDatabase

db = SqliteDatabase(db='test.db')
db.query('SELECT ...')
db.insert('INSERT INTO ...')
db.update('UPDATE ...')
db.delete('DELETE ...')
...
```

**Args:**

**db** The sqlite database file.

**class** mandb.**MySQLDatabase** (\**args*, \*\**kwargs*)

Subclass of *Database*, wrapper for MySQL

usage:

```
from mandb import MySQLDatabase

db = MySQLDatabase(host='localhost', port=3306, db='test',
                   user='root', passwd='123456', charset='utf8')
db.query('SELECT ...')
db.insert('INSERT INTO ...')
db.update('UPDATE ...')
db.delete('DELETE ...')
...
```







# CHAPTER 1

---

## Release history

---

### Version 0.1.4, May 08 2017

- Bug fix: `Database._execute` does not format sql by args when kwargs and args are empty.

### Version 0.1, Mar 31 2017

- Support `MySQLdb`, `sqlite3` and `DBUtils`.
- Make `DBUtils` support autocommit by default.



## CHAPTER 2

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



**m**

mandb, [5](#)



### C

`close()` (mandb.Database method), 1  
`connect()` (mandb.Database method), 1

### D

`Database` (class in mandb), 1  
`delete()` (mandb.Database method), 2

### E

`execute()` (mandb.Database method), 1  
`execute_lastrowid()` (mandb.Database method), 1  
`execute_rowcount()` (mandb.Database method), 2  
`executemany()` (mandb.Database method), 2  
`executemany_lastrowid()` (mandb.Database method), 2  
`executemany_rowcount()` (mandb.Database method), 2

### G

`get()` (mandb.Database method), 1

### I

`insert()` (mandb.Database method), 2  
`insertmany()` (mandb.Database method), 2  
`is_closed()` (mandb.Database method), 1  
`iter()` (mandb.Database method), 1

### M

`mandb` (module), 1  
`MandbEception`, 1  
`MySQLDatabase` (class in mandb), 2

### Q

`query()` (mandb.Database method), 1

### R

`rollback()` (mandb.Database method), 1  
`Row` (class in mandb), 1

### S

`SqliteDatabase` (class in mandb), 2

### U

`update()` (mandb.Database method), 2  
`updatemany()` (mandb.Database method), 2