

SQLite - PHP

安装

自 PHP 5.3.0 起默认启用 SQLite3 扩展。可以在编译时使用 **--without-sqlite3** 禁用 SQLite3 扩展。

Windows 用户必须启用 php_sqlite3.dll 才能使用该扩展。自 PHP 5.3.0 起，这个 DLL 被包含在 PHP 的 Windows 分发版中。

如需了解详细的安装指导，建议查看我们的 PHP 教程和它的官方网站。

PHP 接口 API

以下是重要的 PHP 程序，可以满足您在 PHP 程序中使用 SQLite 数据库的需求。如果您需要了解更多细节，请查看 PHP 官方文档。

序号	API & 描述
1	public void SQLite3::open (filename, flags, encryption_key) 打开一个 SQLite 3 数据库。如果构建包括加密，那么它将尝试使用的密钥。 如果文件名 <i>filename</i> 赋值为 ':memory:' ，那么 SQLite3::open() 将会在 RAM 中创建一个内存数据库，这只会 <i>session</i> 的有效时间内持续。 如果文件名 <i>filename</i> 为实际的设备文件名称，那么 SQLite3::open() 将使用这个参数值尝试打开数据库文件。如果该名称的文件不存在，那么将创建一个新的命名为该名称的数据库文件。 可选的 <i>flags</i> 用于判断是否打开 SQLite 数据库。默认情况下，当使用 SQLITE3_OPEN_READWRITE SQLITE3_OPEN_CREATE 时打开。
2	public bool SQLite3::exec (string \$query) 该例程提供了一个执行 SQL 命令的快捷方式，SQL 命令由 <i>sql</i> 参数提供，可以由多个 SQL 命令组成。该程序用于对给定的数据库执行一个无结果的查询。
3	public SQLite3Result SQLite3::query (string \$query) 该例程执行一个 SQL 查询，如果查询到返回结果则返回一个 SQLite3Result 对象。
4	public int SQLite3::lastErrorCode (void) 该例程返回最近一次失败的 SQLite 请求的数值结果代码。
5	public string SQLite3::lastErrorMsg (void) 该例程返回最近一次失败的 SQLite 请求的英语文本描述。
6	public int SQLite3::changes (void) 该例程返回最近一次的 SQL 语句更新或插入或删除的数据库行数。
7	public bool SQLite3::close (void) 该例程关闭之前调用 SQLite3::open() 打开的数据库连接。

8 **public string SQLite3::escapeString (string \$value)**
该例程返回一个字符串，在 SQL 语句中，出于安全考虑，该字符串已被正确地转义。

连接数据库

下面的 PHP 代码显示了如何连接到一个现有的数据库。如果数据库不存在，那么它就会被创建，最后将返回一个数据库对象。

```
<?php
class MyDB extends SQLite3
{
    function __construct()
    {
        $this->open('test.db');
    }
}
$db = new MyDB();
if(!$db){
    echo $db->lastErrorMsg();
} else {
    echo "Opened database successfully\n";
}
?>
```

现在，让我们来运行上面的程序，在当前目录中创建我们的数据库 **test.db**。您可以根据需要改变路径。如果数据库成功创建，那么会显示下面所示的消息：

```
Open database successfully
```

创建表

下面的 PHP 代码段将用于在先前创建的数据库中创建一个表：

```
<?php
class MyDB extends SQLite3
{
    function __construct()
    {
        $this->open('test.db');
    }
}
$db = new MyDB();
if(!$db){
    echo $db->lastErrorMsg();
} else {
```

```
        echo "Opened database successfully\n";
    }

    $sql =<<<EOF
        CREATE TABLE COMPANY
        (ID INT PRIMARY KEY     NOT NULL,
        NAME          TEXT      NOT NULL,
        AGE           INT        NOT NULL,
        ADDRESS       CHAR(50),
        SALARY        REAL);
    EOF;

    $ret = $db->exec($sql);
    if(!$ret){
        echo $db->lastErrorMsg();
    } else {
        echo "Table created successfully\n";
    }
    $db->close();
?>
```

上述程序执行时，它会在 **test.db** 中创建 COMPANY 表，并显示下面所示的消息：

```
Opened database successfully
Table created successfully
```

INSERT 操作

下面的 PHP 程序显示了如何在上面创建的 COMPANY 表中创建记录：

```
<?php
class MyDB extends SQLite3
{
    function __construct()
    {
        $this->open('test.db');
    }
}
$db = new MyDB();
if(!$db){
    echo $db->lastErrorMsg();
} else {
    echo "Opened database successfully\n";
}

$sql =<<<EOF
    INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY)
```

```
VALUES (1, 'Paul', 32, 'California', 20000.00 );

INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY)
VALUES (2, 'Allen', 25, 'Texas', 15000.00 );

INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY)
VALUES (3, 'Teddy', 23, 'Norway', 20000.00 );

INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY)
VALUES (4, 'Mark', 25, 'Rich-Mond ', 65000.00 );
EOF;

$ret = $db->exec($sql);
if(!$ret){
    echo $db->lastErrorMsg();
} else {
    echo "Records created successfully\n";
}
$db->close();
?>
```

上述程序执行时，它会在 COMPANY 表中创建给定记录，并会显示以下两行：

```
Opened database successfully
Records created successfully
```

SELECT 操作

下面的 PHP 程序显示了如何从前面创建的 COMPANY 表中获取并显示记录：

```
<?php
class MyDB extends SQLite3
{
    function __construct()
    {
        $this->open('test.db');
    }
}
$db = new MyDB();
if(!$db){
    echo $db->lastErrorMsg();
} else {
    echo "Opened database successfully\n";
}

$sql = <<<EOF
SELECT * from COMPANY;
```

```
EOF;

$ret = $db->query($sql);
while($row = $ret->fetchArray(SQLITE3_ASSOC) ){
    echo "ID = ". $row['ID'] . "\n";
    echo "NAME = ". $row['NAME'] . "\n";
    echo "ADDRESS = ". $row['ADDRESS'] . "\n";
    echo "SALARY = ". $row['SALARY'] . "\n\n";
}
echo "Operation done successfully\n";
$db->close();
?>
```

上述程序执行时，它会产生以下结果：

```
Opened database successfully
ID = 1
NAME = Paul
ADDRESS = California
SALARY = 20000

ID = 2
NAME = Allen
ADDRESS = Texas
SALARY = 15000

ID = 3
NAME = Teddy
ADDRESS = Norway
SALARY = 20000

ID = 4
NAME = Mark
ADDRESS = Rich-Mond
SALARY = 65000

Operation done successfully
```

UPDATE 操作

下面的 PHP 代码显示了如何使用 UPDATE 语句来更新任何记录，然后从 COMPANY 表中获取并显示更新的记录：

```
<?php
class MyDB extends SQLite3
{
    function __construct()
    {
```

```
$this->open('test.db');
    }
}
$db = new MyDB();
if(!$db){
    echo $db->lastErrorMsg();
} else {
    echo "Opened database successfully\n";
}
$sql =<<<EOF
    UPDATE COMPANY set SALARY = 25000.00 where ID=1;
EOF;
$ret = $db->exec($sql);
if(!$ret){
    echo $db->lastErrorMsg();
} else {
    echo $db->changes(), " Record updated successfully\n";
}

$sql =<<<EOF
    SELECT * from COMPANY;
EOF;
$ret = $db->query($sql);
while($row = $ret->fetchArray(SQLITE3_ASSOC) ){
    echo "ID = ". $row['ID'] . "\n";
    echo "NAME = ". $row['NAME'] . "\n";
    echo "ADDRESS = ". $row['ADDRESS'] . "\n";
    echo "SALARY = ". $row['SALARY'] . "\n\n";
}
echo "Operation done successfully\n";
$db->close();
?>
```

上述程序执行时，它会产生以下结果：

```
Opened database successfully
1 Record updated successfully
ID = 1
NAME = Paul
ADDRESS = California
SALARY = 25000

ID = 2
NAME = Allen
ADDRESS = Texas
SALARY = 15000

ID = 3
```

```
NAME = Teddy
ADDRESS = Norway
SALARY = 20000
```

```
ID = 4
NAME = Mark
ADDRESS = Rich-Mond
SALARY = 65000
```

Operation done successfully

DELETE 操作

下面的 PHP 代码显示了如何使用 DELETE 语句删除任何记录，然后从 COMPANY 表中获取并显示剩余的记录：

```
<?php
class MyDB extends SQLite3
{
    function __construct()
    {
        $this->open('test.db');
    }
}
$db = new MyDB();
if(!$db){
    echo $db->lastErrorMsg();
} else {
    echo "Opened database successfully\n";
}
$sql = <<<<EOF
DELETE from COMPANY where ID=2;
EOF;
$ret = $db->exec($sql);
if(!$ret){
    echo $db->lastErrorMsg();
} else {
    echo $db->changes(), " Record deleted successfully\n";
}

$sql = <<<<EOF
SELECT * from COMPANY;
EOF;
$ret = $db->query($sql);
while($row = $ret->fetchArray(SQLITE3_ASSOC) ){
    echo "ID = ". $row['ID'] . "\n";
    echo "NAME = ". $row['NAME'] . "\n";
    echo "ADDRESS = ". $row['ADDRESS'] . "\n";
    echo "SALARY = ". $row['SALARY'] . "\n\n";
}
```

```
}  
echo "Operation done successfully\n";  
$db->close();  
?>
```

上述程序执行时，它会产生以下结果：

```
Opened database successfully  
1 Record deleted successfully  
ID = 1  
NAME = Paul  
ADDRESS = California  
SALARY = 25000  
  
ID = 3  
NAME = Teddy  
ADDRESS = Norway  
SALARY = 20000  
  
ID = 4  
NAME = Mark  
ADDRESS = Rich-Mond  
SALARY = 65000  
  
Operation done successfully
```

[← SQLite – Java](#)[SQLite – Perl →](#)[✎ 点我分享笔记](#)