



整数类型

主要有：

int, tinyint, smallint, mediumint, bigint

整数类型所占空间（字节）：

int: 占 4 个字节，即 32 位；

tinyint: 占 1 个字节，即 8 位；最多能存储 256 个数字，默认范围是-128~127

bigint: 占 8 个字节，即 64 位；

类型	字节	最小值 (带符号的/无符号的)	最大值 (带符号的/无符号的)
TINYINT	1	-128 0	127 255
SMALLINT	2	-32768 0	32767 65535
MEDIUMINT	3	-8388608 0	8388607 16777215
INT	4	-2147483648 0	2147483647 4294967295
BIGINT	8	-9223372036854775808 0	9223372036854775807 18446744073709551615

整数类型字段的设定形式：

类型名 **【(M)】** **【unsigned】** **【zerofill】**

说明：

- 1, M 表示设定该整数的“显示长度”，即 select 输出的时候，123 可能显示为 000123（假设 M 为 6）
- 2, unsigned 用于设定该整数为“无符号数”，其实就是没有负数。
- 3, zerofill 用于设定是否填充“0”到一个数字的左边，此时，需与设定的长度 M 配合使用。
- 4, 如果设置了 zerofill，则自动也就表示同时具备了 unsigned 修饰

举例：

```
mysql> create table tab_int(f1 int, f2 tinyint, f3 bigint);
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> create table tab_int2(
-> f1 int unsigned ,
-> f2 tinyint zerofill,
-> f3 bigint(10) zerofill
-> );
Query OK, 0 rows affected (0.00 sec)
```

小数类型

有 3 中：float, double, decimal。

float: 单精度浮点型，使用 4 个字节存储数据，其精度大约只有 6-7 个有效数字数位；

double: 双精度浮点型，使用 8 个字节存储数据，其精度大约只有 20 个有效数字数位；

decimal: 定点小数类型，整数部分最长可以有 65 位，小数部分最长可以有 30 位。一般设置格式为：decimal(总位数，小数部分位数)

```
24 //演示小数类型的使用：
25 create table tab_xiaoshu(
26     f1 float,
27     f2 double,
28     f3 decimal(10, 3)/*表示总位数为10,小数位数为3*/
29 );
30
31 insert into tab_xiaoshu (f1,f2,f3)
32     values(123.456, 123.456, 123.456);
33
34 insert into tab_xiaoshu (f1,f2,f3)
35     values(123.456789, 123.456789, 123.456789);
36 insert into tab_xiaoshu (f1,f2,f3)
37     values(123456789, 123456789, 1234567);
38
```

Php 操作 mysql 数据库基本函数复习

通常，php 网页中完成有关数据库的操作，首先，需要如下代码：

```
$link = mysql_connect("数据库服务器地址","用户名","密码"); //连接数据库系统
mysql_query("set names 网页文件编码名"); //设定“连接编码”；
//也可以这样做：mysql_set_charset("网页文件编码名");
mysql_query("use 数据库名"); //选定要使用的数据库
//也可以这样做：mysql_select_db("数据库名");
```

然后，才开始正式执行要完成的数据库操作任务（语句）：

```
$result = mysql_query("select / delete / update / insert / desc / show tables / drop .....");
```

其他个别补充：

mysql_error(): 获取 mysql 执行失败时的错误信息。它通常应用与如下场合：

```
$result = mysql_query("select / delete / update / insert / create / dropt ....所有 sql 语句");
if( $result === false){
```

```
    echo "语句执行失败，请参考提示信息：".mysql_error();
```

```
}
```

```
else{
```

```
    //这里是成功的情况，那就后续处理。。。
```

```
}
```

对于 mysql_query() 这个函数，可以执行“几乎任何 sql 语句”，在应用中，通常需要区分为 2 情况：

情况 1：执行没有数据返回的语句，比如：insert, update, delete, create table, create database, drop ...

这种语句，使用 mysql_query() 函数，执行的结果只有 true 和 false 返回：

返回 true：表示执行成功；

返回 false：表示执行失败；

情况 2：执行有数据返回的语句：select, show tables, show databases, desc 表名（显示表结构）；

此时，使用 mysql_query() 函数，返回的结果分为：

如果失败：返回的是 false；

如果成功：返回的是“结果集”——“数据集”——就是一行一行数据，就是有字段的一个“表结构”。它需要我们进一步去“取出”数据，才能使用其中的数据；

示例如下：

```
<?php
//通常，php网页中完成有关数据库的操作，首先，需要如下代码：
//$link = mysql_connect("数据库服务器地址","用户名","密码"); //连接数据库系统
$link = mysql_connect("localhost", "root", "123"); //连接数据库系统

//mysql_query("set names 网页文件编码名"); //设定“连接编码”；
mysql_query("set names utf8"); //设定“连接编码”；
//也可以这样做：mysql_set_charset("网页文件编码名");

//mysql_query("use 数据库名"); //选定要使用的数据库
mysql_query("use php39"); //选定要使用的数据库
//也可以这样做：mysql_select_db("数据库名");

//然后，才开始正式执行要完成的数据库操作任务（语句）：
//$result = mysql_query("select / delete /update / insert / desc / show tables /drop .....");

$num = rand(0, 355); //随机生成0-355之间的数字
$sql = "insert into tab_int2 (f1, f2, f3) values(123, $num, 123)";
$result = mysql_query( $sql );
if($result === false){
    echo "执行失败，请参考： " . mysql_error();
}
else{
    echo "执行成功！";
}

?>
```

```
$link = mysql_connect("localhost", "root", "123"); //连接数据库系统
mysql_query("set names utf8"); //设定“连接编码”；
mysql_query("use php39"); //选定要使用的数据库
```

```
$sql1 = "select * from tab_temp1";
$result = mysql_query($sql1);
if($result === false){
    echo "执行失败，请参考： " . mysql_error();
}
else{
    //此时，$result就是“结果集”（数据集）；
    echo "执行成功！数据如下： ";
    //echo $result;
```

```
while( $rec = mysql_fetch_array( $result )){
    echo "<br />f1: " . $rec['id'];
    echo "——f2: " . $rec['name'];
    echo "——f3: " . $rec['age'];
}
```

从select语句执行后获得的结果集中取出数据：

“fetch 函数”的 3 种形式的辨析：

假设 `mysql_query(“select id, age, name ”)` 执行的结果集为 `$result`，其该表中的数据类似这样：

id	name	age
1	user1	18
2	张三	28
4	user4	38

`$result = mysql_query(“select id, age, name ”);`

`mysql_fetch_assoc($result)`: 得到的数组类似这样：

`array("id" => 1, "name" => "user1", "age" => 18);`

I

`mysql_fetch_row($result)`: 得到的数组类似这样：

`array(0 => 1, 1 => "user1", 2 => 18);`

`mysql_fetch_array($result)`: 得到的数组类似这样：

`array("id" => 1, "name" => "user1", "age" => 18, 0 => 1, 1 => "user1", 2 => 18);`

```
<?php
$link = mysql_connect("localhost", "root", "123"); //连接数据库系统
mysql_query("set names utf8"); //设定“连接编码”；
mysql_query("use php39"); //选定要使用的数据库

$sql1 = "desc tab_temp1";
$result = mysql_query($sql1);
if($result === false){
    echo "执行失败，请参考： " . mysql_error();
}
else{
    //此时，$result就是“结果集”（数据集）；
    echo "执行成功！数据如下： ";
    //echo $result;
    echo "<table border='1'>";

    echo "<tr>";
    echo "<td>Field</td>";
    echo "<td>Type</td>";
    echo "<td>Null</td>";
    echo "<td>Key</td>";
    echo "<td>Default</td>";
    echo "<td>Extra</td>";
    echo "</tr>";

    while( $rec = mysql_fetch_array( $result ) ){
        //mysql_fetch_array()会取出该结果集中的“一行数据”，并取得该行数据后赋值给$rec；
        //此$rec就是一个数组，其下标就是字段名；
        //在此while循环中，mysql_fetch_array()会一次次（一行行）取出结果集中的所有数据；
        //然后，在这里就可以处理该数组$rec了；
    }
}
```



```

    echo "执行失败，请参考：" . mysql_error();
}
else{
    //此时，$result就是“结果集”（数据集）；
    echo "执行成功！数据如下：";
    //echo $result;
    echo "<table border='1'>";

    echo "<tr>";
    echo "<td>Field</td>";
    echo "<td>Type</td>";
    echo "<td>Null</td>";
    echo "<td>Key</td>";
    echo "<td>Default</td>";
    echo "<td>Extra</td>";
    echo "</tr>";

    while( $rec = mysql_fetch_array( $result ) ){
        echo "<tr>";
        echo "<td>" . $rec['Field'] . "</td>";
        echo "<td>" . $rec['Type'] . "</td>";
        echo "<td>" . $rec['Null'] . "</td>";
        echo "<td>" . $rec['Key'] . "</td>";
        echo "<td>" . $rec['Default'] . "</td>";
        echo "<td>" . $rec['Extra'] . "</td>";
        echo "</tr>";
    }
    echo "</table>";
}
?>

```

扩展 php 中操作 mysql 数据的几个函数：

\$n1 = mysql_num_rows(结果集); //获得该结果集的数据行数；

\$n2 = mysql_num_fields(结果集); //获得该结果集的数据列数；

\$name = mysql_field_name(结果集, \$i); //获得结果集的第 i 个字段的名称！i 从 0 开始算起

```
//此时，$result就是“结果集”（数据集）；
echo "执行成功！数据如下：";
//echo $result;
echo "<table border='1'>";
$field_count = mysql_num_fields($result);//获得列数
    echo "<tr>";
    for($i = 0; $i < $field_count; ++$i);
    echo "<td>Field</td>";
```

```
//此时，$result就是“结果集”（数据集）；
echo "执行成功！数据如下：";
//echo $result;
echo "<table border='1'>";
$field_count = mysql_num_fields($result);//获得列数
echo "<tr>";
for($i = 0; $i < $field_count; ++$i){
    $field_name = mysql_field_name($result, $i);
    echo "<td>" . $field_name . "</td>";
}
//echo "<td>Type</td>";
//echo "<td>Null</td>";
//echo "<td>Key</td>";
//echo "<td>Default</td>";
//echo "<td>Extra</td>";
echo "</tr>";

while( $rec = mysql_fetch_array( $result ) ){
/*
```

```
$sql1 = "show databases;";
$result = mysql_query($sql1);
if($result === false){
    echo "执行失败，请参考：" . mysql_error();
}
else{
```



```

mysql_query("use php39"); //选定要使用的数据库

$sql1 = "select * from tab_temp1"; //这里可以是任意返回结果集的sql语句
//比如select, desc, show。。。

$sql1 = "select * from user_info";
$sql1 = "select * from tab_int2";
$sql1 = "desc tab_temp1";
$sql1 = "show tables";
$sql1 = "show databases;";
$result = mysql_query($sql1);
if($result === false){
    echo "执行失败，请参考： " . mysql_error();
}
else{
    //此时，$result就是“结果集”（数据集）；
    echo "执行成功！数据如下： ";
    //echo $result;
    echo "<table border='1'>";
    $field_count = mysql_num_fields($result); //获得列数
    echo "<tr>";
    for($i = 0; $i < $field_count; ++$i){
        echo "<td>";
        $field_name = mysql_field_name($result, $i); //获得第i列的名字
        echo "<td>" . $field_name . "</td>";
    }
    echo "</tr>";

    while( $rec = mysql_fetch_array( $result ) ){
        /*
        echo "<tr>";
        //对这个结果集继续进行“列遍历”
        for($i = 0; $i < $field_count; ++$i){
            $field_name = mysql_field_name($result, $i); //获得第i列的名字
            echo "<td>" . $rec[$field_name] . "</td>";
        }
        echo "</tr>";
        */
    }
    echo "</table>";
}

```

```
$dbname = "php39";  
$link = mysql_connect("localhost", "root", "123"); //连接数据库系统  
mysql_query("set names utf8"); //设定“连接编码”;  
mysql_query("use $dbname"); //选定要使用的数据库
```

```
<?php  
$dbname = "php39";  
if(!empty( $_GET['db'] )){//如果db这个get数据不为空  
    $dbname = $_GET['db'];  
}
```