mysql数据库

今时今日，数据库系统已经成为各个动态网站上 web 应用程序的重要组成部分。

由于非常敏感和机密的数据有可能保存在数据库中，所以对数据库实施保护就显得尤为重要了。

要从数据库中提取或者存入数据，就必须经过连接数据库、发送一条合法查询、获取结果、关闭连接等步骤。

目前，能完成这一系列动作的最常用的查询语言是结构化查询语言 Structured Query Language (SQL)。

PHP 本身并不能保护数据库的安全。

设计数据库

第一步一般都是创建数据库，除非是使用第三方的数据库服务。

当创建一个数据库的时候，会指定一个所有者来执行和新建语句。通常，只有所有者（或超级用户）才有权对数据库中的对象进行任意操作。如果想让其他用户使用，就必须赋予他们权限。

应用程序永远不要使用数据库所有者或超级用户帐号来连接数据库，因为这些帐号可以执行任意的操作，比如说修改数据库结构（例如删除一个表）或者清空整个数据库的内容。

应该为程序的每个方面创建不同的数据库帐号，并赋予对数据库对象的极有限的权限。

仅分配给能完成其功能所需的权限，避免同一个用户可以完成另一个用户的事情。这样即使攻击者利用程序漏洞取得了数据库的访问权限，也最多只能做到和该程序一样的影响范围。

PHP连接MySQL

(PHP 5 >= 5.1.0, PHP 7, PHP 8, PECL pdo >= 0.2.0)

PDO::query — 执行 SQL 语句，以 PDOStatement 对象形式返回结果集

```php
<?php
$servername = "127.0.0.1:33060"; //mysql的地址
$username = "root";
$password = "root";
$dbname = "magedu";

try {
    $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
    echo "连接成功\r";
} catch (PDOException $e) {
    echo $e->getMessage();
}
```

> 允许MySQL远程登录
>
> GRANT ALL PRIVILEGES ON . TO 'root'@'%' IDENTIFIED BY '' WITH GRANT OPTION;

> FLUSH PRIVILEGES;

PHP执行SQL语句

```
$sql = "select * from magedu_students";
foreach ($conn->query($sql) as $row) {
  print $row['name'] . "\t";
  print $row['age'] . "\t";
}
```

PHP使用预编译的方式执行SQL语句

```
$sql2 = $conn->prepare("select * from magedu_students where age=?");
$age = 27;
$sql2->bindParam(1, $age);
$sql2->execute();
$res = $sql2->fetchAll();
print_r($res);
```

PDO::prepare方法

预编译防止SQL注入的原理是提前编译SQL语句，将所有的用户输入都当做数据，而非语法。

mysql提供预编译

预编译原理：提前进行sql语句的编译，在sql语句执行之前就确定sql的语义，保证用户输入的内容只当作变量执行