 如果您正在运行使用MySQL的Web应用程序，那么它把密码或者其他敏感信息保存在应用程序里的机会就很大。保护这些数据免受黑客或者窥探者的获取是一个令人关注的重要问题，因为您既不能让未经授权的人员使用或者破坏应用程序，同时还要保证您的竞争优势。幸运的是，MySQL带有很多设计用来提供这种类型安全的加密函数。本文概述了其中的一些函数，并说明了如何使用它们，以及它们能够提供的不同级别的安全。双向加密 就让我们从最简单的加密开始：双向加密。在这里，一段数据通过一个密钥被加密，只能够由知道这个密钥的人来解密。MySQL有两个函数来支持这种类型的加密，分别叫做ENCODE()和DECODE()。下面是一个简单的实例：mysql> INSERT INTO users (username, password) VALUES (joe, ENCODE(guessme, abracadabra)).Query OK, 1 row affected (0.14 sec) 其中，Joe的密码是guessme，它通过密钥abracadabra被加密。要注意的是，加密完的结果是一个二进制字符串，如下所示：mysql> SELECT * FROM users WHERE username=joe. --------- ---------- | username | password | --------- ---------- | joe | &iexcl.?i??!? | --------- ---------- 1 row in set (0.02 sec) abracadabra这个密钥对于恢复到原始的字符串至关重要。这个密钥必须被传递给DECODE()函数，以获得原始的、未加密的密码。下面就是它的使用方法：mysql> SELECT DECODE(password, abracadabra) FROM users WHERE

username=joe. ---------- | DECODE(password, abracadabra) | ---------- | guessme | ---------- 1 row in set (0.00 sec) 应该很容易就看到它在Web应用程序里是如何运行的在验证用户登录的时候，DECODE()会用网站专用的密钥解开保存在数据库里的密码，并和用户输入的内容进行对比。假设您把PHP用作自己的脚本语言，那么可以像下面这样进行查询：$query = "SELECT COUNT(*) FROM users WHERE username=$inputUser AND DECODE(password, abracadabra) = $inputPass".?> 提示：虽然ENCODE()和DECODE()这两个函数能够满足大多数的要求，但是有的时候您希望使用强度更高的加密手段。在这种情况下，您可以使用AES_ENCRYPT()和AES_DECRYPT()函数，它们的工作方式是相同的，但是加密强度更高。单向加密 单向加密与双向加密不同，一旦数据被加密就没有办法颠倒这一过程。因此密码的验证包括对用户输入内容的重新加密，并将它与保存的密文进行比对，看是否匹配。一种简单的单向加密方式是MD5校验码。MySQL的MD5()函数会为您的数据创建一个“ 指纹 ”并将它保存起来，供验证测试使用。下面就是如何使用它的一个简单例子：mysql> INSERT INTO users (username, password) VALUES (joe, MD5(guessme)).Query OK, 1 row affected (0.00 sec)mysql> SELECT * FROM users WHERE username=joe. ---------- ---------- | username | password | ---------- ---------- | joe | 81a58e89df1f34c5487568e17327a219 | ----------

---------- 1 row in set (0.02 sec) 现在您可以测试用户输入的内容是否与已经保存的密码匹配，方法是取得用户输入密码的MD5校验码，并将它与已经保存的密码进行比对，就像下面这样：mysql> SELECT COUNT(*) FROM users WHERE username=joe AND password=MD5(guessme). ---------- | COUNT(*) | ---------- | 1 | ---------- 1 row in set (0.00 sec) 或者，您考虑一下使用ENCRYPT()函数，它使用系统底层的crypt()系统调用来完成加密。这个函数有两个参数：一个是要被加密的字符串，另一个是双（或者多）字符的“ salt ”。它然后会用salt加密字符串；这个salt然后可以被用来再次加密用户输入的内容，并将它与先前加密的字符串进行比对。下面一个例子说明了如何使用它：mysql> INSERT INTO users (username, password) VALUES (joe, ENCRYPT(guessme, ab)).Query OK, 1 row affected (0.00 sec)mysql> SELECT * FROM users WHERE username=joe. ---------- ---------- | username | password | ---------- ---------- | joe | ab/G8gtZdMwak | ---------- ---------- 1 row in set (0.00 sec)结果是mysql> SELECT COUNT(*) FROM users WHERE username=joe AND password=ENCRYPT(guessme, ab). ---------- | COUNT(*) | ---------- | 1 | ---------- 1 row in set (0.00 sec) 提示：ENCRYPT()只能用在*NIX系统上，因为它需要用到底层的crypt()库。 幸运的是，上面的例子说明了能够如何利用MySQL对您的数据进行单向和双向的加密，并告诉了您一些关于如何保护数据库和其他敏感数据库信息安全的理念。 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com