

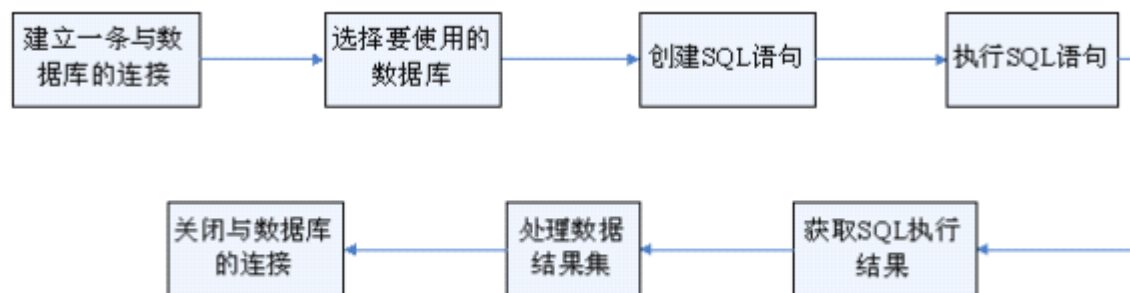
第13章 使用PHP操作MySQL数据库

- 前一章介绍了MySQL的基本用法和基础知识，其中各种管理操作和用法，都是通过直接操作MySQL来完成。这一章，将介绍如何通过PHP完成对MySQL数据库的操作。使用PHP操作MySQL数据库，是PHP进行Web编程的必然要求，同时也是本书的重点内容之一。本章将介绍操作MySQL的PHP函数，以及这些函数在使用PHP进行Web开发的具体用法。



13.1 操作MySQL的PHP函数

- PHP提供了完整的操作MySQL数据库的函数，这些函数覆盖从连接数据库、执行SQL语句、处理数据结果集到关闭数据库的方方面面。PHP通过这些函数，使基于MySQL数据库的Web开发高效而简单。通常，



PHP 访问 MySQL 数据库的步骤



13.1.1 连接和关闭数据库的函数

- 在用PHP执行具体的MySQL操作之前，首先要让PHP程序连接到一个MySQL数据库服务器。这一步骤类似使用提示符登录到MySQL。PHP提供函数mysql_connect() 打开一个到MySQL服务器的连接，该函数的语法如下所示。
- `mysql_connect(string $server, string $user_name, string $password, [bool $new_link, int $client_type]);`
- 函数mysql_connect() 有5个参数：
- 第一个参数\$server代表要连接的MySQL数据库服务器，可以给这个参数附加端口号。
- 第2个参数\$user_name是连接MySQL服务器的用户名。
- 第3个参数\$password就是该用户所使用的密码。
- 第4个参数\$new_link是可选的，它表示是否在下一次使用同样的参数调用mysql_connect()时，建立一个与MySQL的新的连接，该参数是BOOL型变量。
- 第5个参数\$client_type表示客户端连接MySQL服务器的类型，这个参数在实际应用中使用的比较少，这里不做详细介绍。



13.1.2 执行SQL语句的函数

- 函数mysql_query()在当前已建立连接的数据库中，执行一条SQL语句，该函数的语法如下所示。
- mysql_query(string \$sql);
- 参数\$sql是一个SQL语句字符串，该语句可以不以分号结尾。该函数执行成功，会返回一个资源表示符，失败时返回FALSE。
- 说明：函数mysql_query()仅对SHOW、SELECT、DESCRIBE等语句返回一个资源标识，对于其他SQL语句，mysql_query()在执行成功时返回TRUE，出错时返回FALSE。如果查询一个当前连接用户没有权限管理的表，mysql_query()也会返回FALSE。



13.1.3 处理查询结果集的函数

- 在PHP程序具体应用中，当一个 SQL语句执行完毕后，通常需要对查询的结果集做处理，以满足 Web应用的需要。处理数据结果集的PHP函数主要有以下几个。
- 函数mysql_affected_rows ()，取得前一次 MySQL 操作所影响的记录行数。如果函数执行成功，返回上一次 SQL语句执行所影响的行数，否则返回 -1。
- 函数mysql_fetch_row(\$result)，参数\$result是执行mysql_query ()之后返回的资源标识符。该函数从查询结果集中返回一行数据。该函数返回值是一个数组，其中每个元素对应一行结果记录的字段值。依次调用该函数可以返回结果集中的下一行，如果没有更多行，函数返回 FALSE。



13.1.4 获取字段信息的函数

- 函数mysql_fetch_field可以取得结果集的字段信息，并将其作为对象返回。该函数的语法如下所示。
- `object mysql_fetch_field ($result [, int field_offset]);`
- 参数\$result是执行mysql_query()之后返回的资源标识符。可选参数\$field_offset用来指定字段偏移量，如果没有指定该参数，函数mysql_fetch_field()将获取下一个取得的字段。函数mysql_fetch_field()返回一个对象，其属性存储了字段信息，该对象的属性如下所示。
- name，字段名。
- table，该字段所在的表名。
- max_length，该字段最大长度。
- not_null，如果该属性值为1，表示该字段不能为NULL。
- primary_key，如果该属性值为1，表示字段列是primarykey。



13.1.4 获取字段信息的函数

- `unique_key`，如果该属性值为1，表示该字段是 `uniquekey`。
- `multiple_key`，如果该属性值为1，表示该字段是 `non-uniquekey`。
- `numeric`，如果该属性值为1，表示该字段是 `numeric`。
- `blob`，如果该属性值为1，表示该字段是 `BLOB`。
- `type`，该字段的类型。
- `unsigned`，如果该属性值为1，表示该字段是无符号数。
- `zerofill`，如果该属性值为1，表示该字段填充0。



13.1.5 数据库操作的其他函数

- 这小节介绍几个其他的操作数据库函数，它们是以下几个函数。
- 函数mysql_select_db(\$database)，连接上MySQL数据库之后，可以使用该函数选择一个数据库。参数 \$database表示要选择的数据库名称。
- 函数mysql_num_rows(\$result)，用来取得结果集的行数目，即结果集中的记录数。参数 \$result是执行mysql_query()之后返回的资源标识符。不过，该函数仅对SELECT语句有效，要取得INSERT、UPDATE或DELETE语句执行后所影响的行的数据，需要使用函数mysql_affected_rows()。
- 函数mysql_error()，返回最近一次MySQL操作产生的错误文本信息。如果没有出错，函数返回空字符串。



13.2 使用PHP连接MySQL数据库

- 上一节介绍了PHP操作MySQL的主要函数，这节通过具体实例，来了解这些函数在PHP程序中的具体用法。



13.2.1 建立和数据库的连接

- 在PHP程序中操作数据库，首先要连接数据库。
- 这段代码使用函数mysql_connect()尝试连接本地的MySQL数据库，函数mysql_connect()中参数\$host的值为localhost，表示建立一个到本地的MySQL数据库连接。.php中，函数mysql_connect()连接MySQL服务器使用的用户名是root，密码是admin，读者可能需要根据自己的实际设置改变这两个参数（以下同），以便程序可以正确连接到MySQL数据库。接着，程序判断函数mysql_connect()的返回值，如果返回值为FALSE，程序会提示一个“数据库连接失败的”信息，同时使用函数mysql_error()将具体的错误信息输出到Web页面。这里使用了语言结构die()，它的功能类似于exit，输出一段信息并立即中断程序的执行。
- 如果函数mysql_connect()连接本地MySQL数据库服务器成功，代码会向页面输出一个“数据库连接成功”的提示信息。如果本地MySQL已启动，并且程序执行正常，



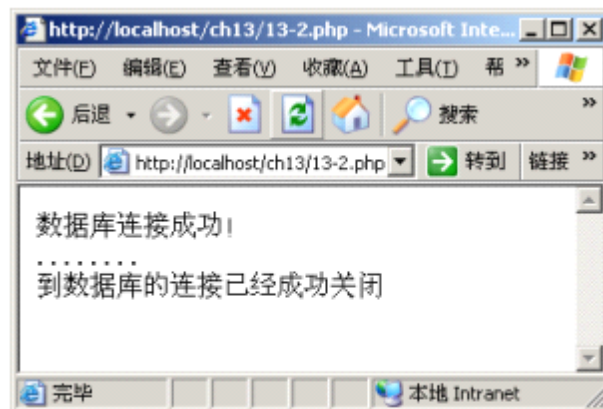
13.2.2 持久连接和临时连接

- 除mysql_connect () 之外，PHP提供了另外一个函数，也可以打开到MySQL服务器的连接，这个函数是mysql_pconnect ()。该函数的语法和函数mysql_connect () 类似，它返回的是到MySQL服务器的持久连接，如果出错会返回FALSE。该函数和mysql_connect () 非常相似，但也有两个主要区别。首先，mysql_pconnect () 会查找一个在同一主机上使用相同用户名和密码已经建立的连接，如果找到，该函数就返回此连接而不是创建一个新连接。另外一个区别是，当PHP程序执行完成后，使用mysql_pconnect () 建立的MySQL连接不会被关闭，mysql_close () 也不会关闭由mysql_pconnect () 建立的连接。区别由函数mysql_pconnect () 建立的连接，可以把由函数mysql_connect () 建立的连接称作临时连接。



13.2.3 关闭连接

- 在PHP程序中，关闭数据库连接可以使用函数 `mysql_close()`。事实上，当PHP脚本程序执行结束之后，会自动关闭到MySQL的连接，但还是推荐在程序中明确调用 `mysql_close()` 来关闭数据库连接。



关闭到 MySQL 服务器的连接



13.3 使用PHP显示MySQL数据库的数据

- 在PHP建立了到MySQL的连接之后，就可以执行SQL语句来查询数据库中的数据。然后，通过PHP函数处理查询后的结果集，以便后续程序使用，或通过整理，将这些数据显示到Web页面上。本节通过示例代码，向读者介绍如何使用PHP函数完成从数据库获取数据、处理数据以及向Web页面显示数据。



13.3.1 在PHP执行SQL语句

- 要从数据库获取数据，首先PHP要执行一条对表操作的SQL语句，包括SELECT、INSERT、UPDATE或DELETE语句。一般情况下，在PHP中执行SELECT语句，会从表中查找出一些记录行。而执行其他语句，只会返回语句是否执行成功的信息。
- 通过前面的介绍，读者已经知道，在PHP中调用函数mysql_query()将会通知MySQL执行一条SQL语句。



使用 mysql_error() 获取错误信息。



13.3.2 用PHP处理数据结果集

- 当一个SQL语句在程序中成功执行之后，可以使用 `mysql_fetch_array()` 来获取具体的查询结果，即使用该函数获取记录的字段值。



显示由 `mysql_fetch_array()` 返回的结果集



13.3.3 向Web页面显示数据

- 上一小节已经将从程序中找到数据显示到Web页面，但只是输出了数组结构，并不符合实际应用的需求。这小节将学习，如何通过把PHP代码嵌入HTML文档中，实现一个比较美观的查询结果页面。



The screenshot shows a Microsoft Internet Explorer window titled "13-7.php - Microsoft Internet Explorer". The address bar displays "http://localhost/ch13/13-7.php". The main content area displays a table with the following data:

用户ID	用户名称	来自城市	注册时间
1	Charles	NewYork	2007-03-25 12:50:00
2	David	London	2007-03-25 21:40:37
3	Jenny	Beijing	2007-03-26 08:37:06
4	Jack	NewYork	2007-03-26 11:13:12
5	Mary	London	2007-03-26 15:22:57
6	Lily Cameron	NewYork	2007-03-26 19:01:21

把数据库数据显示到 Web 页面



13.3.4 数据分页显示的原理及实现

- 分页显示是Web编程中最频繁处理的环节之一。所谓分页显示，就是通过程序将结果集一段一段地来显示。实现分页显示，需要两个初始参数：每页显示多少记录和当前是第几页。再加上完整的结果集，就可以实现数据的分段显示。至于其他功能，比如上一页、下一页等均可以根据以上信息加以处理得到。
- 在12.3.9小节，讲述了使用LIMIT子句对查询结果做限定。比如要取得某表中的前10记录，可以使用如下SQL语句。
- `SELECT * FROM a_table LIMIT 0,10;`
- 如要查找第11条到第20条记录，使用的SQL语句如下所示。
- `SELECT * FROM a_table LIMIT 10,10;`
- 如要查找第21条到第30条记录，使用的SQL语句如下所示。
- `SELECT * FROM a_table LIMIT 20,10;`



13.4 用PHP维护MySQL数据库的数据

- 上一节介绍了如何使用PHP函数操作数据库的查询结果集，并将结果集显示在页面上。本节讲述通过Web页面获取数据，来向MySQL数据库添加或更新数据。



13.4.1 从页面获取数据并插入数据库

- 本小节介绍如何获取Web页面数据，并将获取的数据添加到数据库中。首先建立一个Web页面，供用户输入数据之用，



提交用户信息的页面



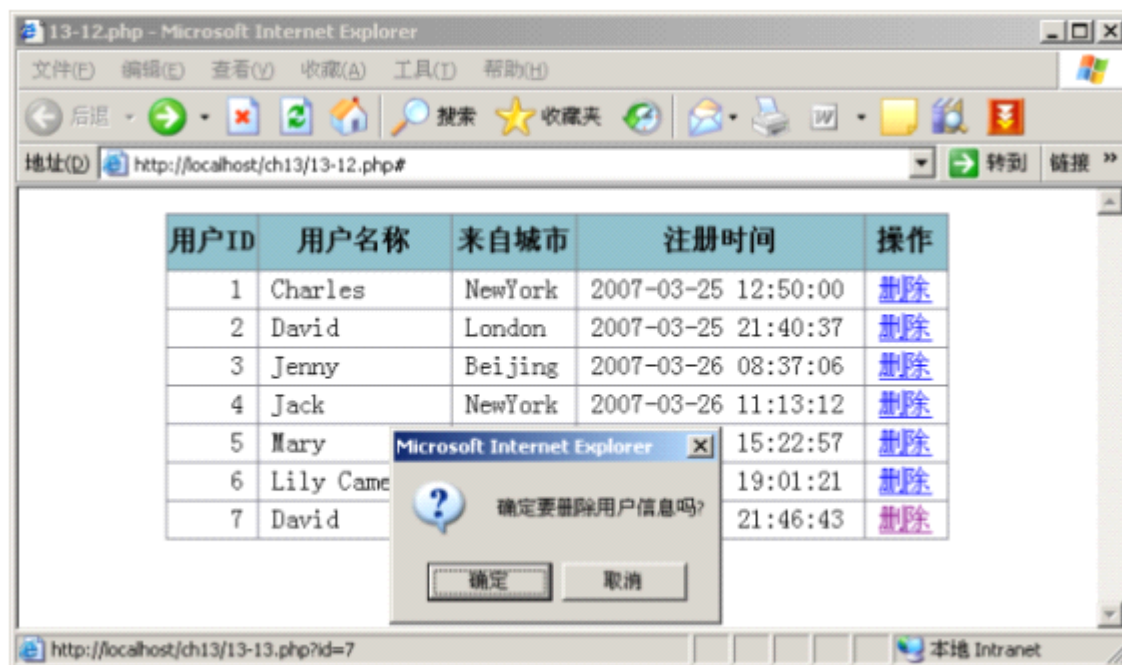
13.4.2 根据表单内容修改数据库数据

- 通过程序数据修改数据库数据，和通过程序向数据库插入数据很类似。从程序角度看，一个是通过PHP程序执行插入操作（INSERT语句），另外一个执行是更新操作（UPDATE语句）。
- 通过PHP程序实现更新操作，首先需要有一个显示数据的Web页面。另外，要根据不同的URL参数，该Web页面显示不同用户的信息，因此，该页面应该由内嵌在HTML文档中的PHP程序完成。然后，在这个页面修改用户数据，最后提交表单，由程序完成数据库数据的修改。



13.4.3 通过PHP删除数据库数据

- 前两小节介绍在程序中完成插入和更新数据的操作，这小节讲述通过PHP程序操作数据库的最后一个内容——删除数据库数据。



删除确认对话框



13.5 PHP操作MySQL常见错误信息及分析

- PHP操作数据库是使用PHP开发Web程序的基本部分，也是最重要的部分。几乎所有用PHP开发的Web程序或应用，都无一例外地需要操作数据库。因此，PHP程序中，对数据库操作部分的调试和错误排错，就显得非常重要。本节，将介绍几种在PHP程序操作MySQL时，比较常见的错误，以及对这些错误的分析，以期给读者在实际开发中有个参考。



13.6 小结

- 本章首先从语言级别上介绍了PHP操作MySQL的常用函数，这些函数如下所示。
- `mysql_connect()`：建立和MySQL数据库的连接
- `mysql_close()`：关闭MySQL数据库连接。
- `mysql_select_db()`：选择一个数据库。
- `mysql_query()`：执行一条SQL语句。
- `mysql_num_rows()`：获取结果集的行数目。
- `mysql_fetch_field()`：获取字段信息。
- `mysql_affected_rows()`：取得前一次 MySQL 操作所影响的记录行数。
- `mysql_fetch_row()`：该函数从查询结果集中返回一行数据。



13.6 小结

- `mysql_fetch_array()`：该函数从结果集中返回一行作为关联数组，或普通数组，或二者兼有。
- `mysql_fetch_assoc()`：该函数从结果集中返回一行作为关联数组。
- `mysql_error()`：返回最近一次MySQL操作产生的错误文本信息。
- 然后，每个函数都通过1至2个完整的示例代码，从连接、显示和维护三个方面演示这些函数的用法，其中还介绍了Web程序分页显示数据的原理及实现。最后，简要阐述了使用PHP操作MySQL时常见的错误及解决办法。

