

## 第12章 MySQL数据库和SQL基础

- 前几章介绍的主要内容是PHP语言本身，仅用PHP语言开发Web程序是远远不能满足实际需求的。Web程序中的各类数据需要靠数据库存储，只有在数据库配合的基础上，PHP才能发挥其最大功效。从本章开始，就向读者介绍数据库的有关知识。通常，使用PHP进行Web开发时，都是以MySQL作为数据库，二者可以完美地结合起来。本章通过介绍MySQL数据库管理系统，来完成对数据库基本概念及SQL基础知识的学习。同时，也会介绍一些MySQL操作方面的基本内容。



## 12.1 MySQL数据库简介

- 数据库是以结构化格式组织的数据的集合，这些数据集合是需要通过某种程序或软件创建和维护的，MySQL就是这些众多软件中的一员，SQLServer、Oracle和Sybase也是其中的成员。读者在这里需要明白一点，MySQL本身并不是数据库，它只是用来创建、维护和管理数据库的计算机软件，这些软件被称作数据库管理系统（DBMS）。通常情况下，技术人员就是通过数据库管理系统来维护数据库中的数据。
- MySQL是一款免费软件，它基于开放软件的理念，提供免费和低成本的数据解决方案。重要的是，它在性能、安全和稳定性方面，完全可以满足大多数Web开发的需要。MySQL的特点就是灵活而不失强大，和Oracle等DBMS相比，它本身并不庞大，但仍然集中了大量特性，使其快速高效。因此，对于中小规模的数据库需求来说，它已足够。
- MySQL是基于关系数据库的，当前，大多数数据库都是关系数据库。关系数据库是数据库类别中的一种，它将数据组织成表，并表现为表与表之间的关系。通过这种关系，数据库管理系统可以从不同的表中提取某种特定的数据集合。关于表的概念在下节会有更多的介绍。





## 12.2.1 表、字段、记录和键的概念

- 这一小节介绍关系数据库中的一些基本概念，主要涉及到以下内容。
- 表。
- 字段。
- 记录。
- 键。

Users

UserId	Name	Gender	RegTime
1	小昭	female	20070722
2	小乐	male	20070928
3	小白	male	20071001

表的实例——用户信息



## 12.2.2 登录和退出MySQL

- Windows系统中，如果在没有设置环境变量PATH的情况下登录MySQL，可以使用Windows命令提示符。在命令提示符界面下，进入MySQL安装目录的bin目录中，使用语法格式如下所示的命令。
- `mysql -u user_name -p password -h host`
- 其中参数说明如下：
- 选项参数-u用来指定用户名user\_name。
- 选项-p用来指定该用户名登录MySQL所使用的密码password。
- 选项-h用来指定所要连接的数据库服务器host。



## 12.2.3 建立和删除数据库

- 在第1章已经介绍过如何安装和启动MySQL数据库。在Windows下可以通过命令提示符进入MySQL系统，或者使用其他软件，如使用phpMyAdmin来管理MySQL（参见附录4）。本书以在Windows命令提示符下使用MySQL为例，介绍MySQL及其有关操作。
- 要使用MySQL，首先需要创建一个数据库。在MySQL中，使用命令SHOW DATABASES（可以全部使用小写字母）可以查看在当前MySQL系统中的所有数据库。如果是新安装的MySQL，使用这个命令后，应该看到如下所示的结果。



## 12.2.4 表的建立

- 在MySQL中，首先使用命令USE database\_name，可以转到要管理或查看信息的数据库，其中database\_name是数据库名称。然后使用SHOW TABLES命令查看某个数据库下的所有表。例如查看数据库mysql中的表，所用SQL语句及执行结果如下所示。

```
mysql> USE mysql
Database changed
mysql> SHOW TABLES;
+-----+
| Tables_in_mysql |
+-----+
| columns_priv     |
| db               |
| func             |
| help_category    |
| help_keyword     |
| help_relation    |
| help_topic       |
| host             |
| proc             |
| tables_priv      |
| user             |
+-----+
11 rows in set (0.00 sec)
```



## 12.2.5 建立索引

- 如果一个表的数据量很大，例如有上百万行的时候，MySQL查找某条（或某些）特定的记录时会花费很长时间。可以通过在表中建立索引，来减少MySQL查找记录所使用的时间。索引最有效的使用方法是，为那些要频繁查找的字段建立索引。MySQL使用下面的语法为一个表中的某些列（即字段）建立索引。
  - `CREATE INDEX index_name ON table_name (`
  - `column_1,`
  - `column_2,`
  - `...`
  - `)`







## 12.3.1 查看表结构

- 前面介绍了查看某个数据库中所有表的命令：SHOW TABLES。如果要查看某个表的结构，先使用USE命令，转入到某个数据库，然后使用DESCRIBE命令，示例语句如下所示。

```
mysql> DESCRIBE users;
```

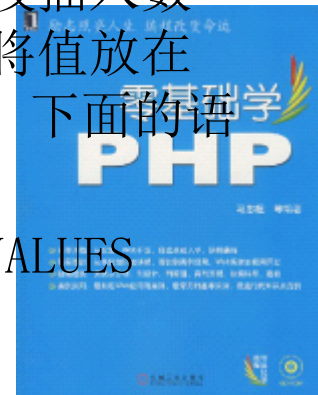
Field	Type	Null	Key	Default	Extra
id	int(11)		PRI	0	
name	varchar(30)	YES		NULL	
created_time	datetime	YES		NULL	

```
3 rows in set (0.02 sec)
```



## 12.3.2 插入数据

- 一个表创建好后，就可以向其中存储数据。向表中插入一条数据（记录）是最基本的数据库操作之一。在 MySQL 中，使用 INSERT 语句向表中插入一条数据。其语法如下所示。
- `INSERT INTO table_name  
(column_1, column_2, ..., column_n)`
- `VALUES (value_1, value_2, ..., value_n);`
- 这个语法表示向表 `table_name` 中插入一条数据，并且指定字段 `column_1` 的值为 `value_1`，`column_2` 的值为 `value_2`，依次类推。这里要注意的是，如果是向数值型字段插入数据，可以不带单引号，但对于其他字段，就必须将值放在单引号中。以第 12.2.5 小节创建的表 `users` 为例，下面的语句向表 `users` 插入一条记录。
- `INSERT INTO users (id, name, created_time) VALUES  
(1, 'Charles', '2007-03-25 12:50:00');`



### 12.3.3 查询数据

- 查询数据，即浏览数据库中的数据，可以使用SELECT语句。SELECT语句是数据库操作中最常用的语句，因为数据库大部分时间提供的是数据读取服务。SELECT语句的语法如下所示。
- `SELECT column_1, column_2, ..., column_n  
FROM table_name [WHERE clause];`
- 该语句查询表table\_name中某些字段的值组成的一条或多条记录，其中WHERE子句是可选的，它表示按某种条件查询表中特定的记录，这个子句的用法将在12.3.4小节介绍。下面是通过SELECT语句从表users中查找所有数据的示例。



## 12.3.4 条件查询

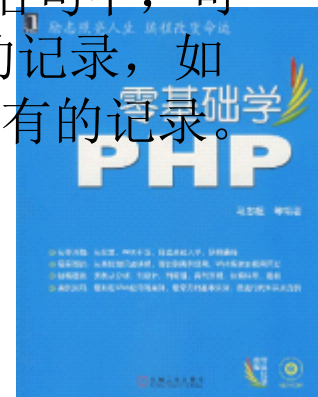
- 在实际应用中，通常不需要查找表中的所有记录，而是查找满足某些条件的特定记录。在SQL中，可以在SELECT语句中使用WHERE子句指定查询条件，从表中查找出特定的行。例如，下面的SQL语句找出表users中用户名叫Jenny的用户信息。

```
mysql> SELECT * FROM users WHERE name='Jenny';
+----+-----+-----+
| id | name  | created_time |
+----+-----+-----+
| 3  | Jenny | 2007-03-26 08:37:06 |
+----+-----+-----+
1 row in set (0.03 sec)
```



## 12.3.5 更新数据

- 数据更新是指对数据库中的某些记录做修改，更改记录字段的值。在MySQL中使用UPDATE语句完成对表中数据的修改，UPDATE语句的用法如下所示。
- `UPDATE table_name SET  
column_1=value_1,column_2=value_  
2, ..., column_n=value_n [WHERE clause];`
- `table_name`是指要做更新操作的表名，在UPDATE语句中，使用SET子句制定要更新的字段及其值。UPDATE语句中，可以使用WHERE子句中的限定条件，来指定要修改的记录，如果没有使用WHERE子句，那么MySQL将修改表中所有的记录。下面是使用UPDATE更新表users的SQL语句示例。



## 12.3.6 删除数据

- 使用DELETE语句删除表中的数据，其用法如下所示。
- DELETE FROM table\_name [WHERE clause];
- 这句SQL表示从表table\_name中删除数据，WHERE子句表示删除满足限定条件的记录，WHERE语句是可选的。
- 删除数据是比较简单的操作，但同时是非常危险的操作。在MySQL的DELETE语句中，如果没有使用WHERE子句，那么表中的数据会在没有任何提示确认的情况全部被删除！因此这个操作是相当危险的，尤其是在使用root用户登录MySQL并且表的数据没有备份的情况下，所以删除操作一定要谨慎执行。



## 12.3.7 对查询结果做排序

- MySQL中，执行SELECT语句所生成结果的默认顺序，由其出现的顺序决定。SQL支持将查询结果按特定的顺序排列。使用ORDER BY子句完成对查询结果的排序，排序有降序和升序两种方式，ORDER BY子句中的关键字ASC表示升序排序，DESC表示降序排序方式。下面是使用ORDER BY子句的对查询结果做排序的示例。

```
mysql> SELECT * FROM users WHERE created_time>'2007-03-26 00:00:00'  
-> ORDER BY id DESC;
```

```
+-----+-----+-----+  
| id | name          | created_time          |  
+-----+-----+-----+  
| 6 | Lily Cameron | 2007-03-26 19:01:21 |  
| 5 | Mary         | 2007-03-26 15:22:57 |  
| 4 | Jack         | 2007-03-26 11:13:12 |  
| 3 | Jenny        | 2007-03-26 08:37:06 |  
+-----+-----+-----+  
4 rows in set (0.00 sec)
```



## 12.3.8 对查询结果做分组

- MySQL支持对查询结果的分组，以汇总相关数据。假设表users中有字段city，用来表示各用户所在的城市，此时，查询表users的结果如下所示。

```
mysql> SELECT city FROM users GROUP BY city;
```

```
+-----+
```

```
| city |
```

```
+-----+
```

```
| Beijing |
```

```
| London |
```

```
| NewYork |
```

```
+-----+
```

```
3 rows in set (0.00 sec)
```



## 12.3.9 对查询结果做限定

- 除了WHERE子句，MySQL还提供了另外两个SQL通用方法来支持查询条件，他们是LIMIT和HAVING。这两个子句主要是为已有的查询结果设置限定条件，从已有的数据结果中，根据LIMIT或HAVING子句指定的条件，进一步选出所需数据。
- LIMIT用来限定查询返回的记录条数。如在LIMIT语句的最后加上LIMIT 3，表示返回查询结果中的最前3行记录。而LIMIT 0, 10，表示从查询结果中的第1行开始，返回10行记录。下面是使用LIMIT子句的SQL语句及其查询结果。



## 12.3.10 MySQL的数据类型

- 在创建一个表时，要指定字段的数据类型，MySQL根据字段的数据类型来决定如何存储数据。本小节将介绍MySQL常用的数据类型。

- MySQL常用数据类型

数据类型	说明
INT	整数值。MySQL支持有符号或无符号整数值，用UNSIGNED INT指定无符号整数值。
DECIMAL	指定数字值的精度和范围，如DECIMAL(5,2)。
REAL	浮点数值。
CHAR	定长字符类型。定义为CHAR类型的字段不能保存比其指定长度更长的字符串，多于部分将不会存入表中，短于指定长度的字符串，会用空白补足指定的长度后存入表中。
VARCHAR	变长字符串，指定长度范围在1-255之间。
TEXT	文本类型，存放更长的文本型数据。
DATE	日期值。
TIME	时间值，存放的是与日期无关的一天中的任何时间。
DATETIME	MySQL所支持的，用于存放日期和时间的类型。



### 12.3.11 MySQL运算符

- 在前面讲述WHERE子句的小节中，已经使用了验证两个值是否相等的运算符=。在MySQL中常用的运算符是比较运算符和逻辑运算符。使用逻辑运算符，可以使WHERE语句指定的条件更加灵活实用。同时，可以使用括号改变MySQL运算符的优先顺序，从而建立起比较复杂的条件查询。

- MySQL运算符

运算符	说明
=	比较运算符，比较两个值是否相等。
!=或<>	比较运算符，比较两个值是否不相等。
<	比较运算符，表示小于。
>	比较运算符，表示大于。
<=	比较运算符，表示小于或等于。
>=	比较运算符，表示大于或等于。
+	算术运算符，求和。
-	算术运算符，求差。
*	算术运算符，求积。
/	算术运算符，求商。
AND	逻辑运算符，与。
OR	逻辑运算符，或。
NOT	逻辑运算符，非。



## 12.3.12 MySQL中的模式匹配

- SQL中的模式匹配是指根据一个不完整的字符串，查找字段的值，例如使用' cand%'，可以匹配candy、candle等。在MySQL中使用LIKE运算符完成模式匹配。下列SQL演示了从users表中查找名字以字母J开头的用户。

```
mysql> SELECT * FROM users WHERE name LIKE 'J%';
+----+-----+-----+-----+
| id | name  | created_time | city |
+----+-----+-----+-----+
| 3  | Jenny | 2007-03-26 08:37:06 | Beijing |
| 4  | Jack  | 2007-03-26 11:13:12 | NewYork |
+----+-----+-----+-----+
2 rows in set (0.01 sec)
```



## 13.3.13 使用MySQL函数

- MySQL支持SQL常用的函数，如集合函数、字符串处理函数和日期函数等。本小节将介绍一些编程过程中经常遇到的MySQL函数，它们包括以下所示函数。
- 函数COUNT，用来统计表中记录个数或这列中值的个数。
- 函数MAX，返回某列中的最大值。
- 函数MIN，返回某列中最小值。
- 函数SUM，将指定列的值求和。

```
mysql> SELECT COUNT(*) FROM users;  
+-----+  
| COUNT(*) |  
+-----+  
|      6 |  
+-----+  
1 row in set (0.01 sec)
```



### 13.3.13 使用MySQL函数

- 函数AVG，计算指定列的平均值。
- 函数LENGTH，计算字符串的长度。
- 函数SUBSTRING，截取子字符串。
- 函数YEAR，返回指定日期的年份，范围在1000到9999之间。类似的还有MONTH、HOUR等函数。
- 函数UNIX\_TIMESTAMP，返回一个UNIX时间戳。
- 函数DATE\_FORMAT，将一个日期格式化。
- 函数NOW，返回MySQL服务器系统的当前日期和时间。



## 12.4 小结

- 本章介绍了什么是数据库，以及和数据库相关的各个概念，如表、字段、记录和索引的概念。本章通过以 MySQL 为对象，介绍了在 MySQL 中进行各类管理的操作，它们包括如下内容。
- 在 Windows 下登录和退出 MySQL。
- 数据库的建立与删除。
- 表的建立和索引字段的建立。
- 本章还通过 MySQL，介绍了 SQL 基础语法及其在 MySQL 系统中的具体应用，包括以下内容。
- 向表中插入数据。
- 查询数据。
- 更新、删除数据。
- 对查询结果做排序、分组。
- 对查询结果做限定。





## 12.4 小结

- 此外还介绍了MySQL的其他基础知识，包括以下内容。
- MySQL的数据类型。
- MySQL的运算符。
- MySQL的模式匹配
- 在MySQL中使用函数。
- 事实上，SQL语言本身和MySQL的内容都是相当丰富的，仅用一章内容不可能做到面面俱到。本章只是介绍了MySQL最基本的一些用法，以及涉及到PHP编程方面的某些内容，这些内容都相当基础，它们只是冰山一角。更丰富的内容，还需读者通过专门的资料继续探索和学习。

