php

MYSQL数据库相关

一、MYSQL数据库简介

相关术语

1.什么是数据库?

数据库（database) 就是一个由一批数据构成的有序集合，这个集合通常被保存为一个或多个彼此相关的文件。

2.什么是关系型数据库?

数据被分门别类的存放在一些结构化的数据表(table)中，而数据表之间又往往会形成种种内在的交叉关系。存在于数据表之间的这种关系(relation)使数据库又被称为关系型数据库;

3.关系型数据库系统

MySQL、Oracle、Microsoft SQL Server 和IBM DB2都是关系型数据库系统(database system)。除了管理数据，一个这样的系统还包括用来管理各种关系数据库的程序。一个合格的关系数据库系统不仅要确保各种数据的存储情况安全可靠，还必须能够处理对现有数据进行查询、分析和排序以及对新数据进行保存等诸多命令。

4.数据表、记录、字段、查询、SQL、索引

数据表(table)即用来实际存放有关数据的框架结构。

这种数据表里的每一行被称为一条数据记录(data record),简称”**记录**”,每条记录的结构和格式是由人们在定义该数据表时决定的。例如，在某个用户表里，每条记录可能包含着用户的姓名，出生日期，注册时间等多个**字段**（field)。每个字段对自己所能存储的信息类型又有着一定的要求（例如，它必须是一个有着某种特定格式的数字或者是一个字符个数不得超过某个预定义最大值的字符串）。

查询 是人们用各种SQL指令构造出来的，SQL指令负责具体完成筛选和提取结果数据的工作

SQL （Structured Query Lanuage) 结构化查询语言;这种语言已发展为人们在构造数据库查询命令的一个标准。

二、SQL语言

SQL语言简介

SQL (structured Query Language) 结构化查询语言;

主要用途是构造各种数据库系统操作指令，如 SELECT、INSERT、UPDATE、DELETE;

SQL命令可以分类以下三大类别:

DML(Data Manipulation Language 数据处理语言）：这类命令主要包括 SELECT、INSERT、UPDATE、DELETE等用来从数据表读出数据，把数据存入数据表或是对数据表里的现有记录进行修改的命令;

DDL(Data Definition Language 数据定义语言）：这类命令主要包括CREATE TABLE、ALTER TABLE 等用来定义和改变数据库结构的命令;

DCL(Data Control Language 数据控制语言）：这类命令主要包括GRANT、REVOKE以及另外几个用来帮助人们设置和调整MySQL访问控制机制的SQL命令;

查询(SELECT)

1.简单查询

SELECT * FROM tablename

2.限制查询结果中的数据列个数

SELECT column1,column2 FROM tablename

3.确定数据表里有多少条数据记录

SELECT COUNT(id) FROM tablename

WHERE子句

WHERE子句设置查询条件，过滤掉不需要的数据行。

例如查询年龄大于20的记录： SELECT * FROM usertable WHERE age>20

WHERE子句可包括各种条件运算符：

1.比较运算符

运算符	含义
=	等于
>	大于
>=	大于等于
<	小于
<=	小于等于
<>	不等于

例：SELECT * FROM user WHERE uid = 10

SELECT * FROM user WHERE uid <= 10

2.逻辑运算符

运算符	含义
AND	如果组合的条件都是TRUE,返回TRUE
OR	如果组合的条件其一是TRUE,返回TRUE
NOT	如果条件是FALSE,返回TRUE

3.范围运算符 (BETWEEN...AND...)

判断表达式值是否在指定的范围

例： SELECT * FROM products WHERE price BETWEEN 1000 AND 2000

等同于： price >= 1000 AND price <= 2000

4.列表运算符 (IN)

判断表达式是否为列表中的指定项

例：SELECT * FROM user WHERE uid IN (1, 3, 5)

SELECT * FROM user WHERE uid NOT IN (1, 3, 5)

5.模式匹配符 (LIKE)

模式匹配符常用于模糊查找，它判断列值是否与指定的字符串格式相匹配。可用于char、varchar、text等类型查询。

可使用以下通配字符：

百分号%：可匹配任意类型和长度的字符。

下划线_：匹配单个任意字符，它常用来限制表达式的字符长度。

例如： 查找以Publishing结尾，使用LIKE '%Publishing'

查找第二字母是 A的 ， 使用 LIKE '_A%'

关联查询

通过连接运算符可以实现多个表查询。

连接是关系数据库模型的主要特点，也是它区别于其它类型数据库管理系统的一个标志。 在关系数据库管理系统中，表建立时各数据之间的关系不必确定，常把一个实体的所有信息存放在一个表中。当检索数据时，通过连接操作查询出存放在多个表中的不同实体的信息。连接操作给用户带来很大的灵活性，他们可以在任何时候增加新的数据类型。为不同实体创建新的表，然后通过连接进行查询。

例：SELECT empe_name, dept_name FROM empe, dept WHERE empe.dept_id = dept.id

SELECT empe_name, dept_name FROM empe LEFT JOIN dept ON empe.dept_id = dept.id

限制查询

LIMIT子句用于强制SELECT语句返回指定的记录数。

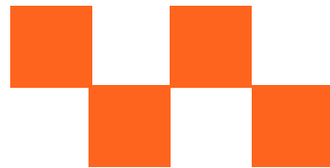
LIMIT接受一个或两个数字参数。参数必须是一个整数常量。如果给定两个参数，第一个参数指定第一个返回记录行的偏移量，第二个参数指定返回记录行的最大数目。注意：初始记录行的偏移量是0而不是1。

例：检索前5个记录行

```
SELECT * FROM table LIMIT 5;
```

检索记录行 6-15

```
SELECT * FROM table LIMIT 5,10;
```



查询结构排序

使用ORDER BY子句对查询返回的结果排序。

ORDER BY子句的语法格式为： ORDER BY {column_name [ASCIDESC]} [,...n]

其中ASC表示升序，为默认值，DESC为降序

例：

SELECT * FROM user ORDER BY uid ASC

SELECT * FROM user ORDER BY uid DESC

SELECT * FROM user ORDER BY regdate DESC,username ASC;

增删改

1.插入数据记录(INSERT)

例：INSERT INTO user (username,password) VALUES ('admin','123456')

一次插入多条记录

INSERT INTO user (username, password)
VALUES ('user1', '123456'), ('user2','123456'),

2.修改数据记录（UPDATE）

例：UPDATE user SET username = 'admin1', passwd ='12345678' WHERE uid = 10

3.删除数据记录（DELETE）

例：DELETE FROM user WHERE uid = 10

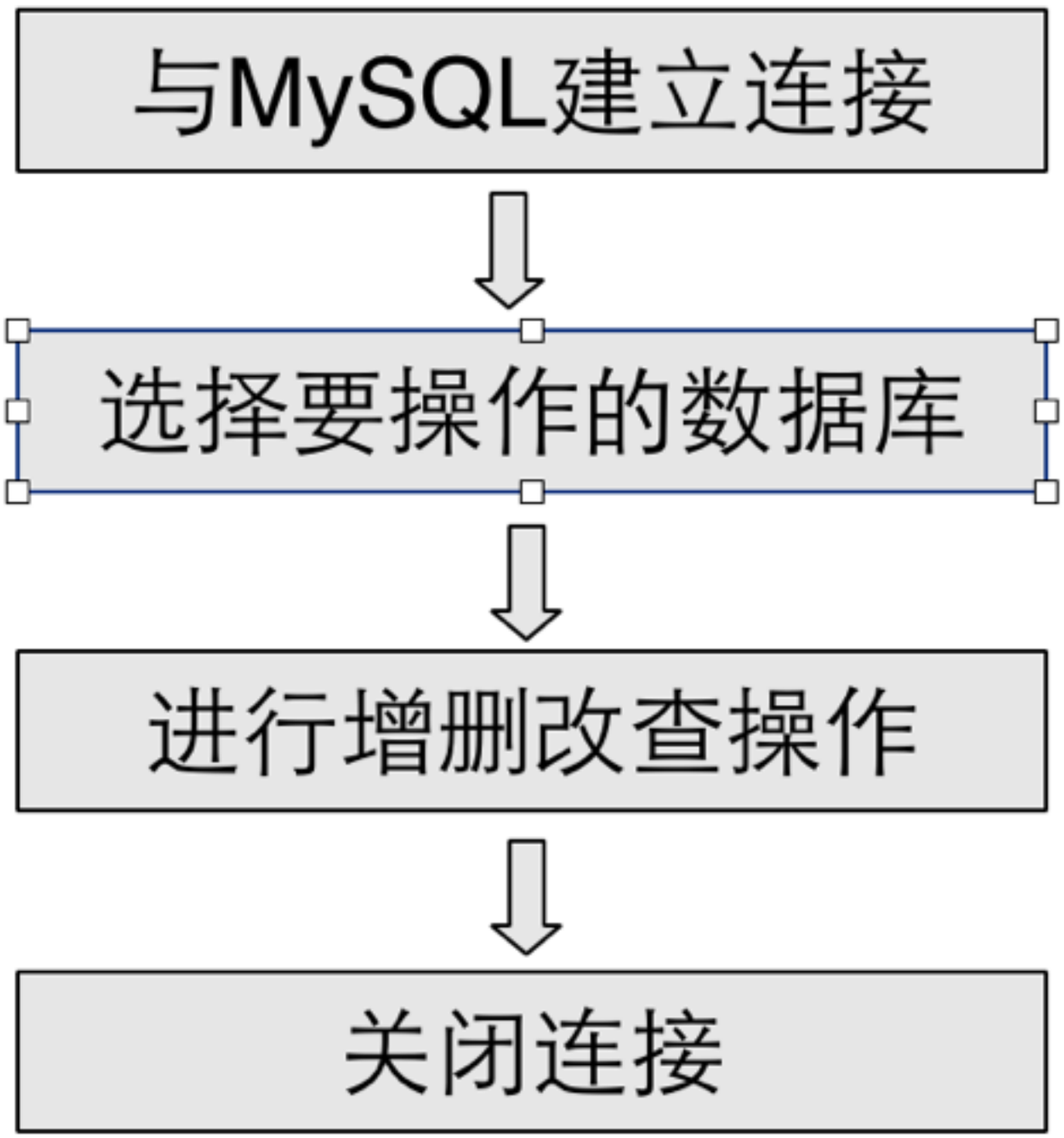
三、数据库操作

编程接口

PHP5开始，PHP向程序员提供了两种MySQL应用程序编程接口:一种是从PHP早期版本一直就有的mysql功能模块;另一种是从PHP5才开始有的mysqli接口;

mysql功能模块不是PHP的一个集成组件。要想使用这个功能扩展模块，PHP的Linux版本必须在编译时加上一个—with-mysql选项。PHP的windows版本通过一个DLL文件提供了相应的扩展,不管使用是哪一种操作系统，都必须在php.ini文件里启用这个扩展以确保PHP能够找到所有必要的DLL。

操作数据库的步骤



链接MYSQL数据库

通过mysql功能模块连接MySQL服务器的办法是调用mysql_connect()函数，它需要提供3项信息：MySQL服务器的主机名、MySQL用户名和密码。如果MySQL服务器与PHP运行在同一台计算机上，可以使用localhost作为它的主机名。

例：$conn = mysql_connect("localhost", "root", "123456");

关闭MYSQL数据库

查询MySQL服务器结束后，应当关闭连接。不过关闭连接不是必需的，因为PHP的垃圾回收机制会处理这个问题。mysql_close()函数关闭可选参数link_id对应的连接。如果没有指定link_id,则认为是最近打开的连接。

例：mysql_close();

选择MYSQL数据库

与MySQL服务器建立连接后，就可以使用各种mysql_xxx()函数去执行SQL命令。但为了避免每次调用mysql_xxx()函数都要指定目标数据库，最好先用mysql_select_db()函数（它相当于SQL命令USE databasename)为后续操作选定一个默认数据库。

例：mysql_select_db("mycompany");

执行SQL命令

为了执行SQL命令，需要把它们作为一个字符串传递给mysql_query()函数。如果想访问的不是当前数据库，就需要调用mysql_db_query()函数来添加SQL命令并明确给出那个数据库名称，这两个函数的最后一个参数（连接的ID号码，即mysql_connect()的返回值）都是可选的，只有与MySQL服务器建立了多个连接的时候才需要给出这个参数。

例：$result = @mysql_query("SELECT * FROM user");

// 自 PHP 4.0.6 起不提倡使用此函数。不要用此函数

$result = @mysql_db_query("mycompany", "SELECT * FROM product");

如果SQL命令执行成功，mysql_query()函数将返回PHP资源的引用指针(一个Resource id #2格式的字符串）；否则将返回FALSE,并生成一条出错消息;

mysql_query()函数可以用来执行任何一种SQL命令，比如 SELECT(查询)、INSERT(插入新记录）、UPDATE(修改现有记录）、DELETE(删除现有记录）、CREATE TABLE（创建新数据表）、ALTER TABLE(修改数据表结构）等。

获取显示数据

1. mysql_fetch_row()

mysql_fetch_row()函数将以一个普通数组的形式返回一条结果，它的各个字段需要以$row[n]的方式进行访问。

2. mysql_fetch_array()

mysql_fetch_array()函数将以一个关联数组的形式返回一条结果，它的各个字段需要以$row[n]或$row["colname"]的方式进行访问。

3. mysql_fetch_assoc()

mysql_fetch_assoc()函数也将以一个关联数组的形式返回一条结果记录，但它的各个字段只能以$row["colname"]的方式进行访问。

4. mysql_fetch_object()

mysql_fetch_object()函数以一个对象的形式返回一条结果记录，

它的各个字段需要以$row->colname的方式进行访问。

这4个函数的共同点是：每次调用将自动返回下一条结果记录,但如果已经到达结果数据表的末尾，则返回FALSE。

5.mysql_free_result()

PHP会把查询的结果一直保存到脚本执行结束。如果要提前释放某次查询结果（例如在某个脚本里已经进行了大量查询），可以用mysql_free_result()函数提前释放它。

6. mysql_num_rows()

mysql_num_rows()函数用于获取查询返回的记录数;

7. mysql_insert_id()

mysql_insert_id()函数用于获取INSERT 操作产生的 ID ;

8. mysql_affected_rows ()

mysql_affected_rows ()函数用于获取前一次 MySQL 操作所影响的记录数 ;

读取数据例子

例：读取数据

```
//连接MySQL
$conn = @mysql_connetc(“localhost”, “root”, “123456”);
if($conn == FALSE){
    echo “数据库连接失败!“;
    exit;
}
//选择数据库
mysql_select_db(“test”);
//执行SQL
$result = @mysql_query(“SELECT * FROM user”);
```

```
//显示查询结果
if(mysql_num_rows($result) > 0){
      While($row = mysql_fetch_array($result)){
       echo $row["username"];
       echo "<br />";
      }
}
//释放结果集
mysql_free_result($result);
//关闭连接
mysql_close();
```

插入数组例子

```
$conn = @mysql_connetc("localhost","root","123456")or die("数据库连接败!");
mysql_select_db("test");
$username = 'admin';
$password = '123456';
$reg_time = time();
$sql = "INSERT INTO user (username, password, reg_time) VALUES ('$username', '$password',
        $reg_time)";
$result = mysql_query($sql);
If($result) {
        echo "添加成功";
}
mysql_close();
```

更新数据例子

```
$conn = @mysql_connetc("localhost", "root", "123456") or die("数据库连接失败!");
mysql_select_db("test");
$newname = 'root';
$sql = "UPDATE user SET username='$newname' WHERE id = 1";
$result = mysql_query($sql);
if(mysql_affected_rows() > 0) {
    echo "编辑成功";
}
mysql_close();
```

删除数据

```
$conn = @mysql_connetc("localhost","root","123456") or die("数据库连接失败!");
mysql_select_db("test");

$sql = "DELETE FROM user WHERE id = 1";
$result = mysql_query($sql);
if(mysql_affected_rows() > 0) {
        echo “删除成功”;
}
mysql_close();
```

谢　谢