昨日回顾 2

数组 3

数组查找： 3

顺序查找 3

二分查找 4

数据库基础 4

数据库的分类： 4

关系数据库： 4

数据库存储的本质和表现： 5

关系数据库常见术语： 6

数据库系统的基本结构 7

配置phpmyadmin 7

数据库操作的基本模式（流程） 8

mysql系统级操作和基本语法规定 8

登录/退出基本操作 8

备份恢复数据库 9

基本语法规定 9

语句结束符 10

数据库中命名规则 10

php中操作数据库的基本代码和流程 10

基本流程： 10

数据定义语句 11

数据库定义 11

语法形式 11

修改数据库： 12

删除数据库： 12

其他数据库相关语句 12

昨日回顾

数组

数组遍历

遍历基本语法

foreach(数组$arr as 键变量$key => 值变量$value )

{

//这里就是循环体，可以使用$key, $value

//$key, $value的值会随着遍历的进行而一次次取得数组的元素的数据（键名和值）

//此遍历语法也是循环语句，同样可以break和continue

}

数组指针和遍历原理

一个数组内部具有一个指针（并非c语言中的指针），只是一个理念上的。

数组的指针也就是实现数组遍历的内部机制。

数组指针操作：

key(), current(), next(), prev(), reset(), end(), each()

数组遍历的流程：

开始（指针初始化）-〉判断指针位置〉有效〉取得该元素数据〉移动指针到下一个位置〉进入循环体执行。

for+next+reset遍历数组

$len = count($arr);

reset($ar); //指针初始化

for($i = 0; $i< $len; ++$i)

{

$key = key($arr);

$value = current($arr);

//这里开始做事情。。。。

next($arr);

}

while+each()+list()遍历数组

reset($arr）；

while( list($key, $value) = each( $arr ) ){

//这里开始做事情。。。。（循环体）

}

foreach 遍历细节探讨

1，是标准循环语句，可以中断；

2，正常循环时，值变量使用值传递方式获取数据；

3，可以对值变量加上引用符号“&”来实现引用传值；

4，正常遍历循环时，默认是在原数组上进行遍历，但如果在遍历过程中试图改变数组（改变一项值或增减数组项）或对数组进行某种指针操作（reset, rurrent, next, prev, key, end )，则此时foreach内部会先将该数组进行“拷贝”一份，原来的数组就会去实现该改变，而新拷贝出来的数组仍然按“既定规划”进行遍历循环（此时认为遍历的是新的那份数组，其实就是未改变之前的那个）

5，如果遍历中值变量使用的是引用传值，则如论如何都是在原数组上进行。

数组排序思想介绍

冒泡排序

选择排序

数组

数组查找：

就是从一个数组中找一个元素的数据（可能是找下标，也可以是找数据值）

数组的查找通常有2种需求：

1：判断要找的数据是否存在。

2：找出要找的数据的位置（下标）

顺序查找

从一个数组中按顺序找出一个元素（下标或值）

需求1：判断要找的数据是否存在

$v1 = 10;

function search1( $arr, $v1){

foreach($arr as $value ){

if( $c1 == $value ){

return true;

}

}

return false;

}

需求1：找出要找的数据的位置（下标）

$v1 = 10;

function search2( $arr, $v1){

foreach($arr as $key => $value ){

if( $c1 == $value ){

return $key; //找到，返回位置（下标）

}

}

return false;

}

//特别注意以下写法：

if （ ($m = search2( $arr, 10)) === false){

echo “没找到。”

}

else{

echo “找到了，位置为：$m”

}

二分查找

二分查找的前提：

1，针对一个已经进行了排序的数组（即里面的数据已经是有序了）

2，是连续的索引数组，比如下标为：0, 1, 2, 3, ......

比如：

$arr2 = array( 3, 4, 5, 15, 19, 21, 25, 28, 30, 30, 33, 38, 44, 51, 52, 55, 60, 77, 80, 82, 83);

数据库基础

数据库的分类：

* 层次数据库：类似xml（html）的文本结构。
* 网状数据库：
* 关系数据库：

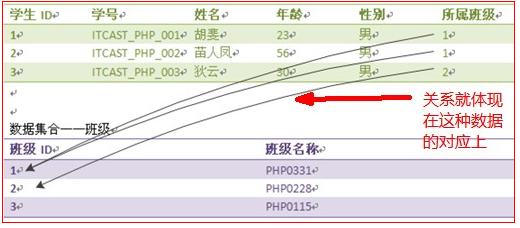
主流数据库产品：



关系数据库：

ＳＱＬ：　Ｓｔｒｕｃｔ　Ｑｕｅｒｙ　　Ｌａｎｇｕａｇｅ（结构化查询语言）　，其实是一个“国际化”标准。

各种关系数据库，其实都是实现（并扩充）该标准的一种产品。

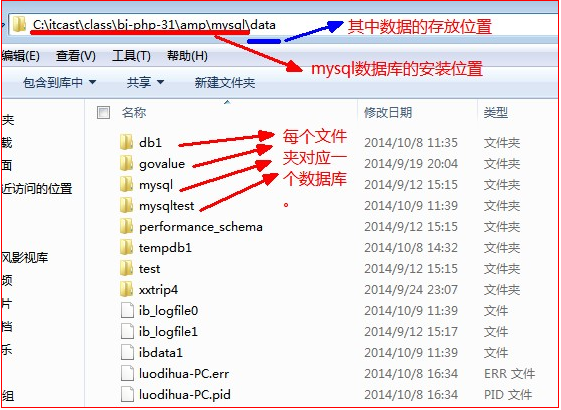


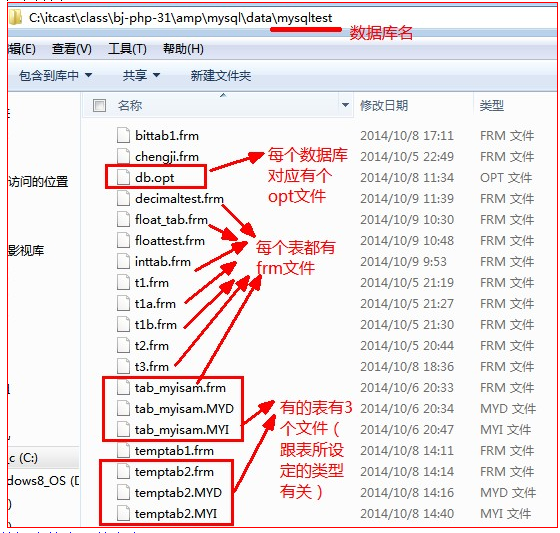
数据库存储的本质和表现：

本质就是将大量的“数据值”通过数据库管理系统进行某种复杂但是逻辑结构上清晰有序的存储

表现：

其实也是一些文件存储而已。



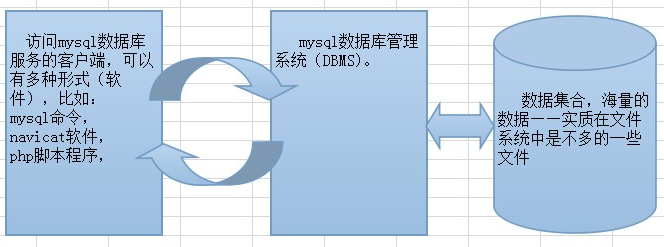


关系数据库常见术语：

* 数据data：
* 数据库database：指一个具体的存储数据的“逻辑名称”，通常一个产品（项目／网站），会使用一个数据库来存储其所有数据
* 数据库管理系统dbms：database management system：通常就是指数据库软件产品
* 表（数据表）table：
* 字段field，列column：
* 行row，记录record：

数据库系统的基本结构

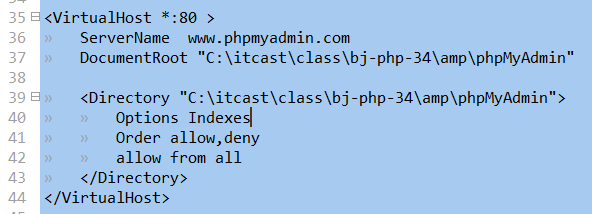
客户端 mysql管理系统 数据库文件



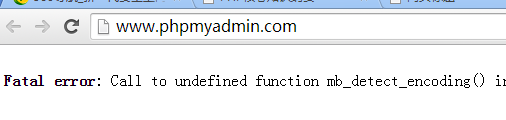
配置phpmyadmin

hosts文件中解析域名：www.phpmyadmin.com

httpd-vhost.conf文件中设定一个站点：



浏览，后发现：



这里需要去打开php的一个模块(在php.ini文件中) ：

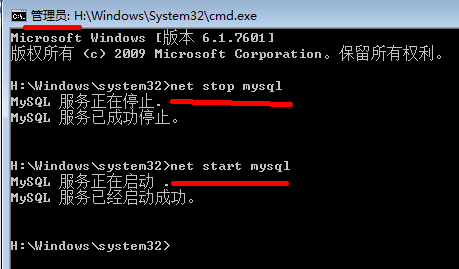


数据库操作的基本模式（流程）

* 建立连接（认证身份）
* 客户端向服务器端发送sql命令
* 服务器端执行命令，并返回执行的结果
* 客户端接收结果（并显示）
* 断开连接

mysql系统级操作和基本语法规定

* 命令行模式：**net start/stop mysql**



* 服务模式：控制面板〉管理工具〉服务〉mysql >启动/停止

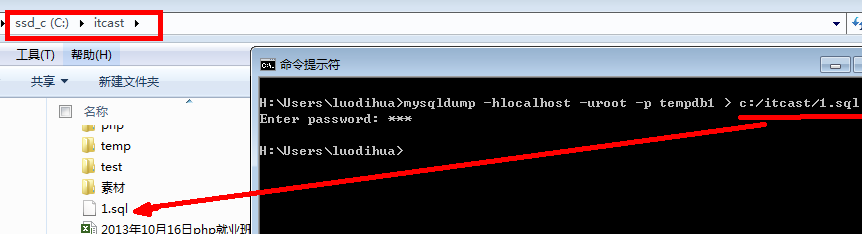
登录/退出基本操作

* 登录：**mysql [-h服务器地址] -u登录名 -P端口号 -p**
* 或登录：**mysql [--host=服务器地址] --user=用户名 --port=端口 --password**
* 退出：quit; 或 exit;
* **注意：登录数据库系统后，需要使用“set names 编码名；”来设定当前连接数据库的“所在的环境变量”，即当前跟数据库打交道的“客户端”本身的编码。通常来说：**
  + **cmd客户端中是固定的使用gbk编码，**
  + **而php网页中，是该网页文件的编码（现在主流都是utf8）**。

备份恢复数据库

备份数据库：

**mysqldump -h服务器地址 -u登录名 -p 要备份的数据库名 > 要保存为的文件**

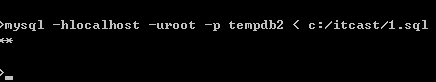


恢复数据库：

将刚才的数据恢复到另一个数据库（tempdb2)中:

**mysql -h服务器地址 -u登录名 -P端口号 -p 数据库名 < 文件名**

注意：通常该数据库名是需要先建立（存在）：



基本语法规定

* 单行注释： #注释内容
* 单行注释： -- 注释内容（注意，两个“--”之后有一个空格）
* 多行注释： /\*注释内容\*/

语句结束符

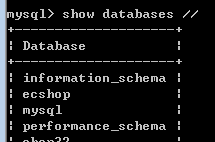
默认是分号（英文）

还可以自己设定，设定命令为：delimiter

形式：delimiter 新的符号（可以是多个字符）

然后在此语句之后，就要使用新的设定的符号来结束一条语句。





数据库中命名规则

mysql本身不区分大小写。

但在某些区分大小写的操作系统中，数据库名和表名会区分大小写。

推荐使用“下划线命名法”来对各种标识符命名：单词都用小写字母，单词之间用“\_”分开。

php中操作数据库的基本代码和流程

基本流程：

1，连接数据库：

$mylink = mysql\_connect(“localhost”, ‘root’, ‘123’);

2，设定连接编码（通常是utf8）

mysql\_set\_charset(“utf8”); //也可以使用：mysql\_query(“set names utf8”);

3，选择数据库（如有需要）

mysql\_select\_db(“数据库名”); //也可以使用：mysql\_query(“use 数据库名”);

4，执行sql命令。

$result = mysql\_query( “几乎任何sql语句 ”);

返回的结果通常需要分两种情形进行处理：

4.1：如果是无返回数据的语句：

4.1.1 如果$result为true，表示执行成功

4.1.2 如果$result为false，表示执行失败

4.2：如果是有返回数据的语句：

4.2.1 如果$result为false，表示执行失败

4.2.2 否则，执行成功，需要继续从结果中取出数据并显示出来。

补充sql语句：

set names gbk；

use 数据库名；

show databases：

desc 表名： 显示一个表的“结构信息”，返回的其实也是结果集（类似select语句）

补充php操作mysql的函数：

$record = mysql\_fetch\_array( 结果集$result );

$n1 = mysql\_num\_rows( 结果集$result ): 获取结果集的行数

$n2 = mysql\_num\_fields( 结果集$result ): 获取结果集的列数

mysql\_field\_name( 结果集$result， $i ): 获取结果集中的第i个字段名（i从0开始）

细看今天的代码文件：《3php\_mysql\_base.php》

数据定义语句

数据库定义

语法形式

create database [if not exists ] 数据库名 [charset 字符集] [collate 字符排序规则]；

说明：

1，if not exists：用于判断是否存在该数据库名，如果存在则不执行该语句

2，字符集： 意图数据存储到本数据库中的时候所使用的字符编码名称，通常utf8，也可以gbk。

3，字符排序规则通常不设置，而是使用所设定的字符集的默认规则（每个字符集都有一个默认的排序规则）；

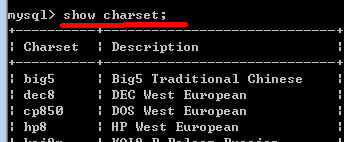
什么叫排序规则：设定一个字符集中的所有字符怎么排列先后顺序的规则。

“中”，“国”，“人”：

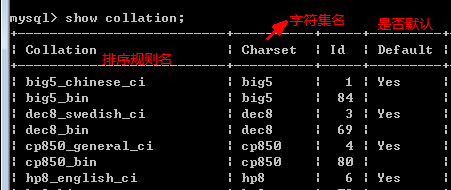
举例：



显示mysql中的所有可用字符集：



显示mysql中的所有可用排序规则：



修改数据库：

* alter database 数据库名 character set=新字符集 collate=新校对集;

删除数据库：

drop database 数据库名；

其他数据库相关语句

选择（进入）某数据库：　　use 数据库名；

通常，要进行数据中的数据表和数据的操作，都必须先“进入”该数据库。



显示所有数据库： show databases;

显示某个数据库的“创建语句”：

show create database 数据库名；