二分查找算法

二分查找算法的前提：

1，针对的是索引数组；

2，针对的是已经排好序的数组；

有关二分查找算法的效率（性能）问题的一点说明：

1000 个数据，约 10 次找出；

100 万个数据，约 20 次找出；

10 亿个数据，约 30 次找出

40 亿个数据，约 32 次找出；

数据库介绍

分类

- 层次数据库
- 网状数据库
- 关系数据库

主流数据库：

SQL：这是关系数据库的“语言标准”，STRUCT　QUERY　LANGUAGE(结构化查询语言）。

关系数据库的特点：表跟表之间的数据可以建立一定的“对应关系”。
几个基本概念：

- 数据 data：
- 数据库 database：用于存储一个项目/产品/软件所使用的各种数据的一个逻辑单位。

- 数据库管理系统 dbms：其实就是指一个具体的数据库软件产品：database management system
- 表（数据表）table：用于存储一种数据的“结构形式”，基本就是“行列对齐”的“外观样子”。
- 字段 field，列 column：一个数据表中的一个纵列，其有一个名字，又叫“字段”
- 行 row，记录 record：

数据存储的“表现”：

我们通过代码（语句），可以对各种数据进行各种操作，但其实，在文件表现，其实只是几个文件名而已，具体在数据库的存储目录中：

1，每个数据库，会对应一个文件夹；

2，每个数据表，会对应一个或几个文件

特别注意：

上述只是演示，我们的数据库的各种“操作”，可能会在文件夹系统中有所反应，但：绝对不要去手工对这些文件进行操作！！！

mysql 数据库操作的基本模式（和流程）：

- 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 数据库的系统级操作及基本语法规定

启动/停止 mysql 数据库服务：

- 命令行模式（需要管理员身份）：
  - **启动：net　start　mysql**
  - **停止：net　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 mysql 系统

- 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**
  - 前两个语法，可以使用的前提都是：设定好了环境变量；
- 退出：quit; 或 exit;——是指已经登录（进入）了 mysql 之后。
- 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“环境编码名”**，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 gbk 编码，
  - 而 php 网页中，是该网页文件的编码（现在主流都是 utf8）

数据库的备份和恢复：

备份：就是将一个数据库，完整地“转换为”一个可以随时“携带和传送”的文件。

语法：

mysqldump -h 服务器地址 -u 登录名 -p 数据库名 > 文件名

恢复： 就是讲一个备份的数据库文件，完整地还原为一个可以使用的数据库。

语法：

mysql -h 服务器地址 -u 登录名 -p 数据库名 < 文件名

注意：

1，这两个命令，都是在“没有登录 mysql”的时候使用。

2，其中 mysqldump 命令还要求为管理员身份。

3，通常，恢复，就是指恢复原来数据库中的所有表数据信息及其他信息，而数据库名可以是原来的名字或新的名字。

基础语法规定

注释：

有如下 3 中注释：

1， 单行注释： #注释内容

2， 单行注释： -- 注释内容（注意：--后面有一个空格）

3， 多行注释： /* 注释内容*/

语句行：

默认情况下，以一个英文分号作为一条语句的结束！

而且，常规的操作中，都是“一次执行一条语句”；

但：

mysql 中，可以可以人为设定语句结束符，做法如下：

delimiter 新的结束符

此行之后 ，就可以使用新的结束符了：

大小写问题

1，mysql 语言内部本身不区分大小写；
2，但，mysql 的某些命令执行会生成文件(夹)，此时他们就可能会区分大小写：
2.1：在文件(夹)名称 区分大小 写的系统中，这些名字也会区分大小写，比如 unix，linux 系统；
2.2：在文件(夹)名称不区分大小写的系统中，他们同样不区分大小写，比如 window 系统。

命名问题

可以自己命名的名字，称为标识符，包括：数据库名， 表名，字段名，视图名，函数名，过程名，变量名，用户名，，等等。

可以命名标识符的字符比常规的语言多，但特别建议只用：字母数字和下划线，并不用数字开头。

非常规字符或系统关键字虽然可以作为标识符使用，但最好要包在反引号（数字 1 左边那个反撇 ` ）中，并且不推荐。

对数据库名，表名，和视图名，在 window 系统中不区分大小写，而其他系统中区分，建议全使用小写，并采用下划线分割法。

对其他自己命名的标识符（字段名，函数名，过程名），不区分大小写，但也建议全使用小写，并采用下划线分割法

数据库定义语句

创建数据库：

形式：

create database 数据库名 【charset 字符编码名称】 【collate 排序规则】；

说明：

1，字符编码名称是用于设定当前数据库中存储的字符内容以什么编码来存储。

2，collate 排序规则用于设定其中的字符内容的“大小关系”（先后顺序）：

对于英文，基本没有任何问题，比如：

“abc” “abd”： 后者大；

“d”; “abc”： 前者大；

.....

所有的对于英文字符的比较，本质上都是“单个字符”的比较。

但，对于中文，就成问题了，比如：

“中国”， “印度”： 谁大？

“中”， “美”： 谁大？

排序规则，就是用于设定类似这种字符的大小关系或先后顺序的一种规定！

实际我们的代码中（应用级别），只是一个名字：排序规则名

而且，通常每种字符编码（字符集），都有一个默认的排序规则，所以，通常都不写的。

显示 mysql 中的所有可用的字符编码（供共 39 种）：

显示 mysql 中的所有可用排序规则（共约 200 种）：

实际应用中，我们使用某个字符集（字符编码），然后可用的与之对应的排序规则其实可选项很少，通常只有 2 个。当然，我们一般也都不使用它。

删除数据库：

形式

drop database 【if exists 】 数据库名；

说明：

1，if exists 是用于一种“安全运行”的考虑。如果数据库不存在，也不会报错。否则会报错。

修改数据库（字符编码)：

基本上，就是修改数据库的“属性”而已：只有 2 个：

修改编码；

修改排序规则；

alter database 数据库名 charset 新的编码名 collate 新的排序规则名

显示所有数据库：

show databases；

显示一个数据库的创建语句：

形式：

show create database 数据库名；

“进入”（选择）某个数据库：

use 数据库名；

“fetch 函数”的 3 种形式的辨析：

假设 mysql_query(“select id, age, name ”)执行的结果集为$result，其该表中的数据类似这样：

id	name	age
1	user1	18
2	张三	28
4	user4	38

$result = mysql_query(“select id, age, name ”);

mysql_fetch_assoc($result):得到的数组类似这样：

array(“id” => 1, “name” => “user1”, “age”=>18);

mysql_fetch_row($result):得到的数组类似这样：

array(0=> 1, 1=> “user1”, 2=>18);

mysql_fetch_array($result):得到的数组类似这样：

array(“id” => 1, “name” => “user1”, “age”=>18, 0=> 1, 1=> “user1”, 2=>18);

扩展 php 中操作 mysql 数据的几个函数：

$n1 = mysql_num_rows(结果集); //获得该结果集的数据行数；

$n2 = mysql_num_fields(结果集);//获得该结果集的数据列数；

$name = mysql_field_name(结果集, $i); //获得结果集的第 i 个字段的名字！i 从 0 开始算起