一、数据库简介

1、概念

数据库中持久存储数据的介质。

计算机中的数据库，就是能够高效的组织与管理数据的介质(内存、硬盘)

2、数据库的发展史

网状数据库

是指以处理记录类型为节点的网络模型来存储与管理数据的。

层次数据库

层次数据库是网状数据库之后的一种数据库，将数据组织成一种有序有向的树状结构。

关系数据库

就使用二维表模型来存储与管理数据库的。

3、关系型数据库(纯理论)

关系型数据库就是使用关系模型来组织与管理数据的数据库。

关系模型就是二维表模型，而一个关系型数据库是由多个二维表及表之间的联系组成的数据库的集合。

关系：关系就是具有行列结构的表格，每个关系都有一个关系名，对应数据库中的数据表。

元组：元组可以理解为二维表中的一行，也就是数据表中的一条记录。

属性：属性可以理解为二维表中的一列，也就是数据表中的一个字段

域：是限制属性的取值范围，也就是对列的一种约束。

关键字：能够唯一标识元组(记录)，一组属性(字段)

4、常见的关系型数据库

关系型数据库

大型数据库：Oracle(甲骨文)、DB2(IBM公司)

中型数据库：SQL Server(微软公司)、**MySQL**(甲骨文)

小型： Access(微软公司)

非关系型数据库noSQL

MangoDB、Redis

5、SQL简介

**S**tructure **Q**uery **L**anguage结构化查询语言，主要用于对数据进行操作

DML Data Manuplation Language 数据操作语言 update、insert、delete

DQL Data Query Language 数据查询语言 select

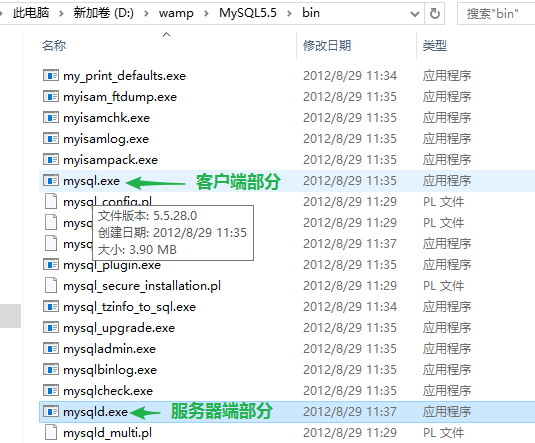
DDL Data Defination Language 数据库定义语言 create

DCL Data Control Language 数据控制语言 grant

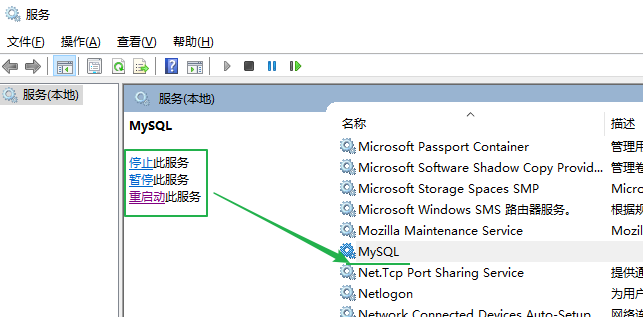
TCL Transaction Control Language 事务控制语言

二、访问数据库

1、MySQL服务器的管理



可以通过Window的服务管理器：



2、登陆MySQL数据库服务器

工具：客户端部分mysql.exe文件，运行环境是cmd窗口。

语法：

mysql -h服务器的地址 -P端口号 -u用户名 -p密码

说明：

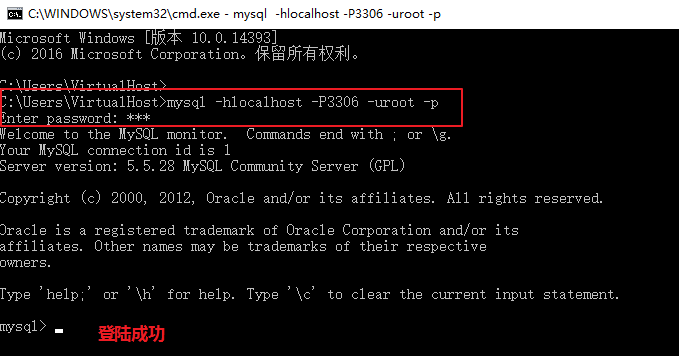
-h mysql服务器的地址 localhost表示本机 127.0.0.1表示本机，如果是本机可以省略

-P 端口号，默认是3306 如果是默认可以省略

-u 用户名 默认的帐户root是管理员帐户

-p 密码 不建议在-p后直接输入密码(注意)

示例：



3、MySQL的层次结构

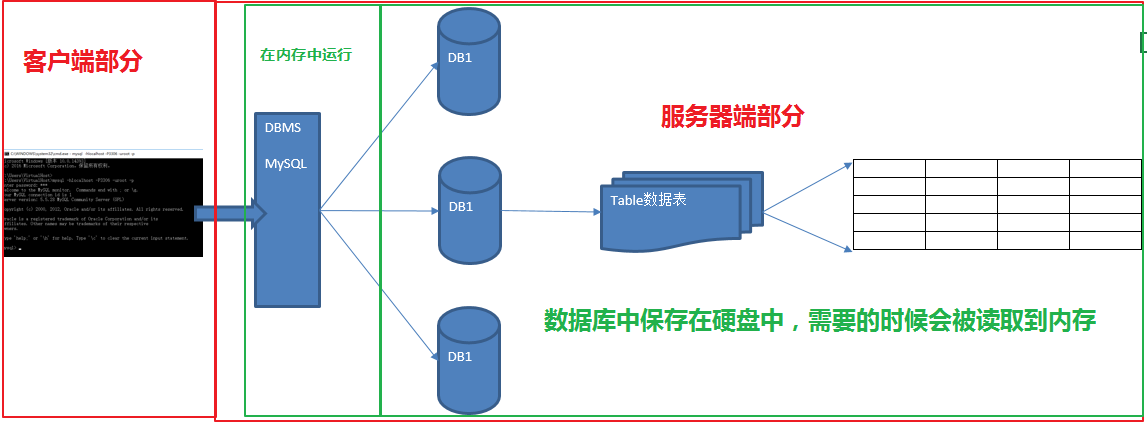
DBMS DataBase Management System数据库管理系统

DB DataBase 数据库

Table 数据表(对应现实中的表格)

Record 数据表中的一行

Fields 数据表中的一列



三、数据库的基本操作

1、查看数据库

语法：

show database**s**;

说明：

查看当前有多少个数据库存在。

示例：



test是一个测试数据库，可以删除。其他余三个数据库MySQL的系统数据库。

2、创建数据库

语法：

create database 数据库名 【库选项】

说明：

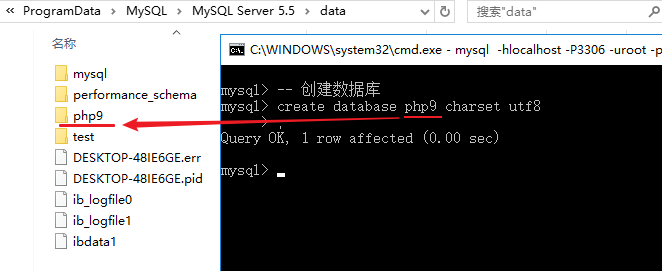
用于创建数据库，数据库名不要与MySQL的关键字重复，如果真的需要使用关键字，必须使用反引号(``)将数据库名括起来。

【库选项】

charset 用于设置存储字符集

collate 用于设置校验集

示例：



当创建一个数据库后，会在硬盘中创建同名的一个文件夹。

MySQL的语句规则：

**每一条语句后必须使用;结束。分号与加车的作用是将客户cmd窗口中输出的命令或数据传递远程的服务器。**

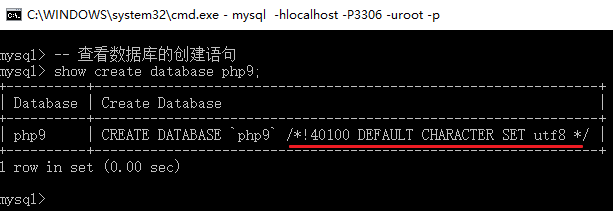
3、查看数据库的创建语句

语法：

show create database 数据库名;

说明：

用于查看数据库的创建语句，可能获取库选项的相关信息。

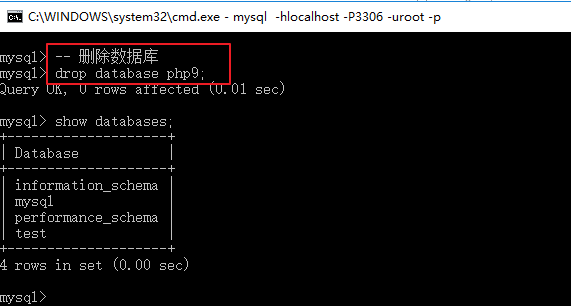


4、删除数据库

语法：

drop database 数据库名;

示例：



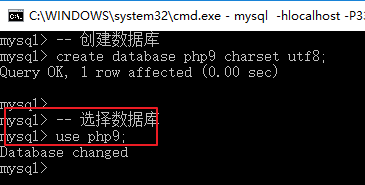
5、选择数据库

语法：

use 数据库名;

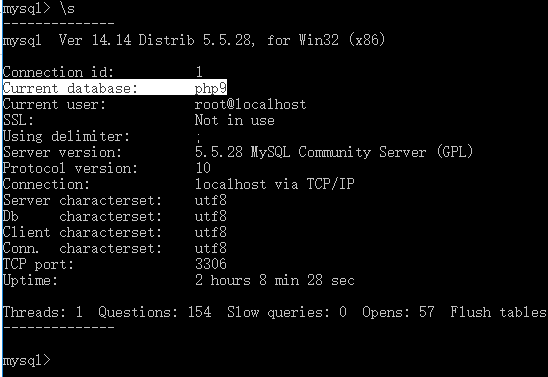
说明：

由于表是存储在某个具体的库中的，所以要对表操作，需先进入到指定的数据库



扩展：

\s 可以查看数据库的信息，其中包含当前所在的数据库信息



四、数据表的基本操作

1、设置客户端的通讯编码

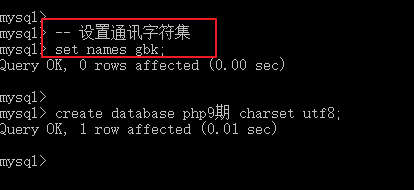
语法：

set names 客户端的字符集

说明：

用于明确的告诉服务器，当前客户端与之通讯的编码是什么。

示例：



2、查看数据表

语法：

show table**s**;

说明：

查看当前数据库中有多少个数据表存在

示例：



3、创建数据表

语法：

create table 表名(

字段名 列类型 【列属性】,

字段名 列类型 【列属性】,

….

)【表选项】

说明：

列类型就是所定义的列的取值限制（对应程序中的数据类型）

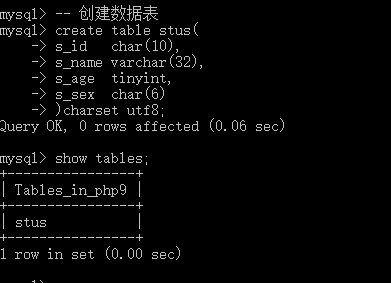
【表选项】

charset 设置存储字符集

collate 设置校验集

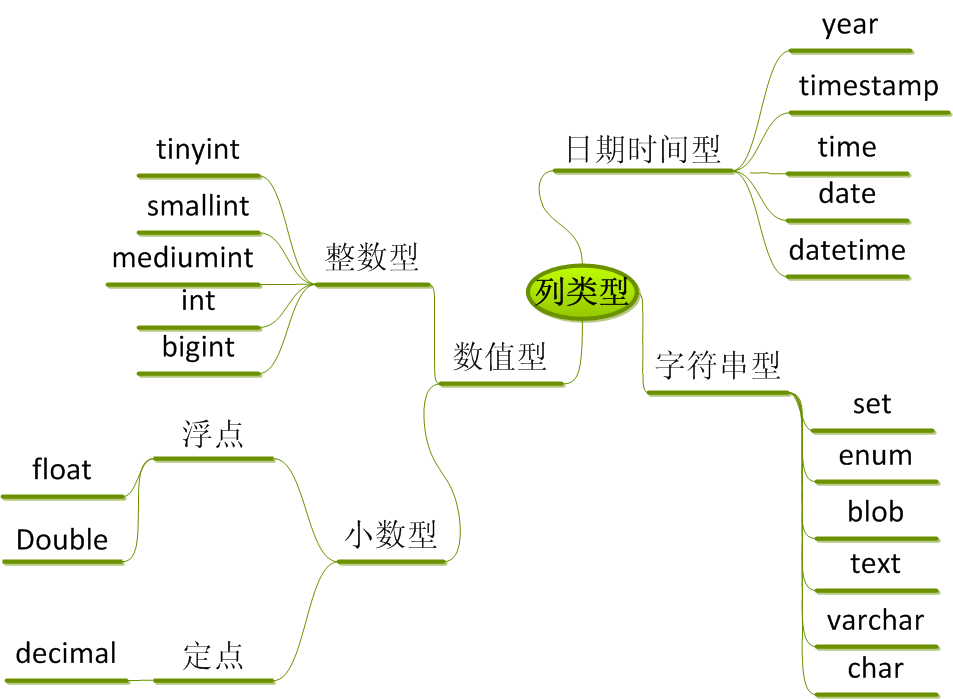
engine 存储引擎

示例：



提示：

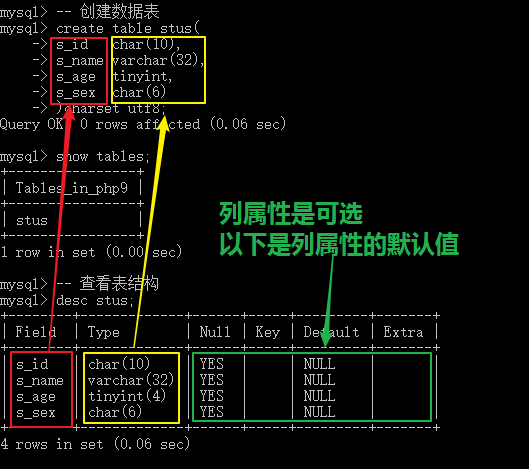
当创建一个表后，仅是创建了一个表的结构(表头)



4、查看表结构

语法：

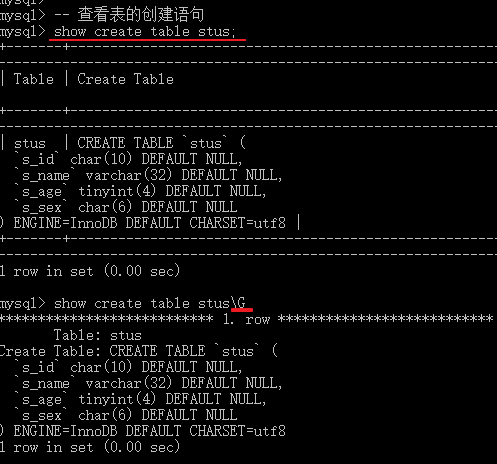
desc 表名;



5、查看表的创建语句

语法：

show create table 表名;

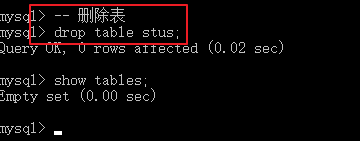


6、删除表

语法：

drop table 表名;

示例：



五、数据的基本操作

1、增加数据

语法：

insert into 表名【(字段列表)】 values(值列表)

说明：

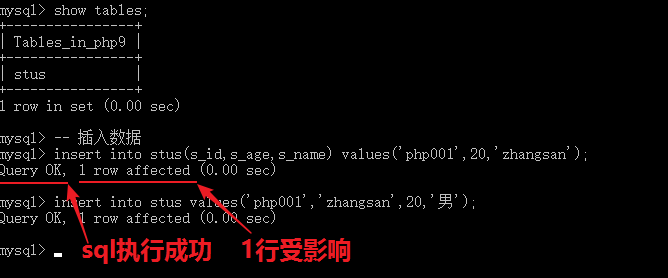
字段列表中字段名不需要加引号

值列表中的字符型的值必须加引号。

字段列表中字段的个数要与值列表中值的个数一致，而且顺序也一要致。

字段列表也可以省略，如果省略那表示全部字段插入数据。值列表中值的顺序要与表中的字段的顺序一致。

示例：



2、查看数据

①、语法：

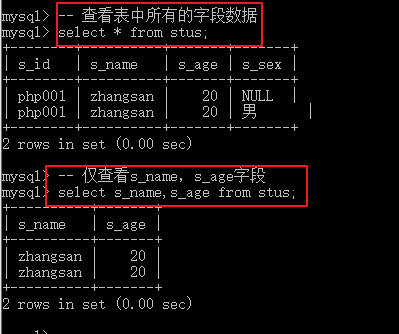
select \*|字段列表 from 表名 【where子句】【group by子句】【having子句】【order by子句】【limit子句】

说明：

\* 所有的字段

字段列表 用于查看部分字段。

示例：



②、where子句

【where子句】

根据where子句的表达式，对记录进行筛选。即是表达式，就会涉及到运算符。

MySQL的运算符：

>、<、>=、<=、<>、=、and、or、not、between m and n、in(值列表)、is null、like

= 用于判断是不相等，同也是赋值运算符

and、or 逻辑与、逻辑或

between m and n 字段的值介于m与n之间

in(值列表) 字段的值等于值列表中的某一个

is null 专用于判断null值的运算符

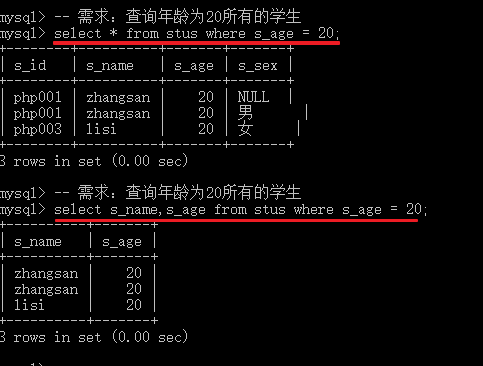
like 用于模糊查询，必须与两个占位符进匹配使用

占位符：%、\_

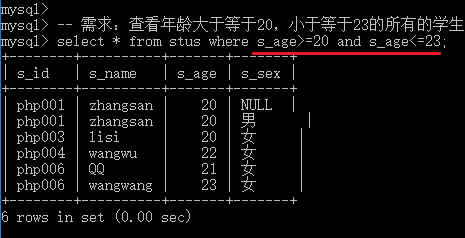
% 代表当前位置及其后0个或多个字符

\_ 代表当前位置的1个字符

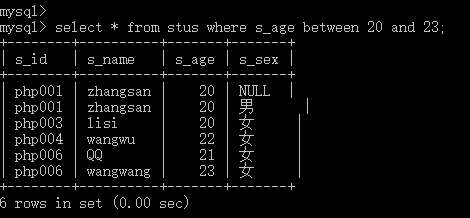
示例： =



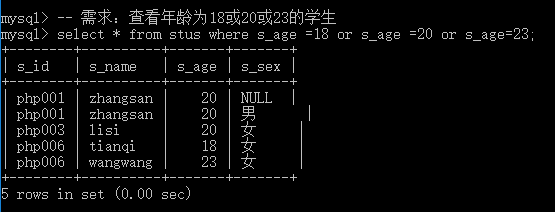
示例：and



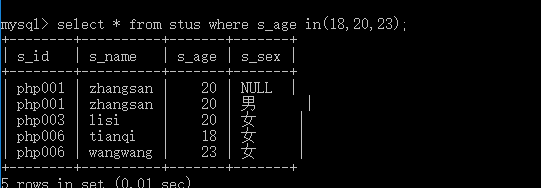
示例：between m and n



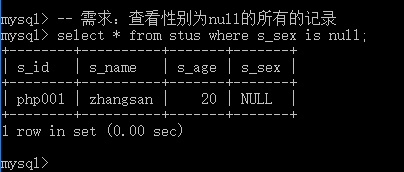
示例：or



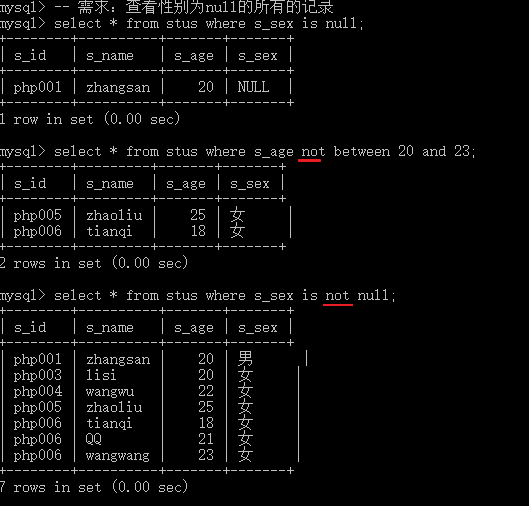
示例：in



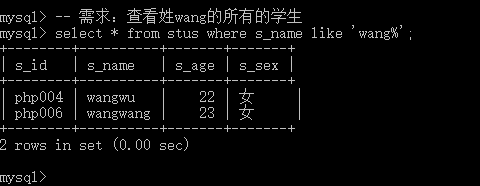
示例：is null



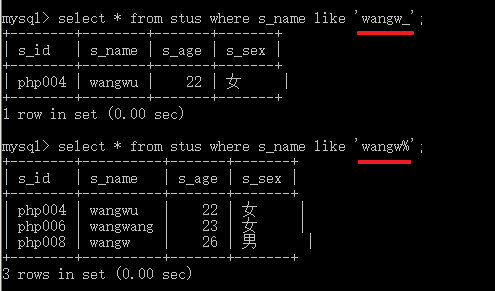
示例：not



示例：like %



示例：like \_



3、修改数据

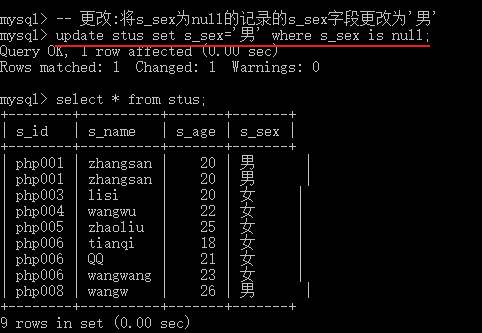
语法：

update 表名 set 字段1=值, 字段2=值,… 【where 子句】

说明：

根据where子句的条件匹配到的记录进行修改。如果省略where表示所有的记录都匹配。

示例：



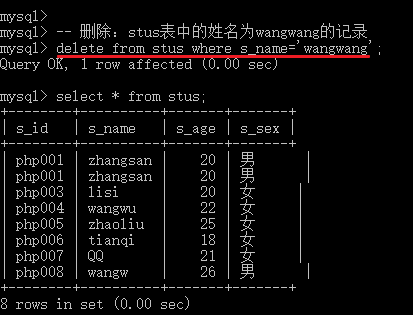
4、删除数据

语法：

delete from 表名 【where子句】

说明：

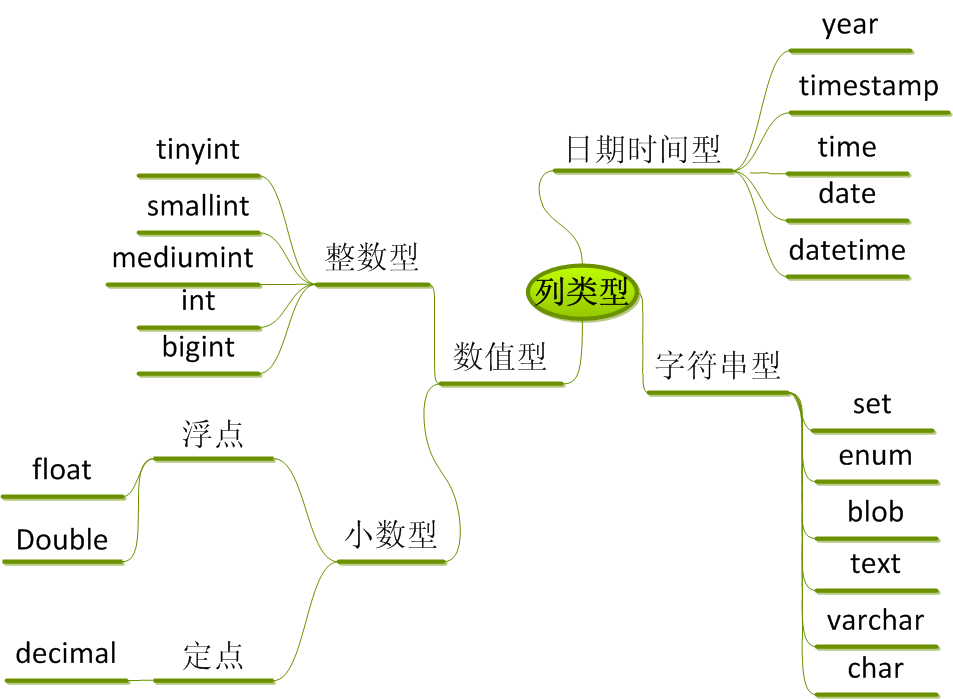
将where子句中匹配到的记录进行删除，



对于结构的查看：使用的是show

对于数据的查看：使用的是select关键字

六、列类型——详解



1、数值型

①、整数型

整数型是一个系列，都是整数，唯一的区别是所能表示的数大小为同而已。

|  |  |  |  |
| --- | --- | --- | --- |
| **类型** | **字节** | **最小值（有符号）** | **最大值（有符号）** |
| **最小值（无符号）** | **最大值（无符号）** |
| TINYINT | 1 | -128 | 127 |
| 0 | 255 |
| SMALLINT | 2 | -32768 | 32767 |
| 0 | 65535 |
| MEDIUMINT | 3 | -8388608 | 8388607 |
| 0 | 16777215 |
| INT | 4 | -2147483648 | 2147483647 |
| 0 | 4294967295 |
| BIGINT | 8 | -9223372036854775808 | 9223372036854775807 |
| 0 | 18446744073709551615 |

在使用整数来表示实现中数据时，有时现实中数据不可能出现负数。所以可以使用unsinged关键字来将有符号位的整型类型，更改为无符号位。

②、小数型

**小数型又分为浮点与定点(了解)：**

**浮点类型分为float与double**

|  |  |  |  |
| --- | --- | --- | --- |
| **类型** | **字节** | **最小值** | **最大值** |
| Float | 4 | -3.402823466E+38 | 3.402823466E+38 |
| Double | 8 | -1.7976931348623157E+308 | 1.7976931348623157E+308 |

语法：

float(m,d)

double(m,d)

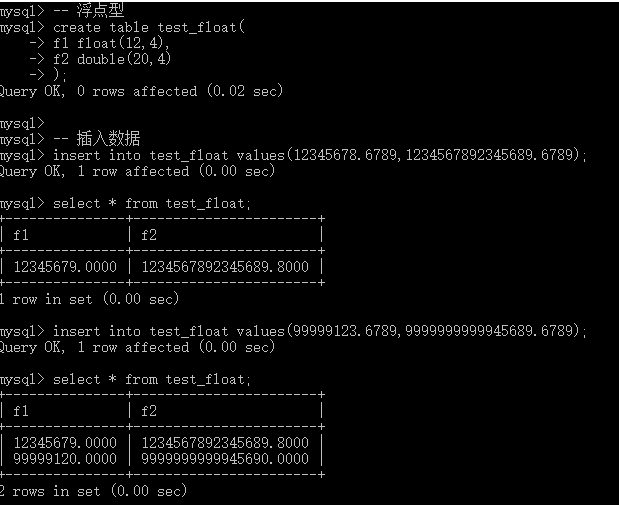
说明：

m 表示的是总位数

d 表示的是小数位

整数位 m-d

示例：



说明：

对于浮点型会出现精度丢失的问题。

对于float类型仅能精确到5至6位，多于的位会使用0代替

对于double类型仅能精确到14至15位，多于的位会使用0代替

decimal(定点型)：

语法：

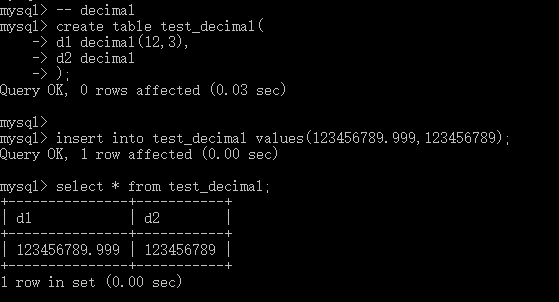
decimal(m,d)

说明：

m 表示总位数，m最大的取值为65

d 表示小数位，d最大的取值为30

示例：



2、字符型

①、定长

语法：

char(len)

说明：

len表示字符的个数。最大是255。

|  |  |  |  |
| --- | --- | --- | --- |
| 表的存储字符集 | 设置的列属性 | 实际存储的数据 | 实际所占的空间 |
| Utf8 | Char(10) | 5 | 10\*3=30 |

②、变长

语法：

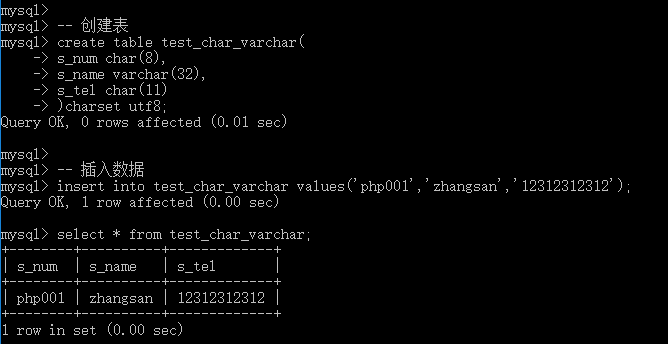
varchar(max)

说明：

max表示字符的个数。理论是65535(会受很多条件限制)

|  |  |  |  |
| --- | --- | --- | --- |
| 表的存储字符集 | 设置的列属性 | 实际存储的数据 | 实际所占的空间 |
| Utf8 | varchar(10) | 5 | 5\*3=30 |

示例：



③、文本类型

text也是一个系列，包含

tinytext、text、mediumtext、longtext，唯一的不同就是所能存储的字符的多少不同。

④、blob类型

blob也是一系列，包含

tinyblob、blob、mediumblob、longblob，唯一的不同就是所能存储的数据的多少不同。

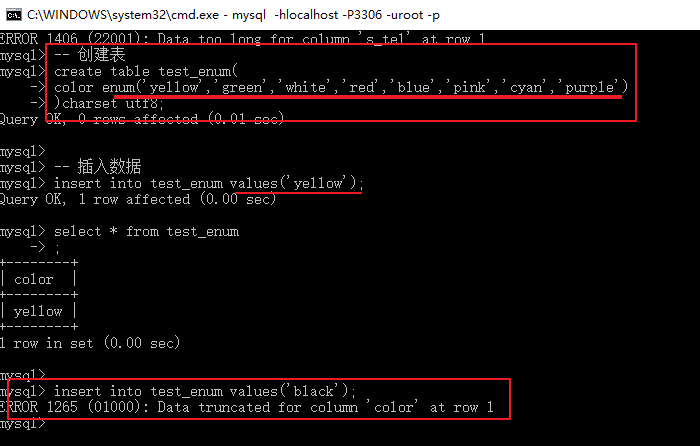
blog可以存储二进制数据

⑤、枚举类型

用于规范数据，只允许用户从多个预定的数据中，选取一个。

语法：  
 enum(值列表)

示例：

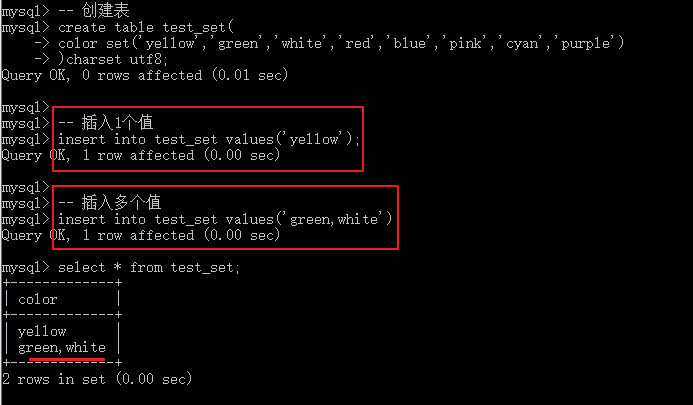


⑥、集合类型

用于规范数据，只允许用户从多个预定的数据中，选取多个。

语法：  
 set(值列表)

示例：



3、日期时间型

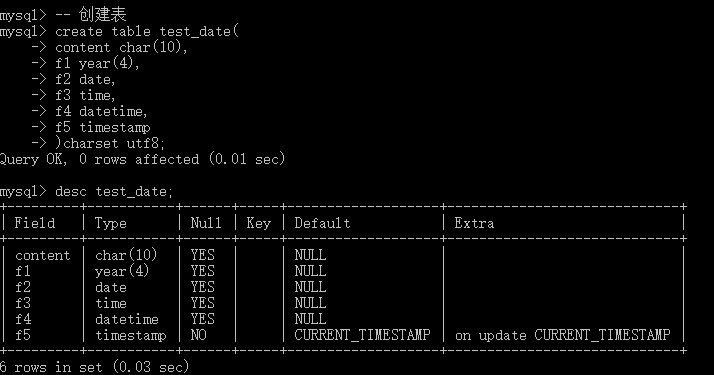
year(2或4) 年份 2表示2位的年份;4表示4位的年份

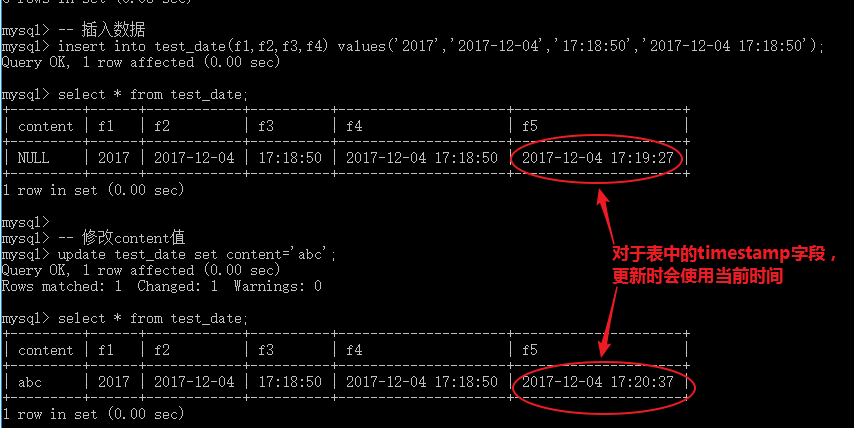
date 日期 格式： '2017-12-4'

time 时间 格式： '17:14:59'

datetime 日期时间型 格式： '2017-12-4 17:14:59'

timestamp 时间戳，主要的作用是用于记录数据表中记录最后一次被修改的时间。





七、字符集与校对集

1、字符集

①、查看MySQL所支持的字符集

语法：

show charset;



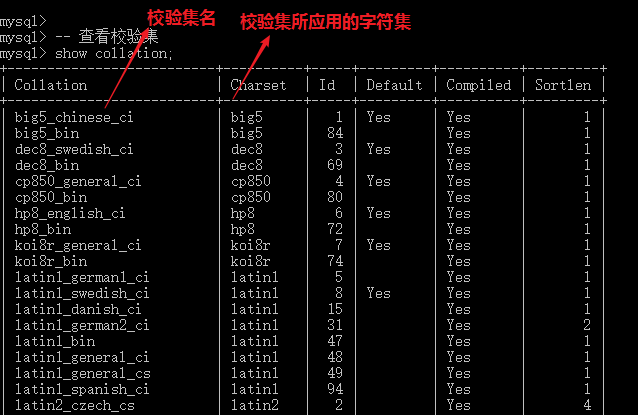
2、校验集

校验集就是某一种字符集中的比较规则。也就是是否区分大小写。

查看校验集

语法：

show collation;



提示：

虽然有很多校验集，但通过观察发现这么多的校验集是以三种方式结尾：

\_cs CaseSensitive 大小写敏感(区分大小写)

\_ci CaseInsensitive 大小写不敏感(不区分大小写)

\_bin 二进制 区分大小写

核验集主要是体现在比较上。

