## mysql数据库

## 第一章

一、简介

mysql数据库是一种c/s结构的软件：客户端/服务端，若想访问服务器必须通过客户端。

二、交互方式

１．客户单链接认证：链接服务器，认证省身份：mysql -hPup

linux: ./mysql -hlocalhost -P3306 -urrot -p;



２.发送sql命令

show databases;

３．服务器接收sql命令，处理sql指令：返回操作结果

４．客户端接收结果：显示结果

５．断开连接

第二章　sql 基本操作

将sql的基本操作根据操作对象进行分类，分为三类：库操作、表操作、数据操作。

一、库操作：对数据库的增删改查

１．新建数据库

creat database 数据库名字　[库选项]

库选项：用来约束数据库

字符集设定:charset/character set 数据库的字符集

校对集设定：collate 具体的校对集

注释：＃　--

注意：数据库名称不能用关键字或者保留字。如果非要使用关键字或者保留字，要用反引号括起。



当数据库的sql语句执行之后，发生了什么。

1. 在数据库系统中会增加相应的数据库信息
2. 会在保存数据的文件夹下创建一个对应的数据库名字的文件夹。
3. 每个数据库下都会有一个opt文件，存放字符集。
4. 查看数据库
   * 1. 查看所有数据库

show database;



* + 1. 查看部分指定当然数据库：模糊查询

show databases like “pattern”; -- pattern 是匹配模式

%代表匹配多个字符

\_：表示匹配单个字符





* + 1. 查看数据库的创建语句

show create database 数据库名



1. 更新数据库
   * 1. 数据库名字不可以修改
     2. 数据库的修改仅库选项：字符集、校对集（校对集依赖字符集）

alter database 数据库名　charset utf8;



1. 删除数据库

drop database 数据库名



1. 表操作：表和字段是密不可分的
2. 新增数据库表

create table [if not exists] 表名(

字段名字　数据类型，

字段名字　数据类型

)[表选项];

If not exists :如果表名不存在，就创建否则不执行创建代码

表选项：控制表的表现

字符集：charset/character set 具体字符集

校对集：collate 具体校对集

存储引擎：ｅｎｇｉｎｅ　具体的存储引擎



1. 任何一个表的设计都需要指定数据库

方案一：显示的制定所属的数据库

create tabel 数据库名.表名；



方案二：隐式的指定所属数据库：先进入到某个数据库环境，然后这样创建表自动归属到某个指定的数据库。



1. 查看数据表
   * 1. 查看所有表

show tables;



* + 1. 查看部分表：模糊匹配

show tables like “patter”;



* + 1. 查看表的创建语句

show craete table 表名;



* + 1. 查看表结构

desc 表名;

describe 表名;

show columns from 表名;



1. 修改数据表
   * 1. 修改表名字

rename table 老表名　to 新表名;



* + 1. 修改表选项

alter table 表名　表选项;



* + 1. 新增字段

alter table 表名 add [column] 字段名　数据类型　[列属性] [位置];

位置：　字段名可以存放表中任意位置‘

first:第一个位置

after: 那个字段之后，　after 字段名; 默认是最后一个字段。





* + 1. 修改字段

alter table 表名　modify 字段名　字段属性　位置



* + 1. 重命名字段

alter table 表名　change 旧字段名　新字段名　数据类型　[属性] [位置]



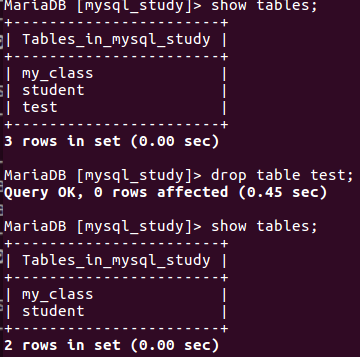
* + 1. 删除字段

alter table 表名 drop 字段名;



* + 1. 删除数据库表

drop table 表名１,表名２.....;



注意：一定要小心删除表，容易负法律责任。

1. 数据操作
2. 新增数据
   * 1. 方案一

给全表字段插入数据，不需要指定字段列表，要求数据的值出现的顺序必须与表中设计的字段出现的顺序一致，凡是非数值数据，都需要使用引号（建议单引号）包裹。

insert into 表名　values(值列表)[,(值列表)];



* + 1. 方案二

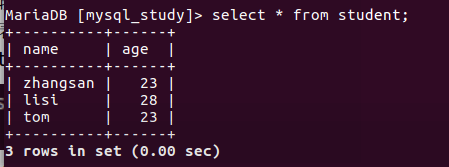
给部分字段插入数据，需要选定数据列表，字段列表的的出现顺序与字段的顺序没有关系， 但是值列表的顺序必须与选定的顺序一致。

insert into student(字段列表)　values(值列表);



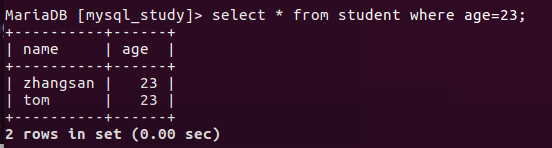
1. 查看数据（最基本的）
   * 1. 查看全部数据

select \* from 表名；



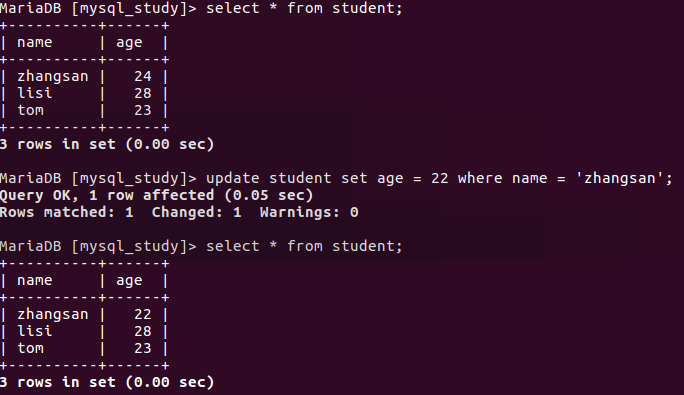
* + 1. 查看指定数据

select \*/字段列表　from 表名　where条件;



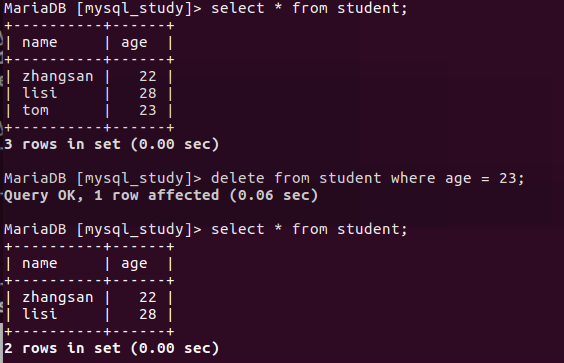
1. 更新数据

updata 表名　set 字段 = 值　[where 条件];

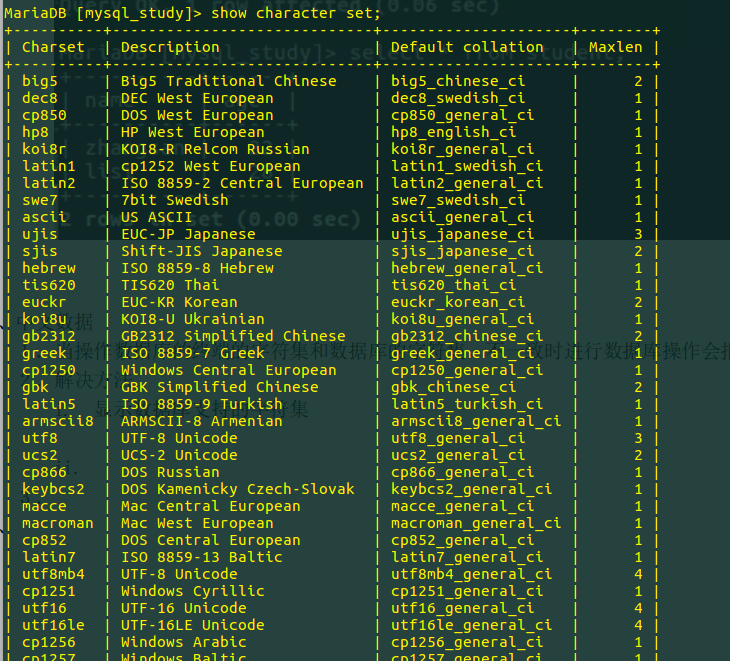


1. 删除数据

delete from 表名　[where条件];

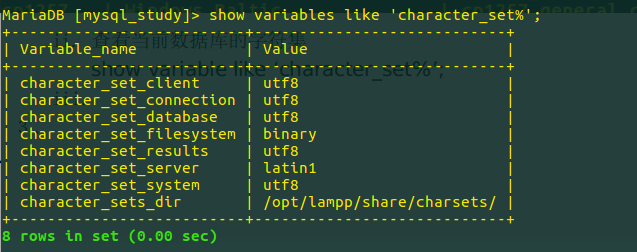


1. 中文数据
2. 当操作数据库的终端的字符集和数据库的字符集，不一致时进行数据库操作会报错。
3. 解决方法
   * 1. 显示数据库支持的字符集



* + 1. 查看当前数据库的字符集

show variables like ‘character\_set%’;



* + 1. 设置数据库处理来自客户端的数据的字符集

如果能把客户端的字符编码改成与数据库一样那就改成一样，如果不能那就更改数据库的字 符编码。

set character\_set\_client = gbk/utf8;



* + 1. 设置数据库返回给客户端的字符集

set character\_set\_results = gbk/utf8;



* + 1. 快速设置字符集

set names gbk/utf8;

1. 校对集：数据比较的方式
2. 校对集有三种格式

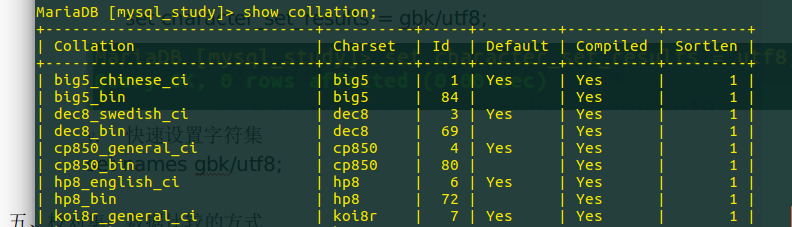
\_bin : binary,二进制比较，取出二进制位，一位一位的比较，区分大小写

\_cs : case sensitive 大小写敏感区分大小写

\_ci : case insensitive 大小写不敏感，不区分大小写

1. 查看数据库所支持的校对集

show collation;

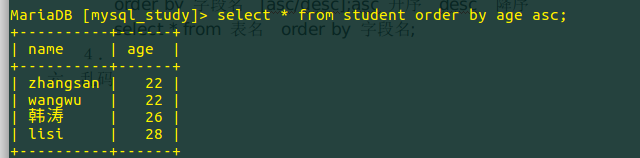


1. 当数据产生比较时，校对集才会生效

根据某个字段进行排序

order by 字段名　[asc/desc];asc 升序　desc　降序

select \* from 表名　order by 字段名;

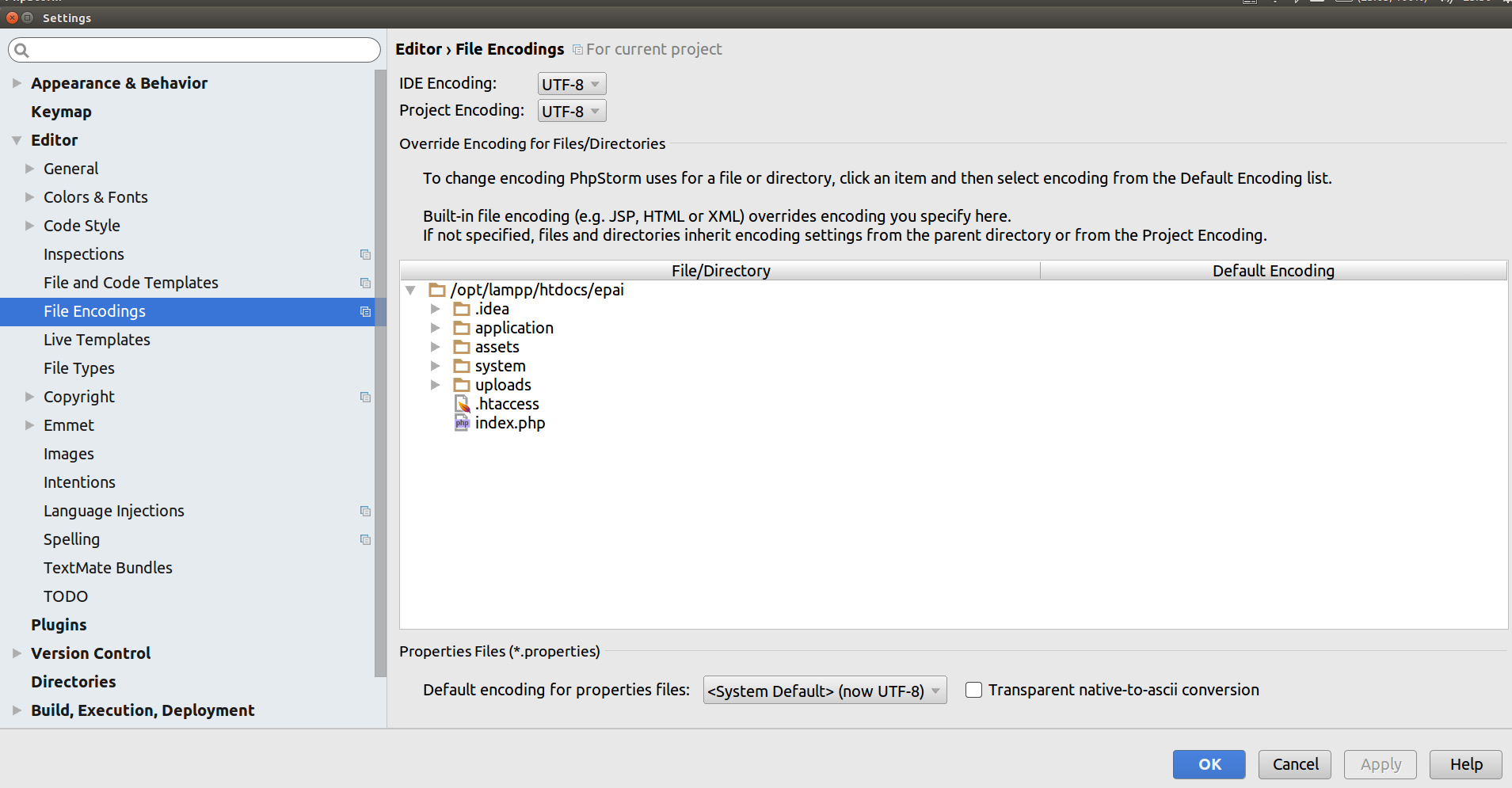


1. 修改校对集必须在表没有数据的情况下才生效。
2. 乱码

动态的网站由三部分构成，浏览器、apache 、数据库服务器。三个部分都有自己的字符集，数据在三个部分之间传递：很容易产生乱码。为了解决乱码要三码合一

1. Web乱码
   * 1. header(content-type:text/html;charset=utf-8);
     2. <mate charset=’utf-8’>
2. php 、mysql 的编码都设置成相同的。
   * 1. php 的编码可以在php的idea里更改，以phpstorm为例：

file -> setting -> Editor -> File Encodings



* + 1. Mysql在建立数据库是或者创建表时都可以指定字符集





1. 如果php 与　mysql的字符集相同那就没必要做过多的字符集转换的设置，如果不同可以参考

数据库中文数据的内容进行设置。（本章第四节）

1. 字段的数据类型和属性
2. 字段的数据类型

所谓的数据类型：对数据进行统一的分类，从系统的角度出发为了能够使用统一的方式进行管理，更好的利用有限的空间。

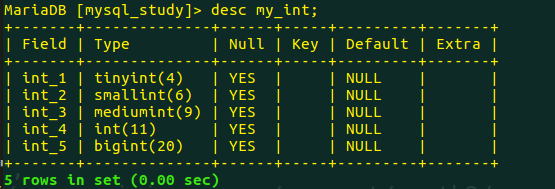
sql 中将数据类型分成了三大类：数值类型、字符串类型、时间类型。

1. 整型
   * 1. tinyint : 迷你整型，使用一个字节存储，表示的状态最多为２５６．
     2. smallint：小整型，使用2个字节存储，表示的状态最多为２^16种。
     3. mediumint :中整型，使用３个字节存储，表示的状态最多为２^24种。
     4. int : 标准整型，使用４个字节存储（常用），表示的状态最多为２^３２种。
     5. bigint: 大整型，使用８个字节存储（常用），表示的状态最多为２^６４种。
     6. unsigned 无符号的意思。
     7. zerofill 补０填充

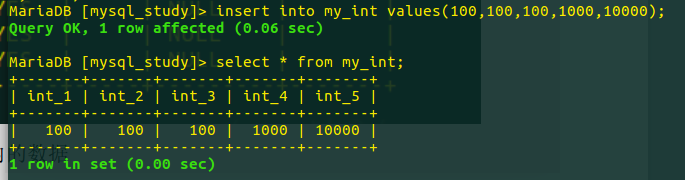


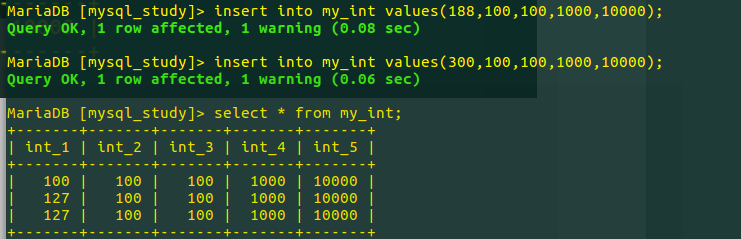
创建一个整型表



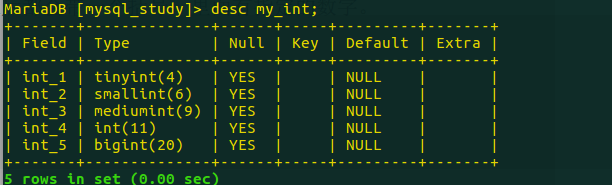


插入数据，必须插入与之相对应范围内的数据，如果范围或者类型不对会得到不一样的效果。

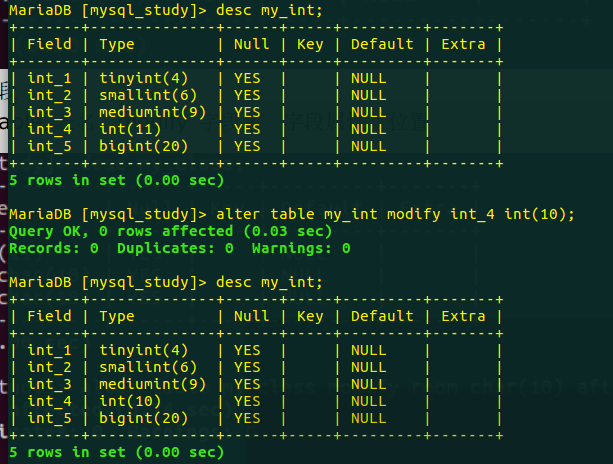




查看表结构时，每个字段的数据类型之后都会带一个括号，里面有指定的数字。

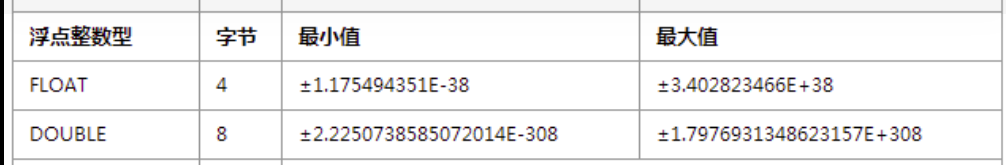


括号里面的数字就是显示宽度，-127就是四位，100就是三位。显示位数没有实际意义，并不影响数据的大小，数据的大小仅仅和数据类型有关，数据的显示位数是用户可以控制的。



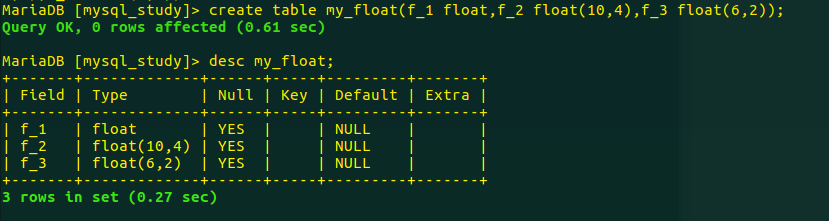
在指定０填充时，如果数据的位数小于数据的显示位数时，系统会使用０进行填充。

1. 浮点型：小数点浮动，精度有限，而且会丢失精度。
   * 1. float : 单精度，占用四个字节存储数据，精度范围大概为７位左右。
     2. double : 双精度，占用八个字节存储数据，精度范围大概为１５位左右。

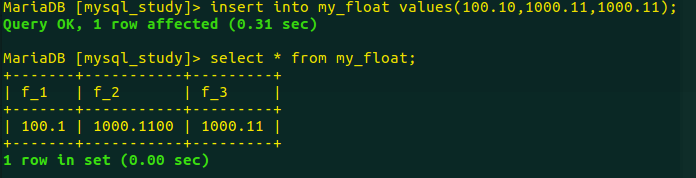


创建浮点数表

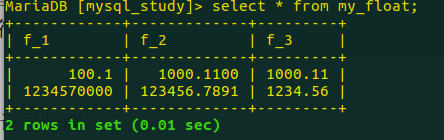
create table my\_float();



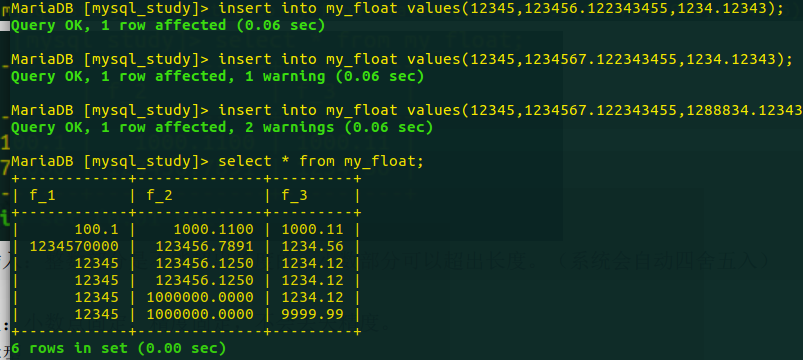
插入数据可以是小数或者是科学计数。







浮点数的插入：整数部分是不能超出长度的，小数部分可以超出长度。（系统会自动四舍五入）

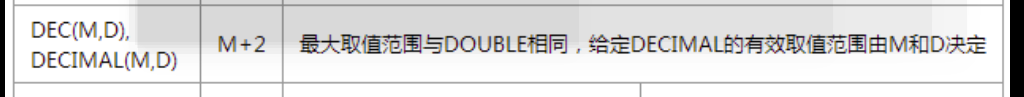


从结果可以看出浮点数的存储都有进位，所以进度不准确。

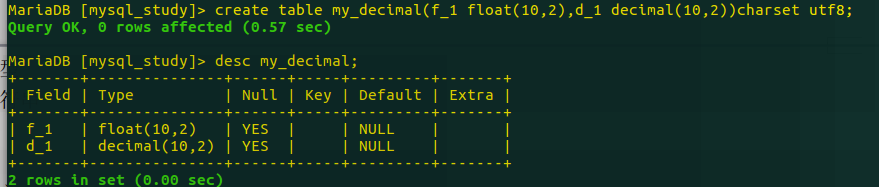
1. 定点型：小数点固定，精度固定，不会丢失精度。

保证整数部分不会被四舍五入（不会丢失精度），小数部分有可能（理论上也不会丢失精度）

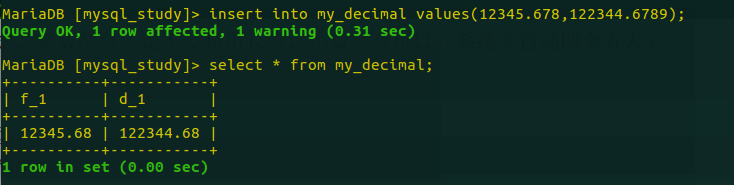
整体最大长度为６５，小数部分长度为３０，整数部分长度为３５．



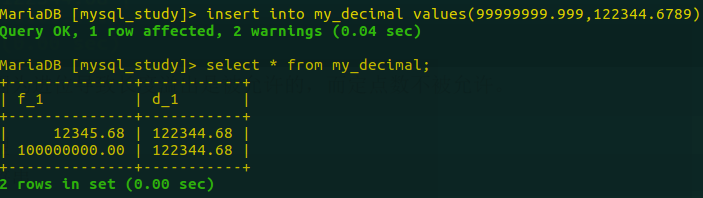
创建定点数表，用一个浮点数作对比

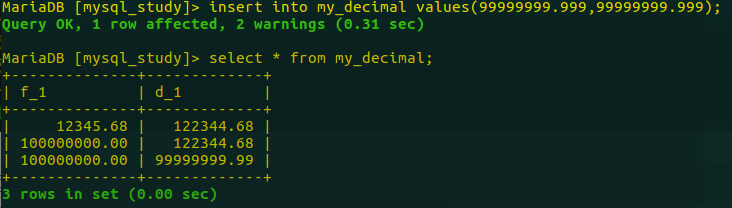


插入数据，整数部分一定不要超出长度，小数部分可以（系统会自动四舍五入）



浮点数因为进位导致长度溢出是被允许的，而定点数不被允许。





数量级大，而精度要求不高的使用浮点数。

数量级大，精度要求也高的用定点数。

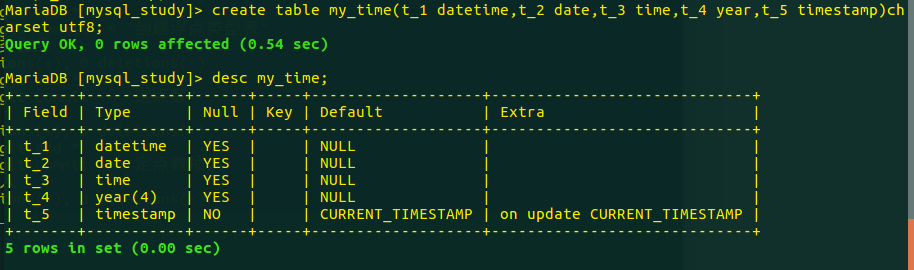
1. 时间类型



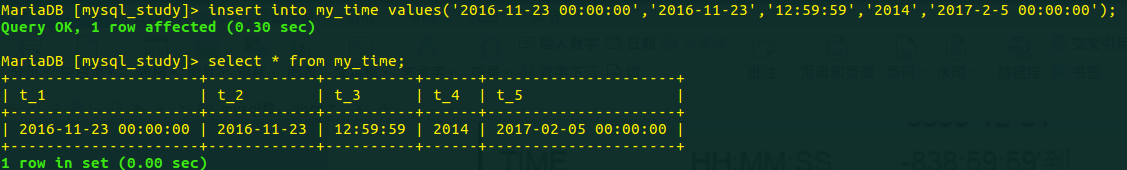
mysql 没有真正意义上的时间戳。

创建数据库

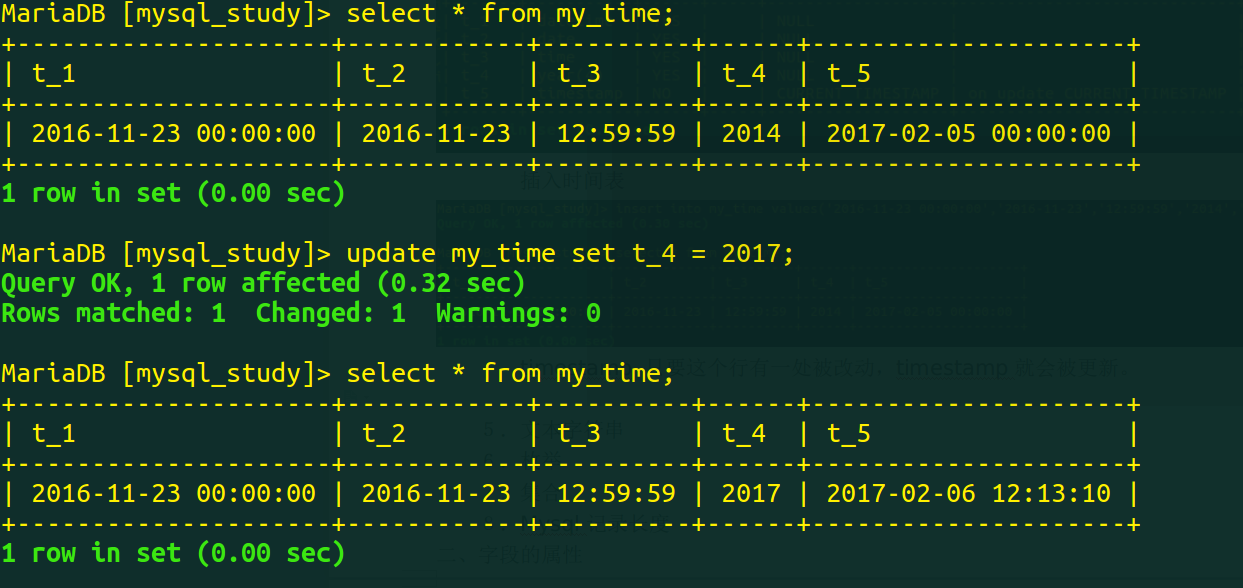
创建时间表



插入时间表



timestamp 只要这个行有一处被改动，timestamp就会被更新。



1. 文本字符串
2. 枚举
3. 集合
4. Mysql记录长度
5. 字段的属性
6. 空属性
7. 列描述
8. 默认值

一、