

# MySQL（二）

## 1. 浮点型

浮点型，不精确的一个大概的小数。

### 1.1 两种浮点型数

float：单精度型。

double：双精度型。

字段名称	有符号数范围	无符号数范围	占硬盘空间
float	(-3.402 823 466 E+38, -1.175 494 351 E-38), 0, (1.175 494 351 E-38, 3.402 823 466 351 E+38)	0, (1.175 494 351 E-38, 3.402 823 466 E+38)	4 个字节
double	(-1.797 693 134 862 315 7 E+308, -2.225 073 858 507 201 4 E-308), 0, (2.225 073 858 507 201 4 E-308, 1.797 693 134 862 315 7 E+308)	0, (2.225 073 858 507 201 4 E-308, 1.797 693 134 862 315 7 E+308)	8 个字节

### 1.2 浮点数的范围

可以在定义浮点数时设置 M 和 D 两个参数：

float(M,D)

double(M,D)

M 代表总数字的位数，最大值是 255。

D 代表其中小数的位数。

比如：float(7,4) 代表最大可以保存 7 位数据，其中 4 位是小数。也就是范围为：0.0000~999.9999

● 在 phpMyAdmin 中如何为浮点型设置范围：

名字 类型 长度/值 默认 排序规则 属性 空 调整权限 A 注释 Virtuality 移动字段

float\_1 FLOAT 3,2 无

float(3,2)

预览 SQL 语句 保存

然而并没有什么用：

#	名字	类型	排序规则	属性	空	默认	注释	额外	操作
1	id	int(10)		UNSIGNED	否	无		AUTO_INCREMENT	修改 删除 主键
2	name	varchar(30)	utf8_general_ci		否	无			修改 删除 主键
3	float_1	float			否	无			修改 删除 主键
4	double_1	double			否	无			修改 删除 主键

解决办法：使用 SQL 语句来决定这个问题。

[ 编辑内嵌 ]

#	名字	类型	排序规则	属性	空	默认	注释	额外	操作
1	id	int(10)		UNSIGNED	否	无		AUTO_INCREMENT	修改 删除 主键 唯一 索引 空间 全文搜索 非重复值 (DISTINCT)
2	name	varchar(30)	utf8_general_ci		否	无			修改 删除 主键 唯一 索引 空间 全文搜索 非重复值 (DISTINCT)
3	float_1	float			否	无			修改 删除 主键 唯一 索引 空间 全文搜索 非重复值 (DISTINCT)
4	double_1	double			否	无			修改 删除 主键 唯一 索引 空间 全文搜索 非重复值 (DISTINCT)

结构

名字 类型 长度/值 默认 排序规则 属性 空 调整权限 A 注释 Virtuality 移动字段

float\_1 FLOAT 3,2 无

预览 SQL 语句 保存

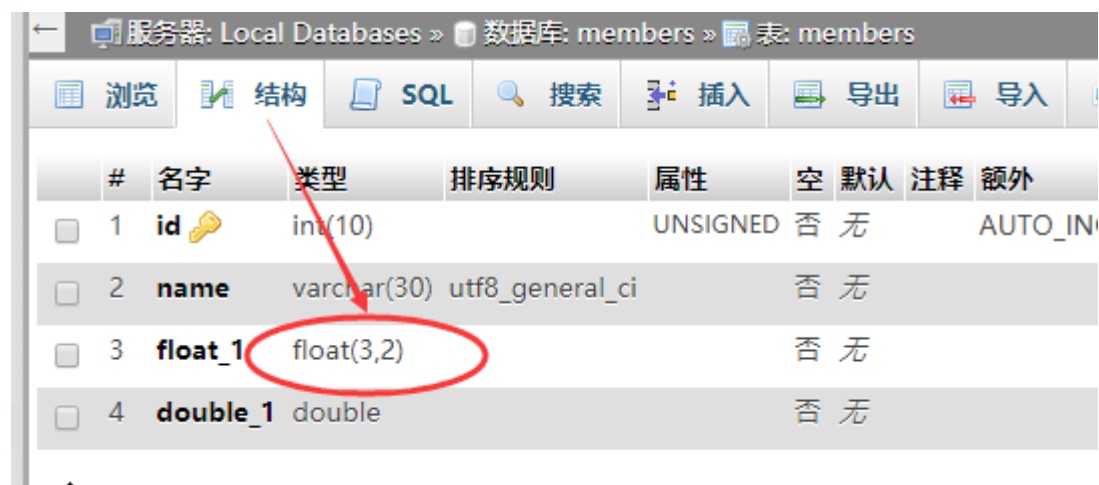
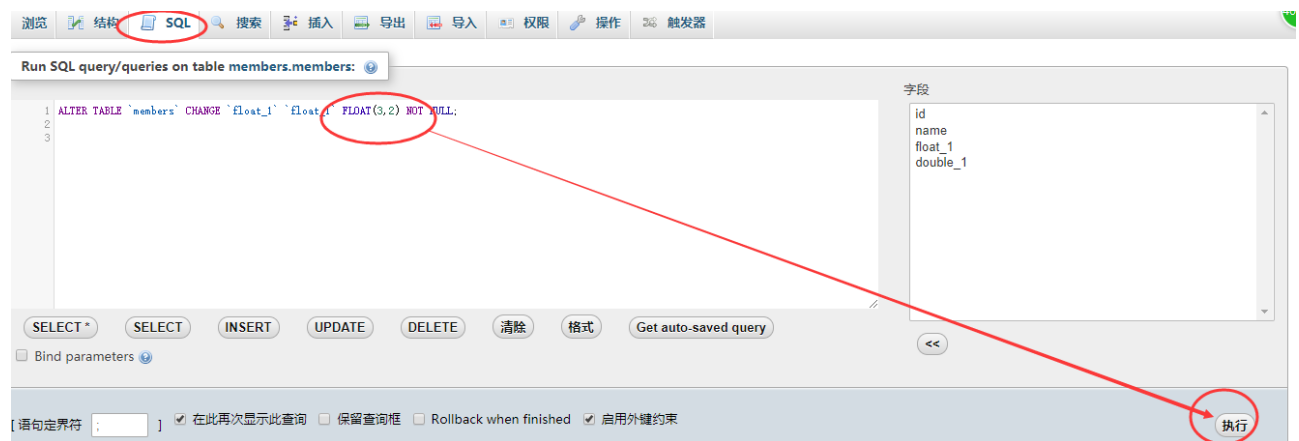
复制以下指令：

预览 SQL 语句

```
ALTER TABLE `members` CHANGE `float_1` `float_1` FLOAT(3) NOT NULL;
```

关闭

打开 SQL 面板：



## 1.3 浮点数特点

1. 保存的数据并不精确只是一个大概的数。

price  
8888880

保存 8888881 时变成了

2. 小数会四舍五入。

price  
2.55556

保存 2.55555 时变成了

说明：由于保存的小数并不精确，只是一个近似值，所以不适合精确数字的情况。

## 2. 定点数

定点数，保以保存精确的小数。

### 2.1 定点数的范围

decimal(M,D)，M 代表总的数字位数【最大为 65】，D 代表其中的小数位。

比如：decimal(5,2)代表共 5 位数字，其中 2 位是小数，比如：888.88

### 2.2 phpMyAdmin 中定义定点数

注意：phpMyAdmin 中设置范围的问题：



实际变成了



也就是说 phpMyAdmin 中设置这种长度时有问题，解决办法：使用 SQL 语句来解决：



名字	类型	长度/值	默认	排序规则	属性	空	调整权限	A.I 注释	Virtuality	移动字段
price	DECIMAL	10,2				<input type="checkbox"/>	<input checked="" type="checkbox"/>			

预览 SQL 语句 保存

查看 SQL 语句并复制:

预览 SQL 语句

```
ALTER TABLE `members` CHANGE `price` `price` DECIMAL(10) NOT NULL;
```

关闭

粘贴到 SQL 中并修改这个指令执行:

Run SQL query/queries on table members.members:

```
1 ALTER TABLE `members` CHANGE `price` `price` DECIMAL(10,2) NOT NULL;
```

字段: id, name, float\_1, double\_1, float\_2, price

SELECT \* SELECT INSERT UPDATE DELETE 清除 格式 Get auto-saved query

☐ Bind parameters

[ 语句定界符: ] ☒ 在此再次显示此查询 ☐ 保留查询框 ☐ Rollback when finished ☒ 启用外键约束

执行

使用 SQL 就可以了:



## 2.3 应用场景

用在需要精确的小数时，比如价格。

## 3. 字符串

### 3.1 几种字符串

字段名称	长度	用途
char	最大 255 个字符。	255 个字符以下的字符串。
varchar	最大 65535 个字节。	65535 个字符以下的字符串。
tinytext	最多 255 个字节。	大段文本时，比如新闻、文章、论文等。
text	最多 65535 个字节。	
mediumtext	最多 16777215 个字节。	
longtext	最多 4294967295 个字节。	

示例：定义一个只能保存 300 个汉字的字段：

名字	类型	长度/值	默认	排序规则	属性	空	索引
jieshao	VARCHAR	300	无			<input type="checkbox"/>	---

## 3.2 char 和 varchar 的区别

### 3.2.1 最大长度

char 最多能保存 255 个 **字符**【无论中文还是英文还是任何符号】。

比如：可以保存 255 个汉字、英文字母、数字、符号等等。

varchar 最多能保存 65535 个字节【UTF8 编码时，一个字符占 3 个字节;GBK 编码时一个字符占 2 个字节】。

最多保存多个个字符呢？

utf8 时

大约是  $65535/3$  个=2 万多个字符

gbk 时

大约是  $65535/2$  个=3 万多个字符

### 3.2.2 占硬盘空间

- **char：定长字符串**

char 也叫做定长字符串，指的是在创建表时，char 字段占用硬盘空间的大小就已经固定了。比如，我们定义 name char(10)，代表 name 字段将占 10 个字符的硬盘空间，那么具体是多少个字节要看字段是什么编码的，如果是 utf8 编码，那么一个字符占 3 个字节，所以 char(10) 将占 30 个字节，无论内容够不够 10 个字节都占 30 个字节，比如只存 abc 三个字符也要占 30 个字节。

- **varchar：变长字符串**



**varchar** 也叫做变长字符串，指的是字段占用硬盘空间的大小并不是固定的，而是由内容决定的，等于内容的长度+1 个字节。

比如，我们定义 **name varchar(10)**，代表最多保存 10 个字符，如果表是 **utf8** 编码，则最多占用空间 30 个字节，但是，如果内容不够 10 个字节，比如内容是 **abc**，那么只占三个字符 9 字节再+1，共 10 占 10 个字节。

#### ● 对比

表是 **utf8** 编码：

**name char(10)**          如果内容是 **abc**，占用硬盘空间为 30 个字节。

**name varchar(10)**      如果内容是 **abc**，占用硬盘空间为 10 个字节。

总结：

1. **char** 定义时是多少，就占多少，无论内容多长。
2. **varchar** 定义时定义的是最大长度，而实际占有的空间由内容的长度+1 决定，--》节省空间

如何选择使用哪个？

1. 如果字符串的长度都是固定的可以用 **char**
2. 如果字符串内容的长度不固定，比如商品名称，每件商品的名字长度都不同，所以可变确定，这时使用 **varchar**

### 3.3 char、varchar 和 text 应该如何选择

选择的原则：

1. **varchar**：长度不固定时使用，比如商品名称、用户名称、文章标题、家庭住址、简短的评论等。
2. **char**：长度固定时使用，应用场景较少，比如 IP 地址、成语、MD5 之后的密码等。
3. **text**：大文本时使用，比如：文章的内容、小说的内容、论文、商品描述、帖子内容等。



## 4. 枚举

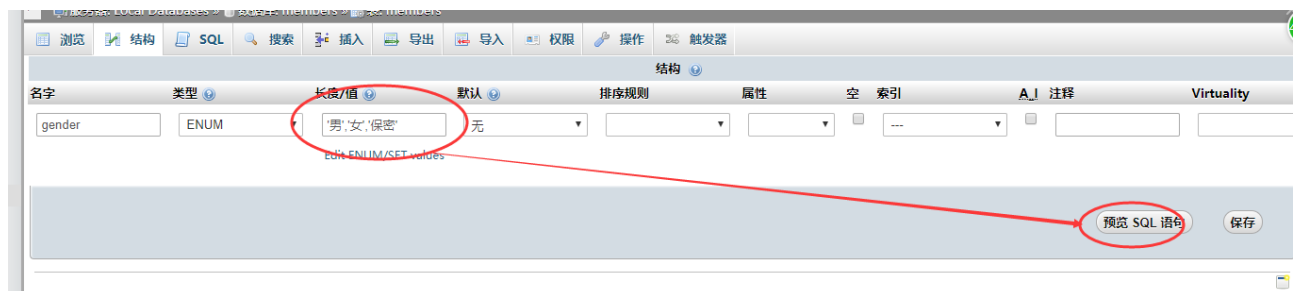
### 4.1 什么是枚举？

枚举类型【enum】，在定义字段时就预告规定好固定的几个值，然后插入记录时值只能这几个固定好的值中选择一个。

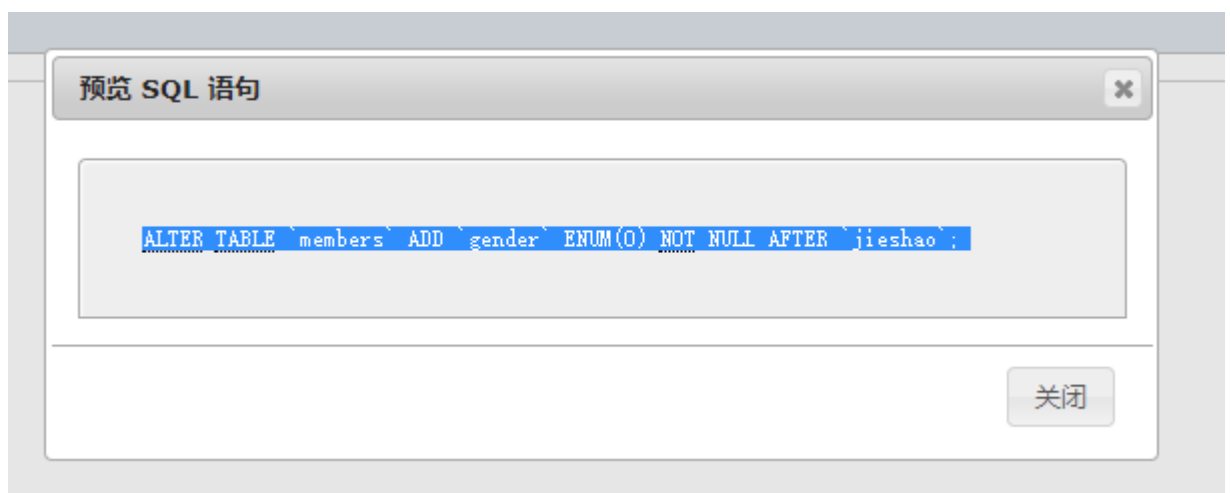
语法：

```
gender enum('男','女','保密')
```

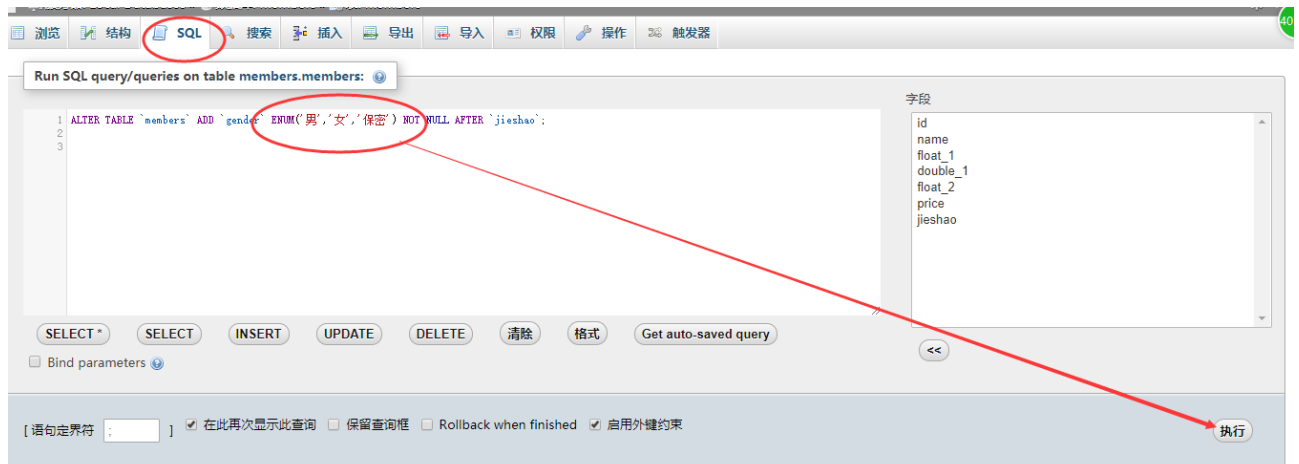
### 4.2 phpMyAdmin 中使用 enum



查看 SQL 语句并复制



到 SQL 面板中粘贴并修改执行：



这个字段里面的值，只能是这三个值之一。

## 4.3 特点

1. 一个 enum 最多可以设置 65535 个值
2. 这个字段最终只占 1 到 2 个字节【要看设置值的多少】【节省空间】

## 4.4 应用场景

当值是几个固定可选时，比如：性别、星期、月份、表示状态时【比如是、否】。

## 5. 集合

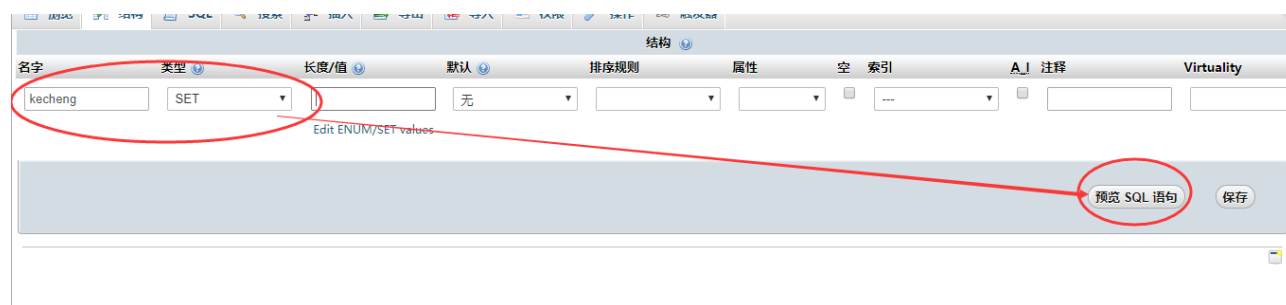
### 5.1 什么是集合

集合【set】，在定义字段时预先规定好几个值，然后在添加记录时可以从选出多个做为值，多个值之间用,隔开。

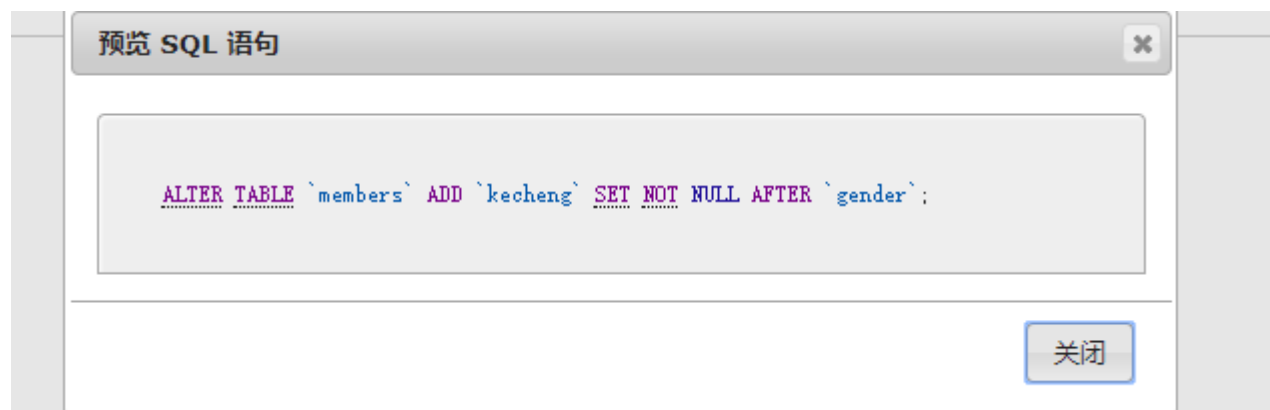
语法：

```
字段名 set ('值 1', '值 2', '值 3' ....)
```

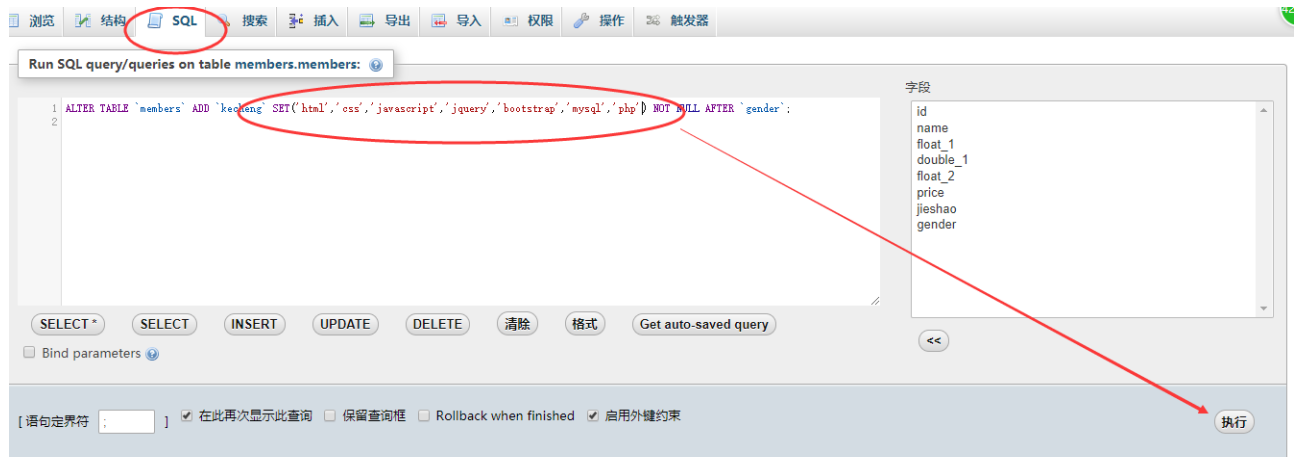
### 5.2 在 phpMyAdmin 中设置集合



复制 SQL:



粘贴到 SQL 面板中并修改执行：



Local Databases > 数据库: members > 表: members

#	名字	类型	排序规则	属性	空	默认	注释
1	id	int(10)		UNSIGNED	否	无	
2	name	varchar(30)	utf8_general_ci		否	无	
3	float_1	float(3,2)			否	无	
4	double_1	double			否	无	
5	float_2	float			否	无	
6	price	decimal(10,2)			否	无	
7	jieshao	varchar(300)	utf8_general_ci		否	无	
8	gender	enum('男', '女', '保密')	utf8_general_ci		否	无	
9	kecheng	set('html', 'css', 'javascript', 'jquery', 'bootst...	utf8_general_ci		否	无	

添加记录时可以从这些值里面选择多个:

选项

	id	name	float_1	double_1	float_2	price	jieshao	gender	kecheng
<input type="checkbox"/>  编辑  复制  删除	1	比尔盖	6565.23	9843.54	0	0.00		男	
<input type="checkbox"/>  编辑  复制  删除	2	孙司空	3.56	333	8888900	9.99		保密	html,css,javascript,jquery,bootstrap,php

↑

☐ 全选

选中项:

 编辑  复制  删除  导出

## 5.3 特点

1. 集合中最多定义 64 个值

2. 占硬盘空间数量为：1，2，3，4 或者 8 个字节，由值的个数决定

## 6. 时间类型

MySQL 提供了几个专门用来保存时间的类型。

### 6.1 几种时间类型

字段名称	格式
date	年-月-日 。 比如 2010-10-10
datetime	年-月-日 时:分:秒 。 比如：2010-10-10 10:10:10
timestamp	年-月-日 时:分:秒 。 比如：2010-10-10 10:10:10
time	时:分:秒 。 比如：10:10:10
year	年 。 比如：2010

添加一个日期字段：

允许为空

可以在表中保存日期：

id	name	float_1	double_1	float_2	price	jieshao	gender	kecheng	birthday
1	比尔盖	6565.23	9843.54	0	0.00		男		2017-08-14
2	孙司空	3.56	333	8888900	9.99		保密	html,css,javascript,jquery,bootstrap,php	2017-08-10

## 6.2 datetime 和 timestamp 的区别

### 1. 范围不同

datetime 保存时间的范围: '1000-01-01 00:00:00' 到 '9999-12-31 23:59:59'

timestamp 保存时间的范围: '1970-01-01 00:00:01' 到 '2038-01-19 03:14:07'

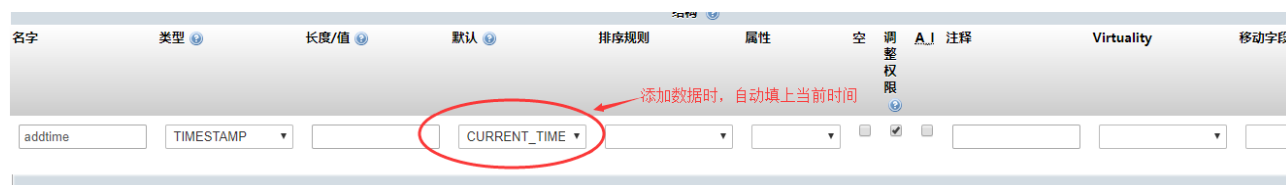
### 2. 存储空间不同

Data Type	Storage Required Before MySQL 5.6.4	Storage Required as of MySQL 5.6.4
<u>YEAR</u>	1 byte	1 byte
<u>DATE</u>	3 bytes	3 bytes
<u>TIME</u>	3 bytes	3 bytes + fractional seconds storage
<u>DATETIME</u>	8 bytes	5 bytes + fractional seconds storage
<u>TIMESTAMP</u>	4 bytes	4 bytes + fractional seconds storage

## 6.3 自动当前时间

可以在默认属性中设置 current\_timestamp, 这时, 这个字段的值会被自动添加入当前时间。

只对 timestamp 和 datetime 这两种时间类型有效。





## 7. 字段属性

每个字段除了可以设置名称、数据类型之外还可以设置其他信息，比如属性。

### 7.1 默认值

可以为每个字段设置一个默认值，当没有为这个字段设置值时，就用这个默认值填充：

名字	类型	长度/值	默认	排序规则
age	INT	11	定义 : 10	

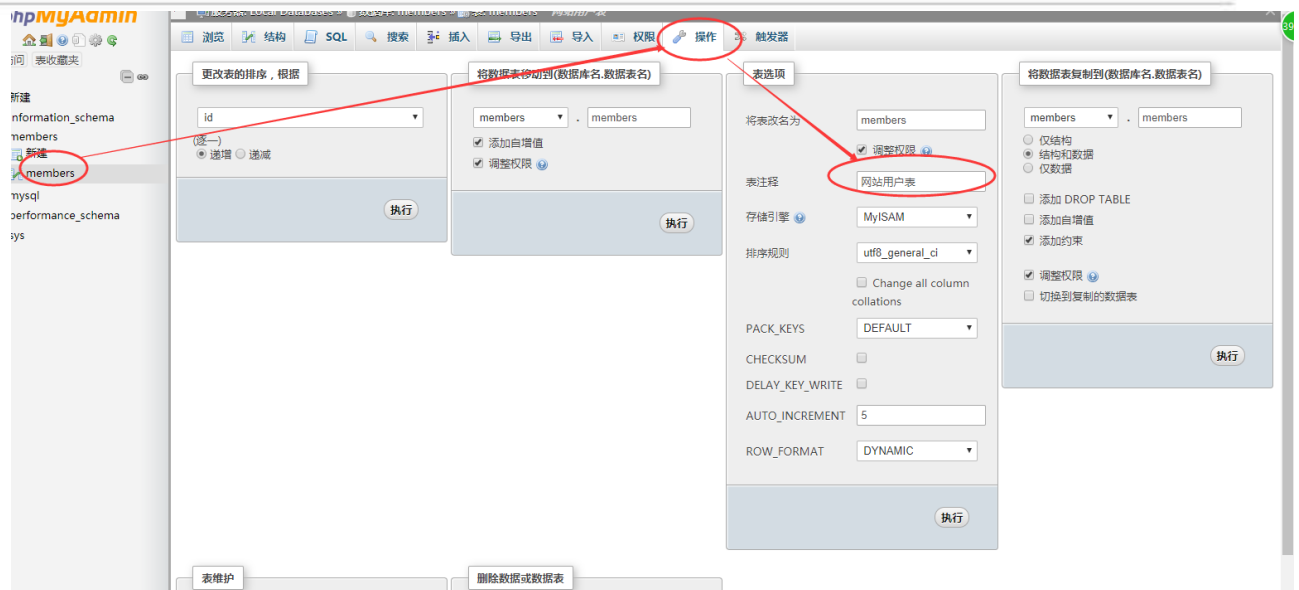
默认10岁

### 7.2 注释

每个字段可以使用注释对字段进行注解说明：

名字	类型	长度/值	默认	排序规则	属性	空	调整权限	A.I	注释	Virtuality
age	INT	11	定义 : 10						年龄字段	

可以为表也添加注释：



## 7.3 是否为 null

可以设置一个值中的值是否可以 null 【空】:

勾选: 可以为 null。

不勾选: 不允许为 null。





## 8. 编码和校对规则

不同的编码支持不同的字符集，所以要根据表中要保存的数据来选择编码。

### 8.1 支持中文的编码

gb2312: 简体中文。

big5: 繁体中文。

GBK: 简体+繁体中文。

utf8: 万国码，包含所有其他字符集。

优点：包含所有字符集。

缺点：同样的字符占的硬盘空间更大。

我们一般选择 utf8 编码。

### 8.2 库、表、字段的编码

可以分别为库、表和每个字段设置不同的编码和校对规则，一般我们都设置为统一的。

技术：当只有库设置编码时，库中所有表和字段默认和库编码一致。

- 库可以设置编码：



新建数据库

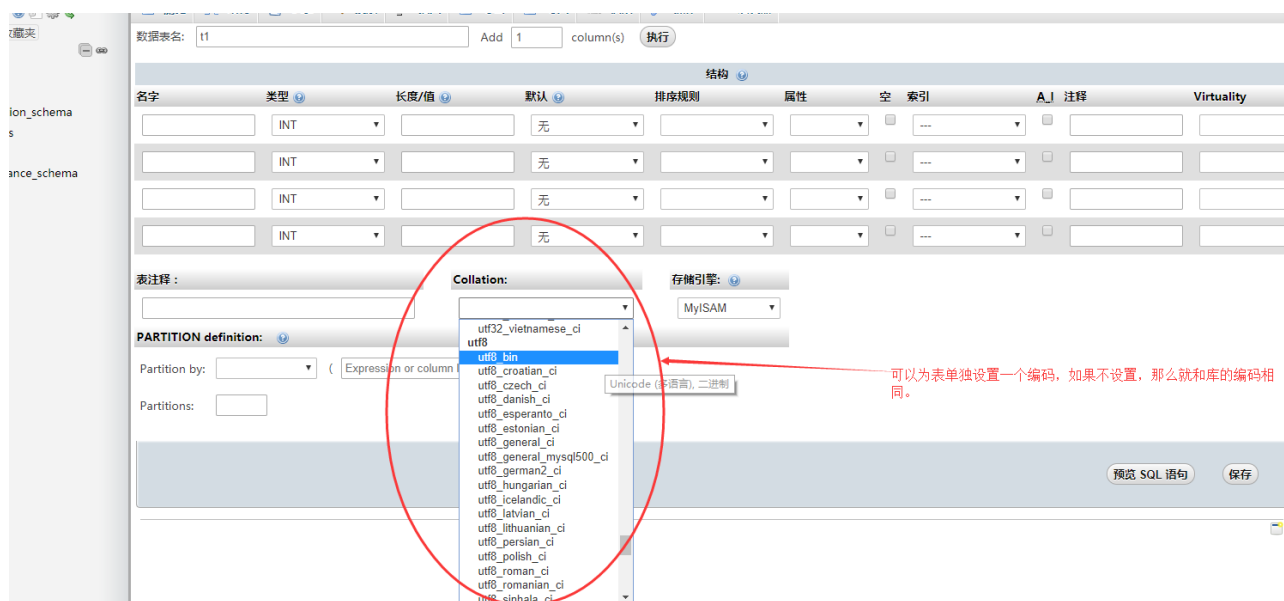
数据库名

排序规则

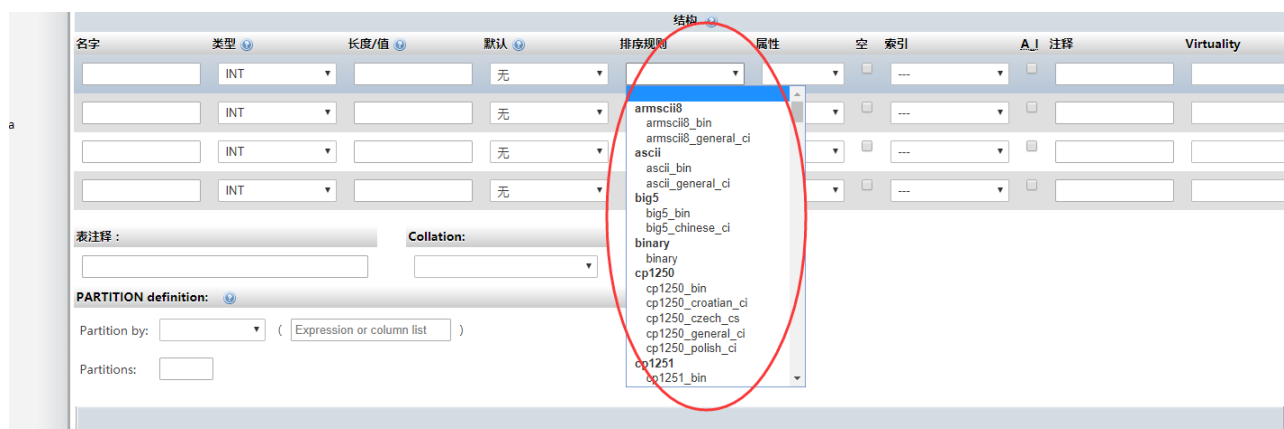
创建

数据库	排序规则	操作
-----	------	----

- 表也可以设置编码，如果不设置就和库的编码相同：



- 表中每个字段还可以单独设置一个编码，如果不设置就和表的编码一致



总结：首先我们为库设置一个编码，这时库中所有的表和字段将都是这种编码，如果有某个表或者某个字段需要设置不同的编码，那么可以单独为其设置。

## 8.3 校对规则

校对规则就是排序、比较时的规则，常用的两个：

utf8\_general\_ci: 区分大小写，比如 a 等于 A。

utf8\_bin: 区分大小写，比如 a 并不等于 A。

## 9. 表的尺寸限制

MySQL 中规定一个表中所有字段的尺寸加到一起不能超过 65535 个字节。

### 9.1 计算表的字节

表名: members    编码: utf8

n1 tinyint	--->	1	
n2 int	--->	4	
n3 varchar(200)	--->	600	
n4 varchar(2000)	--->	60000	
n5 varchar(2000)	--->	6000	<---失败无法创建这个字段了，尺寸超了

### 9.2 在表中保存大字段

当表中需要使用超过 65535 个字节时，应该把 varchar 改成 text。因为 text 字段比较特殊，它中的数据并不是保存在表中的，无论 text 中有多少的数据，它都只占这个表的 9~12 个字节，所以一个表中可以有大量的 text，当然注意 text 速度比较慢，当不保存大字符串时不要使用。

## 10. 案例、CMS 系统建表

创建两张表，表名分别是：authors【作家表】、articles【文章表】。

author【作家表】:

字段名称	字段类型	字段含义	是否为空	是否主键	备注
id	int (10)	编号	否	是	自动增长
username	varchar (50)	作家账号名称			
passwd	char (32)	密码			md5,32 位
user_type	enum('个人','媒体')	作家类型			个人、媒体
author_domain	varchar (255)	作家领域			
nicheng	varchar (50)	作家昵称			
head_img	varchar (255)	作家头像			图片名称
personal_introduction	varchar (255)	作家签名			
province	varchar (25)	所在省			
city	varchar (30)	所在市			
county	varchar (30)	所在县			
phone_num	bigint	手机号码			
invite_code	varchar (64)	邀请码			
author_state	tinyint (1)	作者状态			1-正常 0-禁用 2-已经选择完作家类型 3-已经完善作家信息
operator_name	varchar (50)	运营者姓名			
operator_ids	char (18)	运营者身份证			
operator_img	varchar (255)	运营者照片			
add_time	datetime	添加时间			
last_update_time	datetime	最后更新时间			
last_update_user	varchar (50)	最后更新用户			

articles【文章表】:

字段名称	字段类型	字段含义	是否为空	是否主键	备注
------	------	------	------	------	----

id	int (10)	编号	否	是	自动增长
article_type_id	int (10)	文章类型 id			
author_id	tinyint (10)	作者编号			
title	varchar(255)	标题			
description	varchar(255)	文章简述			
article_content	text	文章正文内容			
tags	varchar(255)	标签			
article_img	varchar(255)	文章图片			
video_url	varchar(255)	视频链接			
read_num	mediumint	阅读次数			
favorable_num	mediumint	好评次数			
difference_assessment	mediumint	差评次数			
add_time	datetime	添加时间			
state	tinyint (1)	文章状态			
is_top	tinyint (1)	是否置顶			
last_update_time	datetime	最后更新时间			
last_update_user	varchar (50)	最后更新人			

## 11. 今日总结

