

目录

- 1. 数据类型: ..... 1
  - 1.1 常见信息种类..... 1
  - 1.2 字符类型..... 1
  - 1.3 数值类型..... 2
    - 1.3.1 整数型: 只能存整数..... 2
    - 1.3.2 浮点型: 存储有小数点的数..... 3
  - 1.4 日期时间类型..... 3
    - 1.4.1 类型格式..... 3
    - 1.4.2 时间函数..... 4
    - 1.4.3 日期时间字段 datetime 与 timestamp 的区别..... 5
    - 1.4.4 year 类型..... 5
  - 1.5 枚举类型..... 6
    - 1.5.1 enum 单选 ..... 6
    - 1.5.2 set 多选..... 6
  - 1.6 课后练习..... 7

## 1. 数据类型：

### 1.1 常见信息种类

数值型：体重、身高、成绩、工资

字符型：姓名、工作单位、通信地址

枚举型：兴趣爱好、性别、专业

日期时间型：出生日期、注册时间

### 1.2 字符类型

- 定长：char(字符个数)
  - ✓ 最大字符个数255
  - ✓ 不够指定字符个数时在右边用空格补全
  - ✓ 字符个数超出时，无法写入数据。
- 变长：varchar(字符个数)
  - ✓ 按数据实际大小分配存储空间
  - ✓ 字符个数超出时，无法写入数据。
- 大文本类型：text/blob
  - ✓ 字符数大于65535存储时使用

```
mysql> create table t1(name char(5), email varchar(15));
mysql> desc t1;
mysql> create table t2( name char, email varchar(3) );    #char类型不指存储几个字符，
默认存储一个
mysql> desc t2;
mysql> insert into t2 values("a", "bac");                #成功
mysql> insert into t2 values("aa", "bacd");               #失败
```

```
mysql> insert into t2 values("b", "bacd"); #失败
mysql> select * from t2;
```

## 1.3 数值类型

### 1.3.1 整型：只能存整数

类 型	名称	有符号范围	无符号范围
tinyint	微小整数	-128~127	0 ~ 255
smallint	小整数	-32768~32767	0 ~ 65535
mediumint	中整型	$-2^{23} \sim 2^{23}-1$	$0 \sim 2^{24}-1$
int	大整型	$-2^{31} \sim 2^{31}-1$	$0 \sim 2^{32}-1$
bigint	极大整型	$-2^{63} \sim 2^{63}-1$	$0 \sim 2^{64}-1$
unsigned	使用无符号存储范围		

创建一张表 t3，用于存储学生信息(用户名，年龄，等级)，tinyint 类型，unsigned 无符号存储(0~255)，默认有符号存储(-128~127)

```
mysql> create table t3(name char(15), age tinyint unsigned, level tinyint);
mysql> insert into t3 values("bob", 21, 7); #成功
mysql> insert into t3 values("tom", -1, -129); #失败，条件不满足
mysql> insert into t3 values("tom", 0, -129); #失败，超出范围
mysql> insert into t3 values("tom", 0, -127); #成功
存储小数，会四舍五入
mysql> insert into t3 values("jim", 21.5, 3); #21.5四舍五入存为22
mysql> select * from t3;
mysql> insert into t3 values("jim", 21.5, 3.43); #3.43四舍五入存为3
mysql> select * from t3;
```

### 1.3.2 浮点型：存储有小数点的数

类 型	名 称	有符号范围	无符号范围
float	单精度	-3.402823466E+38 到 -1.175494351E-38	1.175494351E-38 到 3.402823466E+38
double	双精度	-1.7976931348623157E+308到 -2.2250738585072014E-308	2.2250738585072014E-308 到 1.7976931348623157E+308

float(7,2) #7 指整个浮点数的最大位数，2 指 7 位数字中有两位是小数位，则取值范围 为：  
-99999.99 ~ 99999.99

float(5,3) #5 指整个浮点数的最大位数，3 指 5 位数字中有三位是小数位，则取值范围为： -  
99.999 ~ 99.999

float(数字 1,数字 2) #数字 1： 总的位数      数字 2： 小数位的个数

```
mysql> create table t4(name char(10), pay float(5,2));
mysql> insert into t4 values("john", 1000.88);           #失败，超出范围
mysql> insert into t4 values("john", 999.88);             #成功
mysql> insert into t4 values("john", -999.99);            #成功
mysql> select * from t4;
mysql> insert into t4 values("john3", 218);               #存储整数，小数位默认补0
```

## 1.4 日期时间类型

### 1.4.1 类型格式

**日期 date**  
✓ 范围：0001-01-01 ~ 9999-12-31  
✓ 格式：yyyymmdd 19491001

**年 year**  
✓ 范围：1901~2155  
✓ 格式：yyyy 1998 2000

**时间 time**  
✓ 格式：HH:MM:SS

**日期时间 datetime**  
✓ 范围：1000-01-01 00:00:00~ 9999-12-31 23:59:59  
✓ 格式：yyyymmddhhmmss

**日期时间 timestamp**  
✓ 范围：1970-01-01 00:00:00 ~ 2038-01-19 00:00:00  
✓ 格式：yyyymmddhhmmss

创建与日期时间相关的表，指定名称，年份，上课时间，生日，聚会时间

```
mysql> create table t5(name char(15), s_year year, uptime time, birthday date, party
datetime);
mysql> insert into t5 values("bob", 1990,083000, 20231120, 20230214183000);
mysql> select * from t5;
```

#### 1.4.2 时间函数

- 函数服务内置的命令，可以使用时间函数给字段赋值

类 型	用 途
curtime( )	获取当前的系统时间
curdate( )	获取当前的系统日期
now( )	获取当前系统日期和时间
year( )	获取年
month( )	获取月
day( )	获取日
date( )	获取日期
time( )	获取时间

```
mysql> select curtime();           #获取当前的系统时间
mysql> select curdate();          #获取当前的系统日期
```

```
mysql> select now();           #获取当前的系统日期和系统时间
mysql> select year(now());     #从当前系统时间中只取出年份
mysql> select month(now());    #从当前系统时间中只取出月份
mysql> select day(now());      #从当前系统时间中只取出天数
mysql> select date(now());     #从当前系统时间中只取出年月日
mysql> select time(now());     #从当前系统时间中只取出时分秒
```

根据时间函数在 t5 表中插入一条数据

```
mysql> insert into t5 values("tom",2000,time(now()),curdate(),now());
mysql> select * from t5;
```

### 1.4.3 日期时间字段 datetime 与 timestamp 的区别

关于日期时间字段：当未给 timestamp 字段赋值时，自动以当前系统时间赋值，而 datetime 值为 NULL（空）

创建 t6 表，指定姓名，约会时间，聚会时间，验证 timestamp 和 datetime 的区别

```
mysql> create table t6(name char(10), meeting datetime, party timestamp);
mysql> insert into t6 values("bob", now(), now());           #两个字段都有值
mysql> select * from t6;
```

t6 表中重新插入一条数据，只插入 name 和 meeting 字段的值，party 字段采用默认值

```
mysql> insert into t6(name,meeting) values("bob", 20231120224058);
mysql> select * from t6;  #party字段同样有值，字段类型为timestamp，用当前系统时间
```

t6 表中重新插入一条数据，只插入 name 和 party 字段的值，meeting 字段采用默认值

```
mysql> insert into t6(name,party) values("john", 19731001223000 );
mysql> select * from t6;  # meeting字段类型为datetime，没有指定时间，默认为空
NULL)
```

### 1.4.4 year 类型

要求使用 4 位赋值

当使用 2 位数赋值时: 01-99

01 ~ 69 视为 2001 ~ 2069

70 ~ 99 视为 1970 ~ 1999

插入数据, 只给 t5 表中的 s\_year 字段赋值

```
mysql> show tables;
mysql> desc t5;
mysql> select s_year from t5;
mysql> insert into t5(s_year) values(03),(81);
mysql> select s_year from t5;
```

#查看t5表中s\_year字段的数据, 验证结果

## 1.5 枚举类型

字段的值不能自己输入, 必须在设置的范围内选择(有单选和多选之分)

### 1.5.1 enum 单选

格式: 字段名 enum (值 1, 值 2, 值 N)

仅能在列表里选择一个值

### 1.5.2 set 多选

格式: 字段名 set(值 1, 值 2, 值 3)

在列表里选择一个或多个值

创建 t7 表, 指定字段: 姓名(name), 性别(sex), 爱好(likes)

```
mysql> create table t7(name char(15), sex enum("boy", "girl", "no"), likes set("eat",
"money", "game", "music"));
mysql> desc t7;
mysql> insert into t7 values('bob','boy','eat,game,music');
mysql> select * from t7;
```

#成功

```
mysql> insert into t7 values('bob','man','girl,book');      #字段sex的类型中没有man,
存储失败
mysql> insert into t7 values('bob','no','girl,book');      #字段likes的类型中没有girl和
book,存储失败, 使用类型enum(单选), set(多选), 值必须在其范围之内
```

## 1.6 课后练习

创建一个员工信息表为 stuinfo, 表字段要求如下: (对应字段的数据类型自己考虑, 表字段顺序没要求)

--员工 ID 号, 姓名, 年龄, 性别, 身高, 出生日期, 邮箱, 手机号, 入职时间, 爱好, 通信地址, 学历, 婚姻, 工资, 上班时间