



链滴

Mysql 数据库入门基础

作者: [rxz](#)

原文链接: <https://ld246.com/article/1505901579631>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

Mysql 烦断肠的字符集/反复安装失败

作为一个新手在操作数据库的时候，最烦的就是安装数据库，mysql相对来说，比较简单，不象oracle也许你安装坏了，可能要重做系统。但是mysql在安装时候最烦的就是字符集不统一的问题和安装失败，再次安装怎么都安不上。

一、对于解决mysql反复安装失败的问题

1、重新点击安装程序，因为安装过一次，所以这个程序就会出现**修复**和**卸载**两个选项我们可以选择**卸载**。

2、如果上一步，未解决，就需要去C盘，和安装目录下手动删除mysql的文件夹，然后再次执行1中操作。

以上两步操作足够制服安装失败的问题，安装过程中也许会出现端口占用的问题，此时修改一下3306口就可以，以后再使用的时候记住自己设置的端口。

二、对于解决mysql字符集不统一导致乱码的问题

mysql的字符集有很多，数据表的/数据库的/客户端的/服务器的/文件系统的。。。等等，因为我们文要使用gbk/gbk132/utf-8的字符集，但是数据库默认的不是

1、第一个设置字符集的地方在安装数据库的时候，这一步最为关键，设置的是默认字符集。

2、第二个在创建数据库的时候

3、第三个在创建表的时候【不过一般不会写，会跟随默认的字符集】

如果以上两步都没有设置，没关系，以下操作轻松搞定字符集

1、打开mysql的终端查询字符集设置：

```
mysql> SHOW VARIABLES LIKE '%char%';
+-----+-----+
| Variable_name      | Value                                     |
+-----+-----+
| character_set_client | utf8                                     |
| character_set_connection | utf8                                     |
| character_set_database | utf8                                     |
| character_set_filesystem | binary                                   |
| character_set_results | utf8                                     |
| character_set_server | utf8                                     |
| character_set_system | utf8                                     |
| character_sets_dir   | C:\Program Files\MySQL\MySQL Server 5.6\share\charsets\ |
+-----+-----+
```

2、所有的字符集列表，**直接使用一下语句修改**，如下：

```
mysql> SET character_set_client = utf8 ;
mysql> SET character_set_connection = utf8 ;
mysql> SET character_set_database = utf8 ;
mysql> SET character_set_results = utf8 ;
mysql> SET character_set_server = utf8 ;
mysql> SET collation_connection = utf8 ;
mysql> SET collation_database = utf8 ;
mysql> SET collation_server = utf8 ;
```

Mysql 常见的运算符

mysql中同样提供了常用的运算符，用于数据的运算，以下内容来源

1、算数运算符

算数运算符常用在对数值数据做算术运算，例如：一个要给员工发薪水，我们记录的是考勤和基本薪：user_info这张表存储了基本工资base_salary，考勤天数attendance_day，应上班天数all_day,员的薪资就是 base_salary*(attendance_day/all_day)此时算术运算就起到了作用。

```
mysql> select 1+2; 加
mysql> select 2-1; 减
mysql> select 2*3; 乘
mysql> select 5/3; 除
mysql> SELECT 5 DIV 2; 商
mysql> select 5%2,mod(5,2);模
```

2、比较运算符

比较运算符经常会用到，在查询的时候，用做条件的判断，范围的判断，都要用到比较运算符。

等于

```
mysql> select 1=0,1=1,null=null;
```

不等于

```
mysql> select 1<>0,1<>1,null<>null;
```

完全等于

```
mysql> select 1<=>1,2<=>0,0<=>0,null<=>null;
```

小于

```
mysql> select 'a'<'b','a'<'a','a'<'c',1<2;
```

小于等于

```
mysql> select 'bdf'<='b','b'<='b',0<1;
```

大于

```
mysql> select 'a'>'b','abc'>'a',1>0;
```

大于等于

```
mysql> select 'a'>='b','abc'>='a',1>=0,1>=1;
```

BETWEEN AND :在某个范围内

```
mysql> select 10 between 10 and 20, 9 between 10 and 20;
```

IN : 在某个范围内

```
mysql> select 1 in (1,2,3), 't' in ('t','a','b','l','e'), 0 in (1,2);
```

IS NULL

```
mysql> select 0 is null,null is null;
```

IS NOT NULL

```
mysql> select 0 is not null, null is not null;
```

LIKE : 模糊匹配

```
mysql> select 123456 like '123%', 123456 like '%123%', 123456 like '%321%';
```

REGEXP :正则匹配

```
mysql> select 'abcdef' regexp 'ab', 'abcdefg' regexp 'k'; -- 1 0 表示abcdef中匹配到ab一次, abc  
efg匹配到k 0次
```

** 3、逻辑运算符 **

非

```
mysql> select not 0, not 1, not null; -- 1 0 (null)
```

```
mysql> select ! 0, ! 1, ! null; -- 1 0 (null)
```

与

```
mysql> select (1 and 1), (0 and 1), (3 and 1), (1 and null);
```

```
mysql> select (1 && 1), (0 && 1), (3 && 1), (1 && null);
```

或

```
mysql> select (1 or 0), (0 or 0), (1 or null), (1 or 1), (null or null);
```

```
mysql> select (1 || 0), (0 || 0), (1 || null), (1 || 1), (null || null);
```

异或

```
mysql> select (1 xor 1), (0 xor 0), (1 xor 0), (0 xor 1), (null xor 1);
```

```
mysql> select (1 ^ 1), (0 ^ 0), (1 ^ 0), (0 ^ 1), (null ^ 1);
```

** 4、位运算符**

位与

```
mysql> select 2&3;
```

```
mysql> select 2&3&4;
```

位或

```
mysql> select 2|3;
```

位异或

```
mysql> select 2^3;
```

位取反

```
mysql> select ~1,~18446744073709551614;
```

位右移

```
mysql> select 100>>3;
```

位左移

```
mysql> select 100<<3;
```

** 5.运算符优先级顺序 **

最高优先级 :=

1 ||, OR, XOR

2 &&, AND

3 BETWEEN, CASE, WHEN, THEN, ELSE

4 =, <=>, >=, >, <=, <, <>, !=, IS, LIKE, REGEXP, IN

5 |

6 &

7 <<, >>

8 -, +

9 *, /, DIV, %, MOD

10 ^

11 - (unary minus), ~ (unary bit inversion)

12 !, NOT

最低优先级 BINARY, COLLATE

常用的必须熟悉，不常用的，暂时记者，也许有一天就有用了。。

Mysql 常用的函数

MySQL数据库中提供了很丰富的函数。MySQL函数包括数学函数、字符串函数、日期和时间函数、件判断函数、系统信息函数、加密函数、格式化函数等。通过这些函数，可以简化用户的操作。以下
[容来源](#)

**一、数学函数

ABS(x) 返回x的绝对值
BIN(x) 返回x的二进制 (OCT返回八进制, HEX返回十六进制)
CEILING(x) 返回大于x的最小整数值
EXP(x) 返回值e (自然对数的底) 的x次方
FLOOR(x) 返回小于x的最大整数值
GREATEST(x1,x2,...,xn)返回集合中最大的值
LEAST(x1,x2,...,xn) 返回集合中最小的值
LN(x) 返回x的自然对数
LOG(x,y)返回x的以y为底的对数
MOD(x,y) 返回x/y的模 (余数)
PI()返回pi的值 (圆周率)
RAND()返回 0 到 1 内的随机值,可以通过提供一个参数(种子)使RAND()随机数生成器生成一个指定的。
ROUND(x,y)返回参数x的四舍五入的有y位小数的值
SIGN(x) 返回代表数字x的符号的值
SQRT(x) 返回一个数的平方根
TRUNCATE(x,y) 返回数字x截短为y位小数的结果

二、聚合函数(常用于GROUP BY从句的SELECT查询中)

AVG(col)返回指定列的平均值
COUNT(col)返回指定列中非NULL值的个数
MIN(col)返回指定列的最小值
MAX(col)返回指定列的最大值
SUM(col)返回指定列的所有值之和
GROUP_CONCAT(col) 返回由属于一组的列值连接组合而成的结果

三、字符串函数

ASCII(char)返回字符的ASCII码值
BIT_LENGTH(str)返回字符串的比特长度
CONCAT(s1,s2...,sn)将s1,s2...,sn连接成字符串
CONCAT_WS(sep,s1,s2...,sn)将s1,s2...,sn连接成字符串,并用sep字符间隔
INSERT(str,x,y,instr) 将字符串str从第x位置开始, y个字符长的子串替换为字符串instr, 返回结果
FIND_IN_SET(str,list)分析逗号分隔的list列表, 如果发现str, 返回str在list中的位置
LCASE(str)或LOWER(str) 返回将字符串str中所有字符改变为小写后的结果
LEFT(str,x)返回字符串str中最左边的x个字符
LENGTH(s)返回字符串str中的字符数
LTRIM(str) 从字符串str中切掉开头的空格
POSITION(substr,str) 返回子串substr在字符串str中第一次出现的位置
QUOTE(str) 用反斜杠转义str中的单引号
REPEAT(str,srchstr,rplcstr)返回字符串str重复x次的结果
REVERSE(str) 返回颠倒字符串str的结果
RIGHT(str,x) 返回字符串str中最右边的x个字符
RTRIM(str) 返回字符串str尾部的空格
STRCMP(s1,s2)比较字符串s1和s2
TRIM(str)去除字符串首部和尾部的所有空格
UCASE(str)或UPPER(str) 返回将字符串str中所有字符转变为大写后的结果

四、日期和时间函数

CURDATE()或CURRENT_DATE() 返回当前的日期
CURTIME()或CURRENT_TIME() 返回当前的时间

DATE_ADD(date,INTERVAL int keyword)返回日期date加上间隔时间int的结果(int必须按照关键字行格式化),如: SELECT DATE_ADD(CURRENT_DATE,INTERVAL 6 MONTH);

DATE_FORMAT(date,fmt) 依照指定的fmt格式格式化日期date值

DATE_SUB(date,INTERVAL int keyword)返回日期date加上间隔时间int的结果(int必须按照关键字行格式化),如: SELECT DATE_SUB(CURRENT_DATE,INTERVAL 6 MONTH);

DAYOFWEEK(date) 返回date所代表的一星期中的第几天(1~7)

DAYOFMONTH(date) 返回date是一个月的第几天(1~31)

DAYOFYEAR(date) 返回date是一年的第几天(1~366)

DAYNAME(date) 返回date的星期名, 如: SELECT DAYNAME(CURRENT_DATE);

FROM_UNIXTIME(ts,fmt) 根据指定的fmt格式, 格式化UNIX时间戳ts

HOUR(time) 返回time的小时值(0~23)

MINUTE(time) 返回time的分钟值(0~59)

MONTH(date) 返回date的月份值(1~12)

MONTHNAME(date) 返回date的月份名, 如: SELECT MONTHNAME(CURRENT_DATE);

NOW() 返回当前的日期和时间

QUARTER(date) 返回date在一年中的季度(1~4), 如SELECT QUARTER(CURRENT_DATE);

WEEK(date) 返回日期date为一年中第几周(0~53)

YEAR(date) 返回日期date的年份(1000~9999)

```
返回当前日期: SELECT FROM_UNIXTIME(UNIX_TIMESTAMP());
mysql> SELECT FROM_UNIXTIME(UNIX_TIMESTAMP());
+-----+
| FROM_UNIXTIME(UNIX_TIMESTAMP()) |
+-----+
| 2017-09-20 17:44:50           |
+-----+
```

五、加密函数

AES_ENCRYPT(str,key) 返回用密钥key对字符串str利用高级加密标准算法加密后的结果, 调用AES_DECRYPT的结果是一个二进制字符串, 以BLOB类型存储

AES_DECRYPT(str,key) 返回用密钥key对字符串str利用高级加密标准算法解密后的结果

DECODE(str,key) 使用key作为密钥解密加密字符串str

ENCRYPT(str,salt) 使用UNIXcrypt()函数, 用关键词salt(一个可以惟一确定口令的字符串, 就像钥一样)加密字符串str

ENCODE(str,key) 使用key作为密钥加密字符串str, 调用ENCODE()的结果是一个二进制字符串 它以BLOB类型存储

MD5() 计算字符串str的MD5校验和

PASSWORD(str) 返回字符串str的加密版本, 这个加密过程是不可逆转的, 和UNIX密码加密过使用不同的算法。

SHA() 计算字符串str的安全散列算法(SHA)校验

```
mysql> select md5('123213')
+-----+
| md5('123213')           |
+-----+
| 8ad3fac6c6b3528499d347d924443abb |
+-----+
```

** 六、MySQL控制流函数: **

CASE WHEN[test1] THEN [result1]...ELSE [default] END 如果testN是真, 则返回resultN, 否返回default

CASE [test] WHEN[val1] THEN [result]...ELSE [default]END 如果test和valN相等, 则返回resultN
否则返回default
IF(test,t,f) 如果test是真, 返回t; 否则返回f
IFNULL(arg1,arg2) 如果arg1不是空, 返回arg1, 否则返回arg2
NULLIF(arg1,arg2) 如果arg1=arg2返回NULL; 否则返回arg1

这些函数的第一个是IFNULL(), 它有两个参数, 并且对第一个参数进行判断。如果第一个参数不是NULL, 函数就会向调用者返回第一个参数; 如果是NULL,将返回第二个参数。

七、格式化函数

DATE_FORMAT(date,fmt) 依照字符串fmt格式化日期date值
FORMAT(x,y) 把x格式化为以逗号隔开的数字序列, y是结果的小数位数
INET_ATON(ip) 返回IP地址的数字表示
INET_NTOA(num) 返回数字所代表的IP地址
TIME_FORMAT(time,fmt) 依照字符串fmt格式化时间time值

```
mysql> SELECT DATE_FORMAT(NOW(),'%W,%D %M %Y %r');
+-----+
| DATE_FORMAT(NOW(),'%W,%D %M %Y %r') |
+-----+
| Wednesday,20th September 2017 05:54:02 PM |
+-----+
```

其中最简单的是FORMAT()函数, 它可以把大的数值格式化为以逗号间隔的易读的序列。

八、类型转化函数

为了进行数据类型转化, MySQL提供了CAST()函数, 它可以把一个值转化为指定的数据类型。类型: BINARY,CHAR,DATE,TIME,DATETIME,SIGNED,UNSIGNED

九、系统信息函数

DATABASE() 返回当前数据库名
BENCHMARK(count,expr) 将表达式expr重复运行count次
CONNECTION_ID() 返回当前客户的连接ID
FOUND_ROWS() 返回最后一个SELECT查询进行检索的总行数
USER()或SYSTEM_USER() 返回当前登陆用户名
VERSION() 返回MySQL服务器的版本

...待续。。。