



链滴

# MySQL 如何使主键大小写敏感

作者: [houjie](#)

原文链接: <https://ld246.com/article/1516798710390>

来源网站: 链滴

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

最近在日常使用中,遇到了一个问题.

在python里去重处理好了的数据,插入时却遇到了1064错误

1062 Duplicate entry 'xxx' for key primary

但是实际上是没有重复的. 又不想用ignore忽略这个错误.

遂仔细检查了一下数据,发现有有的数据在忽略大小写的情况下是一致的

即有 'AAA' 和 'aaa' 这样的数据

也就是说mysql的主键不是大小写敏感的

## 如何使MySQL的主键大小写敏感?

MySQL 默认查询是不区分大小写的,主键也是不区分大小写的.

如果需要区分,必须在建表的时候,使用**Binary**标示敏感的属性.

以下摘自 MySQL 5.7 官方文档 [10.1.10.8 The Binary Character Set](#)

### 10.1.10.8 The Binary Character Set

The **binary** character set is the character set of binary strings, which are sequences of bytes. The **binary** character set has one collation, also named **binary**. Comparison and sorting are based on numeric byte values. The effect is that lettercase and accent differences are significant in comparisons. That is, the **binary collation is case-sensitive and accent sensitive**.

(也就是说, **binary**排序规则是区分大小写和区分重音的。)

```
mysql> SET NAMES 'binary';
```

```
mysql> SELECT CHARSET('abc'), COLLATION('abc');
```

```
+-----+-----+
| CHARSET('abc') | COLLATION('abc') |
+-----+-----+
| binary        | binary          |
+-----+-----+
```

```
mysql> SELECT 'abc' = 'ABC', 'a' = 'ä';
```

```
+-----+-----+
| 'abc' = 'ABC' | 'a' = 'ä' |
+-----+-----+
| 0             | 0         |
+-----+-----+
```

For information about the differences between the **binary** collation of the **binary** character set and the **\_bin** collations of nonbinary character sets, see [Section 10.1.8.5, "The binary Collation Compared to \\_bin Collations"](#) .

To convert a string expression to a binary string, any of these constructs are equivalent:

```
BINARY expr
CAST(expr AS BINARY)
CONVERT(expr USING BINARY)
```

If `expr` is a character string literal, the `_binary` introducer may be used to designate it as a binary string. For example:

```
_binary 'a'
```

The `_binary` introducer is permitted for hexadecimal literals and bit-value literals as well, but unnecessary; such literals are binary strings by default.

## 解决方案 - 字段大小写敏感

### 1: 创建表时字段指定binary

```
create table table_name (  
id varchar(32) binary;  
)
```

### 2: 修改列指定校对规则

```
MODIFY COLUMN `x` varchar(100) BINARY CHARACTER SET utf8 COLLATE utf8_general_ci
```

## BINARY 与 COLLATE

mysql中控制数据库名和表名的大小写敏感由参数`lower_case_table_names`控制，\*\*为0时表示区分大小写，为1时，表示将名字转化为小写后存储，不区分大小写。\*\*字段名通常都是不区分大小写的，字段值呢？**字段值的大小写由mysql的校对规则来控制。**提到校对规则，就不得不说字符集。**字符集一套符号和编码，校对规则是在字符集内用于比较字符的一套规则**，比如定义'A' < 'B'这样的关系的规则。不同的字符集有多种校对规则，一般而言，校对规则以其相关的字符集名开始，通常包括一个语言，并且以`_ci`（大小写不敏感）、`_cs`（大小写敏感）或`_bin`（二元）结束。

比如 utf8字符集，

`utf8_general_ci`表示不区分大小写，这个是utf8字符集默认的校对规则；

`utf8_general_cs`表示区分大小写，

`utf8_bin`表示二进制比较，同样也区分大小写。

校对规则通过关键字`collate`指定，比如创建数据库d1，指定字符集为utf8，校对规则为utf8\_bin

```
CREATE DATABASE d1 DEFAULT CHARACTER SET utf8 COLLATE utf8_bin;
```

通过上述语句说明数据库d1中的数据按utf8编码，并且是对大小写敏感的。

## 解决方案 - 查询大小写敏感

有时候我们建库时，没有指定校对规则校对时字符大小写敏感，但是我们查询时，又需要对字符比较小写敏感，例如只想要ab打头的字符串。没关系，mysql提供了`collate`语法，通过指定`utf8_bin`校对规则即可。

```
select * from test where c1 like 'ab%' collate utf8_bin;
```

```
+-----+  
| c1 |
```

```
+-----+
| abc |
+-----+
```

这里还有另外一种方法，通过**binary**关键字，将串转为二进制进行比较，由于大小写字符的二进制肯不同，因此可以认为是区分大小的一种方式。

```
select * from test where binary c1 like 'ab%';
```

```
+-----+
| c1 |
+-----+
| abc |
+-----+
```

最后要说明一点的是校对规则与索引存储的关系。因为校对规则会用于字符串之间比较，而**索引是基比较有序排列的**，因此校对规则会影响记录的索引顺序。

## MySQL 其他大小写敏感

在MySQL的配置文件my.ini中增加一行：

```
lower_case_table_names = 0
```

其中 0：区分大小写，1：不区分大小写

MySQL在Linux下数据库名、表名、列名、别名大小写规则是这样的：

1. 数据库名与表名是严格区分大小写
2. 表的别名是严格区分大小写的；
3. 列名与列的别名在默认情况下均是忽略大小写的；
4. 变量名也是严格区分大小写的；

MySQL在Windows下都不区分大小写

## 参考资料

[[MySQL主键不区分大小写问题、查询不区分大小写问题](#)

](<http://blog.csdn.net/u011127019/article/details/51404862>)

[mysql大小写敏感与校对规则](#)

[Mysql大小写敏感的问题](#)