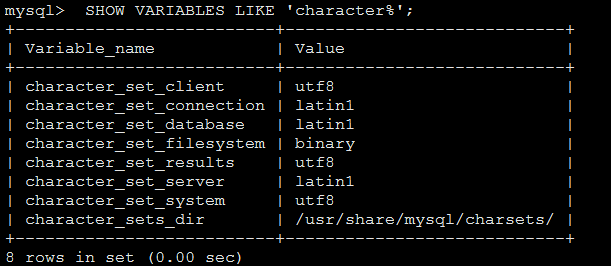
## 设置数据库编码为utf-8

<http://www.blogjava.net/qileilove/archive/2013/06/14/400541.html>

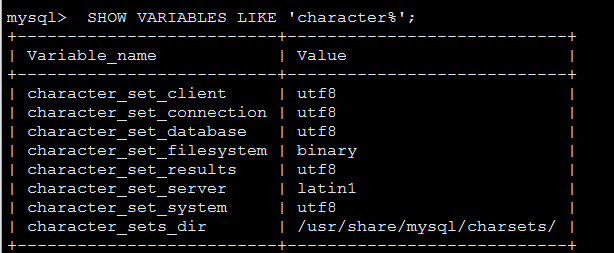
1.mysql> SHOW VARIABLES LIKE 'character%';



1. 设置编码命令(mysql服务重启后当前设置的编码规则失效)

|  |
| --- |
| SET character\_set\_client = utf8; SET character\_set\_connection = utf8; SET character\_set\_database = utf8; SET character\_set\_results = utf8; SET character\_set\_server = utf8; |

3.再进行查看即可



## 2.SET NAMES UTF8;

MySQL命令行输入“SET NAMES UTF8;”，然后执行“show variables like “character\_set\_%”;”，发现原来为latin1的那些变量“character\_set\_client”、“character\_set\_connection”、“character\_set\_results”的值全部变为utf8了，原来是这3个变量在捣蛋。查阅手册，上面那句等于：

1. SET character\_set\_client = utf8;
2. SET character\_set\_results = utf8;
3. SET character\_set\_connection = utf8;

看看这3个变量的作用：

信息输入路径：client→connection→server；  
信息输出路径：server→connection→results。

换句话说，每个路径要经过3次改变字符集编码。以出现乱码的输出为例，server里utf8的数据，传入connection转为latin1，传入results转为latin1，utf-8页面又把results转过来。如果两种字符集不兼容，比如latin1和utf8，转化过程就为不可逆的，破坏性的。所以就转不回来了。

但这里要声明一点，“SET NAMES UTF8”作用只是临时的，MySQL重启后就恢复默认了。

## 数据库表创建默认的引擎和编码

mysql> show create table user\_table;

