<http://blog.csdn.net/sunboyzsm/article/details/17511511>

* **问题引入：**

        我们经常会遇到一些向MySQL数据库中插入中文，但是select出来的时候，却发现是乱码的情况。如我们向表a出入这样一段记录：i

insert into a values('你好helloworld你好','helloworld');

可能当你访问它的时候，会发现他的结果变成如下图所示：



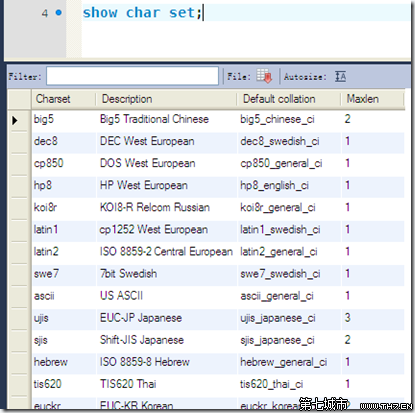
那怎么样才能解决这种问题呢？通过下文对MySQL中字符集的一些操作，你将会得到答案！

* **查看库、表字符集命令：**

        要解决字符集的问题，首先要知道现在的系统、数据库、表、客户端等使用什么样的字符集，以及系统支持什么字符集等，下面介绍一些获取相关信息的语句：

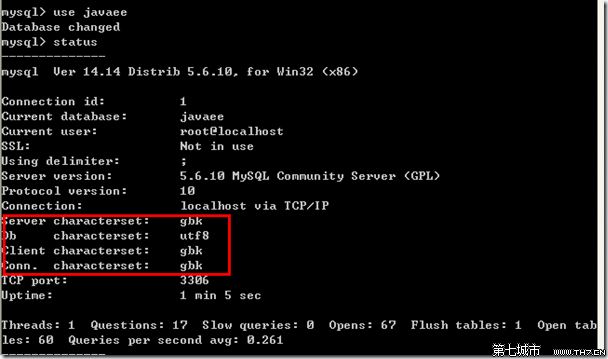
**1.查看数据库支持的所有字符集**

show character set;或者show char set;

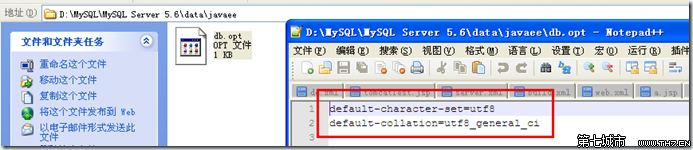


**2.查看当前状态，里面当然包括字符集的设置：**

status或者/s



其中Db characterset对应的是数据库目录下的db.opt文件内容：



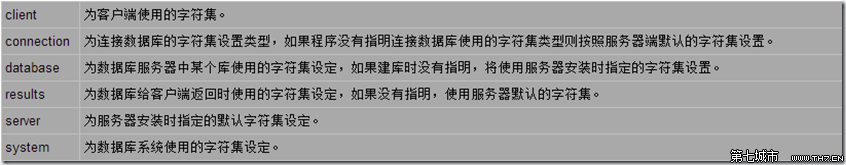
**3.查看系统字符集设置，包括所有的字符集设置：**

show variables like '%char%';

得出如何所示结果：



其中的含义如下：



关于connection相关的字符集的官方文档：

* What character set is the statement in when it leaves the client?

The server takes the character\_set\_client system variable to be the character set in which statements are sent by the client.

* What character set should the server translate a statement to after receiving it?

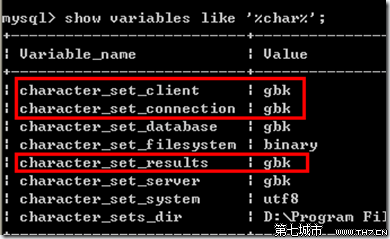
For this, the server uses the character\_set\_connection and collation\_connection system variables. It converts statements sent by the client from character\_set\_client to character\_set\_connection (except for string literals that have an introducer such as \_latin1 or \_utf8). collation\_connection is important for comparisons of literal strings. For comparisons of strings with column values, collation\_connection does not matter because columns have their own collation, which has a higher collation precedence.

* What character set should the server translate to before shipping result sets or error messages back to the client?

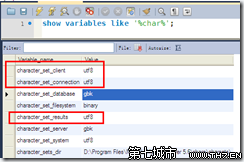
The character\_set\_results system variable indicates the character set in which the server returns query results to the client. This includes result data such as column values, and result metadata such as column names and error messages.

     从上文中可以看出character\_set\_connection、character\_set\_client、character\_set\_results三个字符集什么时候用到。从实际上可以看到，当客户端连接服务器的时候，它会将自己想要的字符集名称发给mysql服务器，然后服务器就会使用这个字符集去设置character\_set\_connection、character\_set\_client、character\_set\_results这三个值。如cmd是用gbk，而mysql workbench是用utf8.

CMD：

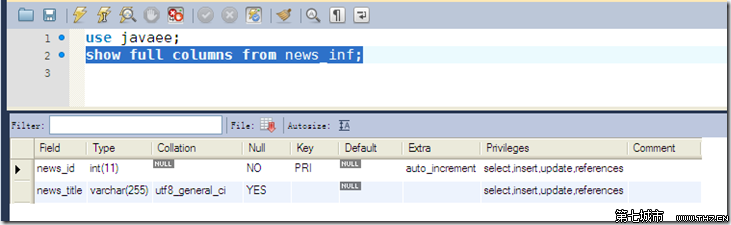


MySql WorkBench：

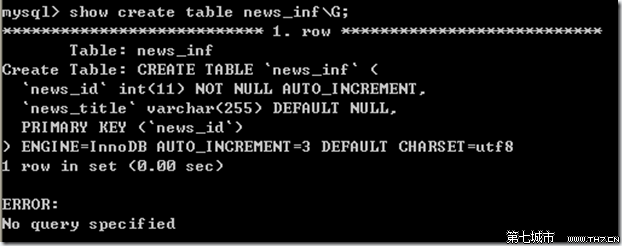


**4.查看数据表中字符集设置：**

show full columns from tablename;

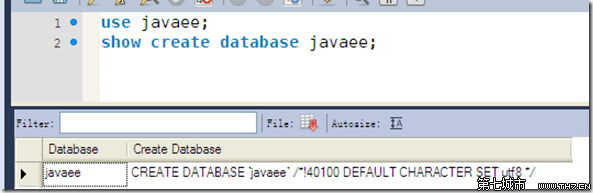


show create table tablename/G;



**5.查看数据库编码：**

show create database dbname;



* **创建时指定字符集：**

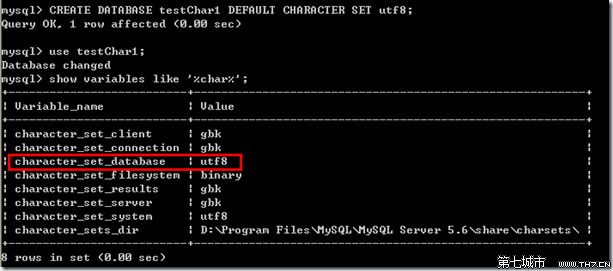
        知道了怎么查找字符集的相关信息之后，我们就要懂得怎么在创建指定对象的时候，为该对象匹配相应的字符集。

**1.服务器级：**

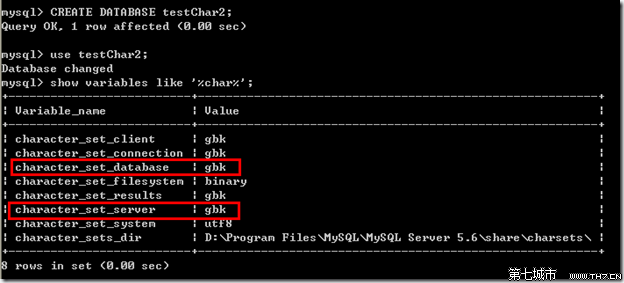
在安装MySQL时可以设置服务器的默认编码格式，也可对my.ini做修改，修改[mysqld]里面的character\_set\_server=utf8，则可设置character\_set\_server的值。

**2.数据库级：**

CREATE DATABASE db\_name DEFAULT CHARACTER SET utf8;



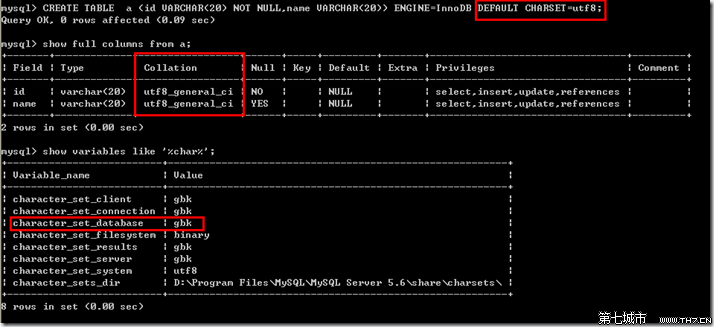
注意，如果不指定默认的字符集，则系统会根据character\_set\_database的值进行设置，如：



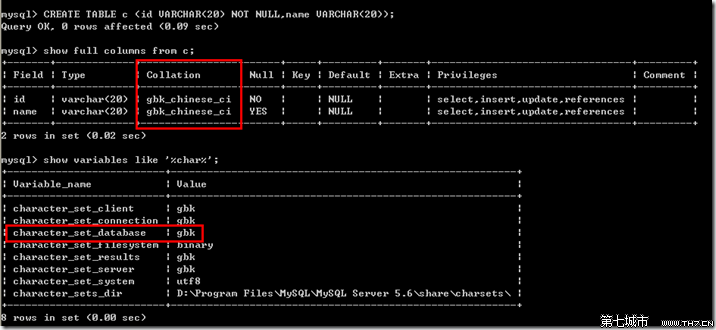
**3.表级：**

CREATE TABLE `db\_name`.`tb\_name` (id VARCHAR(**20**) NOT NULL,name VARCHAR(**20**) ) ENGINE=InnoDB DEFAULT CHARSET=utf8;

从下图可看出，定义表的默认字符集为utf8，即使character\_set\_database为gbk，但是表的列都未utf8



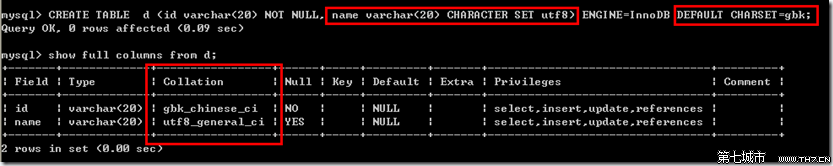
但要注意，如果没有定义表的默认字符集，则他会按照character\_set\_database的值来设置，如图所示：



**4.列级：**

CREATE TABLE `db\_name`.`tb\_name` ( id varchar(**20**) NOT NULL, name varchar(**20**) CHARACTER SET utf8 );

从下图可以看到，整个表的默认字符集为gbk，所以没有指定字符集的列都用默认的字符集，而指定了字符集的列name，则使用指定的字符集utf8。



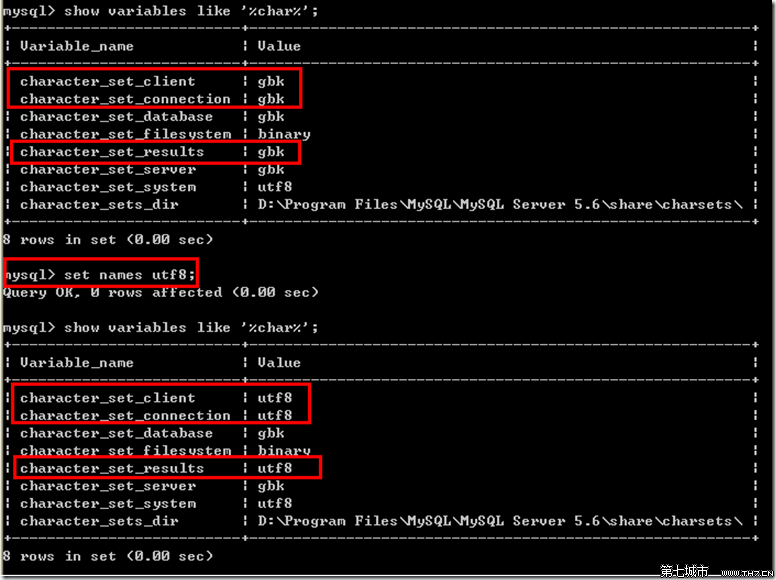
* **修改字符集命令**

        如果已经是创建好的对象，那又应该如何处理呢。我们就应该对指定对象就行修改字符集的操作。

**1.修改character\_set\_connection、character\_set\_client、character\_set\_results三值：**

对于某一个连接来说，可以使用：

SET NAMES 'charset\_name' [COLLATE 'collation\_name']



命令

SET NAMES 'charset\_name' [COLLATE 'collation\_name']

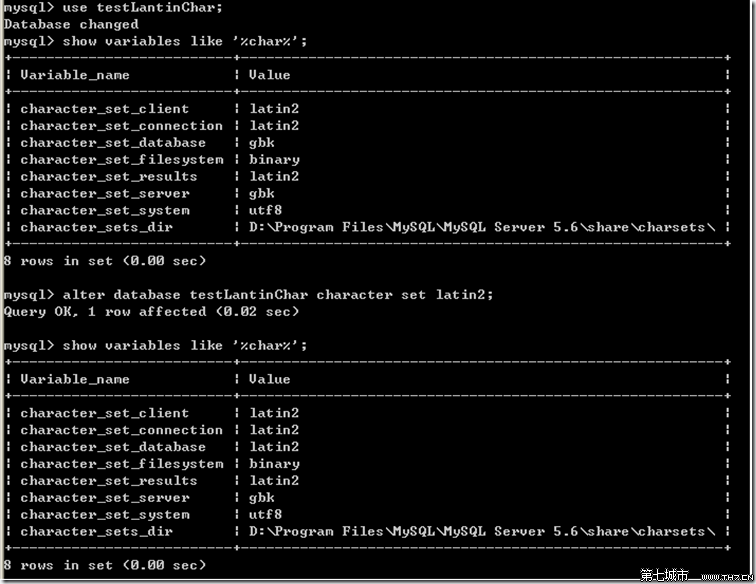
相当于

SET character\_set\_client = charset\_name; SET character\_set\_results = charset\_name; SET character\_set\_connection = charset\_name;

另外、还可以修改配置文件，对[mysql]下增加default-character-set=utf8，配置成你想要的字符集。（个人尝试在my.ini里面配置过，没有成效，不知道是不是被使用的客户端想要的字符集给覆盖掉了呢？）

**2.修改character\_set\_database字段：**

ALTER DATABASE db\_name [[DEFAULT] CHARACTER SET charset\_name] [[DEFAULT] COLLATE collation\_name]



**3.修改character\_set\_server字段：**

最简单的方法是直接改my.ini配置文件里面[mysqld]的字段，增加character-set-server=gbk，然后重启mysqld，则可改为你想要的字符集。

**4.修改表的字符集：**

ALTER TABLE tbl\_name [[DEFAULT] CHARACTER SET charset\_name] [COLLATE collation\_name]

**5.修改列的字符集：**

col\_name {CHAR | VARCHAR | TEXT} (col\_length) [CHARACTER SET charset\_name] [COLLATE collation\_name]

例如：

ALTER TABLE t1 MODIFY col1 VARCHAR(**5**) CHARACTER SET latin1 COLLATE latin1\_swedish\_ci;

  转自：<http://zhidao.baidu.com/link?url=E5kHt1YHeMOFxI6p_5myGoGmHw0wwVi2KEOomo2tzKrHn9KihKaH0Cb5eGb8Bz015hKlP1oBYqOa-Ka2MV1K4SnR0aWTBnRrqEJiWqKMale>

* **参考资料：**

MySQL的Character Set Support:  http://dev.mysql.com/doc/refman/5.6/en/charset.html

mysql常用查看库，表字符集命令:  http://bjlfp.blog.163.com/blog/static/773684612012298455765/

MySQL 插入数据时，中文乱码问题的解决: http://www.cnblogs.com/sunzn/archive/2013/03/14/2960248.html

<http://blog.csdn.net/fengbingyang/article/details/17149769>

1 从<http://dev.mysql.com/downloads/mysql/>  下载MySQL Community Server的Windows 平台mysql-5.6.14-win32.zip压缩包

2 将文件解压至F:\mysql-5.6.14-win32

3 切换至目录F:\mysql-5.6.14-win32，将文件my-default.ini复制一份重命名为my.ini放到当前目录下

4 打开my.ini进行MySQL配置，主要设置以下几项：

**[plain]** [view plaincopy](http://blog.csdn.net/fengbingyang/article/details/17149769)[在CODE上查看代码片](https://code.csdn.net/snippets/99111)

1. [mysqld]
2. #这一句解决有IPV6协议的计算机上默认采用IPV6协议导致无法从程序连接数据库的问题
3. bind-address = 127.0.0.1
4. #设置MySQL Server的字符集
5. character-set-server=utf8
7. [client]
8. #设置MySQL客户端的字符集
9. default-character-set=utf8

5 将MySQL的F:\mysql-5.6.14-win32\bin目录加入系统环境变量

6 在CMD控制台将目录切换到F:\mysql-5.6.14-win32\bin，运行： mysqld -install 安装MySQL的服务，然后到服务列表中启动它

7 在CMD控制台下运行mysqladmin -uroot -p password 123456并回车，设置root用户新密码为123456。接下来输入密码的提示是指输入原密码，初始root密码为空

8 在CMD控制台下运行mysql -uroot -p，根据提示输入密码成功后就可以使用MySQL了