最近，在项目组使用的mysql数据库中，插入数据出现乱码，关于这个问题做了下总结，我们从最基本的地方说起，到错误产生的深层次原因和解决办法。

**基本概念**

• 字符(Character)是指人类语言中最小的表义符号。例如’A'、’B'等；  
• 给定一系列字符，对每个字符赋予一个数值，用数值来代表对应的字符，这一数值就是字符的编码(Encoding)。例如，我们给字符’A'赋予数值0，给字符’B'赋予数值1，则0就是字符’A'的编码；  
• 给定一系列字符并赋予对应的编码后，所有这些字符和编码对组成的集合就是字符集(Character Set)。例如，给定字符列表为{’A',’B'}时，{’A'=>0, ‘B’=>1}就是一个字符集；  
• 字符序(Collation)是指在同一字符集内字符之间的比较规则；  
• 确定字符序后，才能在一个字符集上定义什么是等价的字符，以及字符之间的大小关系；  
• 每个字符序唯一对应一种字符集，但一个字符集可以对应多种字符序，其中有一个是默认字符序(Default Collation)；  
• MySQL中的字符序名称遵从命名惯例：以字符序对应的字符集名称开头；以\_ci(表示大小写不敏感)、\_cs(表示大小写敏感)或\_bin(表示按编码值比较)结尾。例如：在字符序“utf8\_general\_ci”下，字符“a”和“A”是等价的；

**MySQL字符集设置**

• 系统变量：  
– character\_set\_server：默认的内部操作字符集  
– character\_set\_client：客户端来源数据使用的字符集  
– character\_set\_connection：连接层字符集  
– character\_set\_results：查询结果字符集  
– character\_set\_database：当前选中数据库的默认字符集  
– character\_set\_system：系统元数据(字段名等)字符集  
– 还有以collation\_开头的同上面对应的变量，用来描述字符序。

• 用introducer指定文本字符串的字符集：  
– 格式为：[\_charset] ’string’ [COLLATE collation]  
– 例如：  
       SELECT \_latin1 ’string’;  
       SELECT \_utf8 ‘你好’ COLLATE utf8\_general\_ci;  
– 由introducer修饰的文本字符串在请求过程中不经过多余的转码，直接转换为内部字符集处理。

**MySQL中的字符集转换过程**

1. MySQL Server收到请求时将请求数据从character\_set\_client转换为character\_set\_connection；  
2. 进行内部操作前将请求数据从character\_set\_connection转换为内部操作字符集，其确定方法如下：  
       - 使用每个数据字段的CHARACTER SET设定值；  
       - 若上述值不存在，则使用对应数据表的DEFAULT CHARACTER SET设定值(MySQL扩展，非SQL标准)；  
       - 若上述值不存在，则使用对应数据库的DEFAULT CHARACTER SET设定值；  
       - 若上述值不存在，则使用character\_set\_server设定值。

3. 将操作结果从内部操作字符集转换为character\_set\_results。

  我们现在回过头来分析下我们产生的乱码问题：  
          a 我们的字段没有设置字符集，因此使用表的数据集  
          b 我们的表没有指定字符集，默认使用数据库存的字符集  
          c 我们的数据库在创建的时候没有指定字符集，因此使用character\_set\_server设定值  
          d 我们没有特意去修改character\_set\_server的指定字符集，因此使用mysql默认  
          e mysql默认的字符集是latin1，因此，我们使用了latin1字符集，而我们character\_set\_connection的字符集是UTF-8，插入中文乱码也再所难免了。

**常见问题解析**  
• FAQ-1 向默认字符集为utf8的数据表插入utf8编码的数据前没有设置连接字符集，查询时设置连接字符集为utf8  
     – 插入时根据MySQL服务器的默认设置，character\_set\_client、character\_set\_connection和character\_set\_results均为latin1；  
     – 插入操作的数据将经过latin1=>latin1=>utf8的字符集转换过程，这一过程中每个插入的汉字都会从原始的3个字节变成6个字节保存；  
     – 查询时的结果将经过utf8=>utf8的字符集转换过程，将保存的6个字节原封不动返回，产生乱码。参考下图：   
• 向默认字符集为latin1的数据表插入utf8编码的数据前设置了连接字符集为utf8（**我们遇到的错误就是属于这一种**）  
     – 插入时根据连接字符集设置，character\_set\_client、character\_set\_connection和character\_set\_results均为utf8；  
     --插入数据将经过utf8=>utf8=>latin1的字符集转换，若原始数据中含有\u0000~\u00ff范围以外的Unicode字符，会因为无法在latin1字符集中表示而被转换为“?”(0×3F)符号，以后查询时不管连接字符集设置如何都无法恢复其内容了。转换过程如下图：    
  
   
**检测字符集问题的一些手段**  
   • SHOW CHARACTER SET;  
    • SHOW COLLATION;  
    • SHOW VARIABLES LIKE ‘character%’;  
    • SHOW VARIABLES LIKE ‘collation%’;  
    • SQL函数HEX、LENGTH、CHAR\_LENGTH  
    • SQL函数CHARSET、COLLATION  
   
**使用MySQL字符集时的建议**  
    • 建立数据库/表和进行数据库操作时尽量显式指出使用的字符集，而不是依赖于MySQL的默认设置，否则MySQL升级时可能带来很大困扰；  
    • 数据库和连接字符集都使用latin1时，虽然大部分情况下都可以解决乱码问题，但缺点是无法以字符为单位来进行SQL操作，一般情况下将数据库和连接字符集都置为utf8是较好的选择；  
    • 使用mysql CAPI（mysql提供C语言操作的API）时，初始化数据库句柄后马上用mysql\_options设定MYSQL\_SET\_CHARSET\_NAME属性为utf8，这样就不用显式地用SET NAMES语句指定连接字符集，且用mysql\_ping重连断开的长连接时也会把连接字符集重置为utf8；  
    • 对于mysql PHP API，一般页面级的PHP程序总运行时间较短，在连接到数据库以后显式用SET NAMES语句设置一次连接字符集即可；但当使用长连接时，请注意保持连接通畅并在断开重连后用SET NAMES语句显式重置连接字符集。  
   
**其他注意事项**  
    • my.cnf中的default\_character\_set设置只影响mysql命令连接服务器时的连接字符集，不会对使用libmysqlclient库的应用程序产生任何作用！  
    • 对字段进行的SQL函数操作通常都是以内部操作字符集进行的，不受连接字符集设置的影响。  
    • SQL语句中的裸字符串会受到连接字符集或introducer设置的影响，对于比较之类的操作可能产生完全不同的结果，需要小心！  
   
**总结**  
 根据上面的分析和建议，我们解决我们遇到问题应该使用什么方法大家心里应该比较清楚了。对，就是在创建database的时候指定字符集，不要去通过修改默认配置来达到目的，当然你也可以采用指定表的字符集的形式，但很容易出现遗漏，特别是在很多人都参与设计的时候，更容易纰漏。  
   
    虽然不提倡通过修改mysql的默认字符集来解决，但对于如何去修改默认字符集，我这里还是给出一些方法，仅供大家参考。  
   
**MySQL默认字符集**  
MySQL对于字符集的指定可以细化到一个数据库，一张表，一列.传统的程序在创建数据库和数据表时并没有使用那么复杂的配置，它们用的是默认的配置.  
    (1)编译MySQL 时，指定了一个默认的字符集，这个字符集是 latin1；  
    (2)安装MySQL 时，可以在配置文件 (my.ini) 中指定一个默认的的字符集，如果没指定，这个值继承自编译时指定的；  
    (3)启动mysqld 时，可以在命令行参数中指定一个默认的的字符集，如果没指定，这个值继承自配置文件中的配置,此时 character\_set\_server 被设定为这个默认的字符集；  
    (4)安装 MySQL选择多语言支持，安装程序会自动在配置文件中把default\_character\_set 设置为 UTF-8，保证缺省情况下所有的数据库所有表的所有列的都用 UTF-8 存储。  
**查看默认字符集**  
 (默认情况下，mysql的字符集是latin1(ISO\_8859\_1)，如何查看在上面我们已经给出了相关命令  
**修改默认字符集**  
(1) 最简单的修改方法，就是修改mysql的my.ini文件中的字符集键值，  
     如    default-character-set = utf8  
      character\_set\_server =  utf8  
     修改完后，重启mysql的服务  
(2) 还有一种修改字符集的方法，就是使用mysql的命令  
     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 ;  
   
    设置了表的默认字符集为utf8并且通过UTF-8编码发送查询，存入数据库的仍然是乱码。那connection连接层上可能出了问题。解决方法是在发送查询前执行一下下面这句： SET NAMES 'utf8';它相当于下面的三句指令：  
SET character\_set\_client = utf8;  
SET character\_set\_results = utf8;  
SET character\_set\_connection = utf8;

分类: [Database](http://www.cnblogs.com/discuss/category/266317.html)