Oracle9i 字符集设置与数据转换问题的研究

The Research of the Relationship between Characterset and Data Exchange on Oracle9i

朱 毅　谢益武　（大连海事大学信息系统工程研究所 116026）

摘要：数据库中数据的导入，导出和存储数据的一致性是个很关键的问题。字符集的设置与之密切相关。本文从字符集的概念入手，结合实际应用中的问题，着重讨论了 oracle9i 中服务器和客户端字符集设置和数据读取出现乱码问题。

关键词：oracle9i　字符集　数据导出　数据导入　NLS

1　引言

作为一种关系数据库管理系统，oracle 在管理信息系统、企业数据处理、电子商务及电子政务等诸多领域使用日益广泛，因其在数据的安全性和完整性控制方面的优越性能，以及跨越操作系统，多硬件平台的数据相互操作等特点，越来越多的用户使用 oracle 作为其应用数据的后台处理系统。继 oracle7 以后，oracle 公司又陆续推出 oracle8 及 oracle8i，oracle9i 作为其大家族的新成员，在构造大型在线事务处理，支持决策分析，数据仓库等方面体现优势，然而在不同版本 oracle 数据库和不同操作系统的数据导出中，容易出现导入数据乱码的现象，这其中的一个主要原因在于数据库字符集的设置上，本文从以下几个方面着手来阐明 oracle9i 字符集设置与数据导出问题的关系。

2　字符集

字符集就是决定数据库所支持的语言标准，既是指该数据库是支持中文，英文，还是日文，德文等等。目前 oracle9i 支持的字符集约在 60 种左右。在 oracle9i 中，字符集参数采用 NLS（national language support，民族语言支持）表示，它的正确选择，就能使数据库存储、处理、检索等支持本国语言的数据。在 oracle8i，oracle9i 中，允许数据库同时支持多国语言字符集，可以使用户用本国语言与数据库交互，一些不同的字符集是容许进行交换的[1]。

3　字符集之间交换存在的情况

（1）服务器与客户端的字符集相同。这种字符集的使用是典型的数据库应用类型，例如：服务器和客户端都使用 SIMPLFIED CHINESE – CHINA. ZHS16GBK 中文字符集。需要说明的是数据库核心字符集与服务器操作系统的语种是没有关系的。在中文操作系统中可以存储英文，在英文操作系统中也同样可以存储中文，只要字符集设对就可以。

（2）不同字符集可以转换。例如：两种不同的字符集 ZHS16GBK 和 ZHS16CG231280，同样支持中文，这两种字符集是兼容的。字符集的转换由 oracle 自动完成的。

（3）不同字符集不可以转换。例如：ZHS16GBK 与 USTASCII 是两种截然不同的字符集，在字符集 ZHS16GBK 中，可以查询英文，但是在 USTASCII 字符集的数据库中的中文将显示乱码。

（4）单一字符集三层数据库系统结构。如果数据库服务器和应用服务器中使用相同的字符集，则他们之间不存在字符集转化的问题。

（5）对多种字符集的支持。在 oracle8i，oralce9i 中，容许数据库支持多国语言字符集，多用户可以使用本国语言与数据库交互。从 oracle7 开始，oracle 数据库就支持 Unicode 标准编码字符集。Unicode 支持标准的 XML，JAVA，CORBA3. 0 等等。目前的编码为 UTF – 16 和 UTF – 8 编码。oracle8i，oracle9i 支持 Uni-

code3.0 标准,支持 Unicode 编码的数据库字符集类型有 UTF8,AL32UTF8 和 UTFE,优先使用 UTF8。

(6) 支持多语种字符集的三层结构。数据库服务器,应用服务器和客户端都采用 UTF8 字符集,主要用于构造因特网中大型网站,便于不同国家,不同语种的用户浏览[3]。

4 问题的出现与解决

当服务器和客户端之间,或者是服务器和服务器之间的字符集不同的时候,容易引起数据交换,出现问题的现象。以支持 ZHS16GBK, USTASCII 字符集为例(不提示下为中文操作系统)交换出现的问题又可以分为四种:

(1) 服务器支持 ZHS16GBK 字符集,客户端支持 USTASCII 字符集。当在客户端以中文向服务器输入数据时,数据已经出现转化的问题,其他与服务器的字符集相同的用户调用该数据时,将出现乱码。这里我们举个简单的例子,在服务器端建立了一个 gzb,存储了中英文数据,其他客户端读取该表的数据时,出现乱码。例如图 1 所示:

```
SQL > select * from gzb;

ID    NAME          AGE    DEPARTMENT
- - - - - - - - - - - - - - - - - - - - - - - - -
01    ??            24     MARKET
02    ??            25     HR
03    ??            23     PLANNING
04    ???           24     HR
05    SAM           30     MANAGE
06    TOM           35     MARKET
07    sam           30     ???
08    tom           32     ???
```

图 1

同时,客户端可以成功的用 Exp 导出服务器中 gzb 的数据,但利用该导出文件进行 Imp 导入时,系统将提示:

IMP – 00016:不支持要求的字符集转换(从类型 178 到 852)

IMP – 00000:未成功终止导入

(2) 服务器支持 USTASCII 字符集,而客户端支持 ZHS16GBK 字符集。为了数据的安全问题,服务器端的字符集一般在安装完以后,是不允许修改的,当客户端输入中文数据时,由于不同字符集通常是不能转换的(ZHS16GBK 和 ZHS16CG231280 除外),输入到服务器的数据以及从服务器输出的数据,将会出现乱码的问题。同样,可以成功的用 Exp 导出数据,但用该导出文件进行 Imp 导入时,系统提示错误。

(3) 服务器都支持 ZHS16GBK,但是没有汉字的操作系统。汉字在服务器中存储是没有问题的,从客户端查询的时候,汉字也是能正常显示的。但当用 Exp 导出数据,用 Imp 导入数据时,中文将出现乱码。导出时提示的信息为:

Export done in US7ASCII character set and ZHS16GBK character

Server uses ZHS16GBK character set (possible charset conversion)

表示数据库使用的是 ZHS16GBK 字符集,而 Exp 使用的是 US7ASCII 字符集导出,这种方式导出的数据不论导入到什么服务器上,中文都会出现乱码。

(4) 一个服务器支持 ZHS16GBK 字符集,其他服务器支持 USTASCII 字符集。当服务器之间出现数据的导出时,系统提示不能导出,原因是服务器之间的字符集不同。这里有个例外,当其他服务器支持的是 ZHS16CG231280 时,数据可以正常导出,但导入时将出现错误。

我们只需要正确的调整相关的字符集参数,问题就迎刃而解。在 oracle 数据库安装时,系统默认情况下使用操作系统的语种作为数据库的核心字符集使用。如果操作系统为中文平台,则数据库在安装时候默认的字符集为 ZHS16GBK。如果服务器没有中文平台,则数据库默认的字符集为 US7ASCII,当操作系统的语种和数据库核心字符集不一致时,在安装数据库之前必须重新设置 NLS_LANG,这样系统将不再使用默认的字符集,而是以设置的字符集为准[3]。

首先我们需要了解服务器和客户端分别采用什么样的字符集以及和字符集相关的参数。可以查询数据字典 NLS_DATABASE_PARAMETERS 和 V $ NLS_PARAMETERS,相应的 SQL 语句和结果如下:

① SQL > show parameter nls;

② SQL > select * from nls_database_parameters;

③ SQL > select * from v $ nls_parameter;

PARAMETER	VALUE
NLS_LANGUAGE	SIMPLIFIED CHINESE
NLS_TERRITORY	CHINA
NLS_ISO_CURRENCY	CHINA
NLS_NUMERIC_CHARACTERS	
NLS_CALENDAR	GREGORIAN
NLS_DATE_FORMAT	DD – MON – RR
NLS_DATE_LANGUAGE	SIMPLIFIED CHINESE
NLS_CHARACTERSET	ZHS16GBK
NLS_CALENDAR	GREGORIAN
NLS_DATE_FORMAT	DD – MON – RR

这里的 NLS_CHARACTERSET 参数表示系统的字符集,为 ZHS16GBK,表示支持中文。

其次在客户端修改 NLS_LANG 的方法如下:WINDOWS 平台下客户端的 NLS_LANG 定义在操作系统的注册表里。步骤如下:

- 先在"开始"的"运行"里键入 regedit。
- 其次在注册表里选择 HKEY_LOCAL_MACHINE = > SOFTEARE = > ORACLE = > HOME0。
- 最后选择 NLS_LANG 参数,修改其"数值数据"为 ZHS16GBK。

在服务器端,参数 NLA_LANG 同样可以在注册表里修改,同时也可以用 SQL 语句进行修改(一般数据库安装完后不进行修改),这里我们选择使用 SQL 语句修改。步骤如下:

- 在"运行"里键入 sqlplus 。
- "user/password as sysdba"(以系统数据库管理员身份登陆)
- SQL > show parameter nls
- SQL > update props $
  2 > set values $ = "ZHS16GBK"
  3 > where name = "NLS_CHARACTERSET"
- SQL > exit
- 重新启动数据库[2]。

按照上面方法调整后,客户端与服务器的字符集是一致的,上例中客户端再次读取服务器中的中文数据时,数据还原为原始的数据。例如图 2 所示:

ID	NAME	AGE	DEPARTMENT
01	高峰	24	MARKET
02	赵伟	25	HR
03	海霞	23	PLANNING
04	于媛媛	24	HR
05	SAM	30	MANAGE
06	TOM	35	MARKET
07	sam	30	企划部
08	tom	32	组织部

再次用 Exp 和 Imp 工具进行数据导出和导入时,将不会出现乱码和系统错误的问题。

5 结束语

oracle 数据库中,核心字符集的设定是个很关键的问题,它关系到整个数据库中数据导入导出,存储的一致性。我们已经从什么是字符集以及字符集之间转换的几种情况着手,在实际的例子中解决了数据转换中出现乱码和系统错误的问题,这对今后在工程中遇到同样的问题将会有所帮助。

参考文献

1 Loney K. Theriau M. Oracle9i DBA 手册,机械工业出版社,2002。
2 Jess S. Oracle9i for Windows 2000 技术与技巧 ,机械工业出版社,2003。
3 Oracle9i 数据库管理员使用大全,清华大学出版社 ,2004。
4 Oracle9i 开发指南:PL/SQL 程序设计 ,清华大学出版社 ,2004。
5 Oracle9i 高级专家编程 ,清华大学出版社,2004。