MySQL字符集

石磊

2015-10-31

- MySQL字符集

提要

从本质上来说，计算机只能识别二进制代码，因此，不论计算机程序还是其处理的数据，最终都必须转换成二进制码。为了使计算机不仅能做科学计算，也能处理文字信息，人们想出了给每个文字符号编码以便于计算机识别处理的办法。

字符集的由来

字符集概述

- 字符集就是一套文字符号及其编码、比较规则的集合。1960年，美国标准化组织ANSI发布了第一个计算机字符集（AMERICAN STANDARD CODE FOR INFORMATION INTERCHANGE)
- ASCII
- 这个字符集采用7位编码，定义了包括大小写英文字母、阿拉伯数字和标点符号，以及33个控制符号。直到今天它依然是计算机世界里奠基性的标准，并被其后的字符集所兼容。

字符集问题

- 自ASCII之后，为了处理不同的文字，各大计算机公司，各国政府，标准化组织等先后发明了几百种字符集。比如ISO-8859系统，GB2312-80，GBK，BIG5等。
- 由于收录的字符和编码规则各不相同，给计算机软件的开发和移植带来了很大的困难。

Unicode概述

- 为了统一字符编码，国际标准化组织ISO在1984年发起制定新的国标字符集标准，以容纳全世界各种语言文字和符号。这个标准最后叫做universal multiple-octet coded character set 简称UCS
- 标准编号定为ISO-10646，采用4字节（32bit）编码，因此简称UCS-4

- 但是，ISO-10646发布后，遭到了部分美国计算机公司的反对。1988年xerox公司提议制定新的16位编码的统一字符集unicode，并联合Apple、IBM、DEC、Sun、Microsoft、Novell等公司成立unicode协会（the Unicode Consortium），并成立unicode技术委员会（Unicode Technical Committee）专门负责unicode文字的搜集整理和编码，并与1991年推出unicode1.0

不同声音

- 大家逐渐认识到统一的必要性。经过双方谈判，ISO将unicode编码并入ISO-10646，叫做基本多语言文字面（basic multi-lingual plane，BMP)，共有65534个码位。
- ISO-10646的编码空间足以容纳人类所有文字和符号，到其实许多已经很少使用了。超过99%的在用文字都编入了BMP，因此绝大部分情况下unicode的双字节编码方式都能满足需求，而这种双字节编码比起10646的4字节编码，在节省内存和处理时间上都有优势，因此比较流行。

和解

- 但使用unicode后万一需要使用10646其他字面的编码怎么办呢？
- Unicode提出了UTF-16的解决方案：对BMP字面的编码保持2字节不变，对其他字面的文字按一定规则转换为2个2字节编码。
- UTF-16虽然解决了ISO-10646其他字面的编码问题，但当时的计算机和网络时间还是ASCII码的天下，只能粗粒1字节的数据流，UTF-16在离开unicode环境后，传输和处理都有问题。

完善

UTF-8

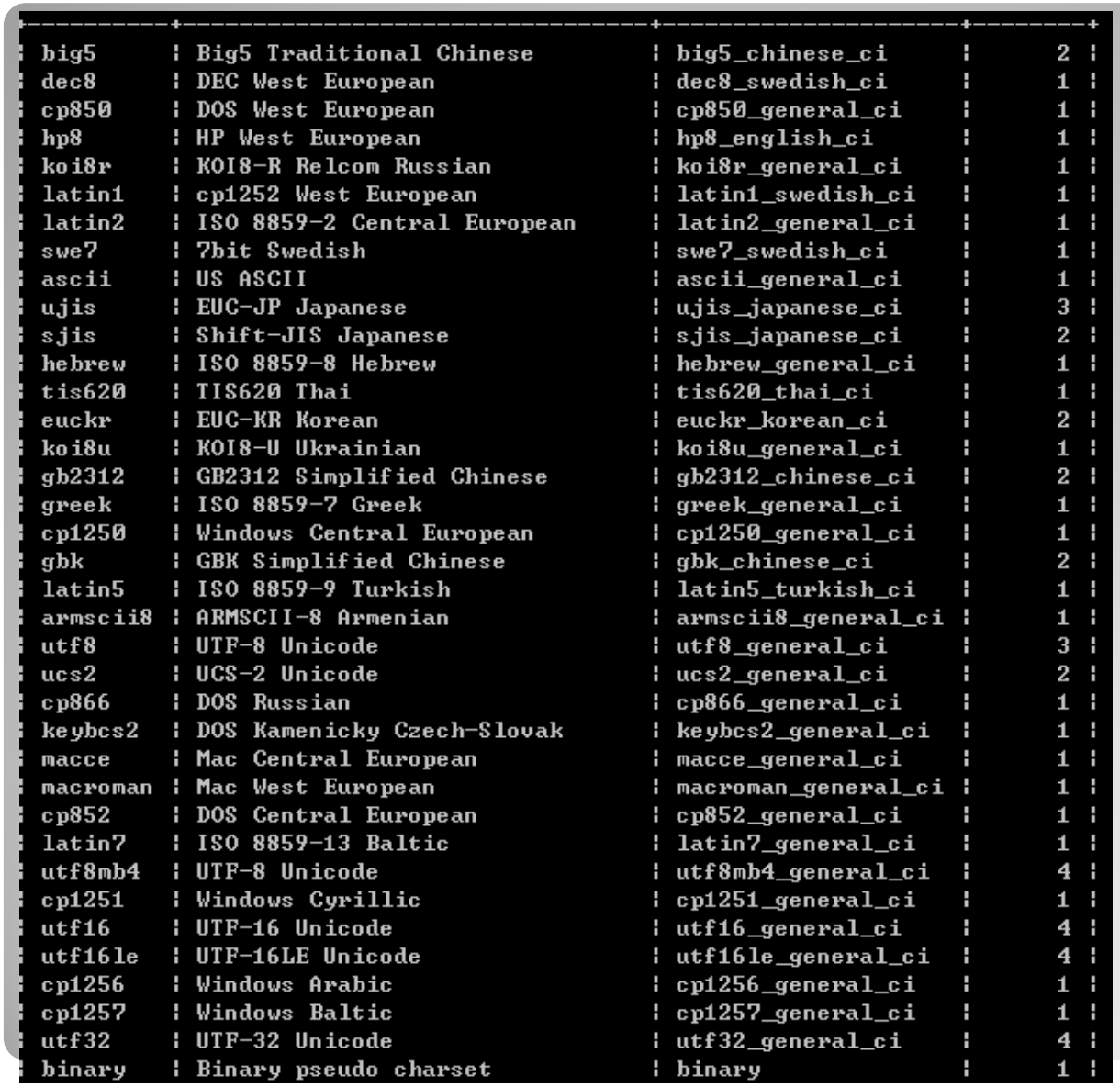
- 于是出现了UTF-8：
- UTF-8按如下规则将ISO-16046转换成1~4个字节编码，其中将ASCII码转换成单字节编码，ISO-16046标准0x0080~0x7FF转换成2字节编码， ISO-16046标准0x0800~0xFFFF转换成3字节编码，其他转换成4字节编码。

中文常见的编码方式

- GB2312-80：1980年发布，2字节编码，收录了6763和常用汉字和682个非汉字图形符号。
- GB13000：1993年发布，除收录GB2312-80外，还收录了部分辅助汉字，共27484个，以及一些偏旁部首等。但其推出后，几乎没有得到业界的支持。
- GBK：1995年。GBK在GB2312-80基础上进行了扩充，还收录了GB13000的全部20902个CJK统一汉字，还增补了52个汉字和一些偏旁部首。GBK完美兼容GB2312.GBK并不是强制性的国家标准，只是一个行业规范，但由于得到了windows95的支持而最为流行。
- GB18030：2000年发布，是GBK的超集，也完全兼容GB13000，制定GB18030是为了解决GBK强制力不够的问题。

怎样选择合适的字符集？

- MySQL目前支持几十种字符集，UTF-8是其支持的唯一unicode字符集，但不支持4字节的扩展部分。选择MySQL字符集，考虑：
- 1. 满足应用支持语音的需求：如果应用要处理各种不同文字，就应该选择unicode。对MySQL来说首选UTF-8
- 2. 如果应用中涉及已有数据的导入，就要充分考虑兼容性。假如已有数据是GBK，就不能选择更老的GB2312-80字符集。
- 3.如果数据库仅需支持中文，数据量大，性能要求高，那就应该选择2字节编码的中文字符集，比如GBK，因为UTF-8的中文编码是3字节。



big5	Big5 Traditional Chinese	big5_chinese_ci	2
dec8	DEC West European	dec8_swedish_ci	1
cp850	DOS West European	cp850_general_ci	1
hp8	HP West European	hp8_english_ci	1
koi8r	KOI8-R Relcom Russian	koi8r_general_ci	1
latin1	cp1252 West European	latin1_swedish_ci	1
latin2	ISO 8859-2 Central European	latin2_general_ci	1
swe7	7bit Swedish	swe7_swedish_ci	1
ascii	US ASCII	ascii_general_ci	1
ujis	EUC-JP Japanese	ujis_japanese_ci	3
sjis	Shift-JIS Japanese	sjis_japanese_ci	2
hebrew	ISO 8859-8 Hebrew	hebrew_general_ci	1
tis620	TIS620 Thai	tis620_thai_ci	1
euckr	EUC-KR Korean	euckr_korean_ci	2
koi8u	KOI8-U Ukrainian	koi8u_general_ci	1
gb2312	GB2312 Simplified Chinese	gb2312_chinese_ci	2
greek	ISO 8859-7 Greek	greek_general_ci	1
cp1250	Windows Central European	cp1250_general_ci	1
gbk	GBK Simplified Chinese	gbk_chinese_ci	2
latin5	ISO 8859-9 Turkish	latin5_turkish_ci	1
armscii8	ARMSCII-8 Armenian	armscii8_general_ci	1
utf8	UTF-8 Unicode	utf8_general_ci	3
ucs2	UCS-2 Unicode	ucs2_general_ci	2
cp866	DOS Russian	cp866_general_ci	1
keybcs2	DOS Kamenicky Czech-Slovak	keybcs2_general_ci	1
macce	Mac Central European	macce_general_ci	1
macroman	Mac West European	macroman_general_ci	1
cp852	DOS Central European	cp852_general_ci	1
latin7	ISO 8859-13 Baltic	latin7_general_ci	1
utf8mb4	UTF-8 Unicode	utf8mb4_general_ci	4
cp1251	Windows Cyrillic	cp1251_general_ci	1
utf16	UTF-16 Unicode	utf16_general_ci	4
utf16le	UTF-16LE Unicode	utf16le_general_ci	4
cp1256	Windows Arabic	cp1256_general_ci	1
cp1257	Windows Baltic	cp1257_general_ci	1
utf32	UTF-32 Unicode	utf32_general_ci	4
binary	Binary pseudo charset	binary	1

MySQL字符集

- MySQL字符集包括字符集（CHARACTER)和校对规则（COLLATION)2个概念：
- 字符集是MySQL存储字符串的方式，
- **校对规则**是MySQL比较字符串的方式。
- 校对规则与其相关的字符集名开始，通常包括一个语言名，并且以_ci(大小写不敏感）_cs(大小写敏感）_bin(二元，比较是基于字符编码的值而与language无关）结束。

MySQL字符集的设置

- MySQL的字符集和校对规则有4个级别的默认设置：服务器级，数据库级，表级和字段级。分别在不同的地方设置。
- 1.服务器级字符集：
- 在配置文件my.cnf中设置：

[mysqld]
default-character-set=utf8
在启动时指定：
mysqld –default-character-set=utf8

注意：如果没有特别指定，Latin1作为服务器默认字符集。

- 数据库字符集：

可以在my.cnf中指定也可以在建库时指定。不指定则使用服务器字符集。

- 表字符集：

在建表时指定，如果不指定则使用数据库字符集。

- 字段字符集：

在建表时指定，如果不指定则使用表字符集。

数据库级字符级

```
mysql> use edu
Database changed
mysql> create table test_character(id int,name char(8))
    -> engine=innodb default charset=utf8 collate=utf8_bin;
Query OK, 0 rows affected (0.02 sec)

mysql> show create table test_character\G
*************************** 1. row ***************************
       Table: test_character
Create Table: CREATE TABLE `test_character` (
  `id` int(11) DEFAULT NULL,
  `name` char(8) COLLATE utf8_bin DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_bin
1 row in set (0.00 sec)
```

表字符集实例

字符集的历史
Mysql常用字符集
（utf8，gbk）
Mysql字符集的设置

小结