1. 字符集和比较规则
2. 概述

计算机中实际存储的是二进制数据，对于字符串数据，通过建立字符串与二进制数据的映射关系来存储，首先要界定字符范围，然后将字符映射为二进制数据的过程称为编码，将二进制数据映射为字符的过程为解码。用字符集来表示某个字符范围的编码规则，即字符集定义了哪些字符映射到哪些二进制数据的规则。

设定了字符集后，如何比较两个字符的大小呢？最容易想到的就是直接比较两个字符对应的二进制数据的大小，这种称为二进制比较规则。但是现实中，很多情况下是更加复杂的，如英文字母不区分大小写时，需要将大小写不同的字符转为大写或小写，再比较二进制数据。

1. 常用的字符集
2. ASCII字符集

共128个字符，包括空格，标点符号，数字，大小写字母等，可使用一个字节来编码，即8位。

1. ISO 8859-1字符集

共256个字符，在ASCII基础上又扩充了128个西欧常用字符，也可以使用一个字节编码。

1. GB2312字符集

收录了汉字等，兼容ASCII字符集，如果该字符在ASCII字符集中，则采用一字节编码，否则采用两字节编码。这种采用不同字节数来表示一个字符的编码方式称为变长编码方式。

1. GBK字符集

扩充了GB2312，兼容GB2312.

1. UTF-8字符集

几乎收录了所有地区的字符，兼容ASCII，采用变长编码方式，编码一个字符使用1-4个字节。

在读取一个字节时，如何区分该字节代表的是单独的字符还是某个字符的一部分呢？由于该字节在0-127之中，即字节的最高位为0，则该字节代表单独的一个字符，否则该字节的最高位为1，超出了127范围，说明是两个字节代表一个字符。

1. MySQL中的字符集和比较规则
2. utf8

MySQL中，字符集表示一个字符所用的最大字节长度会影响系统的存储和性能。utf8mb3只是用1-3个字节表示字符，别名是utf8；utf8mb4使用1-4个字节表示字符，这是正宗的UTF-8字符集。

1. 查看字符集

SHOW CHARSET LIKE 匹配模式;

CHARSET 和 CHARACTER SET 是同义词。ASCII字符集占一个字节，latin1占一个字节，GB2312占两个字节，GBK占两个字节。

1. 查看比较规则

SHOW COLLATION LIKE 匹配模式;

一种字符集可能对应着若干中比较规则，比较规则的命名以其关联的字符集的名称开头，如utf8的比较规则都以utf8开头，后面紧跟着比较规则所应用的语言，如utf8\_polish\_ci表示波兰语的标记规则，特殊的是utf8\_general\_ci表示通用的比较规则。

最后面的后缀表示重音，大小写，二进制。如\_ai（accent insensitive）表示不区分重音，\_as(accent sensitive) 表示区分重音 ，\_ci(case insensitive) 表示不区分大小写， \_cs(case sensitive) 表示区分大小写， \_bin(binary) 表示以二进制方式比较。

1. 字符集和比较规则的应用

字符集和比较规则是相互对应的，不论修改哪一个，都会自动修改对应的另一个。

1. 四个级别的字符集合比较规则
2. 服务器级别

两个系统变量来表示服务器级别的字符集和比较规则：

character\_set\_server : 服务器级别的字符集;

collation\_server : 服务器级别的比较规则;

在启动服务器程序时，可以通过启动选项或者配置文件来修改这两个变量的值。

1. 数据库级别

创建和修改数据库时可以指定该数据库的字符集和比较规则，同样的有两个系统变量来表示数据库级别的字符集和比较规则：

character\_set\_database;

collation\_database;

创建和修改数据库时设置数据库的字符集和比较规则：

CREATE DATABASE test CHARACTER SET utf8 COLLATE utf8\_polish\_ci;

ALTER DATABASE test CHARACTER SET utf8 COLLATE utf8\_polish\_ci;

若创建数据库时不指定字符集和比较规则，则使用服务器级别的字符集和比较规则。

1. 表级别

同理，创建和修改表时指定表的字符集和比较规则，如果不指定，则会继承数据库的。

1. 列级别

对于存储字符串的列，同一个表中的不同的列也可以有不同的字符集和比较规则，我们在从创建或修改列时可以指定。

CREATE TABLE 表名 (

列名 字符串类型 CHARACTER SET 字符集名称 COLLATE 比较规则名称

);

ALTER TABLE 表名 MODIFY 列名 字符串类型 CHARACTER SET 字符集名称 COLLATE 比较规则名称;

同样的，如果不指定列的字符集和比较规则，就会继承表的。根据所要存储的数据，选择设定列的字符集和比较规则。

1. 客户端和服务器通信中的字符集

从机器的角度看，客户端发送的请求和服务器返回的响应实质上就是一个个字节序列，在请求-响应的过程中，经历了多次的字符集转换。

1. 客户端发送请求

一般情况下，客户端编码请求字符串时使用的字符集与操作系统当前使用的字符集一致，这是在类Unix操作系统中的规则，但是在Windows系统中，如果启动MySQL客户端程序时指定了default-character-set启动选项，则会以该字符集对请求的字符串编码。

类Unix系统中使用如下命令来查看当前系统的字符集：

echo $LC\_ALL

1. 服务器接收请求

服务器接收到一个字节序列，服务器将这个字节序列看做是使用系统变量character\_set\_client代表的字符集进行编码的字节序列，这个系统变量是SESSION级别的。可以看到，客户端在编码请求字符串时使用的字符集与服务器在收到字节序列时所认为该字节序列采用的编码字符集是两个独立的字符集。

1. 服务器处理请求

真正处理器请求时，又会将以character\_set\_client对应的字符集编码的字节序列转换为使用SESSION级别变量character\_set\_connection进行编码的字节序列，collation\_connection表示对应的比较规则。

1. 服务器生成响应

服务器处理完请求后，不会直接将结果发送到客户端，而是使用SESSION级别的系统变量character\_set\_results对应的字符集进行编码，然后发送给客户端。

1. 客户端接受响应

类Unix使用操作系统的字符集解释响应的字节序列，Windows系统使用客户端默认的字符集来解释。

1. 客户端默认字符集

每个MySQL客户端都维护着一个客户端默认字符集，客户端在启动时自动检测所在系统当前使用的字符集，按照一定的规则映射为MySQL支持的字符集。如果MySQL不支持系统的字符集，则将客户端默认的字符集设置为MySQL的默认字符集。若指定了default-character-set启动选项，则该选项作为客户端的默认选项。MySQL的默认字符集在5.7之前为latin1, 8.0之后为utfmb4。

连接服务器时，服务器会将character\_set\_client, character\_set\_connection, character\_set\_results初始化为客户端的默认字符集。