



链滴

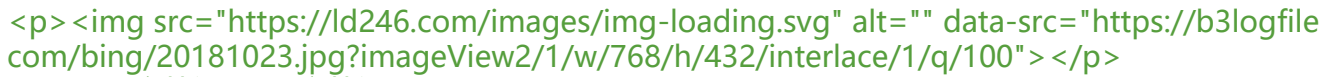
# MySQL 数据库之 - 字符集 - 存储引擎 - 事务

作者: [468336329Zc](#)

原文链接: <https://ld246.com/article/1572593863604>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

字符问题:

位 字节 字符

## ASCII

因为计算机能够计算的是 2 进制的数字，只能识别 2 进制的，而我们现实世界有很多种表达信的方式，有语音，文字，视屏等方式。

那么为了让计算机能够识别我们人类的信息，我们需要对我们的信息进行编码。

美国人因为首先发明电脑，因此他们就搞了套 ascii 码字符集，规定了所有英文字母和常用英文号与数字之间的对应表。

我们知道计算机中一个字节等于 8 位，8 位二进制数可以表示 256 种数字，但是所有英文字母和英文字符只要 128 位就可以完全表示完，

因此 ascii 码中只有规定了那 8 位中第一位只能为 0，这样就只用了后 7 位，能表示 128 种。

但是其他国家不能表示完他们的文字字符，而且其他国家的字符与 ascii 并不统一。

假设美国 ascii 中 65 表示 A 字母，但是俄罗斯规定了 65 表示 "О"。

如果是这样，那么就会很混乱。

## 字符集

Unicode(美国标准信息转换码)。

老美意识到全世界不统一字符就很他娘的混乱，就创造了 Unicode。

这里面给世界上所有字符都规定了一个唯一的数字编号，但是它并没有规定这些字符是如何进行编码二进制的。

**UTF-8**

是一种编码方式，是 Unicode 的实现方式。

它规定了如何将 unicode 中的字符编码，并放进计算机中存储的。

它是一种可变长的字节编码方式。

中国的汉字，它是用 3 个字节表示一个汉字。

我们也有专门的字符集规定了字符之间的转换

gbk (中文) 专门规定了中文的字符转换规则，

**gbk**

它也是规定了 unicode 中中国字符如何编码存进计算机中。

因为只关心中国的字符，采用了 2 个字节表示一个中国字

中国字有 3 万多，2 个字节 2 的 16 次方，完全能够表示完中国的汉字。

```
<code class="highlight-chroma"><span class="highlight-line"><span class="highlight-cl">## 存储引擎:
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">MyISAM 不支持
务，支持全文索引
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">innnoDB 支持事务
不支持全文索引，5.6版本以上支持。
</span></span><span class="highlight-line"><span class="highlight-cl">•Csv
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">•Memory
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">•BlackHole
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">## 事务
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">事务就是一种机制
可以使得执行错误的sql语句回滚。
</span></span><span class="highlight-line"><span class="highlight-cl">
```

```
</span></span><span class="highlight-line"><span class="highlight-cl">一个sql语句就是个事务
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">每次执行一个sql
句之前，都需要开启事务
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">开启指令:start tran
action;
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">默认没开启事务，
认自动提交。开启事务，就需要我们确认提交，需要我们每次执行sql语句之后，需要我们commit
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">查看提交情况  sho
variables like  "%commit%"
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">**事务的作用举例:*
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">比如我给另一用户
钱，有一种情况。当我转钱成功之后，对方的账户上应该加上我转的钱。但是突然断网了，我钱转出
了，但是对方没有加上，这种情况该怎么办呢？
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">事务的作用就凸显
，只有当我钱转出去了，并且确认对方钱加上去了，那么我们确认，才能执行成功，否则就是失败，
撤销操作回滚。
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">***### 事务特性**
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">一致性
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">原子性（要么全
，要么都不做）
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">隔离性（多个事务
时并发发生，）
</span></span><span class="highlight-line"><span class="highlight-cl">
</span></span><span class="highlight-line"><span class="highlight-cl">持久性(对于数据
言，应该是硬盘数据发生变化，而不是内存种的变化)
</span></span></code></pre>
```