书后面会详细介绍。

（2）之前学习过，列表类型的每个条目只能是一个字符串。类似的还有哈希类型的字段值和有序集合类型的值，它们也都只能是字符串。而流类型的每个条目都可以是若干键值对，可以方便我们结构化地存储日志的详情。例如，对于 NGINX 的访问日志，我们可以用键值对分别存储 IP 地址、访问的网址等信息。

此外，流类型的另一个重要用途是作为消息中间件[①]使用。这一部分会在 4.4 节中详细介绍。

3.7.2 命令

1. 增加条目

```
XADD key [MAXLEN [=|~] threshold] *|ID field value [field value ...]
```

作为唯一一个用来向流中增加条目的命令，XADD 命令可以分成两部分：第一部分用来向流中插入新条目，第二部分（上述命令定义中 MAXLEN 所在的指令组）类似 LTRIM 命令，用来要求流最多只保持指定数量的条目。其中，第二部分是可选的。

只看第一部分的话，只需要提供键名、条目 ID 以及若干字段和对应的字段值。例如：

```
redis> XADD nginxLogs * IP 193.180.71.3 status 200
"1616919370007-0"
```

这个示例指明了键名 nginxLogs，并添加了一个包含 IP 和 status 这两个字段的条目。需要单独说明的是命令中的第二个参数“*”，在介绍这个参数前，我们先看一下这条命令的返回值："1616919370007-0"。3.7.1 节介绍过，流中的每个条目都会有唯一的 ID。这个返回值就是通过 XADD 命令新增的条目的 ID。它的格式是：

```
<millisecondsTime>-<sequenceNumber>
```

以“-”分隔的前半部分是增加这条记录时 Redis 服务器的时间戳（即从 UTC 时间 1970 年 1 月 1 日到现在的时间数值，以毫秒为单位）。这一部分保证在不同毫秒插入的条目具有不同的 ID。那么，如果两个条目在同一毫秒插入呢？这个时候后半部分的序列号就派上用场了，在同一毫秒插入的条目会具有不同的序列号，如"1616920548205-0"和"1616920548205-1"。因为序列号在同一毫秒内是递增的，所以才能保证依据条目增加的先后顺序其 ID 不同，后加入的条目的 ID 一定比之前加入的条目的 ID 大。

了解了条目的 ID 后，下面来介绍命令中的“*”。通过命令定义我们知道，第二个参

① 有趣的是，流类型作为消息中间件使用时非常方便和强大，以至于几乎所有文章介绍流类型时都是基于消息中间件的场景来介绍它，而忽视了它作为数据类型的本身特性。Redis 的作者曾经专门写了一篇文章描述这个事情，觉得只关注消息中间件的场景会限制流类型的使用。因此，本书从数据结构本身的用途开始介绍流类型，并在 4.4 节扩展介绍其如何作为消息中间件来使用。