

那有没有办法能够实现可扩展的数据结构设计,不用修改表结构就可以新增字段呢?当然有!许多 NoSQL 数据库使用的列族 (Column Family) 设计就是一种解决方案。列族最早在 Google 的 BigTable 中使用,是一种面向列的稀疏矩阵存储格式,如图 3.17 所示。

Key	联系方式 (Column Family)			课程成绩 (Column Family)		
001	Weibo: li_zhihui	分机: 233		历史: 85		地理: 77
002		分机: 809	QQ: 523		英语: 78	地理: 87
003		分机: 523	QQ: 908	历史: 91	英语: 88	

图 3.17 HBase 列族

这是一张学生的基本信息表,表中不同学生的联系方式各不相同,选修的课程也不同,而且将来还会加入更多联系方式和课程,如果按照传统的关系数据库设计,无论提前预设多少冗余字段都会捉襟见肘、疲于应付。

而使用支持列族结构的 NoSQL 数据库,在创建表的时候,只需要指定列族的名字,无须指定字段 (Column)。那什么时候指定字段呢?可以在数据写入时再指定。通过这种方式,数据表可以包含数百万的字段,这样就可以随意扩展应用程序的数据结构了,并且这种数据库也很方便查询,可以通过指定任意字段名称和值进行查询。

HBase 这种列族的数据结构设计,实际上是把字段的名称和字段的值,以 Key-Value 的方式一起存储在 HBase 中。实际写入的时候,可以随意指定字段名称,即使有几百万个字段也能轻松应对。

## HBase 的高性能存储

传统机械式磁盘的访问特性是连续读写很快,随机读写很慢。这是因为机械磁盘靠电机驱动访问磁盘上的数据,电机要将磁头落到数据所在的磁道上,这个过程需要较长的寻址时间。如果数据存储不连续,磁头就要不停移动,浪费了大量的时间。

为了提高数据写入速度,HBase 使用了一种名为 LSM 树的数据结构进行数据存储。LSM 树的全名是 Log Structed Merge Tree,翻译过来就是 Log 结构合并树。数据写入的时候以 Log 方式连续写入,然后异步对磁盘上的多个 LSM 树进行合并,如图 3.18 所示。