TURING 图灵程序设计丛书

NoSQL

NOT ONLY SQL

数据库入门

[日] 佐佐木达也 著
罗勇 译

- 了解当今最炙手可热的NoSQL新型数据库技术
- 介绍memcached、Tokyo Tyrant、Redis、MongoDB
- 如何基于MySQL应用NoSQL技术特性

人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵程序设计丛书

NoSQL

NOT ONLY SQL

数据库入门

[日] 佐佐木达也 著
罗勇 译

人民邮电出版社
北京

图书在版编目（CIP）数据

NoSQL数据库入门 /（日）佐佐木达也著；罗勇译.
—北京：人民邮电出版社，2012.5
（图灵程序设计丛书）
ISBN 978-7-115-27950-7

Ⅰ. ①N… Ⅱ. ①佐… ②罗… Ⅲ. ①数据库系统
Ⅳ. ①TP311.13

中国版本图书馆CIP数据核字（2012）第069840号

内 容 提 要

本书详细地介绍了NoSQL数据库（非关系型数据库）的种类、用途以及使用方法，并对memcached、Tokyo Tyrant、Redis、MongoDB这4种代表性的NoSQL数据库的特征、适用范围、实现代码进行了深入探讨，并比较了它们的性能。

本书适合有关系型数据库开发经验的软件工程师和程序员阅读。

NoSQL数据库入门

◆ 著　　[日] 佐佐木达也
译　　罗　勇
责任编辑　傅志红
执行编辑　乐　馨

◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街14号
邮编　100061　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京鑫正大印刷有限公司印刷

◆ 开本：800×1000　1/16
印张：13
字数：308千字　　2012年5月第1版
印数：1—4 000册　　2012年5月北京第1次印刷

著作权合同登记号　图字：01-2012-2664号

ISBN 978-7-115-27950-7

定价：45.00元

读者服务热线：(010) 51095186转604　印装质量热线：(010) 67129223
反盗版热线：(010) 67171154

前　言

最近，到处都能听到“NoSQL”，这个词到底是什么意思呢？“NoSQL”到底能给我们带来什么好处呢？

现在，提起数据存储，一般都是针对关系型数据库来说的。但是，关系型数据库并不是万能的，它对于某些处理依然是很吃力的。本书所讲述的 NoSQL 数据库就是为了弥补关系型数据库的不足应运而生的。在适当的情况下使用 NoSQL 数据库，可以为关系型数据库需要耗费大量时间才能完成的处理，提供高速、合理的解决方案。

预想的适用情况

NoSQL 数据库可用于：

- **取代关系型数据库的弱势处理（比如大量数据的写入处理等）**
- **作为关系型数据库之外的另一种选择**

NoSQL 数据库虽然可以替关系型数据库分担一些难题（如大量数据的写入等），但这其中的操作具有相当的难度。因此本书仅仅将 NoSQL 定位为“关系型数据库之外的另一种选择”。

读者对象

本书面向的读者群为有 1 年关系型数据库开发经验的软件工程师和程序员。因为他们对数据量的增大给关系型数据库的检索和更新处理带来的剧烈性能恶化有切身感受，也能更深刻地理解 NoSQL 数据库的优势。

本书将为读者介绍 NoSQL 数据库以及使用 NoSQL 数据库所带来的便利。

本书内容

本书共由 5 章内容组成，下面对各章内容做一个简单的介绍。

第 1 章首先介绍什么是 NoSQL 以及这个词的来源。之后，会介绍关系型数据库的发展历史、关系型数据库的优势和不足，以及 NoSQL 数据库诞生的背景，以便使读者了解 NoSQL 数据库的发展历史。

这一章亦会对 NoSQL 数据库的种类和特征做简单讲解。同时阐述“怎么样才能更好地区别使用关系型数据库和 NoSQL 数据库”，为 NoSQL 数据库的引入奠定基础。

第 2 章将会讲述 memcached、Tokyo Tyrant、Redis 和 MongoDB 这 4 种 NoSQL 数据库，介绍这些 NoSQL 数据库各自的使用背景、特征和用例，以及它们的实际应用。通过本章的介绍，读者能够了解 NoSQL 数据库的基本使用方法。

第 3 章对上述 4 种 NoSQL 数据库的应用实例以及实现代码进行具体的介绍。这一章将使大家对这 4 种 NoSQL 数据库能解决的具体、实际的问题有所了解，让 NoSQL 数据库成为解决问题的选择之一。

虽然本章涉及的各个实例都可以用关系型数据库来实现，但是使用 NoSQL 数据库能够获得更快的响应，同时简单的操作也给使用者带来更大的便利。

第 4 章对上述 4 种 NoSQL 数据库的性能进行比较。本章不仅会进行与基本 CRUD 处理（创建、检索、更新、删除）相关的性能比较，而且还会着眼于一些具体实例，例如像 Tokyo Tyrant 的 addint 方法和 incr 方法的性能比较，Redis 的 list 类型的插入和删除的性能比较，以及 MySQL 的 join 和 MongoDB 的 embed 的性能比较等。

第 5 章对 NoSQL 数据库在实际应用中的问题点，以及 HandlerSocket 解决方案进行介绍。它虽然和 NoSQL 数据库略有一点不同，却是个非常有意思的解决方案，有利于读者们开阔视野。

版 权 声 明

目　　录

NoSQL 数据库的基础知识

本章首先介绍关系型数据库的起源、特征，以及它的优缺点，进而引出 NoSQL 数据库。

然后介绍 NoSQL 数据库的种类，以及如何更好地区别使用关系型数据库和 NoSQL 数据库。

1.1 关系型数据库和NoSQL数据库

1.1.1 什么是NoSQL

大家有没有听说过“NoSQL”呢？近年，这个词极受关注。看到“NoSQL”这个词，大家可能会误以为是“No! SQL”的缩写，并深感诧异：“SQL怎么会没有必要了呢?”但实际上，它是“Not Only SQL”的缩写。它的意义是：**适用关系型数据库[①]的时候就使用关系型数据库，不适用的时候也没有必要非使用关系型数据库不可，可以考虑使用更加合适的数据存储。**

为弥补关系型数据库的不足，各种各样的NoSQL数据库应运而生。

为了更好地了解本书所介绍的NoSQL数据库，对关系型数据库的理解是必不可少的。那么，就让我们先来看一看关系型数据库的历史、分类和特征吧。

1.1.2 关系型数据库简史

1969年，埃德加·弗兰克·科德（Edgar Frank Codd）发表了一篇划时代的论文，首次提出了关系数据模型的概念。但可惜的是，刊登论文的“IBM Research Report”只是IBM公司的内部刊物，因此论文反响平平。1970年，他再次在刊物《*Communication of the ACM*》上发表了题为“A Relational Model of Data for Large Shared Data banks”（大型共享数据库的关系模型）的论文，终于引起了大家的关注。

科德所提出的关系数据模型的概念成为了现今关系型数据库的基础。当时的关系型数据库由于硬件性能低劣、处理速度过慢而迟迟没有得到实际应用。但之后随着硬件性能的提升，加之使用简单、性能优越等优点，关系型数据库得到了广泛的应用。

---

① 关系数据库包括Oracle、Microsoft SQL、DB2、PostgreSQL等，本书主要指MySQL。

1.1.3 数据库的分类

数据库根据不同的数据模型（数据的表现形式）主要分成阶层型、网络型和关系型 3 种。

阶层型数据库

早期的数据库称为阶层型数据库，数据的关系都是以简单的树形结构来定义的。程序也通过树形结构对数据进行访问。这种结构，父记录（上层的记录）同时拥有多个子记录（下层记录），子记录只有唯一的父记录。正因为如此，这种非常简单的构造在碰到复杂数据的时候往往会造成数据的重复（同一数据在数据库内重复出现），出现数据冗余的问题。图 1-1 所示为阶层型数据库。

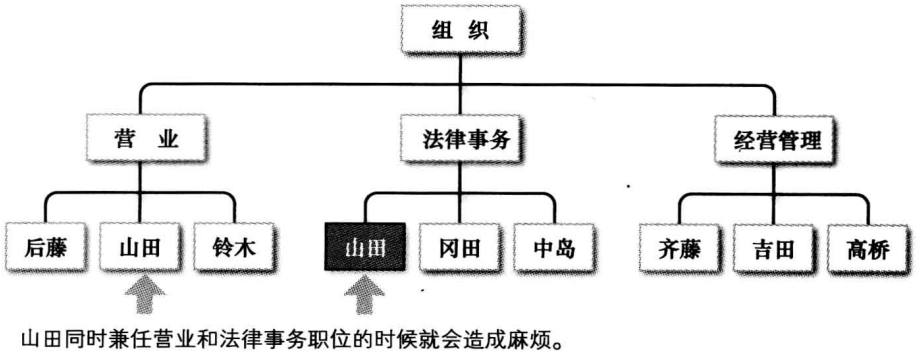

图 1-1 阶层型数据库示例

阶层型数据库把数据通过阶层结构的方式表现出来，虽然这样的结构有利于提高查询效率，但与此相对应的是，不理解数据结构就无法进行高效的查询。当然，在阶层结构发生变更的时候，程序也需要进行相应的变更。

网络型数据库

如前所述，阶层型数据库会带来数据重复的问题。为了解决这个问题，就出现了网络型数据库。它拥有同阶层型数据库相近的数据结构，同时各种数据又如同网状交织在一起，因此而得名。

阶层型数据库只能通过父子关系来表现数据之间的关系。针对这一不足，网络型数据库可以使子记录同时拥有多个父记录，从而解决了数据冗余的问题。图 1-2 所示为网络型数据库。

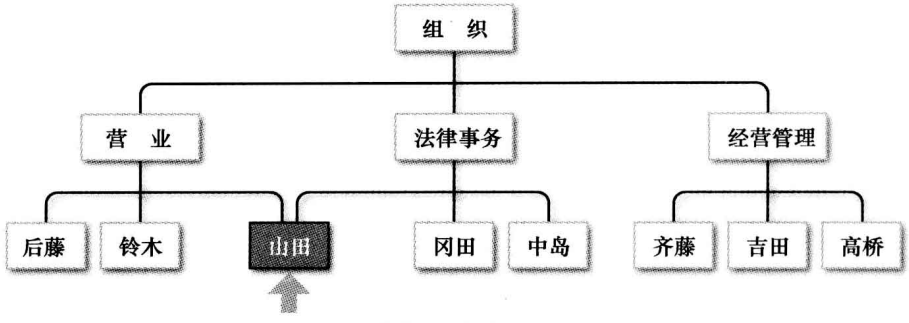

图 1-2 网络型数据库事例

但是，在网络型数据库中，数据间比较复杂的网络关系使得数据结构的更新变得比较困难。另外，与阶层型数据库一样，网络型数据库对数据结构有很强的依赖性，不理解数据结构就无法进行相应的数据访问。

关系型数据库

最后要向大家介绍的是以科德提出的关系数据模型为基础的关系型数据库。关系型数据库把所有的数据都通过行和列的二元表现形式表示出来，给人更容易理解的直观感受。网络型数据库存在着数据结构变更困难的问题，而关系型数据库可以使多条数据根据值来进行关联，这样就使数据可以独立存在，使得数据结构的变更变得简单易行。

对于阶层型数据库和网络型数据库，如果不理解相应的数据结构，就无法对数据进行读取，它们对数据结构的依赖性很强。因此，它们往往需要专业的工程师使用特定的计算机程序进行操作处理。相反，关系型数据库将作为操作对象的数据和操作方法（数据之间的关联）分离开来，消除了对数据结构的依赖性，让数据和程序的分离成为可能。这使得数据库可以广泛应用于各个不同领域，进一步扩大了数据库的应用范围。

参考资料

《RDB技术》（日经BP《ITpro》“技术广场”2004/07/27）
http://itpro.nikkeibp.co.jp/members/NBY/techsquare/20040716/1/
沖冠吾《从零开始学数据建模》（ITmedia《@IT》“Database Expert”2008/9/10）
http://www.atmarkit.co.jp/fdb/index/subindex/basic_modeling.html
铃木浩司《数据库有很多种，你能全都说出来吗?》(朝日互动 ZDNet Japan | builder 2007/09/11)
http://builder.japan.zdnet.com/sp/07database/story/0,3800082819,20356199,00.htm

1.1.4 关系型数据库的优势

通用性及高性能

虽然本书是讲解 NoSQL 数据库的，但有一个重要的大前提，请大家一定不要误解。这个大前提就是“关系型数据库的性能绝对不低，它具有非常好的通用性和非常高的性能”。毫无疑问，对于绝大多数的应用来说它都是最有效的解决方案。

突出的优势

关系型数据库作为应用广泛的通用型数据库，它的突出优势主要有以下几点：

- **保持数据的一致性（事务处理）**
- **由于以标准化为前提，数据更新的开销很小（相同的字段基本上都只有一处）**
- **可以进行 JOIN 等复杂查询**
- **存在很多实际成果和专业技术信息（成熟的技术）**

这其中，能够保持数据的一致性是关系型数据库的最大优势。在需要严格保证数据一致性和处理完整性的情况下，用关系型数据库是肯定没有错的。但是有些情况不需要 JOIN，对上述关系型数据库的优点也没有什么特别需要，这时似乎也就没有必要拘泥于关系型数据库了。

1.1.5 关系型数据库的不足

不擅长的处理

就像之前提到的那样，关系型数据库的性能非常高。但是它毕竟是一个通用型的数据库，并不能完全适应所有的用途。具体来说它并不擅长以下处理：

- **大量数据的写入处理**
- **为有数据更新的表做索引或表结构（schema）变更**
- **字段不固定时应用**
- **对简单查询需要快速返回结果的处理**

下面逐一进行详细的说明。

大量数据的写入处理

在数据读入方面，由复制产生的主从模式（数据的写入由主数据库负责，数据的读入由从数据库负责），可以比较简单地通过增加从数据库来实现规模化。但是，在数据的写入方面却完全没有简单的方法来解决规模化问题[①]。例如，要想将数据的写入规模化，可以考虑把主数据库从一台增加到两台，作为互相关联复制的二元主数据库来使用。确实这样似乎可以把每台主数据库的负荷减少一半，但是更新处理会发生冲突（同样的数据在两台服务器同时更新成其他值），可能会造成数据的不一致。为了避免这样的问题，就需要把对每个表的请求分别分配给合适的主数据库来处理，这就不那么简单了。图 1-3 所示为两台主机问题。图 1-4 所示为二元主数据库问题的解决办法。

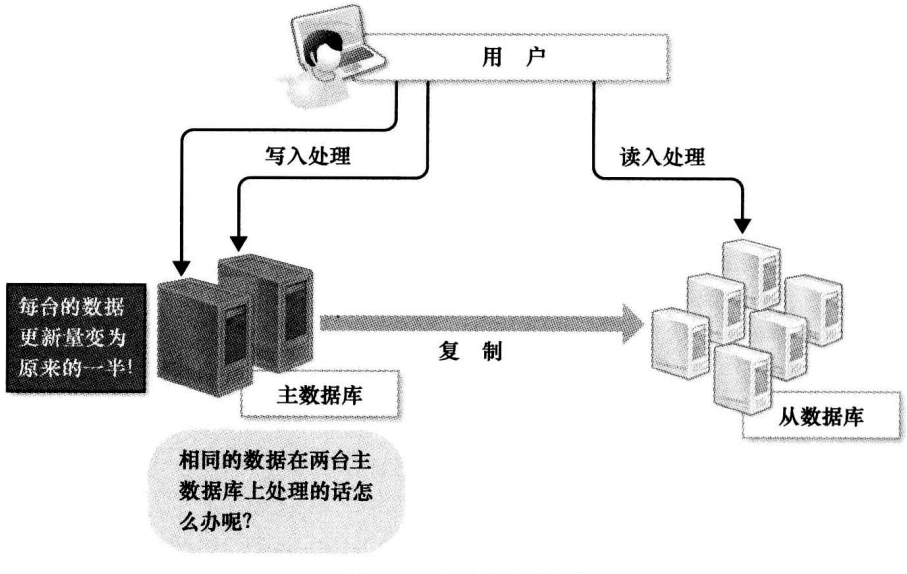

图 1-3 两台主机问题

另外也可以考虑把数据库分割开来，分别放在不同的数据库服务器上，比如将这个表放在这个数据库服务器上，那个表放在那个数据库服务器上。数据库分割可以减少每台数据库服务器上的数据量，以便减少硬盘 I/O（输入/输出）处理，实现内存上的高速处理，效果非常显著。但是，由于分别存储在不同服务器上的表之间无法进行 JOIN 处理，数据库分割的时候就需要预先考虑这些问题。数据库分割之

① 读写集中在一个数据库上让数据库不堪重负，大部分网站开始使用主从复制技术来实现读写分离，以提高读写性能和读库的可扩展性。Mysql 的 master-slave 模式成为了这个时候的网站标配。——译者注

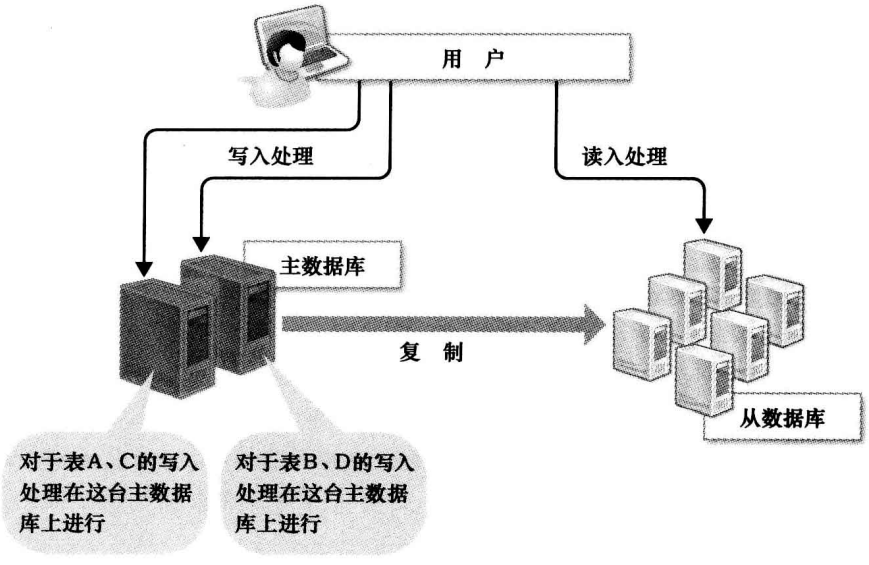

图 1-4 二元主数据库问题的解决办法

后，如果一定要进行 JOIN 处理，就必须要在程序中进行关联，这是非常困难的。图 1-5所示为数据库分割。

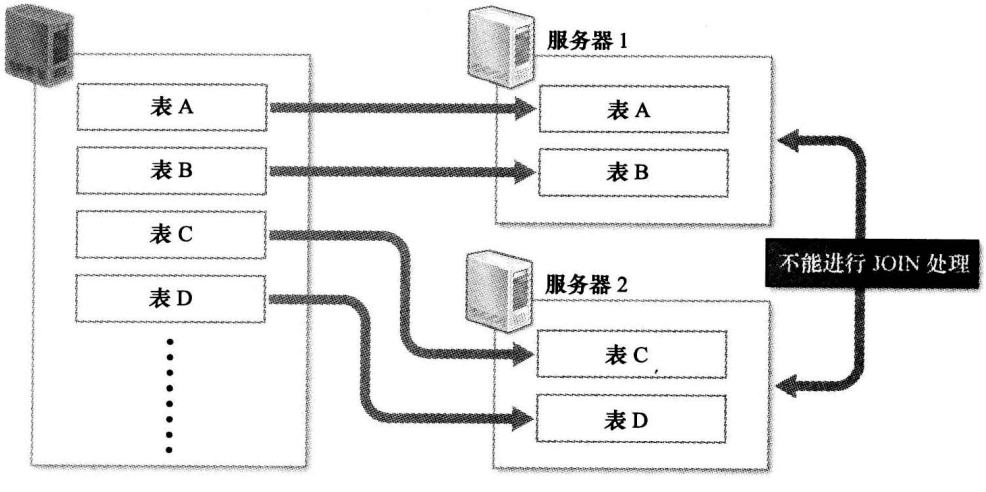

图 1-5 数据库分割

为有数据更新的表做索引或表结构（schema）变更

在使用关系型数据库时，为了加快查询速度需要创建索引，为了增加必要的字段就一定需要改变表结构。为了进行这些处理，需要对表进行共享锁定，这期间数