



高等院校计算机类规划教材  
国家新闻出版改革发展项目库入库项目  
数据科学与大数据技术专业教材丛书

# NoSQL 数据库技术

袁燕妮 编著



北京邮电大学出版社  
[www.buptpress.com](http://www.buptpress.com)

# 大数据顾问委员会

宋俊德 王国胤 张云勇 郑宇  
段云峰 田世明 娄瑜 孙少隣  
王柏

## 大数据专业教材编委会

总主编：吴斌

编委：宋美娜 欧中洪 鄂海红 双锴

于艳华 周文安 林荣恒 李静林

袁燕妮 李劼 皮人杰

总策划：姚顺

秘书长：刘纳新

大数据时代,同时伴随着物联网、人工智能、区块链技术的高速发展,传统关系型数据库在很多业务场景中显得力不从心,NoSQL 数据库系统应用越来越广泛。本书按照北京邮电大学数据科学与大数据技术专业培养方案中课程体系内容设计要求,围绕 OBE 人才培养理念,充分考虑本专业人才培养目标在问题分析能力、研究能力、使用现代工具能力、终身学习能力四方面的要求,主要针对大数据存储与管理能力培养需求设计知识体系内容,理论与实践并重,循序渐进。本书通过具体应用场景示例培养学生的学习兴趣,通过理论与实践相结合的方法使学生掌握主流 NoSQL 数据库的典型特征及适用场景,掌握基于 NoSQL 数据库进行大数据存储与管理的基本技术与方法,了解 NoSQL 数据库技术的新进展。

本书共 12 章,包括 NoSQL 数据库概述、图数据库技术、Neo4j 图数据库、文档数据库技术、MongoDB 文档数据库、键值类数据库技术、Redis 键值数据库、列族数据库技术、Cassandra 列族数据库、NoSQL 数据库访问技术、其他类型的 NoSQL 数据库以及区块链数据存储技术。各章节间关系如图 1 所示。



图 1 本书知识内容组成及各章节关系

读者在学习时,通过第 1 章先对 NoSQL 数据库技术有个整体认识,第 2 章~第 9 章每两章为一个模块,分别先掌握某类数据库的基础理论与知识,再就某一种具体类型的数据库学习掌握其安装及数据操作管理细节知识。第 10 章学习基于 Java、Python 语言的四类数据库编程访问技术。第 11 章主要基于时序数据库、RDF 数据库、搜索引擎学习其他类型的 NoSQL 数据库技术。第 12 章简单介绍区块链存储关键技术。本书以实际应用为出发点,精心组织内容,四类数据库基于典型应用场景设计具体示例,并配有思考与练习题。课件和课堂示例源代码等可通过北京邮电大学出版社网站(<http://www.buptpress.com/>)下载,极大地方便了教

与学。通过实践练习,学生可掌握四大类主流 NoSQL 数据库应用与管理技术基础知识。另外,读者也可扫描本书中的二维码学习知识点相关扩展内容或下载相关资源,以提高学习效率。

学习 NoSQL 数据库技术需要先修的专业课程有:数据科学导论、大数据技术基础、数据库理论与技术等。按照 1 周 2 学时,开设 16 周,共计 32 学时的课程计划,教学进度建议如表 1 所示。

表 1 32 学时教学计划

周	教学内容	实践内容
1	第 1 章 NoSQL 数据库概述	
2	第 2 章 图数据库技术	
3	第 3 章 Neo4j 图数据库	
4	实践:Neo4j 图数据库实践	Neo4j 安装与数据管理操作实践
5	第 4 章 文档数据库技术	
6	第 5 章 MongoDB 文档数据库	
7	实践:MongoDB 文档数据库实践	MongoDB 安装与数据管理操作实践
8	第 6 章 键值类数据库技术	
9	第 7 章 Redis 键值数据库	
10	实践:Redis 键值数据库实践	Redis 安装与数据管理操作实践
11	第 8 章 列族数据库技术	
12	第 9 章 Cassandra 列族数据库	
13	实践:Cassandra 列族数据库实践	Cassandra 安装与数据管理操作实践
14	第 10 章 NoSQL 数据库访问技术	
15	第 11 章 其他类型的 NoSQL 数据库	
16	第 12 章 区块链数据存储技术	

课时安排上或许有些紧张,建议授课老师根据教学内容要求进行调整,可略过部分章节。作者喜欢研讨式教学,所以在每章后面增加了调研分析、思考类习题,希望学生课堂上在老师引领掌握基础理论、技术知识的基础上,课外能够充分发挥自己的能动性,主动深入扩展相关理论与技术知识面,提高自己提出问题、解决问题的能力。老师可以适当针对某些习题采用翻转课堂的方法,组织课堂讨论,也可以将学生的自主学习过程与效果纳入课程考评体系。

本书的出版首先要感谢母校北京邮电大学计算机学院、山东大学计算机学院的培养及一路走来有缘相识的各位恩师的谆谆教导;特别要感谢北京邮电大学数据科学与大数据技术专业负责人吴斌教授、王柏教授的大力支持和鼓励;感谢从事 NoSQL 数据库技术相关研究的科研人员、企业及互联网知识论坛贡献者们,推动了 NoSQL 数据库技术的诞生与发展;还要诚挚感谢合作过、指导我、帮助我、关心我的各位领导、同事、同学和朋友,感谢每个项目团队中一起奋斗过的、努力过的研究生们。作为北京邮电大学数据科学与大数据技术专业新教材系列的编撰者之一,倍感荣耀,这也是我完成此书的强大动力。

我要感谢我的家人,谢谢他们的理解和鼓励,正是他们默默的支持,使我能够静下心来研究梳理 NoSQL 数据库技术相关知识,编著本书。

最后,感谢各位读者选择了本书,希望本书能对读者的学习有所助益。

由于时间仓促,涉及的新技术范围广,加之作者水平有限,书中错误、疏漏及不足之处在所难免,敬请读者批评指正。

袁燕妮  
于北京邮电大学

第 1 章 NoSQL 数据库概述 .....	1
1.1 什么是 NoSQL? .....	2
1.2 NoSQL 的分类与特点 .....	4
1.3 CAP 定理 .....	8
1.4 ACID 与 BASE .....	10
1.5 最终一致性技术基础 .....	11
1.5.1 一致性问题 .....	11
1.5.2 Quorum 的 NWR 策略 .....	11
1.5.3 Paxos 算法简介 .....	12
1.5.4 Raft 算法简介 .....	13
1.5.5 向量时钟机制 .....	16
1.6 数据复制与分片 .....	18
1.7 NoSQL 数据库与云计算 .....	18
1.8 NoSQL 数据库与物联网 .....	19
1.9 NoSQL 数据库与区块链 .....	20
1.10 本章小结 .....	21
1.11 思考与练习题 .....	21
本章参考文献 .....	22
第 2 章 图数据库技术 .....	23
2.1 概述 .....	24
2.2 图关键数据存储逻辑架构 .....	25
2.3 属性图模型 .....	28
2.4 图数据库查询语言 .....	31
2.4.1 Cypher 基础 .....	31

2.4.2 Gremlin .....	36
2.5 图数据库的应用 .....	38
2.6 本章小结 .....	39
2.7 思考与练习题 .....	40
本章参考文献 .....	41
<b>第3章 Neo4j 图数据库</b> .....	<b>42</b>
3.1 Neo4j DB 基础 .....	43
3.1.1 下载与安装 .....	43
3.1.2 数据类型 .....	47
3.2 Neo4j 数据操作基础 .....	48
3.2.1 节点操作 .....	48
3.2.2 关系操作 .....	55
3.2.3 排序与聚合操作 .....	62
3.2.4 路径操作 .....	65
3.2.5 索引操作 .....	68
3.2.6 约束操作 .....	70
3.2.7 存储过程调用 .....	71
3.3 Neo4j 集群技术 .....	73
3.3.1 因果集群 .....	73
3.3.2 高可用性集群 .....	76
3.4 Neo4j 管理与监控 .....	78
3.4.1 图数据导入 .....	78
3.4.2 图数据备份与恢复 .....	81
3.4.3 事务管理 .....	82
3.4.4 监控与日志 .....	84
3.5 本章小结 .....	85
3.6 思考与练习题 .....	86
本章参考文献 .....	87
<b>第4章 文档数据库技术</b> .....	<b>89</b>
4.1 概述 .....	89
4.2 文档数据库存储架构 .....	92
4.2.1 JSON 与 BSON .....	92
4.2.2 MongoDB 数据存储 .....	93

4.2.3 MongoDB 文档唯一性机制 .....	95
4.3 文档数据库管理基础 .....	95
4.3.1 数据库操作 .....	95
4.3.2 集合操作 .....	96
4.3.3 文档操作 .....	97
4.4 文档数据库的应用 .....	98
4.5 本章小结 .....	99
4.6 思考与练习题 .....	100
本章参考文献 .....	101
<b>第 5 章 MongoDB 文档数据库 .....</b>	<b>102</b>
5.1 MongoDB 基础 .....	103
5.1.1 下载与安装 .....	103
5.1.2 数据类型 .....	106
5.1.3 常用操作符 .....	107
5.2 MongoDB 文档操作基础 .....	108
5.2.1 文档 CRUD 操作 .....	108
5.2.2 文档链接引用 .....	115
5.2.3 文档聚合与管道操作 .....	116
5.2.4 MongoDB 索引机制 .....	118
5.3 MongoDB 数据库架构 .....	125
5.3.1 分片机制与集群架构 .....	125
5.3.2 数据冗余复制集 .....	127
5.3.3 分布式文件存储 .....	129
5.3.4 Journaling 日志功能 .....	130
5.4 MongoDB 管理与监控 .....	131
5.4.1 文档数据导入与导出 .....	131
5.4.2 文档数据备份与恢复 .....	131
5.4.3 多文档事务管理 .....	132
5.4.4 数据库监控 .....	133
5.5 本章小结 .....	133
5.6 思考与练习题 .....	134
本章参考文献 .....	134
<b>第 6 章 键值类数据库技术 .....</b>	<b>135</b>
6.1 概述 .....	135

6.2	键值数据存储逻辑架构 .....	137
6.3	键值数据库管理基础 .....	141
6.3.1	持久化管理 .....	141
6.3.2	分区机制 .....	142
6.3.3	集群监控 .....	143
6.4	键值数据库的应用 .....	144
6.5	本章小结 .....	145
6.6	思考与练习题 .....	146
	本章参考文献 .....	146
<b>第7章</b>	<b>Redis 键值数据库</b> .....	<b>147</b>
7.1	Redis DB 基础 .....	147
7.1.1	下载与安装 .....	147
7.1.2	操作命令的分类 .....	149
7.2	Redis 键值管理操作 .....	150
7.2.1	Key 操作 .....	150
7.2.2	字符串 .....	151
7.2.3	列表 .....	153
7.2.4	集合 .....	154
7.2.5	散列 .....	155
7.2.6	有序集合 .....	157
7.2.7	发布与订阅 .....	158
7.3	Redis 集群架构及管理 .....	160
7.4	Redis 管理与监控 .....	161
7.4.1	Redis 数据库配置管理 .....	161
7.4.2	Redis 数据备份与恢复 .....	163
7.4.3	Redis 命令批量执行 .....	164
7.4.4	Redis 图形化管理工具 .....	165
7.5	本章小结 .....	167
7.6	思考与练习题 .....	167
	本章参考文献 .....	168
<b>第8章</b>	<b>列族数据库技术</b> .....	<b>169</b>
8.1	概述 .....	169
8.2	列族数据存储逻辑架构 .....	171

8.3 列族数据库集群架构 .....	175
8.3.1 Cassandra 对等式集群架构 .....	175
8.3.2 Cassandra 分区策略 .....	177
8.3.3 持久化管理 .....	178
8.4 列族数据库的应用 .....	178
8.5 本章小结 .....	179
8.6 思考与练习题 .....	179
本章参考文献 .....	180
<b>第 9 章 Cassandra 列族数据库 .....</b>	<b>181</b>
9.1 Cassandra DB 基础 .....	182
9.1.1 下载与安装 .....	182
9.1.2 数据类型 .....	184
9.1.3 CQL 简介 .....	187
9.2 CQL 数据管理操作 .....	189
9.2.1 键空间操作 .....	189
9.2.2 数据表操作 .....	190
9.2.3 数据 CRUD 操作 .....	193
9.2.4 索引操作 .....	196
9.2.5 函数支持 .....	197
9.3 Cassandra 集群管理 .....	199
9.3.1 读写一致性管理 .....	199
9.3.2 节点状态管理机制 .....	200
9.3.3 节点增删管理 .....	201
9.4 Cassandra 数据库管理 .....	202
9.4.1 数据库配置管理 .....	202
9.4.2 批量导入与导出 .....	203
9.4.3 备份与恢复 .....	205
9.5 本章小结 .....	206
9.6 思考与练习题 .....	206
本章参考文献 .....	207
<b>第 10 章 NoSQL 数据库访问技术 .....</b>	<b>208</b>
10.1 概述 .....	208
10.2 基于 Python 访问 Neo4j .....	210

10.3 基于 Python 访问 MongoDB .....	211
10.4 基于 Python 访问 Redis .....	212
10.5 基于 Python 访问 Cassandra .....	213
10.6 本章小结 .....	214
10.7 思考与练习题 .....	214
本章参考文献 .....	214
<b>第 11 章 其他类型的 NoSQL 数据库 .....</b>	<b>216</b>
11.1 概述 .....	217
11.1.1 时序数据库 .....	217
11.1.2 RDF 数据库 .....	217
11.1.3 搜索引擎 .....	219
11.2 InfluxDB 时序数据库简介 .....	219
11.3 Apache Jena-TDB 简介 .....	220
11.4 Elasticsearch 简介 .....	221
11.5 本章小结 .....	222
11.6 思考与练习题 .....	222
本章参考文献 .....	223
<b>第 12 章 区块链数据存储技术 .....</b>	<b>224</b>
12.1 概述 .....	224
12.2 区块链数据存储技术 .....	226
12.3 区块链技术的应用 .....	227
12.4 本章小结 .....	228
12.5 思考与练习 .....	228
本章参考文献 .....	228

# 第 1 章

## NoSQL 数据库概述

NoSQL(Not Only SQL)泛指非关系型数据库,与传统的用于事务型管理的关系型数据库存在很大的不同,特别是在大数据时代,NoSQL 技术是实现大数据分布式存储、高性能访问与管理,用于支撑互联网+、智慧+等大规模、创新型应用的关键技术,根据不同类型大数据的特点分别提供相适应的存储与管理技术。本章主要从 NoSQL 数据库的基本概念、分类、特点、CAP 等理论及其与云计算、物联网、区块链技术的关系角度从整体上来初步认识 NoSQL 技术体系。

本章内容思维导图如图 1-1 所示。

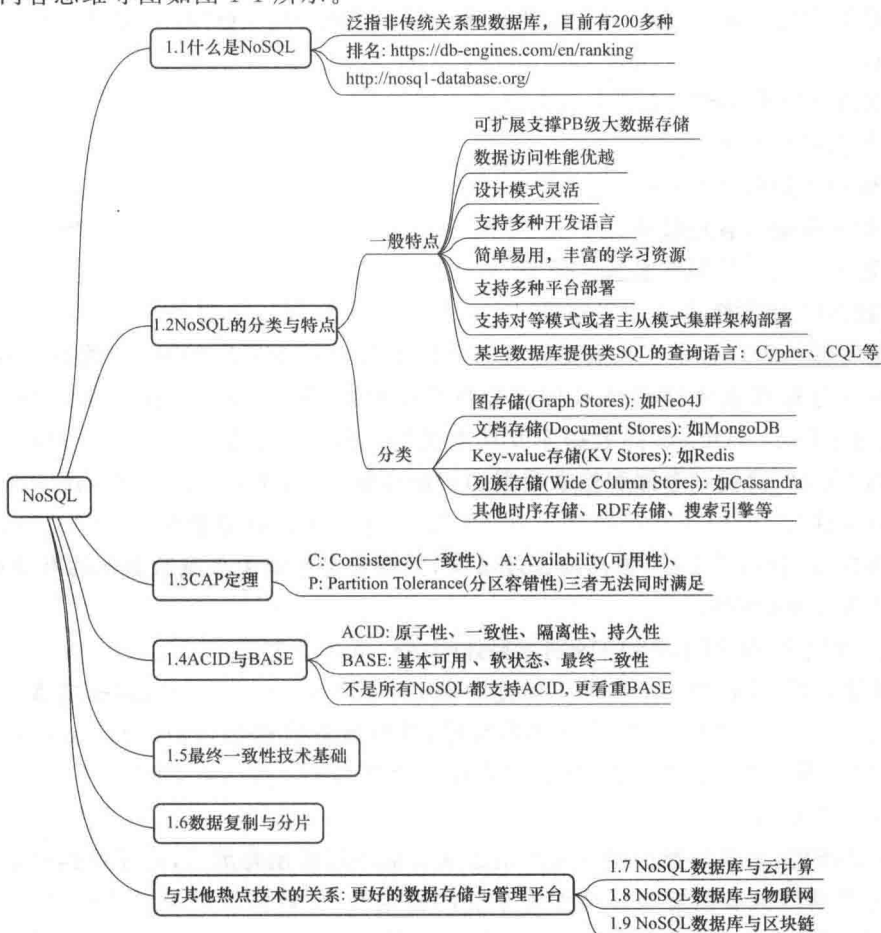


图 1-1 NoSQL 数据库概述章节内容思维导图

## 1.1 什么是 NoSQL?

“Big Data”(大数据)时代的来临已经毋庸置疑,数据已成为各个国家的基础性战略资源,是不同行业企事业单位的重要资产,同时也是不同国家科研组织机构创新的重要出发点和驱动力。大数据正日益对全球生产、流通、分配、消费活动以及经济运行机制、社会生活方式和国家治理能力产生重要影响。特别是全球各个领域纷纷通过“+互联网”“+物联网”“+AI”等技术手段提升现有信息化系统的数据采集与智慧分析支撑能力,业务数据不断累积,经过大数据处理与分析,分析结果反作用于业务系统服务能力的提高,这个过程循环迭代式演进,进一步激发和促进了数据科学与大数据技术的高速发展。大数据本身依托于国民经济各个领域的应用系统而生,服务于社会与个体生产与生活的各个环节和活动,将面临更加严峻、迫切的高效存储与管理新技术的重大挑战。

传统的关系数据库管理系统(Relational Database Management System, RDBMS)面对快速增长的数据规模和日渐复杂的数据类型,渐渐力不从心,无法应对很多 PB 级及以上数据库处理任务。特别是在支撑 Web 2.0 网站应用开发时, RDBMS 暴露了很多难以克服的问题。Web 2.0 网站要根据用户个性化信息来实时生成动态页面和提供动态信息,数据库并发负载非常高。对于关系数据库来说,在 PB 级海量信息中进行 SQL 查询,效率是极其低下甚至是不可忍受的。

大数据技术体系一般包含以下五方面:

- 大数据采集技术;
- 大数据预处理技术;
- 大数据存储与管理技术;
- 大数据分析与挖掘技术;
- 大数据应用技术。

大数据存储与管理技术在大数据处理与分析技术栈中起着承上启下的重要作用。大数据存储主要分为分布式文件存储技术与适合弹性横向扩展部署的 NoSQL 类型数据库技术。NoSQL 是对不同于 RDBMS 的数据库管理系统的统称。NoSQL 一词最早出现于 1998 年, Carlo Strozzi 提出“要找到存储和检索数据的其他高效途径,而不是在任何情况下都把关系数据库当作万金油”。2009 年在亚特兰大举行的“no:sql(east)”讨论会对 NoSQL 给出一个被普遍接受的解释是“非关系型的”数据库管理系统,强调它是传统关系型数据库的有益补充,而不是替代传统关系型数据库。

NoSQL 数据库与 RDBMS 的区别主要体现在以下八个方面。

(1) 存储方式:关系型数据库是以规范的表格形式存储的,表中一行数据包含很多存储属性值的字段。行与行之间很容易关联协作存储,按行提取数据很方便。而 NoSQL 数据库则与其相反,往往纵向按列大块数据存储,或者按照数据本身键值对特点、文档特点、图结构特点进行非关系型存储。

(2) 存储结构:关系型数据库对应的是数据表结构需要预先定义,描述数据存储的形式和内容。预定义结构虽然能够很好地保障系统的稳定性,但数据模型灵活性较差,修改比较困难,在存储大规模数据时,容易存在稀疏列问题。而 NoSQL 数据库基于动态结构,数据存储

结构可以根据需求灵活变化,不要求所有行的数据属性结构都一致。

(3) 存储规范:关系型数据库为了降低数据冗余,需要满足一定的存储范式要求,把数据分割为一系列的关系表以避免重复,获得精简的空间利用。虽然管理起来很清晰,但是单个操作如果涉及多张表的时候,数据管理就显得比较麻烦。而 NoSQL 数据存储为了支撑大数据的高性能访问,大多采用牺牲空间换时间的方式,模式设计适当引入数据冗余,对数据存储没有严格规范要求,更加有利于大规模数据管理操作。

(4) 存储扩展:从可扩展性角度来看,关系型数据库一般采用纵向扩展,也就是说想要提高处理能力,要使用速度更快的计算机,好比轻型卡车拉不动货的时候就换重型卡车拉货。因为数据存储在关系表中,操作的性能瓶颈可能涉及多个表,需要通过提升计算机性能来克服。虽然有很大的扩展空间,但是最终会达到纵向扩展的上限。而 NoSQL 数据库采用的则是横向扩展方式,它的存储集群部署架构天然支持分布式的,可以通过给资源池添加更多的一般数据库服务器来分担负载。好比一只蚂蚁完不成任务的时候,就集结成千上万只蚂蚁一起完成任务。

(5) 查询方式:关系型数据库通过结构化查询语言 SQL 来操作数据库。SQL 支持数据库 CURD 操作的功能非常强大,是业界的标准用法。而 NoSQL 查询以块为单元操作数据,使用的是非结构化查询语言,目前还没有统一标准。

(6) 事务管理:关系型数据库遵循原子性(Atomicity)、一致性(Consistency)、隔离性(Isolation)、持久性(Durability),即 ACID 规则,支持数据强一致性管理,对事务的支持很好。而在 NoSQL 分布式数据库中强调基本可用(Basically Available)、软状态(Soft-state)、最终一致性(Eventual Consistency),即 BASE 原则。虽然有的 NoSQL 数据库也可以用于事务管理,但这并不是 NoSQL 的闪光点。

(7) 性能:关系型数据库为了维护数据的一致性付出了巨大的代价,在面对高并发读写任务时性能非常差,尤其是面对大数据的时候效率非常低。而 NoSQL 存储的格式本质上大都是 Key-Value 类型的,综合利用内存存储、数据副本、负载均衡等机制来保障大数据访问性能,而且遵循数据最终一致性要求,所以高性能是 NoSQL 数据库与生俱来的优势。

(8) 授权方式:关系型数据库通常有 SQL Server、MySQL、Oracle、DB2 等,大多数企业级应用关系型数据库都是付费的,并且价格昂贵,成本很高。而 NoSQL 数据库目前大多数都是开源的,相对成本较低。

总之,NoSQL 数据库除了成本相对低外,突出的优点主要是灵活、可扩展、高性能和功能强大,并提供卓越的用户体验,以上四方面特点分别描述如下。

- 灵活性:NoSQL 数据库通常提供灵活的模型架构,以实现更快速的迭代开发。灵活的数据模型使 NoSQL 数据库成为半结构化和非结构化数据的理想之选。
- 可扩展性:相比较采用更加昂贵和强大的服务器来满足需求的纵向扩展(Scale up)方式,NoSQL 数据库通常采用横向扩展(Scale out)的方式,即大多采用 X86 等相对廉价的服务器搭建集群的方式进行扩展,以满足性能访问需求。
- 高性能:NoSQL 数据库针对特定的数据模型(如列族、文档、键值和图形等)和访问模式进行了优化,这与尝试使用关系数据库完成类似功能相比可实现更高的性能。
- 功能强大:NoSQL 数据库提供功能强大的 API 和数据类型,专门针对其各自的数据模型而构建。

与 NoSQL 数据库的优势相对应的不足之处主要体现在:目前还缺乏统一的标准约束,不

同的数据库之间没有类似 RDBMS 中 SQL 一样的统一数据查询操作语言支持;强调最终一致性而不是强一致性。各个行业以事务管理为核心的应用系统中,主要是对少量业务数据的事务操作,关系型数据库依然处于主导地位,如客户关系管理系统、企业资源管理系统等。

大数据分析驱动的应用系统架构往往包含数据采集层、数据处理层、数据存储层、数据分析与应用呈现层,应用系统需求决定数据存储层技术选型,如系统可靠性要求怎么样、性能访问速度要求如何、数据存储结构满足什么特点等,再综合判断选择什么类型的数据库管理系统去支撑业务功能的设计与实现。

从各个应用领域整体应用系统架构讲,往往是以少量数据事务管理操作为主的 OLTP 型应用系统与批量数据操作为主的分析型应用系统相互服务,形成闭环的综合支撑系统架构,所以很多实际应用系统中一般同时采用 RDBMS 和一种或多种 NoSQL 数据库分别发挥各自所长,综合满足应用系统的业务支撑需求。

除了 RDBMS 和 NoSQL 数据库技术,近几年还出现了 NewSQL 数据库技术,其目标是结合传统关系型数据库与 NoSQL 数据库技术的优点,这类新式的关系型数据库管理系统,针对 OLTP(读-写)工作负载,追求提供与 NoSQL 系统相同的扩展性能,且仍然保持 ACID 和 SQL 等特性。NoSQL 与 NewSQL 目前都还处于快速研究发展阶段。

## 1.2 NoSQL 的分类与特点

近年来 NoSQL 技术迅猛发展,按照数据存储特点,NoSQL 数据库一般分为图存储、文档存储、键值存储、列族存储和其他类型共五大类。参考国际知名的数据库排名网站 DB-Engines Ranking 首页 2019 年 11 月的数据库分类,除 RDBMS 外,NoSQL 数据库的分类如表 1-1 所示。

表 1-1 NoSQL 数据库的分类

类型	Top 3 代表	特点
图存储 (Graph Stores)	Neo4j Microsoft Azure Cosmos DB (Multi-model) OrientDB(Multi-model)	图数据的最佳存储。相比使用传统关系数据库性能更优,存储模式设计与使用更加灵活、简单
文档存储 (Document Stores)	MongoDB Amazon DynamoDB(Multi-model) Couchbase	文档存储一般用类似 JSON (JavaScript Object Notation)的格式存储,存储的内容是文档型的嵌套结构。可以对某些字段建立索引,实现类似关系数据库的某些功能
Key-Value 存储	Redis(Multi-model) Amazon DynamoDB(Multi-model) Microsoft Azure Cosmos DB	可以通过 Key 快速查询到其 Value。支持多种类型值的存储
列族存储 (Wide Column Stores)	Cassandra HBase Microsoft Azure Cosmos DB (Multi-model)	顾名思义,是按列存储数据的。最大的特点是方便存储结构化和半结构化数据,对某一列或者某几列的查询有非常大的性能优势

续表

类型	Top 3 代表	特点
时序存储 (Time Series Stores)	InfluxDB Kdb+ Prometheus	时间序列数据库用于支撑时间序列数据的优化存储, 每个条目都有一个相关的时间戳。时间序列数据可以来自传感器、智能电表等, 或可以存储一个高频股票交易系统的股票价格波动情况
对象存储 (Object oriented Stores)	InterSystems Caché(Multi-model) Versant ObjectDB ObjectStore	通过类似面向对象语言的语法操作数据库, 通过对象的方式存取数据
XML 数据库	MarkLogic Oracle Berkeley DB Virtuoso	可以高效地存储 XML 数据, 并支持 XML 的内部查询语法, 如 XQuery、XPath 等。原生类 XML DB 有 BaseX 等
RDF 存储	MarkLogic(Multi-model) Virtuoso(Multi-model) Apache Jena-TDB	资源描述框架存储是一种信息的描述方法, 最初用于描述元数据。目前主要用于语义网、知识图谱的存储。RDF 存储主要以主语、谓语、宾语三元组形式表示信息
搜索引擎 (Search Engines)	Elasticsearch Splunk Solr	搜索引擎是用于数据内容搜索的 NoSQL 数据库管理系统。除了这种应用的一般优化, 专业化数据库通常还支持复杂搜索表达式、全文搜索、源搜索、搜索结果的排序和分组、空间搜索和高扩展性分布式搜索等功能
事件存储 (Event Stores)	Event Store IBM Db2 Event Store NEventStore	事件存储数据库是记录事件发生的数据库管理系统, 坚持所有的状态变化的事件都对应一个时间戳, 从而创建单个对象的时间序列。一个对象的当前状态可以通过回放, 推断对象从开始到当前时间的所有事件
多值数据库 (Multivalued DBMS)	Adabas UniData, UniVerse jBASE	多值数据库类似于关系数据库系统中的表存储数据, 但可以记录多个值。因为这违背了第一范式, 这些系统有时被称为 NF2(非第一范式)系统
内容存储 (Content Stores)	Jackrabbit ModeShape	多媒体数据库是专门管理数字内容的数据库管理系统, 如文本、图片或视频, 包括它们的元数据。除了通常使用 SQL 或 XPath 存储和查询, 多媒体数据库通常支持的功能是全文检索、版本控制、多媒体数据内容的存储与访问控制
导航数据库 (Navigational DBMS)	IMS IDMS	导航数据库只允许通过链接记录访问数据集

本书将主要基于前四类目前主流 NoSQL DB 来学习相关理论及技术基础知识, 这四类 NoSQL 数据库的特点概述如下。

### 1. 图数据库

世界万事万物之间存在着纷繁复杂的关联关系, 人与人、人与物、物与物通过时间、地域空间、动作事件等发生相互联系, 进行状态变迁。从技术角度来讲, 可以用图论中图的概念来抽

象表示复杂的关系网络。图结构数据如图 1-2 所示。图论、网络科学是图数据库重要的理论基础。图数据库可以高效支撑网状图形结构数据的管理与分析型应用,如微信、微博等社交网络应用场景,金融领域账务往来网络分析应用场景,复杂的交通网络、计算机网络、通信网络、智能电网、有线电视网、物联网等物理网络流量分析、安全分析、监控预警分析等场景。

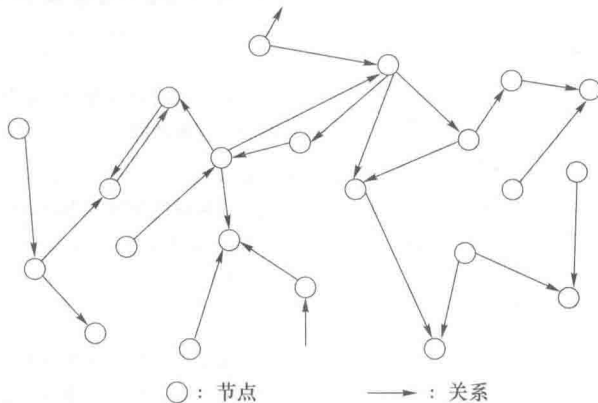


图 1-2 图结构数据示例

以社交平台为例,一个大 V 有少则十几万,多则几千万的粉丝,这些关注关系如果采用 RDBMS 存储,则需要将一条关注关系存为一条数据,每条数据主要包含大 V 标识、粉丝标识及关注时间等信息。整个平台用户数量往往上亿级别,仅仅关注关系轻松破亿、破十亿,甚至上百亿,并且为了保证每条数据的唯一性,还需要设置联合索引,RDBMS 就有些力不从心了。传统的性能优化技术,如根据 ID 哈希将数据分别存储到 100 张表中,可以起到提高查询一个用户有哪些粉丝性能的作用,但是查询一个用户关注了哪些人时仍然需要遍历全表。如果想查询类似好友的好友这种结果的话,RDBMS 需要通过多次表链接查询才能实现,性能往往满足不了业务需求。

应运而生的专门用于网络图数据存储的图数据库突出特点简介如下:

- 采用非关系的图数据结构直接存储数据,数据建模直接用节点与关系表示数据,节点间的关系可以带有描述属性,节点可以有不同类型的类别标签,节点与节点之间也可以存在多种不同类型的关系;
- 支持基于路径的高性能图数据遍历访问方法;
- 可横向扩展,支持分布式集群架构,能够支撑数十亿节点的高效访问;
- 支持 Java、Python 等多种开发语言;
- 支持跨平台部署、简单易用等。

## 2. 文档数据库

从 1989 年起, Lotus 通过其群件产品 Notes 提出了数据库技术的全新概念——文档数据库,文档数据库区别于传统的其他数据库,以文档的形式存储数据。在 RDBMS 中,相关信息被分割成离散的数据段,存放在不同的数据库表中,如有关客户的完整信息被分别存放在客户基本信息表、客户订单表、客户缴费信息表、客户投诉记录表等。文档数据库中文档是处理信息的基本单位。一个文档可以很长、很复杂,内容结构不固定,与字处理文档类似。一个文档相当于关系数据库中表的一条记录,但可以把一个客户相关的属性信息都放在一个文档里。一个描述客户信息的文档示例如下,数据存储方式非常类似 JSON。