

大数据管理系统 原理与技术

王宏志 何震瀛 王鹏 李春静 编著



DATABASE MANAGEMENT SYSTEMS FOR BIG DATA



机械工业出版社
China Machine Press

数据科学与工程丛书

DATABASE MANAGEMENT SYSTEMS FOR BIG DATA

大数据管理系统 原理与技术

王宏志 何震瀛 王鹏 李春静 编著



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

大数据管理系统原理与技术 / 王宏志等编著. —北京: 机械工业出版社, 2019.9
(数据科学与工程丛书)

ISBN 978-7-111-63677-9

I. 大… II. 王… III. 数据处理系统—研究 IV. TP274

中国版本图书馆 CIP 数据核字 (2019) 第 204630 号

本书介绍了多种数据库管理系统的基本概念以及代表性数据库管理系统的使用和优化方法, 覆盖了传统的关系数据库、数据仓库, 以及列族、键值、文档、图等 NoSQL 数据库系统。通过阅读本书, 读者可以较全面地了解支撑大数据应用所需的数据库管理系统的概念、特征和相关技术, 并且可以学习代表性关系数据库系统的使用方法, 将理论和实际相结合。

本书可作为大数据相关专业本科生和研究生教材, 也可供从事大数据相关工作的工程技术人员参考使用。

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 郎亚妹

责任校对: 殷虹

印刷: 北京诚信伟业印刷有限公司

版次: 2020 年 1 月第 1 版第 1 次印刷

开本: 185mm × 260mm 1/16

印张: 21.5

书号: ISBN 978-7-111-63677-9

定价: 69.00 元

客服电话: (010) 88361066 88379833 68326294
华章网站: www.hzbook.com

投稿热线: (010) 88379604
读者信箱: hzsj@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

本书编委会

顾 问：张云泉（中科院计算所）

编写委员会（按姓氏拼音排序）：

韩建军（华育兴业科技有限公司）

何震瀛（复旦大学）

李春静（华育兴业科技有限公司）

王宏志（哈尔滨工业大学）

王昆宇（华育兴业科技有限公司）

王 鹏（华育兴业科技有限公司）

张立臣（华育兴业科技有限公司）

前 言

大数据计算的重要支撑是有效管理数据，研发以数据为核心的应用系统的重要任务之一就是选择数据库管理系统，对其进行配置、编程和调优。由于大数据的多样性，单一的数据库管理系统难以满足所有类型数据管理的需要，因而除传统的成熟关系数据库产品之外，还出现了面向不同数据和应用特点的数据库管理系统，这使得数据库管理系统的选择和使用更加具有挑战性。因而，掌握多种数据库管理系统对于大数据领域从业者来说非常重要。针对这一需求，本书试图介绍面向大数据的数据库管理系统知识。

本书系统介绍了面向大数据的数据库管理系统的基本原理和应用，以及关系数据库、数据仓库、多种 NoSQL 数据库管理系统等。考虑到读者的多样性，本书针对不同种类的数据库介绍了基本原理、使用方法和案例。

本书兼顾深度和广度，选取典型数据库系统进行了介绍。对于关系数据库，介绍了其相关的基本概念以及典型关系数据库系统 MySQL。对于数据仓库，介绍了其基本概念和基于 Hadoop 的数据仓库系统 Hive。本书还重点介绍了 NoSQL 数据库系统，并选取比较典型的键值数据库、列族数据库、文档数据库和图数据库分别介绍了其原理与应用。

本书适用于本科生和研究生的数据库管理系统、大数据管理等相关课程，也可以作为大数据系统开发、分布式系统、数据库系统等课程的补充教材或课外读物。本书也可作为大数据技术方面的职业培训教材，同时，还可供大数据领域的从业人员参考使用。

针对不同层次的教学要求，教师可以从本书中选择不同的内容。如果教学偏重讲授原理，可以着重讲授本书的概念和原理部分，而将后面的具体系统作为实例。如果教学偏重讲授技术和应用，可以着重讲授本书中系统的使用和开发，而对各类数据库管理系统只介绍基本概念，原理部分的深度视学时数而定。需要注意的是，数据库系统原理及使用是相辅相成的：只有深入了解了原理，才能对数据库管理系统进行高效应用，对于大数据管理系统来说尤其如此；反之，只有真正应用了数据库管理系统，才能对相关的原理有深入的认知。

本书可以独立学习，但是建议读者有一些离散数学、计算机系统和数据库管理系统的先修知识。尽管本书对数据库知识进行了简单介绍，但是其深度和广度难以与专门的数据库管理系统教材相比，建议读者在学习第 2 章和第 3 章时与数据库管理系统教材相互参考。具有一些分布式系统的知识对本书第 4 ~ 8 章的学习会有所帮助，建议读者在学习这几章时与分

布式系统教材相互参考。

随着数据量和数据形态的不断变化，面向大数据的数据管理系统一直处于不断变化之中，尽管本书尽可能兼顾深度和广度，但是，限于笔者的水平，在内容安排、表述等方面的不当之处在所难免，敬请读者在阅读过程中不吝提出宝贵建议，以期改进本书。读者的任何意见和建议请发至邮箱 wangzh@hit.edu.cn，与本书相关的信息也会在微信公众号“大数据与数据科学家”(big_data_scientist)发布。

感谢哈尔滨工业大学的李建中教授、高宏教授以及海量数据计算研究中心诸位同事的指导，以及在专业上对我的帮助。

在本书撰写过程中，哈尔滨工业大学的张梦、陈翔等同学在资料翻译、搜集和整理以及文本校对、作图等多个方面提供了帮助和支持，在此表示感谢。

感谢 Neo4j 中国总代理松鼠山科技有限公司对图数据库部分的技术支持，其专业的知识使得这一部分的质量得到了提升。

感谢北京华育兴业科技有限公司的企业专家对本书编写提出了很多建议，并通过“教育部-华育兴业产学合作协同育人”项目在课程和教学资源建设方面提供了大力支持。

非常感谢我的爱人黎玲利副教授一直以来对我的支持，以及在大数据管理领域和我的探讨。在本书写作期间，感谢我的母亲和岳母帮忙料理家务，照顾我的宝宝“壮壮”，使我有时间从事本书的写作。

最后，关于大数据管理方面的研究和本书的写作还得到了国家自然科学基金项目（编号：U1509216，U1866602）和微软亚洲研究院的资助，在此一并表示感谢。

王宏志

2019年10月18日于哈尔滨

目 录

本书编委会

前言

第 1 章 绪论 1

1.1 大数据的基本概念 1

1.2 数据库管理系统 5

1.2.1 数据库管理系统的概念 5

1.2.2 数据库管理系统的发展历史 6

1.2.3 数据库管理系统的要素 10

1.3 大数据对数据库管理系统的需求
和挑战 13

1.4 本书结构 14

第 2 章 关系型数据库管理系统 15

2.1 关系数据库概述 15

2.1.1 关系模型 15

2.1.2 关系数据的存储 17

2.1.3 关系数据库的索引 20

2.1.4 关系数据库中的查询处理
算法 24

2.1.5 并发控制 25

2.1.6 数据库恢复 34

2.2 关系数据库 MySQL 概述 35

2.3 MySQL 应用 36

2.3.1 SQL 概述 36

2.3.2 数据定义语句 37

2.3.3 数据处理语句 38

2.3.4 事务和锁定声明 39

2.3.5 其他 45

2.4 存储过程 46

2.4.1 概述 46

2.4.2 建立存储过程 47

2.4.3 调用存储过程 49

2.4.4 查询存储过程 50

2.4.5 删除存储过程 50

2.5 视图 51

2.6 分区 53

2.7 复制 57

2.8 MySQL 的 Java 客户端 JDBC 62

2.8.1 JDBC 概述 62

2.8.2 JDBC API 63

2.8.3 Java 通过 JDBC API 操作
MySQL 66

第 3 章 数据仓库 Hive 72

3.1 数据仓库概述 72

3.1.1 数据仓库的概念和特征 72

3.1.2 数据仓库的体系结构 73

3.1.3 数据仓库的模型 74

3.1.4 数据仓库关键技术 77

3.1.5 数据仓库与大数据 79

3.2 Hive 概述 80

3.2.1 Hive 存储结构 80

3.2.2 Hive 体系结构 82

- 3.2.3 Hive 的任务执行流程 84
 - 3.3 Hive 的特征 85
 - 3.3.1 一致性 86
 - 3.3.2 可扩展性 86
 - 3.3.3 事务 86
 - 3.4 Hive 的基本概念 87
 - 3.4.1 基本数据类型 87
 - 3.4.2 数据类型转换 89
 - 3.4.3 复杂数据类型 90
 - 3.4.4 文本文件数据编码 91
 - 3.4.5 数据读取模式 92
 - 3.4.6 文件格式与压缩 93
 - 3.4.7 Hive 压缩 93
 - 3.4.8 Hive 关键字 95
 - 3.5 Hive 的使用 97
 - 3.5.1 Hive 命令 97
 - 3.5.2 Hive DDL 100
 - 3.5.3 Hive DML 104
 - 3.5.4 HiveQL 基本查询 108
 - 3.5.5 Hive 函数 118
 - 3.5.6 HiveQL 高级查询 121
 - 3.6 面向大数据的优化策略 133
 - 3.6.1 分桶 134
 - 3.6.2 视图和索引 136
 - 3.6.3 模式设计 139
 - 3.7 Hive 的调优 144
 - 3.7.1 使用 EXPLAIN 查看执行计划 145
 - 3.7.2 Hive 配置管理 147
 - 3.7.3 限制调整 148
 - 3.7.4 JOIN 优化 149
 - 3.7.5 本地模式 155
 - 3.7.6 并行执行 157
 - 3.7.7 严格模式 158
 - 3.7.8 调整 Mapper 和 Reducer 个数 159
 - 3.7.9 JVM 重用 161
 - 3.7.10 动态分区调整 162
 - 3.7.11 推测执行 163
 - 3.7.12 单个 MapReduce 中的多个 GROUP BY 164
 - 3.7.13 虚拟列 164
 - 3.8 Java 通过 JDBC 操作 Hive 165
- 第 4 章 NoSQL 概述 168**
- 4.1 NoSQL 与非关系型数据库 168
 - 4.2 NoSQL 数据模型 169
 - 4.2.1 键值数据库 170
 - 4.2.2 文档数据库 171
 - 4.2.3 列族数据库 172
 - 4.2.4 图数据库 173
 - 4.2.5 四者对比 173
 - 4.3 NoSQL 数据库中的事务 174
 - 4.3.1 CAP 理论 174
 - 4.3.2 BASE 原则 175
 - 4.3.3 一致性协议 176
 - 4.4 NoSQL 关键技术 177
 - 4.4.1 NoSQL 的技术原则 177
 - 4.4.2 存储技术 178
 - 4.4.3 数据划分技术 178
 - 4.4.4 索引技术 179
- 第 5 章 键值数据库 182**
- 5.1 模型结构 182
 - 5.2 特征 183
 - 5.2.1 一致性 183
 - 5.2.2 可扩展性 183
 - 5.2.3 事务 184
 - 5.3 关键技术 184
 - 5.3.1 索引技术 184
 - 5.3.2 查询支持 186
 - 5.4 Redis 186

5.4.1	Redis 数据类型	187	7.2.2	可扩展性	258
5.4.2	Redis 的持久化	196	7.2.3	事务	260
5.4.3	Redis 事务	201	7.2.4	可用性	261
5.4.4	Redis 的发布订阅	205	7.3	MongoDB	261
5.4.5	Redis 的主从复制	208	7.3.1	概述	261
5.5	Redis 的 Java 客户端 Jedis	213	7.3.2	Mongo Shell	262
5.5.1	Jedis 所需要的 jar 包	214	7.3.3	MongoDB 基本操作	269
5.5.2	Jedis 常用操作	214	7.3.4	索引	276
5.5.3	Jedis Pool	215	7.3.5	副本集	279
7.3.6	分片	286	7.4	MongoDB 的 Java 客户端	290
7.4.1	MongoDB 驱动包的获得	290	7.4.1	MongoDB 驱动包的获得	290
7.4.2	Java 操作举例	291	7.4.2	Java 操作举例	291
第 6 章	列族数据库	220	第 8 章	非关系型图数据库	296
6.1	模型结构	220	8.1	图数据库	297
6.2	特征	222	8.1.1	图模型的模型和定义	297
6.2.1	一致性	222	8.1.2	图数据库的应用	298
6.2.2	可用性	223	8.1.3	图管理的关键技术	299
6.2.3	可扩展性	224	8.2	Neo4j 概述	302
6.3	HBase 应用	224	8.2.1	Neo4j 的特点	302
6.3.1	HBase 数据模型	225	8.2.2	Neo4j 的数据模型	304
6.3.2	HBase 体系结构	227	8.2.3	Neo4j 关键技术	312
6.3.3	HBase 基本 Shell 操作	232	8.3	Neo4j 的应用	320
6.3.4	HBase 压缩	234	8.3.1	使用嵌入在 Java 应用程序 中的 Neo4j	320
6.3.5	可用客户端 Java	236	8.3.2	Neo4j 的 Java 客户端环境 配置	320
6.4	架构与设计	244	8.3.3	一个简单的小型图数据库 例子	321
6.4.1	表设计规则	244	8.3.4	属性值	325
6.4.2	RowKey 设计	245	8.3.5	带索引的用户数据库	325
6.4.3	列族的数量	248	8.4	Neo4j 的优化	331
6.4.4	版本的数量	248	8.4.1	索引	331
6.5	HBase 集成	248	8.4.2	批量导入 / 导出	332
6.5.1	HBase 与 Hive 集成	249			
6.5.2	MapReduce 与 HBase 互操作	251			
第 7 章	非关系型文档数据库	255			
7.1	模型结构	255			
7.2	特征	257			
7.2.1	一致性	257			

1.1 大数据的基本概念

一般来说,大数据泛指巨量的数据集。当今社会,互联网和物联网,尤其是移动互联网的发展,显著加快了信息化向社会、经济等各方面以及大众生活的渗透,推动了大数据时代的到来。近年来,人们能明显地感受到大数据来势迅猛。据有关资料显示,1998年,全球网民平均每月使用流量是1MB,2003年是100MB,而2014年是10GB;全网流量累计达到1EB(即10亿GB)的时间在2001年是一年,在2004年是一个月,而在2013年仅需要一天,即一天产生的信息量可刻满1.88亿张DVD光盘。事实上,我国网民数居世界首位,产生的数据量也位于世界前列,这其中包括淘宝网每天超数千万次的交易所产生的超50TB的数据,百度搜索每天生成的几十PB数据,城市里大大小小的摄像头每月产生的几十PB数据,甚至包括医院里CT影像抑或门诊所记录的信息。总之,大到学校、医院、银行、企业的系统行业信息,小到个人的一次百度搜索、一次地铁刷卡,大数据存在于各行各业,连接着大众生活的各个角落。

大数据因自身可挖掘的高价值而受到重视。在国家宽带化战略实施、云计算服务起步、物联网广泛应用和移动互联网崛起的同时,数据处理能力也在迅速发展,数据积累到一定程度,会显示出开发的價值。同时,社会节奏的加快,要求快速反应和精细管理,急需借助数据分析和科学决策,这样,我们便需要对上面所说的形形色色的海量数据进行开发。也就是说,大数据的时代来了。

有学者称,大数据将引发生活、工作和思维的革命;《华尔街日报》也将大数据称为引领未来繁荣的三大技术变革之一;麦肯锡公司的报告指出,数据是一种生产资料,大数据将是下一个创新、竞争、生产力提高的前沿;世界经济论坛的报告认为大数据是新财富,价值堪比石油;等等。因此,大数据的开发和利用将成为各国家抢占的新的制高点。

大数据是相对于一般数据而言的,目前对大数据尚缺乏权威的严格定义,但较普通的解释是“难以用常规的软件工具在容许的时间内对其内容进行抓取、管理和处理的数据集合”。通常用4V来概括大数据的特征:

- Volume(规模性):大数据之“大”,体现在数据的存储和计算均需要耗费海量规模的资源上。规模大是大数据最重要的标志之一,事实上,数据只要有足够的规模就可以

称为大数据。数据的规模越大，通常对数据挖掘所得到的事物演变规律越可信，数据的分析结果也越具有代表性。例如，美国宇航局收集和处理的天气观察、模拟数据达到 32PB；而 FICO 的信用卡欺诈检测系统要监测全世界超过 18 亿个活跃信用卡账户。不过，现在也有学者认为，社会对大数据的关注，应更多地被引导到对数据资源的获得与利用上来，因为对于某些中小型数据的挖掘也有价值，目前报道的一些大数据挖掘的应用例子，不少也只是 TB 级的规模。

- Velocity (高速性)：大数据的另一特点是数据增长速度快，急需及时处理。例如，大型强子对撞机实验设备中包含 15 亿个传感器，平均每秒收集超过 4 亿的实验数据；同样在一秒内，有超过 3 万次用户查询被提交到谷歌，3 万条微博被用户撰写。而人们对数据处理速度的要求也日益严格，力图跟上社会的节奏，有报道称，美国中央情报局就要求利用大数据将分析、搜集数据的时间由 63 天缩短为 27 分钟。
- Variety (多样性)：在大数据背景下，数据在来源和形式上的多样性愈加突出。除结构化形式存在的文本数据之外，网络上也存在大量的位置、图片、音频、视频等非结构化信息。其中，视频等非结构化数据占有很大比例，有数据表明，2016 年，全部互联网流量中，视频数据达到 55%，那么，有理由相信，大数据中 90% 都将是非结构化数据。并且，大数据不仅仅在形式上表现出多元化，其信息来源也表现出多样性：大致可分为网络数据、企事业单位数据、政府数据、媒体数据等。
- Value (高价值性)：大数据价值总量大，但价值稀疏，即知识密度低。大数据以其高价值吸引了全世界的关注，据全球著名咨询公司麦肯锡报告：“如果能够有效地利用大数据来提高效率和质量，预计美国医疗行业每年通过数据获得的潜在价值可超过 3000 亿美元，能够使得美国医疗卫生支出降低 8%。”然而，大数据的知识密度非常低，IBM 副总裁 CTO Dietrich 表示：“可以利用 Twitter 数据获得用户某个产品的评价，但是往往上百万记录中只有很小的一部分真正讨论这款产品。”并且，虽然数据规模与数据挖掘得到的价值之间有相关性，但是两者难以用线性关系表达。这取决于数据的价值密度，同一事件的不同数据集即便有相同的规模（例如对同一观察对象收集的长时间稀疏数据和短时间密集数据），其价值也可以相差很多，因为数据集“含金量”不同，大数据中多数数据是重复的，忽略其中一些数据并不影响对其分析的结果。

注意

大数据之所以难处理不仅在于规模大，更大的挑战是其随时间变化快和类型的多样性，随时间和类型的变化增加了大数据的复杂性，同时也丰富了大数据的内涵。对大数据仅仅冠以“大”这个形容词是不全面的，只不过在大数据 4V 中，规模相对于变化和类型这两个特征量来说容易定量，而且即便是单一类型的数据集，只要具备足够的规模也能称得上是大数据。当然，数据的规模越大，通常对数据挖掘所得到的事物演变规律越可信，数据分析的结果也越有代表性。因此对大数据突出规模大这一特征是可以理解的。

数据分析挖掘需要有足够规模的数据，但前提是这些数据要有一定的时间或空间跨度，

即要具有普遍性。例如，每分钟将一个人的身体数据记录下来以了解其身体状况是有效的，如果将频率改为每秒钟，数据规模有所增加，但其价值并无提升。显然，数据样本密度与被观察对象有关，如风力发电机的很多传感器每毫秒就要检测一次，以检查叶片等的磨损程度。

大数据是无处不在的。大数据包括：

- 数目极庞大的网络数据。有自媒体数据（比如社交网络），有日志数据（比如用户在搜索引擎上留下大数据），还有流量最大的富媒体数据（比如视频、音频）等。例如，淘宝网每天的数据量就超过 50TB；新浪微博晚高峰时每秒钟要接受 100 万次以上的请求；美国 YouTube 网站每分钟就有 72 小时的视频被下载。
- 企事业单位数据和政府数据。一家医院一年能接受包括医疗影像、患者信息在内的 500TB 数据；中国联通每秒钟记录用户上网条数近百万，一个月的数据量大概是 300TB；国家电网信息中心目前累计收集了 2PB 的数据。
- 我们身边的一些公用设施所记录的数据。就监控而言，很多城市的交通摄像头多达几十万个，一个月的数据就达到数十个 PB，另外，基本上所有的超市都覆盖了摄像头，这些都可以是大数据的基本来源并可被挖掘利用；在北京，每天公交一卡通的刷卡记录有 4000 万条，而每天地铁刷卡的记录也有 1000 万条，这些数据可以用来改善北京的交通状况，优化交通路线。
- 国家大型公用设备和科研设备等产生的数据。例如，波音 787 每飞一个来回可产生 TB 级的数据，美国每个月收集 360 万次飞行记录；风力发电机装有测量风速、螺距、油温等的多种传感器，每隔几毫秒就要测量一次，这些数据用于检测叶片、变速箱、变频器等的磨损程度；一个具有风机的风场一年会产生 2PB 的数据，这些数据用于预防和维护，可使风机寿命延长 3 年，极大降低了风机的成本。
- 一些地理位置、基因图谱、天体运动轨迹的数据。总之，任何可以利用数据分析的地方就会有大数据的存在。

毋庸置疑，大数据将带动产业和市场，包括服务器、存储器、联网设备、软件与服务等，但是硬件、软件和服务仅仅是狭义上的大数据产业：通过大数据挖掘，大数据被应用到各行各业，可有效提升生产效益、支撑节能降耗、促进经济发展，因此广义上的大数据产业的产值更多地体现在工业、农业、交通运输、建筑制造等行业。事实上，大数据分析在社会治理和民生服务上的效益更为显著，这远不是 GDP 可以衡量的。也就是说，大数据的社会效益大于经济效益，大数据受到广泛重视也是因其溢出效应明显。

大数据计算是关系国民经济发展与国家安全的重大需求，是把握信息产业的制高点。在大数据中，我们可以获得比其他方式更及时、更精准的统计特征，继而建立相应的数据模型，辅助政策制定者更有效地制定决策、观察反馈、优化调整。总的来说，在各行各业研究大数据都有非凡的意义，大数据可以辅助社会管理、推动科学发展、提高企业效益、改善人民生活，以下是大数据在各个领域的一些具体作用和应用实例。

- 在宏观经济领域，淘宝网根据网上成交额比较高的 390 个类目的商品价格来得出 CPI，比国家统计局公布的 CPI 更早地预测到经济状况。国家统计局统计的 CPI 主要根据刚性物品得出，如食品，百姓都要买，差别不大。可是淘宝网是利用化妆品、电

子产品等购买量受经济影响较明显的商品进行预测，因此淘宝网的 CPI 更能反映经济走势。美国印第安纳大学利用谷歌公司提供的心情分析工具，从近千万条短信和网民留言中归纳出 6 种心情，进而预测道琼斯工业指数，准确率高达 87%。

- 在企业经营领域，华尔街对冲基金依据购物网站的顾客评论，分析企业的销售状况；一些企业利用大数据分析实现对采购和合理库存的管理，通过分析网上数据了解客户需求，掌握市场动向；美国通用电气公司通过对所产生的 2 万台喷气引擎的数据分析，开发的算法能够提前一个月预测需求，准确率达 70%。
- 在农业领域，硅谷的气候公司利用 30 年的气候变化和 60 年的农作物收成变化、14TB 的土壤历史数据、250 万个地点的气候预测数据和 1500 亿例土壤观察数据，生成 10 万亿个模拟气候据点，可以预测下一年的农产品产量以及天气、作物、病虫害和灾害、肥料、收获、市场价格等的变化。
- 在商业领域，商家得到消费者在网上的消费记录后，就可以留意其上网踪迹和消费行为，并适时弹出本公司商品的广告，这样就很容易达成交易，最终的结果是顾客、商家，甚至相关网站都有收益。再比如，沃尔玛将每月 4500 万条网络购物数据与社交网络上产品的大众评分结合，开发出“北极星”搜索引擎，以方便顾客购物，在线购物的人数因而增加了 10% ~ 15%。
- 在金融领域，阿里公司根据淘宝网上中小型公司的交易状况，筛选出财务健康、诚信优良的企业，为其免担保提供贷款达上千亿元，坏账率仅有 0.3%，相较于需要担保的商业银行，坏账率要低很多；华尔街德温特资本市场公司通过分析 3.4 亿条留言判断民众心情，以决定公司股票的买入和卖出，从而获得了较好的收益。
- 在医疗卫生领域，一方面，相关部门可以根据搜索引擎上民众对相关关键词的搜索数据建立数学模型进行分析，得出相应的预测进行预防。例如，2009 年，谷歌公司在甲型 H1N1 爆发前几周，就预测出流感形式，与随后的官方数据相关性高达 97%；而百度公司得出的中国艾滋病感染人群的分布情况，与后期卫生部公布的结果基本一致。另一方面，医生可以借助社交网络平台与患者就诊疗效果和医疗经验进行交流，能够获得在医院得不到的临床效果数据。除此之外，基于对人体基因的大数据分析，可以实现对症下药的个性化诊疗，提高医疗质量。
- 在其他领域，如在交通运输中，物流公司可以根据 GPS 上大量的数据分析优化运输路线，以节约燃料和时间，提高效率；相关部门也会通过对公车上手机用户的位置数据分析，为市民提供交通实时情况。大数据还可以改善机器翻译服务，谷歌翻译器就是利用已经索引过的海量资料库，从互联网上找出各种文章及对应译本，找出语言数据之间的语法和文字对应的规律来达到目的的。大数据在影视、军事、社会治安、政治领域的应用也都有着很明显的效果。总之，大数据的用途是无处不在的。

当然，大数据不仅仅是一种资源，作为一种思维方法，大数据也有着令人折服的影响。伴随大数据产生的数据密集型科学，有学者将它称为第四种科学模式，其研究特点在于：不在意数据的杂乱，但强调数据的规模；不要求数据的精准，但看重其代表性；不刻意追求因果关系，但重视规律总结。如今，这一思维方式广泛应用于科学研究和各行各业，成为从复杂现象中透视本质的重要工具。

对其进行有效管理是有效使用大数据的基础支撑，这就需要面向大数据的数据库管理系统。下面对数据库管理系统进行介绍。

1.2 数据库管理系统

1.2.1 数据库管理系统的基本概念

数据库管理系统 (Data Base Management System, DBMS) 是一种操纵和管理数据库的软件, 用于建立、使用和维护数据库。它对数据库进行统一的管理和控制, 以保证数据库的安全性和完整性。DBMS 是数据库系统的核心。

DBMS 把用户抽象的逻辑数据处理转换为计算机中具体的物理数据处理。有了 DBMS, 用户就可以逻辑地处理数据, 而不必顾及这些数据在计算机中的布局 and 物理位置。用户通过 DBMS 访问数据库中的数据, 数据库管理员也通过 DBMS 进行数据库的维护工作。它可使多个应用程序和用户用不同的方法同时或在不同时刻去建立、修改和查询数据库。DBMS 主要包括如下功能。

1. 数据定义

DBMS 通常提供数据定义语言 (Data Definition Language, DDL), 供用户定义数据库的三级模式结构、两级映像以及完整性和保密等约束。DDL 主要用于建立、修改数据库的库结构。DDL 所描述的库结构仅仅给出了数据库的框架, 而数据库的框架信息被存放在数据字典中。关系数据库的数据定义可参见 2.3.2 节。

2. 数据操作

DBMS 提供数据操作语言 (Data Manipulation Language, DML), 供用户实现对数据的追加、删除、更新、查询等操作。有关关系数据库操作语言的具体介绍可参见 2.3.3 节。

3. 数据库的运行管理

数据库的运行管理功能是指 DBMS 的运行控制、管理功能, 包括多用户环境下的并发控制、安全性检查和存取限制控制、完整性检查和执行、运行日志的组织管理、事务的管理和自动恢复 (即保证事务的原子性)。这些功能保证了数据库系统的正常运行。

4. 数据组织、存储与管理

DBMS 要分类组织、存储和管理各种数据, 包括数据字典、用户数据、存取路径等, 需要确定以何种文件结构和存取方式在存储级上组织这些数据, 以及如何实现数据之间的联系。数据组织和存储的基本目标是提高存储空间利用率, 选择合适的存取方法可提高存取效率。

5. 数据库的保护

数据是信息社会的战略资源, 所以对数据的保护至关重要。DBMS 对数据库的保护主要通过 4 个方面来实现: 数据库的恢复、数据库的并发控制、数据库的完整性控制和数据库安全性控制。此外, 还包括系统缓冲区的管理以及数据存储的某些自适应调节机制等。

6. 数据库的维护

数据库的维护包括数据库的数据载入 / 转换 / 转储、数据库的重组和重构以及性能监控等功能, 这些功能分别由各个实用程序来完成。

7. 通信

DBMS 具有与操作系统的联机处理、分时系统及远程作业输入相关的接口，负责处理数据的传送。网络环境下的数据库系统，还应该包括 DBMS 与网络中其他软件系统的通信功能以及数据库之间的互操作功能。

例如，MySQL 是一种开放源代码的关系型数据库管理系统（RDBMS），它使用最常用的数据库管理语言——结构化查询语言（SQL）进行数据库管理。在 MySQL 中，数据库定义语言的操作对象包括数据库和表，主要操作有 create、alter、drop。MySQL 将数据保存在不同的表中，同时支持 AIX、FreeBSD、HP-UX、Linux、Mac OS、Novell Netware、OpenBSD、OS/2 Wrap、Solaris、Windows 等多种操作系统。MySQL 对数据安全性有着比较全面的保障机制，对数据的保护分为内部安全性保护和外部安全性保护。它提供用于管理、检查、优化数据库操作的管理工具，例如 MySQL Administrator、MySQL Query Browser 和 MySQL Workbench 等。MySQL 将在第 2 章中详细介绍。

MongoDB 是一个基于分布式文件存储的 NoSQL 数据库。它支持的数据结构松散，是类似于 JSON 的 BSON 格式，因此可以存储比较复杂的数据类型。MongoDB 最大的特点是它支持的查询语言非常强大，其语法有点类似于面向对象的查询语言，几乎可以实现类似关系数据库单表查询的绝大部分功能，而且还支持对数据建立索引。MongoDB 的设计目标是高性能、可扩展、易部署、易使用且便于存储数据。MongoDB 面向集合存储，容易存储对象类型的数据。在 MongoDB 中数据被分组存储在集合中，集合类似于 RDBMS 中的表，一个集合中可以存储无限多的文档，采用无模式存储数据是集合区别于 RDBMS 中的表的一个重要特征。MongoDB 支持完全索引，可以在任意属性上建立索引，包含内部对象。MongoDB 的索引和 RDBMS 的索引基本一样，可以在指定属性、内部对象上创建索引以提高查询的速度。除此之外，MongoDB 还提供创建基于地理空间的索引的能力。MongoDB 使用高效的二进制数据存储，包括大型对象（如视频），可以保存任何类型的数据对象。MongoDB 提供了当前所有主流开发语言的数据库驱动包，开发人员使用任何一种主流开发语言都可以轻松编程，实现对 MongoDB 数据库的访问。MongoDB 文件存储格式为 BSON（JSON 的一种扩展）。BSON 是对二进制格式的 JSON 的简称，它支持文档和数组的嵌套。MongoDB 将在第 7 章中详细介绍。

1.2.2 数据库管理系统的发展历史

数据库一词引入于 20 世纪 60 年代中期。该术语与过去基于磁带的系统形成对比，允许共享交互式使用而不是批处理。美国加州的某个系统开发公司首先在特定的技术意义上使用了“数据库”一词。

随着数据规模的增大和硬件的提升，DBMS 的功能和性能已经增长了几个数量级。数据库管理系统的发展可以根据数据模型或结构分为三个时代：导航时代、SQL/关系时代和后关系时代。

1. 导航时代

随着计算机速度的提高和功能的增强，出现了许多通用数据库系统，到 20 世纪 60 年代中期，许多此类系统已投入商业使用。集成数据存储（IDS）的 Charles Bachman 在

CODASYL 内创建了数据库任务组，CODASYL 负责 COBOL 的创建和标准化。1971 年，数据库任务组制定了其标准，该标准通常被称为 CODASYL 方法，基于这种方法的许多商业产品很快进入了市场。

CODASYL 方法依赖于形成大型网络的链接数据集的“手动”导航。应用程序可以通过以下三种方法找到记录：

- 使用主键（称为 CALC 键，通常通过散列实现）；
- 将关系（集合）从一个记录导航到另一个记录；
- 按顺序扫描所有记录。

后来的系统添加了 B 树作为索引（具体介绍可参见 2.1.3 节）以提供备用访问路径。许多 CODASYL 数据库还添加了一种非常简单的查询语言。然而，在最终的统计中，CODASYL 非常复杂，需要大量的培训和努力工作才能生成有用的应用程序。

IBM 公司在 1966 年开发了自己的 DBMS，称为信息管理系统（IMS）。IMS 是为 System/360 上的 Apollo 程序编写的软件。IMS 在概念上与 CODASYL 大体相似，但对其数据导航模型使用严格的层次结构而不是 CODASYL 的网络模型。由于访问数据的方式不同，这两个概念被称为导航数据库，而 Bachman 在 1973 年的图灵奖报告题为“作为导航员的程序员（The Programmer as Navigator）”。IMS 被归类为分层数据库。IDMS 和 Cincom 的 TOTL 归类为网状数据库。IMS 自 2014 年起仍在使用中。

2. SQL/ 关系时代

在导航式数据库得到普遍应用的时候，新的数据管理系统正在酝酿。1970 年，Edgar F. Codd 首次提出关系模型，在该模型中，应用程序按内容查询而不是通过链接查询数据，从而抛弃了传统的导航数据库，引发了数据库管理系统的革命。

Edgar F. Codd 在位于加利福尼亚州圣何塞的 IBM 公司工作，主要负责硬盘系统的开发。在研究过程中，他发现了 CODASYL 方法中导航模型的问题，特别是缺乏“搜索”机制。他在 1970 年发表的论文中描述了一种新的数据库构建方法，最终创建了关系模型。

在这篇论文中，他描述了一个用于存储和使用大型数据库的新系统。Codd 的想法是使用固定长度记录的“表”，而不是将记录存储在 CODASYL 中的某种自由格式的记录链表中，而每个表用于不同类型的实体。当存储“稀疏”数据库时，链表系统的效率非常低，其中任何一条记录的某些数据可以留空。关系模型通过将数据分成一系列规范化表（或关系）来解决这个问题，可选元素从主表移出到仅在需要时占用空间的位置。可以在这些表中自由插入、删除和编辑数据，DBMS 也允许应用程序 / 用户维护表和视图。

关系模型还允许在不重写链接和指针的情况下更新数据库内容。关系部分来自引用其他实体的实体，即所谓的一对多关系（如传统的分层模型），以及多对多关系（如网络模型）。因此，关系模型既可以表达层次和网络模型，也可以表达原生表格模型，其允许 DBMS 根据应用程序需要对这三种模型进行组合建模。

例如，在与用户有关的数据库中存在用户的姓名、登录信息、地址和电话号码信息。在导航方法中，所有这些数据都放在一个记录中，未使用的项目将不会放在数据库中。在关系方法中，数据将被标准化为用户表、地址表和电话号码表，仅当实际提供地址或电话号码时，才会这些可选表中创建记录。

将信息重新链接在一起是该系统的键。在关系模型中，一些信息被用作“键”，用于唯一地定义特定记录。当收集关于用户的信息时，通过搜索该键可以找到存储在可选表中的信息。例如，如果用户的登录名是唯一的，则该用户的地址和电话号码将以登录名作为键进行记录。将相关数据简单地“重新链接”回单个集合难以基于传统的计算机语言进行设计与实现。

正如导航方法需要程序循环以便收集记录一样，关系方法需要循环来收集有关任何一条记录的信息。Codd 的建议是设计一种面向集合的语言，根据这种思想产生了无处不在的 SQL。Codd 使用称为元组演算的数学工具证明了这样的系统可以支持普通数据库的所有操作（插入、更新等），并提供了一个简单的系统，用于在单个操作中查找和返回数据集。

Codd 的论文被加州大学伯克利分校的 Eugene Wong 和 Michael Stonebraker 两位专家选中。他们启动了一个名为 INGRES 的项目，从 1973 年开始，INGRES 交付了第一款测试产品，这款产品在 1979 年得到了广泛使用。INGRES 在很多方面类似于 System R，包括使用 QUEL “语言”进行数据访问。随着时间的推移，INGRES 转向新兴的 SQL 标准。

IBM 本身完成了关系模型 PRTV 的一个测试实现。霍尼韦尔为 Multics 编写了 MRDS，现在有两个新的实现：Alphora Dataphor 和 Rel。通常称为关系的大多数其他 DBMS 实现实际上是 SQL DBMS。

1970 年，密歇根大学开始了 MICRO 信息管理系统的开发。这一系统基于 D. L. Childs 的 Set-Theoretic Data 模型。美国劳工部和美国环境保护署以及来自阿尔伯塔、密歇根大学和韦恩州立大学的研究人员使用 MICRO 来管理非常大的数据集。该系统使用密歇根终端系统在 IBM 大型机上运行，一直到 1998 年。

IBM 在 20 世纪 70 年代早期开始基于 Codd 的概念 System R 研究原型系统。第一个版本在 1974 年 5 月准备就绪，然后在多表系统上开始工作，这个系统可以拆分数据，而不必存储单个的大“块”。客户在 1978 年和 1979 年对随后的多用户版本进行了测试，此时添加了标准化查询语言 SQL。Codd 的想法是建立既可行又优于 CODASYL 的系统，这推动了 IBM 开发 System R 的真正生产版本，即 SQL/DS，以及后来的 DB2。

Larry Ellison 的 Oracle 数据库（或更简单地说，Oracle）是基于 IBM 关于 System R 的论文从另一个角度开发的。尽管 Oracle V1 在 1978 年完成开发，但直到 1979 年，当 Oracle Version 2 击败 IBM 时，才推向市场。

Stonebraker 继续基于 INGRES 开发了一个新的数据库 Postgres，现在称为 PostgreSQL。PostgreSQL 通常用于全局关键任务应用程序（.org 和 .info 域名注册中心将其作为它们的主要数据存储，许多大公司和金融机构也是如此）。

在瑞典，研究人员也阅读了 Codd 的论文并开发了系统，Mimer SQL 是 20 世纪 70 年代中期在乌普萨拉大学开发的。1984 年，围绕此项目形成了一个独立的企业。

实体 - 关系模型作为关系模型的进化版于 1976 年出现，因为它的描述形式比早期的关系模型更为直接，在数据库设计中受到欢迎。后来，实体 - 关系构造被改造为关系模型的数据建模方式，两者之间的差异变得很小。

3. 关系时代的其他尝试

在 20 世纪 70 年代和 80 年代，研究人员尝试构建具有集成硬件和软件的数据库系统。