数据管护技术及应用

于明鹤[1,2]，聂铁铮[3]，李国良[4]
1. 东北大学软件学院，辽宁 沈阳 110169；
2. 广东省普及型高性能计算机重点实验室，广东 深圳 518060；
3. 东北大学计算机科学与工程学院，辽宁 沈阳 110169；
4. 清华大学计算机科学与技术系，北京 100084

摘要
为了对海量数据进行充分和有效的处理、存储以及应用，数据管护技术应运而生。数据管护技术是在数据整个生命周期内，对数据进行的主动并持续的管护，使数据得到最大化的利用，并且大程度地延长数据的使用寿命。围绕数据管护技术的目的、解决方案和应用，系统介绍了数据管护的处理过程和其中的关键技术，并介绍了几种基于数据管护的应用，并对其技术特点进行了对比分析。最后，对数据管护技术的发展前景和未来挑战进行了阐述。

关键词
数据管护；数据清洗；数据集成；元数据管理；溯源管理

中图分类号：TP315 文献标识码：A doi: 10.11959/j.issn.2096-0271.2019048

Data curation technologies and applications

YU Minghe[1,2], NIE Tiezheng[3], LI Guoliang[4]
1. Software College of Northeastern University, Shenyang 110169, China
2. Guangdong Province Key Laboratory of Popular High Performance Computers, Shenzhen 518060, China
3. School of Computer Science and Engineering, Northeastern University, Shenyang 110169, China
4. Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China

Abstract
Data curation is emerged in order to process, store and applied efficiency. Data curation processes active and continuous management the data through the whole lifecycle of it. And utilizing data curation techniques, data could be used to the maximum extent, and the speed of its elimination can be effectively slowed down. The process and key techniques of data curation aroundits goals, solutions and applications were described. For the crucial techniques, existing solutions were analyzed and introduced. In addition, the applications of data curation in the various domains were also introduced and compared. Finally, the development prospect and future challenges were expounded.

Key words
data curation, data cleaning, data integration, metadata management, provenance management

1 引言

数字化信息正以史无前例的速度产生，当前科学界、产业界、社会以及日常生活中大数据方法被广泛使用。这些应用方法依赖于数据的质量及其可用性。在数据驱动的科学研究或数据密集型研究中，科学家们需要大量使用和共享各种海量的数字资源，这就要求对科学数据进行有效的收集、加工、组织、保存、发布等。在数据科学和e-science中，将该类应用称为data curation或digital curation，中文译为“数据策管”“数据监护”“数据管护”等[1]。在图书馆和档案馆领域，同样也使用“数据管护”一词，该数据管护强调的是对数字化数据的维护、保存与增值。在该领域，基于数据管护技术的主要应用为建立开放档案信息系统，如欧盟的电子资源保存与接入网络项目（ERPANET）、美国佛罗里达数字档案项目（FDA）开发的数字资源库（DAITSS）等用于存储数字化信息的资源库。与图书馆学讨论的“数据管护”相比，本文着重强调进行“数据管护”自动化和智能化处理技术，更加面向专门领域的应用，更加注重具体的实现细节。

一个典型的应用案例是，在生物大数据应用中，科学家们可以通过在线访问现有生物和古生物数据集来研究生物多样性。利用长期收集到的大量全球生物数据，人们可以获得与进化过程以及物种向极地迁移相关的科学知识，更进一步地，还可以获得关于气候变化的知识。利用数据管护技术，可以对收集到的生物数据和信息进行管护，并形成一个全球生物的数据管护框架。目前已有一些工作对生物收集工作者进行数据管护问卷调查，他们负责管护美国国家生物多样性建设相关项目的各类标本。调查结果表明，在生物数据的管护中存在着极大的数据多样性，并且为了能够反映生物数据管护的复杂性，还需要增加更多的数据管护问题。为了让这些数据在未来研究中可以对特定的领域专家以外的研究者提供说明与指导，并使数据化的生物数据能够得到长期持续的管护，需要使用专门的元数据创建工具和数据标准，对生物数据管护工作进行长期的维护[2]。

数据管护在学术界和工业界尚没有统一的定义。一般认为，数据管护是指关于数据在其生命周期，即从生成数据和初始存储起，到繁衍变化或者废弃删除的整个过程中的持续管护活动。数据管护的主要目的是使数据在后续研究及重复利用的过程中保持可信任性，另外在商业用途中也需要确保数据的可重塑性。

图1给出了数据管护的基本过程。该过程分为3个阶段：数据收集、数据处理以及数据发布[3]。

（1）数据收集阶段

该阶段完成对原始基础数据的获取工作，主要包括数据加载和数据抽取。

数据加载主要指将外界的原始数据装载到数据管护系统中的过程。原始数据的获取有多种途径。例如，在Web应用中，用户可以通过从网络直接下载、网页爬取以及利用应用程序接口（application programming interface，API）抓取等方

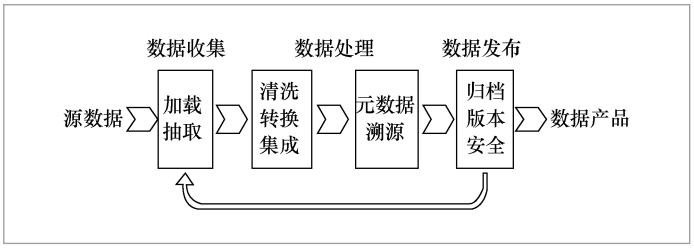

图1 数据管护的基本过程

式得到互联网中的信息，或者将传统以纸质媒介记录的信息通过扫描、人工输入等方式存入数字化存储媒介中。需要将这些来源不同、格式不同的信息加载到用户使用的编程环境下，以便对数据进行进一步的处理。

对于每一个获取到的数据，均需要确定它的质量及整洁度（tidiness），即数据的内容是否有效、数据格式是否易于解析。对于一个数据表格的评判主要有属性值是否存在多值属性、每一行或每一列表示的意义是否相同等。如果存在这些问题，则认为该数据为非整洁的数据，需要进行进一步的处理。

数据抽取是指利用信息抽取技术，从非结构化数据（如网页、文本、新闻、邮件等）中识别有用的信息。与数据加载相比，系统认为数据加载的数据是全部有效的，而数据抽取则是进一步提取更加有效的信息的过程。具体技术有实体抽取（entity extraction）、识别命名实体。例如，使用自然语言处理技术和机器学习算法从网页内容中识别实体（如人物、地点、公司等）。

（2）数据处理阶段

该阶段主要将数据处理成可用的形式，主要包括数据清洗、数据转换和数据集成。

数据清洗是指对上一阶段被判断为低质量或非整洁的数据进行清理，从而避免由于脏数据的存在而导致数据使用者做出不可靠的分析和错误的决定。

数据转换是指对清洗过的数据进行格式转换，以便后续使用。由于同一含义的数据可能有多个来源和多个表达形式，因此，需要将这些数据转换成统一格式，通过过滤、归档或者使用某些正则表达式对数据进行转换，从而将来自多个数据集的同一数据进行合并。然后，用户可以根据自身需求，将这些数据以数据库或文件的方式进行归档存储。例如将PDF、WORD、PPT等文件格式转换成普通文本格式。

数据集成是指将多源异构数据进行合并处理。通过消除异构性，将已有的数据和关系合并成统一的数据格式和统一的语义。在数据集成过程中，还需要抽取元数据和记录溯源信息，进行元数据管理和溯源管理。

数据整合是一种扩展的数据集成处理技术，可以利用数据整合技术发现和增加新的语义，建立新的关联关系。数据整合是一种扩展的数据集成处理，包括关联发现、数据分类等功能[4]。关联发现也称作数据链接，利用相似性函数，如欧氏距离、编辑距离、余弦距离、杰卡德函数等，建立数据实体之间的关联关系。例如，将某一数据链接到WikiData或谷歌知识图谱。数据分类指利用各种分类器，如贝叶斯、支持向量机、决策树、KNN等机器学习算法，将数据实体进行分类或聚类。

（3）数据发布阶段

该阶段主要完成数据归档、数据产品生成。

数据归档是指按照某种存储模型，将数据进行组织和存储。例如，将管护好的数据集保存到关系数据库或NoSQL键值数据库中。

数据发布是指按照产品标准，将数据制作成产品并出版和发行，例如，图书检索数据库、生物数据库、遥感图像数据库、水文地理数据库等。

在数据发布过程中，还需要考虑版本控制、隐私保护和安全控制[5]。

数据管护的典型应用领域如下。

● 建立管护数据库（curated database）。管护数据库是一种结构化的、高质量的数据库，其中的内容是通过大量人力采集管护而成，如对已有的原始数据进行咨询、验证、汇聚，对新的原始数据进行解释和

合并等。许多管护数据库已正式发布，可代替图书馆的字典、百科全书、地名辞典等，起到权威参考书的作用。典型的管护数据库有蛋白质序列数据的UniProt、人口统计数据的CIA World Facebook[6]。

● 保证机器学习与数据挖掘的质量。目前社交推荐技术大多数采用机器学习的算法，根据已知的用户的基本信息，预测其可能感兴趣的事物或对象。利用数据管护技术，可以在选取训练样本时将管护出来的集成数据作为初始数据，使机器得到的数据内容更为丰富，进而提高检测结果的准确率，从而使人们能够精确地将他们的知识添加到机器所需要学习的地方。

● 数据沼泽净化与信息检索。由于现在很多数据源存在设计不良、未有效维护等问题，导致其本身成为一个数据沼泽（data swap），从而降低了对数据的检索能力，用户无法有效地对这些数据进行分析和利用。为了将这类数据转化为可利用、可再生的数据湖（data lake），需要通过数据管护技术对其进行“净化”。一方面对数据集本身进行清洗和分类；另一方面，对新引入的数据进行系统归档，以便使该数据集一直维持在“清澈”状态，方便用户对数据进行检索访问。

● 数据质量保证与数据治理。数据管护者在进行数据管护时，会长期对数据本身进行监督管理，从而使其监管下的数据一直保持在“洁净”状态。例如，对于生物大数据，需要对生物多样性提供有效的保证，通过在物种分布建模过程中使用数据管护技术对数据源进行追踪，保证生态圈的健全性。

2 关键技术

图2给出了数据管护系统框架。在接口层分为管护者（curator）和普通用户2种类型的接口，管护者具有工作流管理的权限，包括数据抽取、数据整合、数据清洗、数据归档和数据发布。普通用户可以对数据进行检索和查询，并将结果以可视化的形式显示。支撑层包括元数据管理、溯源管理、版本控制和访问控制4个模块，通过在4个模块上的操作可以对底层数据进行访问。数据层分为管护数据库和元数据数据库，

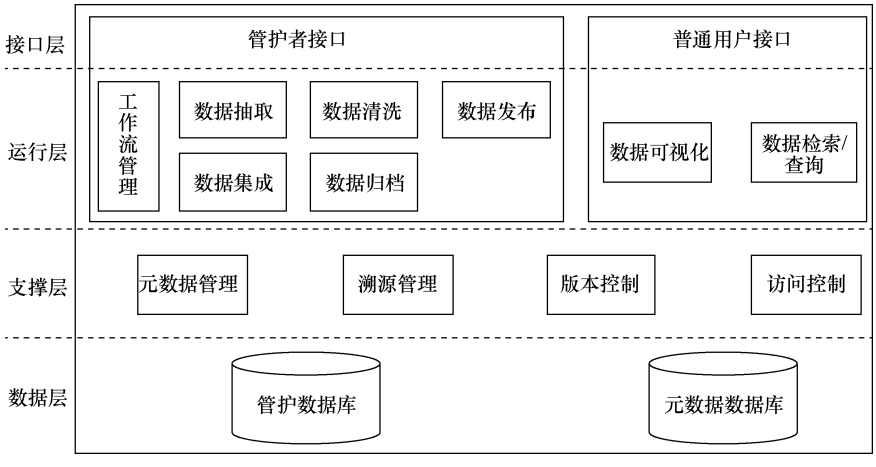

图 2 数据管护系统框架

元数据数据库用于存储该数据集中的所有基本数据，用户权限等操作信息则存放在管护数据库中。

2.1 数据抽取

对于抽取的数据，需要判断其质量及整洁度。目前数据判断的方法主要有2种：第一种方法为视觉化评估，即直接将数据加载到已有的应用软件（如Excel、GoogleSheet等）中，通过人工阅览数据的方式来检查数据的质量和整洁度；第二种方法为程序化评估，即通过编写程序的方式来检查数据的摘要和特定部分，如数据的头部或尾部。

2.2 数据清洗

数据清洗技术包括对脏数据的发现与修复技术。数据清洗可以分为手动和自动2种方式。手动方式指人工对数据进行修正或清理，该方法操作简单，但是存在着效率低、成本高的问题。因此，该方法仅适用于需要清洗的数据量极少的情况。自动清洗指通过编程的方式完成对数据的清洗。在利用程序对数据进行清洗时，首先要确定数据清洗的标准，包括数据中的关键词以及表格的属性标准。基于这些标准，用户可以通过编写相应的程序对数据存在问题的地方进行清洗与修正。当完成清洗程序并结束运行后，还要对清洗结果进行测试，以保证数据得到了有效的处理。测试可以采用与数据质量评估相同的方法。此外，为了保证正确性，在数据清洗之前通常要对原始数据进行备份，以保证进行了错误处理后的数据可以恢复。

目前在学术界和工业界有大量的工作研究和设计相应的数据清洗算法。对于脏数据发现问题，现有算法大多是使用完整性约束进行设计，通过捕获数据库需要确保的完整性规则（如函数依赖[7]、否定约束[8]等）来设计脏数据的判定条件。通过这种基于规则的算法，可以捕获冗余、不一致以及缺失值等情况。当捕捉到这些脏数据后，需要对其进行修复，以保证数据的质量。在修复阶段，需要考虑修复程度与修复效率的平衡问题，现有方法通过制定修复标准以及设计修复模型来实现。对于修复标准，现有算法通常会制定多种关于数据和质量规则的假设[9]，包括信任所声明的完整性约束，将所有不满足约束条件的数据更新并移除错误；信任所给出的数据完整性，允许一定程度的放松约束[10]；对数据和完整性同时变化的可能性进行检测[11]。现有的修复技术大多只能解决一种类型的错误，但已有一些工作开始考虑多种类型错误之间的相互影响，并提出一种数据的整体修复方法。在修复效率方面，现有算法主要采用2种策略来设计修复模型：一种策略是采用全自动机制，如根据某种代价函数，计算原始数据集与修复后的数据集间的距离，使代价最小化；另一类策略则在修复过程中引入人工操作，识别错误并建议修复的方法，或通过机器学习模型来执行自动修复决策[12]。

在现有的数据清洗算法中，基于规则的脏数据发现与修复是目前比较普遍的技术之一。基于规则的算法[7-8,13-14]由于在结构化数据中容易发现规则并以此确定和清洗数据，因此具有很大的优势。但是，对于非结构化或半结构化数据（如JSON或文本数据），由于规则难以捕获，因此无法获得很高的效率。近年来，一些研究引入了人工手段设计数据清洗算法[15-18]。这些采用众包技术的算法通常在监督学习算法中采用主动学习等方法选取更为有用的信息，以提高学习的效果，从而保证数据清洗的

质量。基于众包技术的方法可以通过用户的反馈，避免可能由于完整性约束而导致制定有害规则，从而使清洗结果更满足用户需求。除了上述问题外，目前数据清洗技术还面临的一个问题是如何保证处理海量数据的稳定性。随着大数据时代的来临，如何对大规模、快速增长的数据进行有效的清洗是一个重要的挑战。现有的方法主要采用冗余检测的分块技术[19]、基于采样的清洗技术[20]、分布式数据清洗[21-23]等来解决这一问题。

2.3 数据集成

数据集成技术是指将来自于不同数据源、不同结构的数据整合在一起，以实现对数据内容的扩充。具体而言，数据集成主要分为如下3个部分[24]。

- 模式匹配：对给定的两个或多个模式中的元素生成相应的联系。
- 实体解析：从所有的数据中识别出在现实世界中表示同一实体的多个记录。
- 数据整合：对于同一实体中出现分歧的内容进行解决，并找出正确结论。

在模式匹配阶段，主要完成的是对不同来源或结构的数据实现模式级的匹配，即找出表示同一内容的元素。由于各个数据集来源不同，可能会存在异构的情况。因此，在模式匹配阶段，首先要对异构的数据模式进行调和，针对不同来源的数据生成一个统一的结构；然后，对数据中的属性进行匹配，将各个数据中的属性都与调和后的结构中的属性相映射；最后再对数据中的语义关系进行映射，以保证调和数据能够被正确阐述。

在实体解析阶段，主要是识别出在现实世界中表示同一实体的记录。在近几年的研究中，实体解析作为数据集成的一个核心组件，得到了广泛的关注与研究，目前对实体解析问题的研究主要包括如下3个步骤。

- 在一个或多个属性值上建立块函数，利用该函数把实体切分为几个块，然后，对最不相似的实体对进行过滤，从而减小需要进行匹配的数据数量。
- 使用给定的相似性函数或规则定义记录间的相似性，将满足阈值的记录对认为是可以匹配的。
- 通过聚类算法将记录进行分类，即将表示同一实体的记录放在相同的类别中。

在数据整合阶段，将表示现实世界中同一实体的多个记录进行整合，形成一个简单、一致、干净的数据。在数据整合阶段，最关键的部分是真值发现，即确定来自多个数据源的数据中的真值。现有对数据整合的研究工作主要可以分为3类：第一类是对特定数据（如时态数据、图数据）的整合问题的研究[25-27]；第二类是利用数据特性（例如长尾效应）和专家领域知识等技术对传统整合技术的性能进行优化[28-30]；第三类是新兴的数据整合问题（如知识融合、基于查询的数据整合等）[31]。

数据集成面临的一个主要问题是如何选择数据。由于不同的相关数据源具有大量的重叠数据，目前的研究工作采用概率统计[32]、时间感知选择[33]等技术，设计合理的算法来选择一个好的数据源查询顺序，从而提高响应速度。除此之外，数据集成面临的另一问题是如何对来自不同数据源、不同格式的数据进行整合，参考文献[28,34-35]等采用最小代价模型、轻量级摘要构建、统一的虚拟模式等方法执行对异构多源数据的查询，从而达到异构数据集成的目的。除此之外，在大数据环境下，数据集成所要处理的数据是来自不同数据源的多模式、跨领域的数据。因此，传统的基于某一特定领域并且模式已知的模式匹配技术无法直接应用。对此，可以从寻

找实体在不同领域下的异构特征、识别出与实体相关的完整属性特征集合等方面开始着手研究。

2.4 元数据管理

现有对元数据管理的研究大多以设计和开发元数据管理系统的形式实现对元数据的管理。目前设计元数据管理系统主要有两类方法[36]。第一类方法为存储元数据分析[37-39]，该类方法把元数据视为整个数据的一个全局组件，每一个查询或分析都要通过该组件来执行。第二类方法把数据湖分解为多个数据池，而每一个数据池都是某一特定类型的数据[40]。在这种方法里，数据的存储、元数据管理、查询对于每类数据都是不同的，而这样的方法有助于确保数据的特殊性。

在数据湖应用中，原始数据在没有被查询前，都是以最原始的状态存储的，并且没有任何明确的模式，这被称为“schema-on-read”或延时绑定[41-42]。但是，随着海量数据以飞快的速度涌入数据湖，数据显式模式的缺乏会迅速地导致数据湖变为实用性较低的数据沼泽。因此元数据管理成为数据湖的重要组成部分。另外，一个有效的元数据管理系统也是数据能够被有效地检索、查询和分析的重要保证。元数据可以分为数据集内部和数据集间的元数据等类型[43]。其中，数据集内部的元数据构成了各个数据集的概述轮廓，这些元数据包括描述内容的属性以及数据集的统计性、结构性的信息。而数据集间的元数据表明了不同的数据集或其属性间的关系，这些元数据包括数据约束（如包含依赖）、连接性、亲和性等其他性质。

当前已有很多成熟的元数据管理系统，但是这些管理系统各自独立开发，同一开发者开发的产品具有很好的兼容性，但是在跨系统进行元数据管理时，这些系统的性能不尽如人意。另外，现有的元数据管理技术研究的大部分是表层的元数据，即用于定义数据的数据，忽略了深层的用于数据间的关系的数据。这样的关系元数据需要在半结构化数据中通过社区发现等技术分析得到。通过这些关系元数据，可以更好地保证数据的质量，同时进一步帮助用户理解数据集的含义，以便后续对数据的应用。因此在未来的工作中，可以从上述两个方面着手设计更加完善的元数据管理系统。

2.5 溯源管理

溯源信息通常是指描述最终产品生产过程的任何信息，包含关于实体、数据、处理、活动等在生产过程中的各种元数据。本质上，溯源信息可以当作描述整个生产过程的元数据。而溯源信息的收集（也称作捕获）和处理是十分重要的，例如，可以利用溯源信息进行质量评估、重现性保证以及对最终产品的信任增强。根据溯源管理的范围，按照从一般化到特殊化或检测程度从低到高，可以把溯源的类型分为溯源元数据、信息系统溯源信息、工作流溯源信息和数据溯源信息[44]。

溯源元数据是最基础的溯源信息，包含了在生产过程中所有可能出现的元数据。它给用户提供了对最终产品和生产过程中的任何溯源信息的建模、存储和访问操作的最大程度上的自由，并且还支持对那些内部不允许公开的溯源管理的私有解决方案进行分类操作。另外，溯源元数据不需要对包括溯源操作、溯源信息的数据模型等涉及底层处理的操作加以限制或假设。因此，溯源元数据被定义为描述使用任何数据模型和计算模型的任一生产过程的元数据。

信息系统溯源信息是指在涉及信息传播（例如存储/检索、通信、信息发布）的信息系统中有关处理过程的元数据。尽管每个过程的内部通常是未知的，但是溯源信息可以通过处理过程的输入、输出和相关参数而收集到。

在工作流溯源信息中，工作流可看作一个有向图，其顶点为带有输入、输出或参数的任意函数或模块，边为这些模块之间的预定义数据流或控制流。根据这个处理过程模型，支持工作流溯源的系统利用工作流图中的所有信息，提高了对溯源收集的检测程度。在工作流图的特征发生变化时，这种丰富的信息允许处在不同应用领域的溯源信息具有不同的形式和粒度。

数据溯源信息允许以“最高分辨率”追踪单个数据项的处理，即溯源本身是处于单个数据项及它们所经历的操作的级别[45]。收集数据溯源信息通常应用在结构化数据模型和声明性查询语言中，并且在这一过程中，数据溯源信息还利用了清晰语义，即基于代数、微积分或其他形式化方法。这也是数据溯源在工作流溯源的上一层（即数据溯源的检测程度级别最高）的原因。

对于数据溯源问题，其重要的两个方面是数据标注与版本控制[46]。数据溯源问题可以被解释为标注的一种形式，即用数据的溯源标注数据元素。大多数被管护的数据都是与现有的数据结构的标注相关的，这种标注可以通过关系数据库中的主表所表现，但是有时某些数据的最新性和有效性的重要信息被保存在辅助表中。实际上标注数据本质上是半结构化的，并且通常存在于辅助数据库中。对“内核”数据的查询通常不会识别这种标注数据，这就是管护数据库中产生错误数据和脏数据的主要原因之一。目前有很多数据溯源工作在着手解决这一问题，例如参考文献[47-50]都是基于Polygen模型进行设计的，因而，数据的溯源对查询形式十分敏感。而对于版本控制问题，为了保证溯源过程中引用的版本是合适的，一种直接的方法是将数据的所有版本都存储下来，这样用户就有责任引用正确的版本，并从多个版本的数据中返回正确的查询结果。但是这种方法对于经度查询（如“最近的4个版本都更新了那些内容？”）难以处理。因为对于这类问题，这些方法需要将每一个相关数据至少浏览一次才能回答。为了解决这一问题，可以只存储连续版本间的变量。但是这类方法对于“返回所有存在指定实体的版本”这类问题仍需要遍历所有相关版本来返回结果。为了解决这些问题，参考文献[51]提出了一个归档的方法来平衡这两类方法。参考文献[52-53]也通过面向版本存储引擎、在关系数据库之上添加版本模块等方法解决版本控制问题。

由于溯源管理可以对数据的发展乃至源头加以追踪，在未来的研究中，可以考虑将溯源管理结合在不同场景中进行应用。例如在社交网络中，利用溯源管理技术可以帮助发现和追踪假信息或者谣言的源头。但是由于一些文章缺乏创建者的信息，因此无法直接追踪到其溯源信息。对于这类问题，目前已有一些工作正在开始着手解决[54-55]。除了社交网络外，溯源管理对于区块链技术也有很好的帮助。在区块链中，账本可以被视为一个比特币的溯源记录，这样区块链技术就可以在其他情境下记录溯源信息，如供应链溯源等。

3 典型数据管护系统及工具

本节首先介绍近年来推出的几种具有代表性的数据管护系统和工具，主要内容包括：系统结构、功能模块、应用范围等，

并对目前流行的典型数据管护系统的特点进行比较和分析。

3.1 DBWiki

DBWiki[56]是英国爱丁堡大学开发的一个支持数据管护的多用途数据平台。该平台收集数据并创建管护数据库，该系统还支持版本追踪、起源跟踪、注释等一些数据库系统不常提供的功能。同时，DBWiki将使用的便捷性和百科的灵活性与数据库的鲁棒性和稳定性相结合，提升自身的性能。DBWiki系统结构如**图3**所示。

在平台构建方面，该平台分为数据存储层、数据访问与修改层、请求与响应层。在数据存储层，DBWiki是基于分层数据模型构建的，采用有序、确定的数据树对数据进行建模，并且每一个树节点都有一个唯一标识，每一棵数据树都分配一个模式来描述树中的可能路径。DBWiki目前支持一些常见的数据和模式修改操作，包括增加、删除、修改、重命名等，同时还支持不同数据树间的节点或子树的复制、粘贴。对于每一个树中的节点，都分配一个时间戳，用于列出该节点出现过的各个数据库版本。基于时间戳和创建数据库版本的操作信息，还可以对数据节点提供溯源信息。

在数据库查询方面，DBWiki可以对数据树进行查询，并且将结果切入该系统的Wiki页面中，因此，DBWiki的页面实际上是结合了结构数据视图的超文本的动态内容。目前，DBWiki支持2种查询类型：一种是根据节点标识从数据库中检索节点或相应子树，同时利用查询可能包含的时间戳约束来过滤掉一些子树中的节点；另一种是基于路径表达式，即节点标识序列，路径表达式允许通过位置参考及某一节点的子节点的取值构成约束条件进行过滤。

在用户接口方面，DBWiki是通过网页浏览器与用户进行交互的，并且用户的查询及对数据的操作是使用URL进行编码的。一旦请求的数据被检索，DBWiki将通过HTML生成器生成相应的响应页面。

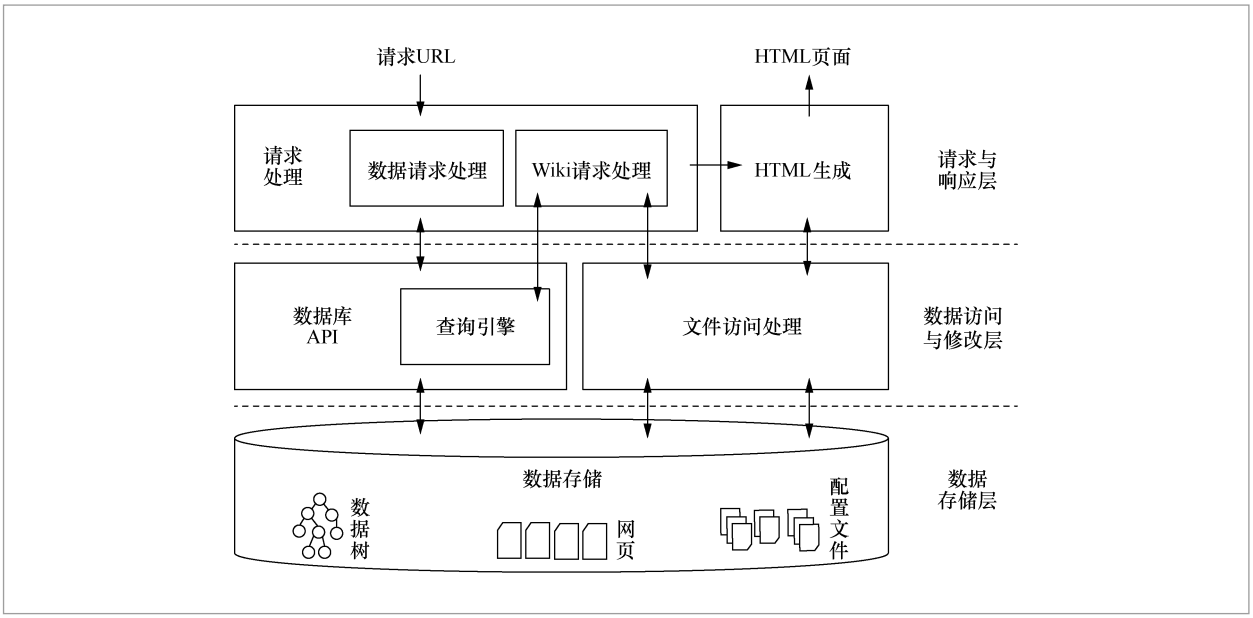

图 3 DBWiki 系统结构

DBWIki的一个设计标准是保持HTML的生成与系统的其他部分相分离，从而实现高度的定制化。HTML的生成主要由3个配置文件引导：第一个配置文件是HTML模板，包含用于存放一些预定义的用户接口组件的占位符；第二个配置文件是CSS文件，用于规定HTML的输出格式；第三个配置文件是布局定义，用于说明如何把树形结构数据映射为HTML页面、表格或列表。所有的文件都可以通过用户接口进行编辑，并且通过这些配置文件使用户在定制网页的感官上更加灵活。

3.2 Vizier

Vizier[57]是一个多模块的数据纠错和管理工具。该系统支持与Python、SQL的无缝集成操作以及自动数据管护和纠错方法。Vizier将Spark作为执行后台，可以处理多种格式的大规模数据集。同时该系统还支持对数据的起源和版本进行管理，从而允许协同和不确定性的管理操作。另外，将数据表方式（spreadsheet-style）接口、记事本（notebook）模块以及系统的可视化集成，从而对局内用户（user in the loop）予以支持与引导，使得该工具非常易于使用。

如**图4**所示，Vizier分为前端和后端两部分。前端的网络用户接口通过RESTAPI（网络API）与Vizier进行交互，从而完成对Vizier记事本工作流的创建、查看和修改。同时该API还提供了分页浏览功能，以便于对工作流中的结果数据集和各自相关元数据进行访问。Vizier的API主要建立在3个后端组件的基础之上，包括VisTrails、Mimir和Spark。其中工作流是通过VisTrails[58]进行管理的，它是一个能综合支持数据和工作流转换的数据探索系统。VisTrails的一个特性是在工

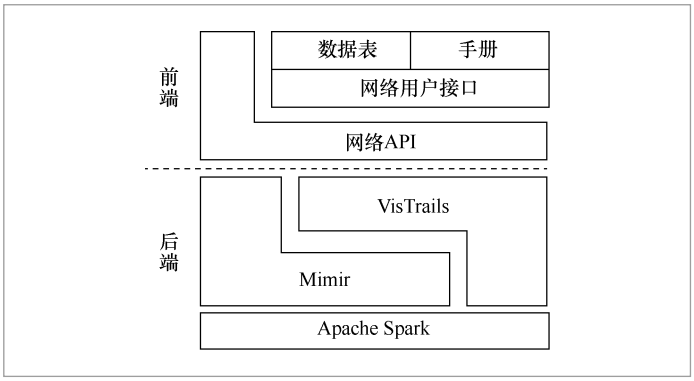

图4 Vizier系统结构

作流步骤之间的数据流都是通过Spark数据帧实现的。因此，工作流操作步骤可以直接被转换为Spark中的操作，并且在Spark本地上直接执行。Mimir组件[59]实现的按需ETL工具Lenses，是Vizier中数据验证和纠错的主要使能器。同时，Mimir组件还实现了有限形式的细粒度溯源管理，用于跟踪工作流中的错误或其他标注。

3.3 Clowder

Clowder[60]是一个开源数据管护系统，它能够支持多研究领域间的多数据类型的长尾数据和元数据的数据管护。机构和实验室可以在本地硬件或远程云计算资源上安装和定制自己的框架实例，为分散的研究人员团队提供共享服务。该系统采用了一个开源数据管护模型，包括有效的工作流、行为准则、邮件列表和聊天通道。利用该系统，数据可以被直接从仪器中读取，或者用户通过手动的方式上传，然后利用Web前端与远程合作者共享。

如**图5**所示，Clowder框架遵从典型的网络应用层次框架，使用HTML/JavaScript 网络前端和JSONRESTful网络服务API作为系统顶层架构。其主要的网络应用是采用PlayFramework和Scala

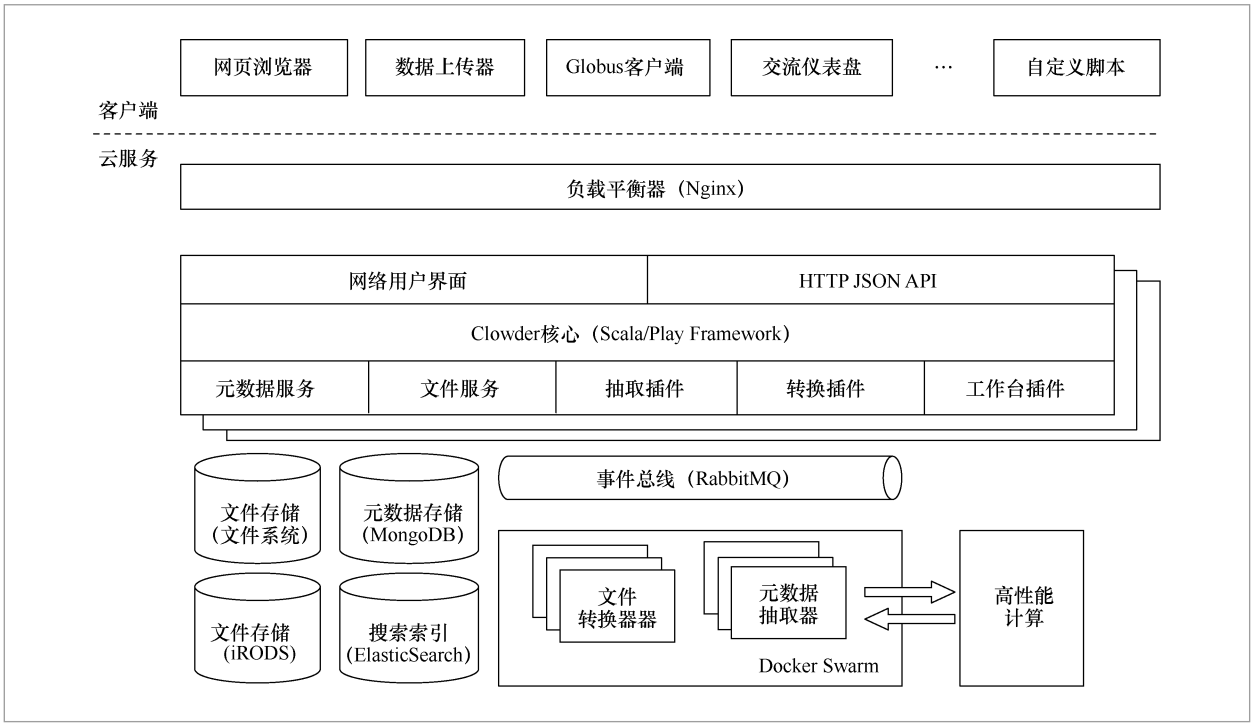

图 5 Clowder 系统结构

程序语言编写，其中Play Framework遵循的是模型-视图-控制器模式。模型在图形用户界面和网络API共享。而Scala特征将会利用依赖注入为一个或多个执行提供服务，这意味着在控制器中的代码不会与后端服务通信中的代码紧密相连。整个系统的所有信息都存储在MongoDB数据库中，其中索引建立在ElasticSearch中，而未处理的文件则存放在文件系统中，RabbitMQ是用于实施元数据抽取的总线。

3.4 MDCS

MDCS是由美国国家标准与技术研究院开发的一个开源的材料数据管护系统。该系统能够将材料数据捕获、共享并将其转换为基于XML的结构化形式。在该系统中，数据是使用XML模式中编码的用户自选择模板组织的，而这些模板被用于创建数据表单。MDCS使用非关系（NoSQL）数据库MongoDB存放文档。利用该系统，实现基于Web的模板驱动的表单数据检索、基于RESTful API调用的数据搜索功能。同时，该系统还能够实现多个MDCS存储库之间的互联，以支持跨搜索库的联合搜索。该系统自2015年发布起，共发布了6个版本，添加了对图像数据、BLOB（binary large objects）等类型数据的支持，资源库的备份与恢复等功能。

3.5 系统比较

本节对近年发布的5种数据管护系统进行了比较，见**表1**。除上述DBWiki[56]、Vizier[57]、Clowder[60]和MDCS外，还包括帮助数据科学家探索和管护数据的CURARE系统[61]以及处理医学影像的MATA系统[62]。

表1 性能对比和分析

比较项	DBWiki	Vizier	Clowder	MDCS	CURARE	MATA
数据模态	多类型数据	多类型数据	多类型数据	多类型数据	多类型数据	多类型数据
数据清洗	√		√		√	
实体识别			√	√	√	√
数据分类	√					
数据标注	√		√	√		√
模式匹配	√	√	√	√	√	√
实体链接	√					√
元数据管理	√	√	√	√	√	√
版本控制	√	√		√		
溯源	√			√		
可视化	√	√	√	√	√	√
检索	√	√	√	√		√
发布	√	√	√	√	√	√

表1比较了上述系统所能实现的功能，包括支持的数据模态，是否能够实现数据清洗、数据抽取（包括实体识别、数据分类、数据标注）、数据整合（包括模式匹配及实体链接）等功能，是否能够对元数据进行处理，是否能够进行版本控制和溯源，以及是否能够实现可视化、检索、发布。从**表1**中可以看出，各个系统都能够支持包括元数据在内的多种类型和格式的数据，并且对这些数据都进行了整合操作，最后将其结果可视化地返回给用户，这是由数据管护的目的所决定的，并且所有系统均支持简单的搜索操作。由于部分系统处理的数据在输入时已保证是干净、有效的，因此不具有数据清洗和抽取的功能。对于版本控制及溯源功能，由于Clowder、CURARE和MATA均存储的是当前的最新数据，并不对历史数据进行追踪，因此不具备版本控制和溯源的功能。

4 问题与挑战

在工业界，数据本身已经被各大公司和机构认为是一种重要资产，这些数据不仅需要存储起来，而且需要对其进一步评估，发现其现有或潜在价值，并且舍弃其中无价值的部分。由于数据海量性以及来源多样性的问题，数据管护技术成为科学界和工业界用于处理数据资产的重要手段，并且逐渐得到广泛的关注[63]。目前，有包括Vizier、Clowder等在内的数据管护软件和资源，利用这些产品可以实现对用户数据的清洗、归档以及管理工作。

尽管这些数据管护产品在不断地上线更新，但是在处理这些数据时还有许多问题需要解决。对于一个公司或机构而言，如果想要实现数据的合理正确管护，需要大量的人力物力投入，如选择合适的人员

策划如何管护数据，并选择合适的数据管护工具实现这一目标，或者自行开发一个符合自身要求的数据管护产品，实现对本公司数据资产的维护与管理。

5 结束语

数据管护技术是对海量数据进行有效处理的重要手段之一。数据管护技术通过对原始数据的清洗、集成、归档等操作，有效地提高了数据的使用率，并减缓了其淘汰的速度。本文围绕着数据管护技术的发展现状，系统介绍了数据管护中的关键技术和解决策略，并介绍了其在不同领域的应用。可以看出，数据管护技术已经在学术界和工业界得到了广泛的研究与应用。最后，本文提出了数据管护技术的发展前景与未来挑战，在今后的研究中将会根据这些问题进行更深入的探索与研究。

参考文献:

[1] 王芳, 慎金花. 国外数据管护(data curation)研究与实践进展[J]. 中国图书馆学报, 2014, 40(4): 116-128.
WANG F, SHEN J H. Advances in data curation abroad: research and practice[J]. Journal of Library Science in China, 2014, 40(4): 116-128.

[2] BISHOP B W, HANK C. Data curation profiling of biocollections[C]// Annual Meeting of the Association for Information Science and Technology, October 14-18, 2016, Copenhagen, Denmark. Hoboken: Wiley, 2016: 1-9.

[3] BOEHMKE B C. Data wrangling with R[M]. Switzerland: Springer Nature, 2016: 1-238.

[4] BEHESHTI S, TABEBORDBAR A, BENATALLAH B, et al. On automating basic data curation tasks[C]// The 26th International Conference on World Wide Web Companion, April 3-7, 2017, Perth, Australia. New York: ACM Press, 2017: 165-169.

[5] SINGH N, SINGH A K. Data privacy protection mechanisms in cloud[J]. Data Science and Engineering, 2018, 3(1): 24-39.

[6] BUNEMAN P, CHENEY J, TAN W C, et al. Curated databases[C]// Symposium on Principles of Database Systems, June 9-11, 2008, Vancouver, Canada. New York: ACM Press, 2008: 1-12.

[7] PBOHANNON, M FLASTER, W FAN, et al. A cost-based model and effective heuristic for repairing constraints by value modification[C]// International Conference on Management of Data, June 14-16, 2005, Baltimore, USA. New York: ACM Press, 2005: 143-154.

[8] CHU X, ILYAS I F, PAPOTTI P. Holistic data cleaning: putting violations into context[C]// International Conference on Data Engineering, April 8-12, 2013, Brisbane, Australia. Piscataway: IEEE Press, 2013: 458-469.

[9] CHU X, ILYAS I F, KRISHNAN S A, et al. Data cleaning: overview and emerging challenges[C]// International Conference on Management of Data, June 26 - July 1, 2016, San Francisco, USA. New York: ACM Press, 2016: 2201-2206.

[10] GOLAB L, KARLOFF H J, KORN F, et al. On generating near-optimal tableaux for conditional functional dependencies[J]. Proceedings of the VLDB Endowment, 2008 1(1): 376-390.

[11] GBESKALES B, ILYAS I F, GOLAB L, et al. On the relative trust between inconsistent data and inaccurate constraints[C]// International Conference on Data Engineering, April 8-12, 2013, Brisbane,

Australia. Piscataway: IEEE Press, 2013: 541-552.

[12] YAKOUT M, ELMAGARMID A K, NEVILLE J, et al. Guided data repair[J]. Proceedings of the VLDB Endowment, 2011, 4(5): 279-289.

[13] WANG J, KRASKA T, FRANKLIN M J, et al. CrowdER: crowdsourcing entity resolution[J]. Proceedings of the VLDB Endowment, 2012, 5(11): 1483-1494.

[14] HAO S, TANG N, LI G, et al. Cleaning relations using knowledge bases[C]// International Conference on Data Engineering, April 19-22, 2017, San Diego, USA. Piscataway: IEEE Press, 2017: 933-944.

[15] MARCUS A, PARAMESWARAN A. Crowdsourced data management: industry and academic perspectives[J]. Foundations and Trends in Databases, 2013, 6(1-2): 1-161.

[16] GOKHALE C, DAS S, DOAN A, et al. Corleone: hands-off crowdsourcing for entity matching[C]// International Conference on Management of Data, June 22-27, 2014, Snowbird, USA. New York: ACM Press, 2014: 601-612.

[17] HAAS D, WANG J, WU E, et al. CLAMShell: speeding up crowds for low-latency data labeling[J]. Proceedings of the VLDB Endowment, 2015, 9(4): 372-383.

[18] MOZAFARI B, SARKAR P, FRANKLIN M J, et al. Scaling up crowd-sourcing to very large datasets: a case for active learning[J]. Proceeding of the VLDB Endowment, 2014, 8(2): 125-136.

[19] ANANTHAKRISHNA R, CHAUDHURI S, GANTI V. Eliminating fuzzy duplicates in data warehouses[C]// International Conference on Very Large Data Bases, August 20-23, 2002, Hong Kong, China. San Francisco: Morgan Kaufmann, 2002: 586-597.

[20] WANG J, KRISHNAN S, FRANKLIN M J, et al. A sample-and-clean framework for fast and accurate query processing on dirty data[C]// International Conference on Management of Data, June 22-27, 2014, Snowbird, USA. New York: ACM Press, 2014: 469-48

[21] KOLB L, THOR A, RAHM E. Dedoop: efficient deduplication with Hadoop[J]. Proceeding of the VLDB Endowment, 2012, 5(12): 1878-1881.

[22] KHAYYAT Z, ILYAS I F, JINDAL A, et al. BigDansing: a system for big data cleansing[C]// International Conference on Management of Data, May 31-June 4, 2015, Melbourne, Australia. New York: ACM Press, 2015: 1215-1230.

[23] CHU X, ILYAS I F, KOUTRIS P. Distributed data deduplication[R]. Waterloo: University of Waterloo, 2016.

[24] HUI J, LI L, ZHANG Z. Integration of big data: a survey[C]// International Conference of Pioneering Computer Scientists, Engineers and Educators, September 21-23, 2018, Zhengzhou, China. Heidelberg: Springer, 2018: 101-121.

[25] LI F, LEE M, HSU W, et al. Linking temporal records for profiling entities[C]// International Conference on Management of Data, May 31-June 4, 2015, Melbourne, Australia. New York: ACM Press, 2015: 593-605.

[26] ZABEDJAN A, AKCORA C G, OUZZANI M, et al. Temporal rules discovery for web data cleaning[J]. Proceedings of the VLDB Endowment, 2015, 9(4): 336-347.

[27] PETERMANN A, JUNGHANNS M, MÜLLER R, et al. Graph-based data integration and business intelligence with BIIIG[J]. Proceedings of the VLDB Endowment, 2014, 4(13): 1577-1580.

[28] LI Q, LI Y, GAO J, et al. Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation[C]// International Conference

on Management of Data, June 22-27, 2014, Snowbird, USA. New York: ACM Press, 2014: 1187-1198.

[29] LI Q, LI Y, GAO J, et al. A confidence-aware approach for truth discovery on long-tail data[J]. Proceedings of the VLDB Endowment, 2014, 8(4): 425-436.

[30] REKATSINAS T, JOGLEKAR M, GARCIA-MOLINA H, et al. SLiMFast: guaranteed results for data fusion and source reliability[C]// International Conference on Management of Data, May 14-19, 2017, Chicago, USA. New York: ACM Press, 2017: 1399-1414.

[31] YU R, GADIRAJU U, FETAHU B, et al. FuseM: query-centric data fusion on structured Web markup[C]// International Conference on Data Engineering, April 19-22, 2017, San Diego, USA. Piscataway: IEEE Press, 2017: 179-182.

[32] SALLOUM M, DONG X L, SRIVASTAVA D, et al. Online ordering of overlapping data sources[J]. Proceedings of the VLDB Endowment, 2013, 7(3): 133-144.

[33] REKATSINAS T, DONG X L, SRIVASTAVA D. Characterizing and selecting fresh data sources[C]// International Conference on Management of Data, June 22-27, 2014, Snowbird, USA. New York: ACM Press, 2014: 919-930.

[34] BONAQUE R, CAO T D, CAUTIS B, et al. Mixed-instance querying: a lightweight integration architecture for data journalism[J]. Proceedings of the VLDB Endowment, 2016, 9(13): 1513-1516.

[35] CHAMANARA J, KÖNIG-RIES B, JAGADISH H V. QUIS: InSitu heterogeneous data source querying[J]. Proceedings of the VLDB Endowment, 2017, 10(12): 1877-1880.

[36] SAWADOGO P, KIBATA T, DARMONT J. Metadata management for textual documents in data lakes[C]// International Conference on Enterprise Information Systems, May 3-5, 2019, Heraklion, Greece. [S.l.]: SciTePress, 2019: 72-83.

[37] STEIN B, MORRISON A. The enterprise data lake: better integration and deeper analytics[J]. Technology Forecast, 2014(1): 1-9.

[38] QUIX C, HAI R, VATOV I. Metadata extraction and management in data lakes with GEMMS[J]. Complex Systems Informatics and Modeling Quarterly, 2016(9): 67-83.

[39] HAI R, GEISLER S, QUIX C. Constance: an intelligent data lake system[C]// International Conference on Management of Data, June 26-July 1, 2016, San Francisco, USA. New York: ACM Press, 2016: 2097-2100.

[40] INMON B. Data lake architecture: designing the data lake and avoiding the garbage dump[M]. [S.l.]: Technics Publications, 2016.

[41] FANG H. Managing data lakes in big data era: what's a data lake and why has it became popular in data management ecosystem[C]// International Conference on Cyber Technology in Automation, Control and Intelligent Systems, June 8-12, 2015, Shenyang, China. Piscataway: IEEE Press, 2015: 820-824.

[42] MILOSLAVSKAYA N G, TOLSTOY A I. Application of big data, fast data, and data lake concepts to information security issues[C]// International Conference on Future Internet of Things and Cloud Workshops, August 22-24, 2016, Vienna, Austria. Piscataway: IEEE Press, 2016: 148-153.

[43] MACCIONI A, TORLONE R. Crossing the finish line faster when paddling the data lake with kayak[J]. Proceedings of the VLDB Endowment, 2017, 10(12): 1853-1856.

[44] HERSCHEL M, DIESTELKA¨MPER R, LAHMAR H B. A survey on provenance:

what for, what form, what from[J]. The VLDB Journal, 2017, 26(6): 881-906.

[45] CHENEY J, CHITICARIU L, TAN W C. Provenance in databases: why, how, and where[J]. Foundations and Trends in Databases, 2009, 1(4): 379-474.

[46] BUNEMAN P, TAN W C. Data provenance: what next[J]. SIGMOD Record, 2018, 47(3): 5-16.

[47] BHAGWAT D, CHITICARIU L, TAN W C, et al. An annotation management system for relational databases[J]. The VLDB Journal, 2005, 14(4): 373-396.

[48] CHITICARIU L, W CH TAN, VIJAYVARGIYA G. DBNotes: a post-it system for relational databases based on provenance[C]// International Conference on Management of Data, June 14-16, 2005, Maryland, USA. New York: ACM Press, 2005: 942-944.

[49] GEERTS F, KEMENTSIETSIDIS A, MILANO D. MONDRIAN: annotating and querying databases through colors and blocks[C]// International Conference on Data Engineering, April 3-8, 2006, Atlanta, USA. Piscataway: IEEE Press, 2006.

[50] BUNEMAN P, CHENEY J, VANSUMMEREN S. On the expressiveness of implicit provenance in query and update languages[J]. ACM Transactions on Database Systems, 2008, 33(4): 1-47.

[51] BUNEMAN P, KHANNA S, TAJIMA K, et al. Archiving scientific data[J]. ACM Transactions on Database Systems, 2004, 29: 2-42.

[52] HUANG S, XU L, LIU J, et al. Orpheusdb: bolt-on versioning for relational databases[J]. Proceeding of the VLDB Endowment, 2017, 10(10): 1130-1141.

[53] MADDOX M, GOEHRING D, ELMORE A J, et al. Decibel: the relational dataset branching system[J]. Proceeding of the VLDB Endowment, 2016, 9(9): 624-635.

[54] LAPPAS T, TERZI E, GUNOPULOSD, et al. Finding Effectors in Social Networks[C]// International Conference on Knowledge Discovery and Data Mining, July 25-28, 2010, Washington, DC, USA. New York: ACM Press, 2010: 1059-1068.

[55] SHAH D, ZAMAN T. Rumors in a network: Who's the culprit[J]. Information Forensics and Security, 2011, 57(8): 5163-5181.

[56] BUNEMAN P, CHENEY J, LINDLEY S, et al. DBWiki: a structured wiki for curated data and collaborative data management[C]// International Conference on Management of Data, June 12-16, 2011, Athens, Greece. New York: ACM Press, 2011: 1335-1338.

[57] BRACHMANN M, BAUTISTA C, CASTELO S, et al. Data debugging and exploration with vizier[C]// International Conference on Management of Data, June 30-July 5, 2019, Amsterdam, The Netherlands. New York: ACM Press, 2019: 1877-1880.

[58] CALLAHAN S P, FREIRE J, SANTOS E, et al. VisTrails: visualization meets data management[C]// International Conference on Management of Data, June 27-29, 2006, Chicago, USA. New York: ACM Press, 2006: 745-747.

[59] YANG Y, MENEGHETTI N, FEHLING R, et al. An on-demand approach to ETL[J]. Proceedings of the VLDB Endowment, 2015, 8(12): 1578-1589.

[60] MARINI L, GUTIERREZ-POLO I, KOOPER R, et al. Clowder: open source data management for long tail data[C]// The Practice and Experience on Advanced Research Computing, July 22-26, 2018, Pittsburgh, USA. New York: ACM Press, 2018: 1-8.

[61] VARGAS-SOLAR B, KEMP G, GALLEGOS I H, et al. Demonstrating data collections curation and exploration with curare[C]// International Conference on Extending Database

Technology, March 26-29, 2019, Lisbon, Portugal. [S.l.:s.n.], 2019: 598-601.

[62] WOLLATZ L, SCOTT M, JOHNSTON S J, et al. Curation of image data for medical research[C]// International Conference on e-Science, October 29 - November 1, 2018, Amsterdam, The Netherlands. Piscataway: IEEE Press, 2018: 105-113.

[63] 杜小勇, 陈跃国, 范举, 等. 数据整理——大数据治理的关键技术[J]. 大数据, 2019, 5(3): 13-22.
DU X Y, CHEN Y G, FAN J, et al. Data wrangling:a key technique of data governance[J]. Big Data Research, 2019, 5(3): 13-22.

作者简介

于明鹤(1989-),女,博士,东北大学软件学院讲师,主要研究方向为大数据、信息检索等。

聂铁铮(1980-),男,博士,东北大学计算机科学与工程学院副教授,主要研究方向数据集成、大数据处理、区块链。

李国良(1980-),男,博士,清华大学计算机科学与技术系教授,主要研究方向为数据清洗、数据整合、众包数据管理等。

收稿日期: 2019-09-21
基金项目: 中国博士后科学基金资助项目(No.2019M651134);广东省普及型高性能计算机重点实验室(2017B030314073);中央高校基本科研业务专项资金资助项目(No.N181703006)
Foundation Items: China Postdoctoral Science Foundation(No.2019M651134), Guangdong Province Key Laboratory of Popular High Performance Computers(2017B030314073), Fundamental Research Funds for the Central Universities(No.N181703006)