Institute for Parallel and Distributed Systems / AS

Leveraging the Data Lake
Current State and Challenges

Corinna Giebler, Christoph Gröger, Eva Hoos, Holger Schwarz, and Bernhard Mitschang

In: Proceedings of the 21st International Conference on Big Data Analytics and Knowledge Discovery (DaWaK 2019)

BIBTEX:
@inproceedings{Giebler2019,
author = {Giebler, Corinna and Gr{\"{o}}ger, Christoph and Hoos, Eva and Schwarz, Holger and Mitschang, Bernhard},
booktitle = {Proceedings of the 21st International Conference on Big Data Analytics and Knowledge Discovery (DaWaK 2019)},
title = {{Leveraging the Data Lake - Current State and Challenges}},
year = {2019},
doi = {10.1007/978-3-030-27520-4_13}
}

Leveraging the Data Lake

Current State and Challenges

Corinna Giebler[1][0000-0002-5726-0685], Christoph Gröger[2][0000-0001-6615-4772], Eva Hoos[2], Holger Schwarz[1], and Bernhard Mitschang[1]

[1] University of Stuttgart, Universitätsstraße 38, 70569 Stuttgart, Germany
{Firstname.Lastname}@ipvs.uni-stuttgart.de
[2] Robert Bosch GmbH, Borsigstraße 4, 70469 Stuttgart, Germany
{Firstname.Lastname}@de.bosch.com

Abstract. The digital transformation leads to massive amounts of heterogeneous data challenging traditional data warehouse solutions in enterprises. In order to exploit these complex data for competitive advantages, the data lake recently emerged as a concept for more flexible and powerful data analytics. However, existing literature on data lakes is rather vague and incomplete, and the various realization approaches that have been proposed neither cover all aspects of data lakes nor do they provide a comprehensive design and realization strategy. Hence, enterprises face multiple challenges when building data lakes. To address these shortcomings, we investigate existing data lake literature and discuss various design and realization aspects for data lakes, such as governance or data models. Based on these insights, we identify challenges and research gaps concerning (1) data lake architecture, (2) data lake governance, and (3) a comprehensive strategy to realize data lakes. These challenges still need to be addressed to successfully leverage the data lake in practice.

Keywords: Data Lakes, State of the Art, Challenges, Industry Case.

1 Introduction

The digital transformation towards capturing and analyzing big data provides novel opportunities for enterprises to improve business and optimize processes [1]. Sensors from the Internet of Things (IoT), for example, enable the continuous gathering of production data, allowing the proactive assessment and the predictive regulation of production processes [1]. Many other novel data sources can be integrated and analyzed to generate new insights for the enterprise, using advanced analytics such as data mining, text analytics or artificial intelligence [2]. In the following, we summarize advanced analytics and traditional business intelligence as *data analytics*. The knowledge gained from data analytics represents a significant competitive advantage for enterprises [3].

Data captured for these data analytics tend to be heterogeneous, voluminous, and complex, and thus pose a challenge on traditional enterprise data analytics solutions based on data warehouses. In order to enable comprehensive and flexible data analytics on these complex data, the concept of the *data lake* emerged in recent years. In a data

lake, any kind of data are available for flexible analytics without predefined use cases [4]. To this end, data are stored in a raw or almost raw format.

However, multiple challenges arise when building and using data lakes. Existing literature on data lakes is vague and inconsistent. Numerous approaches exist to realize selected aspects of a data lake, e.g., governance or data models, but it is unclear whether these approaches are sufficient and where additional concepts are needed.

In this paper, we address this gap. We investigate the current state of the art for data lakes and identify remaining research challenges towards a successful data lake. To this end, we make the following contributions:

- We investigate the current state of the general data lake concept.
- We discuss existing design and realization aspects.
- We identify challenges and research gaps for data lakes.

The remainder of this paper is organized as follows: Section 2 investigates existing data lake literature, while Section 3 discusses different design and realization aspects for data lakes and whether they are sufficiently covered by literature. Based on the gained insights, Section 4 identifies remaining challenges and research gaps. Finally, Section 5 concludes the paper.

2 Current State: Data Lakes in Literature

In order to put data lakes into practice, a uniform understanding of the general concept is needed. We conducted a comprehensive literature review to identify the central characteristics of a data lake. However, it showed that there is no commonly accepted concept. Instead, various definitions and views exist on data lakes, some of which are contradictory. In this section, we summarize our findings and investigate different points of view on the concept of data lakes.

The first person to use the term "data lake" was James Dixon in 2010 [5]. He defined a data lake to store data in a "natural" [5], i.e., *raw*, state compared to a traditional data mart. Large amounts of *heterogeneous data* are added from a single source [6] and users can access them for a *variety of analytical use cases*. The idea of single-source data lakes did not find much acceptance in literature. Nowadays, data lakes have been redefined to contain data from an arbitrary number of sources [7–9]. Dixon's central point of storing raw data is reflected by all investigated definitions. In some definitions, however, this characteristic is extended so that preprocessed data and results from previously performed analyses may additionally be stored in the data lake (e.g., in [10]).

To make storing large amounts of heterogeneous data financially feasible, data lakes must provide inexpensive storage [4, 8]. In many cases, data lakes are directly linked to *Hadoop*[1] and the HDFS [5, 7, 11]. However, the data lake represents a concept, while Hadoop is only one of many possible storage technologies [7, 11]. Dixon and other literature instead proposes a diverse tool landscape for data lakes, i.e., the most appropriate tool to manage and process certain data should be used [6, 12]. For example,

[1] http://hadoop.apache.org/

MongoDB, Neo4J, or other data storage systems could also be used in data lakes [11]. These storage systems could be managed on premise or in the cloud [13].

Beyond data lakes being raw data repositories and the infrastructure they are built on, further characteristics of the concept depend on the exact definition, of which there are many. Often, definitions even contradict each other, especially when it comes to the *role of a data lake*, the *involved user groups*, and *governance and metadata*. In the literature, the *role of a data lake* ranges from pure central data storage [11] to the provider of services related to data management and analysis [14, 15]. For the *involved user groups*, some data lake definitions involve a wide variety of user groups (e.g., [8, 11]), while others name data scientists as the only users of a data lake (e.g., [7]). Regarding *governance and metadata*, some definitions claim a data lake comprises neither governance nor metadata (e.g., [8, 16]), while others state governance as a central aspect of data lakes (e.g., [14, 17]). Metadata and governance ensure that data are reliable and can be accessed and understood. A data lake without governance is said to risk transforming into a *data swamp* [18], where data cannot be used for value creation. Often, data lake concepts that include governance are also referred to as *data reservoir* [18, 19] to differentiate it from the ungoverned data lake. Additionally, data reservoirs are said to enforce more structure on data than a data lake. To achieve this structure, raw data are modeled using appropriate modeling techniques [19]. However, governance is mostly seen as part of a data lake and literature does not differentiate data lake governance from data reservoir governance. Furthermore, data modeling also plays a role in data lakes for data integration and facilitated use of data [15]. Therefore, we will not consider the differentiation between data lake and data reservoir here. Instead, we will use definitions that see governance as part of the data lake.

Some authors also use the term enterprise data lake in order to emphasize the data lake's application in an enterprise environment (e.g., [15, 20, 21]). However, this term is used synonymously to the general data lake term. None of the investigated sources differentiates the characteristics of an enterprise data lake from those of a general data lake. Thus, we will likewise not consider this differentiation in this work.

Overall, literature is split over the concrete characteristics of data lakes. There exists no uniform data lake concept and thus no comprehensive realization strategy. However, there is further literature focusing on particular aspects of data lakes. We examine these design and realization aspects in the following section.

3 Design and Realization Aspects for Data Lakes

Although the literature defining and describing data lakes contains little to no information on their practical realization, various approaches to realize selected aspects of the data lake do exist. Due to space restrictions, we focus on the realization aspects *data lake architecture*, *data lake modeling*, *metadata management*, and *data lake governance*. In the following subsections, we present literature related to these aspects and discuss whether the aspects are sufficiently covered for a practical data lake realization.

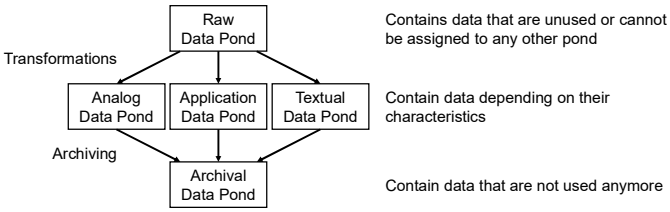

Fig. 1. In the pond architecture, data flow through the different ponds and are always available in only one of them [25].

3.1 Data Lake Architecture

The data lake architecture describes how data are conceptually organized within a data lake. It facilitates the use of a data lake [15] by defining where data can be found in the condition (e.g., raw or pre-processed) needed for a particular use. There exist two variants for the overall architecture of data lakes, namely *zone* and *pond architectures*.

Various alternatives exist for *zone architectures* (e.g., [15, 17, 22–24]). They differ in multiple aspects, such as the number and characteristics of zones. Although there is no commonly accepted zone architecture, the idea always remains the same: Data are assigned to a zone according to the degree of processing that has been applied to them. When data are ingested into the data lake, they are stored in raw format in the raw zone, which is common between all zone architectures. Other zones then condition these data further. Some zones standardize data to fit a common format [15, 23], others cleanse the data [17]. Some zone architectures even include a data-mart-like zone, where data are prepared to fit certain use cases and tools [15]. The advantage of zone architectures is that even if data are available in a transformed and pre-processed format, they can still be accessed as raw data in the raw zone.

The *pond architecture* [25] is another variant of the overall architecture for data lakes (see Fig. 1). Data in the data lake are distributed across five different ponds. However, in contrast to zone architectures, data are only available in one pond at any given point in time. Upon ingestion, data are stored in the raw data pond. Only unused data and data that do not fit into any of the other ponds remain in the raw data pond, all other data flow into the analog, application, or textual data pond. Which pond they flow to depends on the data's characteristics. The analog data pond contains measurement data, such as log files or IoT data. In the application data pond, all data that are generated by applications are stored. The textual data pond contains text data. Other data, like images and videos, remain in the raw data pond. When data are not used anymore, they leave their respective pond and move to the archival data pond. As data flow through the ponds, they are transformed depending on the pond they currently belong to [25]. For example, outliers may be deleted from the analog data pond and textual data may be structured. The advantage of this approach is that data are pre-processed and can easily be analyzed. However, when data leave the raw data pond, they are conditioned and their original format is lost. This contradicts the general idea of a data lake.

In addition to those two general data lake architectures, literature suggests the *lambda architecture* [26] to organize batch and streaming data [11, 27]. The conceptual

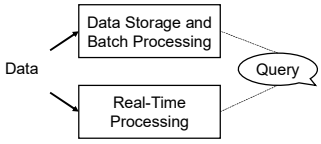

Fig. 2. The lambda architecture [26] enables separate batch and real-time processing.

idea of this architecture is depicted in Fig. 2. Incoming data are copied to two different branches. On one branch, data are stored permanently and periodically processed in batches. On the other branch, incoming data are processed in real-time to deliver quick results. However, in practice the lambda architectures often is adapted (e.g., in [12, 19]). Such adaptions are, e.g., the BRAID architecture [28] or Bolster [29].

While there exist various alternatives for data lake architectures, there is no generally accepted approach. Although zone architectures are more frequently mentioned in literature than the other variants, the definitions of the various zones differ greatly in some cases. To the best of our knowledge, there exist no assessments or comparisons of the different data lake architectures. Additionally, the data lake architectures proposed in literature only cover parts of the data lake. It is not defined how the data lake architectures interact with other aspects of the data lake. For example, it remains unclear how data lake modeling can be done, or what storage technologies can be used. Thus, defining and realizing an adequate data lake architecture is still a challenging task.

3.2 Data Lake Modeling

In the context of data lakes, literature speaks of schema-on-read [11, 15], i.e., data are only transformed when they are retrieved from the data lake for certain use cases. Transforming data requires knowledge of their schema, which in turn necessitates data modeling. However, deferring all modeling to data usage is infeasible [15] as data modeling is needed to ensure certain levels of data quality, data comprehensibility, and data integration [15, 30]. Thus, we need data models that allow data modeling with little effort and, at the same time, maintain the flexibility of the data lake.

There exist a few approaches to data lake modeling. For example, the *data droplets model* [31] models each data object, such as a single document, in the data lake as an RDF graph. These smaller graphs are then combined into an overarching data lake graph according to the relationships between the data objects.

Another approach is to model data in the data lake using *data vault* (see, e.g., in [19]). Data vault originates from the data warehouse context. It provides a flexible and simple way to model data. However, it was designed for structured data. While there exist approaches to integrate semi-structured data into data vault (e.g., [32]), integrating unstructured data is not yet covered.

Overall, literature explicitly mentions these two approaches and provides some ideas on how to model data in data lakes, but it does not offer further guidance. Multiple other data models exist in other contexts, such as 3rd normal form or head-version tables [33].

These data models might also be suitable candidates for data lake modeling but no assessments or best practices do exist so far. There also is no guidance on how to use the different models in a data lake architecture (Section 3.1). Additionally, many existing approaches such as 3rd normal form, head-version tables are available for structured data only. To include semi-structured or unstructured data in these models, further concepts are necessary (such as [34] or [35]). Thus, a comprehensive discussion and assessment of existing data models is still necessary for data lakes.

3.3 Metadata Management

Whenever data from different sources, contexts, and with various schemata are brought together, metadata is necessary to keep track of these data. This also applies to data lakes, where metadata management is a crucial part [36]. Metadata capture information on the actual data, e.g., schema information, semantics, or lineage [10, 14]. They ensure that data can be found, trusted, and used. There exists a large number of different approaches for metadata management. Due to space constraints, this section provides only a general overview over metadata management approaches explicitly for data lakes.

According to data lake literature, *data catalogs* [18] are used to store metadata. Whenever data is added to the data lake, the corresponding metadata has to be added to the catalog. Users are then able to search this catalog and receive additional information on the data, such as schema, relationships, or provenance [37]. However, not all metadata relevant for data lakes are covered by these catalogs [12].

Automatic extraction of metadata is an important topic in data lake environments, as vast amounts of data are ingested and stored. Tools like *GEMMS* [36] can be used that do not only extract, but also annotate the metadata with semantic information and allow querying these metadata. Some data lake concepts even provide an extensive metadata management system to store and query metadata [38]. To extract schematic metadata even from schema-free data sources, schema profiling can be used [39].

There exist various *metadata models* for data lakes (e.g., [36, 40, 41]). Some of them provide only little description and realization details [40], while others are designed for one specific application [41]. A model that is both generic and described in appropriate detail is proposed by Quix et al. [36]. It contains information on both the structure and semantical context of the data the metadata describe.

In addition to structure and semantics of data, metadata on the origin of data is just as important [15]. Lineage metadata describes where data came from, how they were produced, and how they had been processed. However, lineage metadata is not or insufficiently considered in all investigated metadata models for data lakes. Only one metadata model mentions provenance [41], but there is no further explanation on the storage or usage of provenance information.

While metadata management is crucial for data lakes, no comprehensive metadata management strategy covering all data lake metadata is available. However, metadata management is also represented in other contexts, e.g., data warehousing. Therefore, further investigation of approaches beyond the data lake literature is necessary.

3.4 Data Lake Governance

Metadata management is only a part of an overarching data lake governance. Governance comprises all kinds of policies and rules to ensure data quality and rule compliance [18]. Even though some early literature excludes governance from the data lake concept [8, 16], more recent literature views it as a very important aspect [7, 9, 10, 14]. In a data lake, governance has to compromise between control and flexibility [4]. Since many different kinds of data are managed in a data lake, data governance has to consider their differences. For example, master data managed in a data lake has a high need for governance, while IoT data typically needs less control and governance.

In literature, there exist only few approaches to data lake governance, especially for diverse data. There exists a general governance framework [18], giving some guidance on what needs to be considered in data lake governance. For example, the framework defines various roles involved in data lake governance, such as data stewards. In the various zone models (see Section 3.1), some governance principles are applied. For example, one zone model provides a sensitive zone, where sensitive data are encrypted [22]. Other zone models allow to encrypt or tokenize data in the raw zone [17].

However, to the best of our knowledge, none of the existing governance concepts considers the different kinds of data managed in a data lake and their governance requirements. Data lake governance is rudimentarily covered in literature, especially concerning sensitive data. Although there are governance approaches in other contexts, governance for data lakes must meet new requirements, such as compromising between control and flexibility. Thus, a governance concept specifically for data lakes is needed.

4 Challenges and Research Gaps

Even though multiple approaches exist that cover different aspects of data lakes (see Section 3), a comprehensive strategy to realize data lakes is missing. Additionally, it became clear that some aspects are only insufficiently covered by literature. We identified research gaps in three areas of data lakes:

Data Lake Architecture. For *data lake architecture*, the heterogeneity of concepts poses a major problem. There exist no assessments or discussions for the different alternatives. No generally accepted architecture is available, and some of the proposed architectures do not align with the data lake concept (e.g., ponds). Therefore, it is necessary to closely investigate and compare the existing alternatives to identify similarities and shortcomings. Additionally, data lake architectures cover only the conceptual organization of data. No data lake architecture exists that includes other data lake aspects, like data lake modeling or data lake infrastructure. To realize data lakes, a generalized and comprehensive data lake architecture is needed.

Data Lake Governance. The other aspect for which research gaps still remain is *data lake governance*. The data lake poses novel requirements on flexibility and open access that conflict with traditional governance approaches, for instance from data warehousing. Therefore, a comprehensive governance concept developed specifically for data lakes is required. This concept also has to consider the different kinds of data

managed in the data lake, and correspond to their variant data management requirements.

Comprehensive strategy. In addition to these research gaps in the discussed data lake aspects, data lake literature is lacking a *comprehensive design and realization strategy.* Such a strategy considers interdependencies between different data lake aspects, such as data lake architecture and data lake modeling, and combines all aspects into one comprehensive and systematic data lake concept. Thus, it can provide guidance and decision support for the realization of data lakes.

The research gaps in these three areas need to be addressed to allow the definition of a holistic data lake concept and thus to leverage data lakes in practice.

5 Conclusion and Future Work

This paper summarizes our findings on the current state of data lakes. We conducted a comprehensive literature review on data lakes and existing approaches for their design and realization. It turned out that literature on data lakes is often split over the characteristics of a data lake. There exists no universal data lake concept. When it comes to realizing data lakes, research gaps concerning *data lake architecture* and *data lake governance* need to be resolved. Additionally, the lack of a *comprehensive design and realization strategy* that considers interdependencies between data lake aspects constitutes a major challenge to leverage data lakes in practice.

Our future work focuses on overcoming these challenges. Existing concepts for data lakes need to be investigated, categorized, and evaluated. Different data lake architectures have to be compared and evaluated with regard to their suitability to typical data lake use cases. Further approaches for e.g., semantic integration, data curation, or schema extraction have to be considered in a data lake context. Using all of these insights, a comprehensive design and realization strategy can be defined.

References

1. Lee, J., Kao, H.-A., Yang, S.: Service Innovation and Smart Analytics for Industry 4.0 and Big Data Environment. In: Proceedings of the 6th CIRP Conference on Industrial Product-Service Systems (2014).
2. Russom, P.: Big data analytics. TDWI best Practices report, fourth Quarter (2011).
3. Margulies, J.C.: Data as Competitive Advantage. Winterberry Group (October) (2015).
4. Tyagi, P., Demirkan, H.: Data Lakes: The biggest big data challenges. Analytics. 9 (6), 56–63 (2016).
5. Dixon, J.: Pentaho, Hadoop, and Data Lakes, https://jamesdixon.wordpress.com/2010/10/14/pentaho-hadoop-and-data-lakes/.
6. Dixon, J.: Data Lakes Revisited, https://jamesdixon.wordpress.com/2014/09/25/data-lakes-revisited/.
7. Madera, C., Laurent, A.: The Next Information Architecture Evolution: The Data Lake Wave. In: Proceedings of the 8th International Conference on Management of Digital EcoSystems (MEDES) (2016).

8. Fang, H.: Managing data lakes in big data era: What's a data lake and why has it became popular in data management ecosystem. In: Proceedings of the 2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER) (2015).
9. O'Leary, D.E.: Embedding AI and Crowdsourcing in the Big Data Lake. IEEE Intelligent Systems 29 (5), 70–73 (2014).
10. Terrizzano, I., Schwarz, P., Roth, M., Colino, J.E.: Data Wrangling: The Challenging Journey from the Wild to the Lake. In: Proceedings of the 7th Biennial Conference on Innovative Data Systems Research (CIDR) (2015).
11. Mathis, C.: Data Lakes. Datenbank-Spektrum. 17 (3), 289–293 (2017).
12. Gröger, C., Hoos, E.: Ganzheitliches Metadatenmanagement im Data Lake: Anforderungen, IT-Werkzeuge und Herausforderungen in der Praxis. In: Proceedings der 18. Fachtagung Datenbanksysteme für Business, Technologie und Web (BTW) (2019).
13. Lock, M.: Maximizing your Data Lake with a Cloud or Hybrid Approach. Aberdeen Group (2016).
14. IBM Analytics: The governed data lake approach. IBM. (2016).
15. Madsen, M.: How to Build an Enterprise Data Lake: Important Considerations before Jumping In. Third Nature Inc. (2015).
16. Gartner Inc.: Gartner Says Beware of the Data Lake Fallacy, https://www.gartner.com/newsroom/id/2809117, (2014).
17. Patel, P., Wood, G., Diaz, A.: Data Lake Governance Best Practices. DZone Guide to Big Data - Data Science & Advanced Analytics 4, 6–7 (2017).
18. Chessell, M., Scheepers, F., Nguyen, N., van Kessel, R., van der Starre, R.: Governing and Managing Big Data for Analytics and Decision Makers. IBM (2014).
19. Topchyan, A.R.: Enabling Data Driven Projects for a Modern Enterprise. Proc. Inst. Syst. Program. RAS (ISP RAS 2016). 28 (3), 209–230 (2016).
20. Stein, B., Morrison, A.: The enterprise data lake: Better integration and deeper analytics. Technol. Forecast Rethinking Integration 1, (2014).
21. Farid, M., Roati, A., Ilyas, I.F., Hoffmann, H.-F., Reuters, T., Chu, X.: CLAMS: Bringing Quality to Data Lakes. In: Proceedings of the 2016 International Conference on Management of Data (SIGMOD) (2016).
22. Gorelik, A.: The Enterprise Big Data Lake. O'Reilly Media, Inc. (2016).
23. Sharma, B.: Architecting Data Lakes - Data Management Architectures for Advanced Business Use Cases. O'Reilly Media, Inc. (2018).
24. Zikopoulos, P., DeRoos, D., Bienko, C., Buglio, R., Andrews, M.: Big Data Beyond the Hype. McGraw-Hill Education (2015).
25. Inmon, B.: Data Lake Architecture - Designing the Data Lake and avoiding the Garbage Dump. Technics Publications (2016).
26. Marz, N., Warren, J.: Big Data - Principles and best practices of scalable real-time data systems. Manning Publications Co. (2015).
27. Gröger, C.: Building an Industry 4.0 Analytics Platform. Datenbank-Spektrum. 18 (1), 5–14 (2018).
28. Giebler, C., Stach, C., Schwarz, H., Mitschang, B.: BRAID - A Hybrid Processing Architecture for Big Data. In: Proceedings of the 7th International Conference on Data Science, Technology and Applications (DATA) (2018).
29. Nadal, S., Herrero, V., Romero, O., Abelló, A., Franch, X., Vansummeren, S., Valerio, D.: A software reference architecture for semantic-aware Big Data systems. Information and Software Technology 90, 75–92 (2017).

30. Stiglich, P.: Data Modeling in the Age of Big Data. Business Intelligence Journal 19 (4), 17–22 (2014).
31. Houle, P.: Data Lakes, Data Ponds, and Data Droplets, http://ontology2.com/the-book/data-lakes-ponds-and-droplets.html, (2017).
32. Cernjeka, K., Jaksic, D., Jovanovic, V.: NoSQL document store translation to data vault based EDW. In: 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO) (2018).
33. Schnider, D., Martino, A., Eschermann, M.: Comparison of Data Modeling Methods for a Core Data Warehouse. Trivadis. (2014).
34. Gröger, C., Schwarz, H., Mitschang, B.: The Deep Data Warehouse: Link-Based Integration and Enrichment of Warehouse Data and Unstructured Content. In: Proceedings of the 2014 IEEE 18th International Enterprise Distributed Object Computing Conference (EDOC) (2014).
35. Herrero, V., Abelló, A., Romero, O.: NOSQL Design for Analytical Workloads: Variability Matters. In: Proceedings of the 35th International Conference on Conceptual Modeling (ER) (2016).
36. Quix, C., Hai, R., Vatov, I.: Metadata Extraction and Management in Data Lakes With GEMMS. Complex Syst. Informatics Model. Q. 9 (9), 67–83 (2016).
37. Halevy, A., Korn, F., Noy, N.F., Olston, C., Polyzotis, N., Roy, S., Whang, S.E.: Managing Google's data lake: an overview of the Goods system. IEEE Data Engineering Bulletin 39 5–14 (2016).
38. Hai, R., Geisler, S., Quix, C.: Constance: An Intelligent Data Lake System. In: Proceedings of the 2016 International Conference on Management of Data (SIGMOD) (2016).
39. Gallinucci, E., Golfarelli, M., Rizzi, S.: Schema profiling of document-oriented databases. Information Systems 75, 13–25 (2018).
40. Walker, C., Alrehamy, H.: Personal Data Lake with Data Gravity Pull. In: Proceedings of the 2015 IEEE Fifth International Conference on Big Data and Cloud Computing (BDCloud) IEEE (2015).
41. Nogueira, I., Romdhane, M., Darmont, J.: Modeling Data Lake Metadata with a Data Vault. In: Proceedings of the 22nd International Database Engineering Applications Symposium (IDEAS) (2018).