An overview about data integration in data lakes

Conference Paper · June 2022

DOI: 10.23919/CISTI54924.2022.9820576

2 authors:

Julia Couto
Pontifícia Universidade Católica do Rio Grande do Sul
18 PUBLICATIONS **71** CITATIONS

SEE PROFILE

Duncan D. Ruiz
Pontifícia Universidade Católica do Rio Grande do Sul
111 PUBLICATIONS **914** CITATIONS

SEE PROFILE

An overview about data integration in data lakes

Júlia Colleoni Couto
School of Technology
PUCRS University
Porto Alegre, Brazil
julia.couto@edu.pucrs.br

Duncan Dubugras Ruiz
School of Technology
PUCRS University
Porto Alegre, Brazil
duncan.ruiz@pucrs.br

Abstract—Integrating data in data lakes is essential so we can perform more complex analyses. However, data lakes are mainly composed of raw data, from structured, semi-structured, and even unstructured data. It turns out that traditional data integration algorithms usually expect to receive structured data as input, so those different types of data jeopardize big data integration. This paper presents a systematic literature review that generates a broad landscape about data integration in data lakes. We searched for papers in eight well-known search engines, following a structured process. From the 298 papers we retrieved, we selected 22 papers that answer our research questions. We identify examples of data lake integration models, how they calculate the similarity among the datasets, how the models are evaluated, the most common type of data they integrate, and the challenges inherent to the area, which points to future research directions in data integration in data lakes.

Index Terms—data integration, data lake

I. Introduction

Data lakes are central repositories for big data, where the data is kept in the original format until someone needs to query it (Couto et al. [1]). Data can be of the most different formats in data lakes, characterizing the variety of big data. In that sense, we can conclude that integrating those heterogeneous data types can be a demanding problem, since data integration aims to combine different data sources to provide the user a unified view (Lenzerini [2]).

Also, performing integrative queries on heterogeneous datasets is time-consuming for the users (Hai, Quix, and Zhou [3]). In a data lake, this issue is magnified mainly because of the big data characteristics, such as variability, volume, and variety (Searls [4]; Alserafi et al. [5]). That said, data integration in data lakes becomes an interesting and promising research field.

Therefore, we aim to contribute to the data lake integration field by searching the scientific literature to identify the different models implemented for integrating data in data lakes and how other researchers measure the similarity among the datasets. Moreover, we reviewed how they evaluated their models, the most common type of data they integrate, and the challenges faced when integrating data in data lakes.

More precisely, we executed a systematic literature review over eight web search engines, based on the process suggested by Kitchenhan et al. [6]. We started the search with 298 papers, and after the selection process, we identified 22 papers that contributed to answering our research questions.

Our study revealed a broad overview of data integration in data lakes. We classified six groups of models that share similar characteristics for data integration in data lakes: Graph-related, Query processing-based, Data profiling-based, Schema matching, Set similarity-based, and Layered architectures. Among the similarity metrics, the most used are semantic similarity. The models are usually evaluated according to the scalability, execution time, and precision. Regarding the type of data used in the experiments for integration, the most common are CSV-like and relational tables. To conclude, we map the challenges and research opportunities, among which the most common challenge references big data variety. The remaining sections detail our study process and results.

II. Materials and Methods

This section explains the method we followed for performing the literature review. A systematic literature review is a research method that allows us to deeply understand the state of-art of a specific knowledge area, how the area developed, and how it changed over time. We follow the process defined by Brereton et al. [6]. These authors suggest three phases, namely Plan, Conduct, and Document the review, having ten stages among these phases.

In the Planning Phase, we *Specify Research Question*, *Develop the Review Protocol*, and *Validate the Review Protocol*. We followed the PICo (Population, Interest, and Context) and PICO (Population, Intervention, Comparison, and Outcome) methods to specify the research questions (Sacket [7], [8]). One researcher developed the review protocol, and the other reviewed and validated the protocol. Regarding the period to retrieve the papers, we did not stipulate a start year so that we could map the results since the beginning of the publications in the area.

In the Conduct the Reviews Phase, we *Identify Relevant Research*, *Select Primary papers* and *Assess Study Quality*, *Extract the Required Data*, and *Synthesize the Data* with inclusion and exclusion criteria for the papers. To identify the relevant research, we applied the protocol on eight different electronic databases: Scopus, ACM, IEEE Xplorer, Springer, Science Direct, Web of Science, Google Academic, and arXiv. We adapted the search string according to the database.

Then, searching over the internet, we identify a primary paper, to check if the paper would return using our search

2022 17th Iberian Conference on Information Systems and Technologies (CISTI)
22 – 25 June 2022, Madrid, Spain
ISBN: 978-989-33-3436-2

string on the selected databases. Having the retrieved papers from each database, we started performing a quick review to evaluate the quality of the retrieved papers. We defined inclusion and exclusion criteria to identify the papers that would answer our research questions. To be accepted, papers must:

1) Be qualitative or quantitative research about the theme of interest;
2) Present a complete study in electronic format; 3) Be a conference paper or journal.

On the other hand, we rejected the papers that meet at least one of the following exclusion criteria:

1) Incomplete or short paper (less than four pages);
2) Unavailable for download;
3) Do not answer the research questions;
4) Duplicated study;
5) Book or book chapter;
6) Literature reviews;
7) Written in another language than English;
8) 8) Conference proceedings index.

When we identified that the papers' quality met our criteria, we started the selection process by extracting and synthesizing the data to answer the research questions.

Finally, in the Document Review Phase we *Write Review Report* based on the extracted data, and we *Validate the Report*, by a peer review with a senior researcher. We performed the systematic literature review we present in this paper based on this method.

III. Research scope

We developed the SLR to deepen our knowledge about data integration in data lakes. In this section, we present the protocol and the results we achieved. Hereafter we answer the following research questions:

- RQ1: What are the models for data integration in data lakes?
- RQ2: Which similarity metrics are used for data integration?
- RQ3: How do they evaluate data integration models for data lakes?
- RQ4: What type of data do they integrate?
- RQ5: What are the challenges in data integration in data lakes?

We used PICO and PICo methods (see Table I) to help us develop the RQs. The paper we use as a control for the search strings is the following: *Zhu et al. [9] "Josie: Overlap set similarity search for finding joinable tables in data lakes." International Conference on Management of Data*.

TABLE I

PICO and PICo definitions for the SLR about data integration in data lakes

PICO	PICo
Population: Data lakes	Population: Data lakes
Intervention: Data integration	Interest: Models, metrics, evaluation, and challenges
Comparison: -	Context: Data integration
Outcome: Models, metrics, evaluation, and challenges	

TABLE II

Search strings for each electronic database for the SLR about data integration in data lakes

Electronic Database	Search String
ACM	"query": AllField:(("data integration" AND "data lake*")) "filter": ACM Content: DL
arXiv	"data integration" AND "data lake*"
Google Scholar	allintitle: "data integration" "data lake*"
IEEE Xplore	("All Metadata":"data integration") AND ("All Metadata":"data lake*")
Science Direct	Title, abstract, keywords: "data integration" AND ("data lakes" OR "data lake")
Scopus	TITLE-ABS-KEY ("data integration" AND "data lake*")
Springer	https://link.springer.com/search?dc.title=\%22data+integration\%22+AND+\%22data+lake*\%22& date-facet-mode=between&showAll=true\#
Web of Science	https://www.webofscience.com/wos/woscc/ summary/703bae45-ee46-4661-ac06-b5aa14160e54-1b6bd85d/ relevance/1 OR "data integration" AND "data lake*" (All Fields)

We performed the search for the terms "data integration" AND "data lake" through eight different electronic databases. Table II lists the search strings we used for each database.

IV. Results

This section presents our analysis for the papers, and we answer our five RQ. We retrieved 298 papers, and after reading the title, abstract, and keywords, we got 82 papers to analyze fully. Finally, we accepted 22 papers following the SLR process. Table III shows the number of papers we retrieved and the accepted ones per database. This Table shows that most papers came from ACM and Web of Science. The accepted papers were published between the years 2018 to 2021, mostly from 2018 and on (Figure 1). We accepted ten journal papers and 12 conference papers.

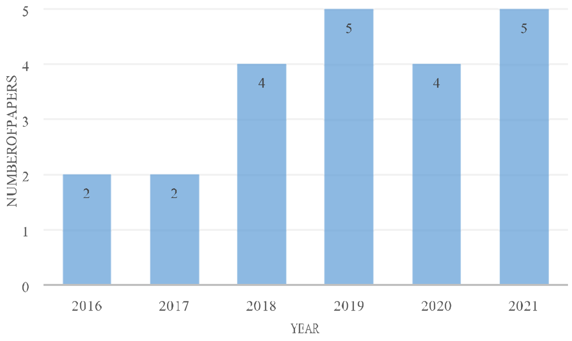

Fig. 1. Papers per year - SLR about data integration in data lakes

V. RQ1: Models for data integration in data lakes

We decided to group the models into six groups: Graph-related, Query processing-based, Data profiling-based, Schema matching, Set similarity-based, and Layered architectures. We group the papers based on the title, abstract, keywords, and important terms in the document. Next, we present the groups and related models.

A. Graph-related

Koutras [21] developed Data as a Language (Daal), which first transforms data into a graph then create documents from the graph. From the documents they construct embeddings, to find semantic relationships (for instance, how two columns from different tables could be joined). Haller and Lenz [27] infer the schema of the data based on the SQL queries performed over data, using, for instance, the joins as points for data integration, to create a knowledge graph. Jovanovic et al. [22] developed an integration manager that generates source-specific metadata when a new data source is registered in their system, which contains a Global Schema Building, for semi-automatic schema alignment and for the data merging process. They rank the candidates to match with terms in a global graph, based on a confidence level representing the degree of similarity between the concepts. The suggestion can be rejected or accepted by the user. The accepted matches are then defined as ”sameAs” edges, compared to the similar concepts in the graph. Alrehamy and Walker [24] developed an ontology-based data integration system named SemLinker. The system is responsible for extracting and maintaining the metadata, handling schema evolution, and finding mappings between the metadata and an ontology.

B. Query processing-based

Hai and Quix [10] developed an approach that translates a subset of second-order tuple-generating dependencies (SO TGDS) into logically equivalent nested TGDS. The method allows data integration through mapping dependencies. Endris et al. [26] developed Ontario, an engine for federated query processing over heterogeneous data in data lakes. Hai, Quix, and Zhou [3] developed Scalable Query Rewriting Engine (SQRE), based on Apache Spark to translate queries for a logical representation, parse the queries according to the source to be queried, and execute queries in different data stores in a data lake, to present the integrated results. Alili et al. [19] developed a model to enrich datasets by searching for related information in external data sources - a service lake. They use it, for instance, to add missing information to the dataset.

TABLE III

Papers per electronic databases for the SLR about data integration in data lakes

Electronic Database	Initial	Accepted papers
ACM	68	5 papers: Hai and Quix [10], Helal et al. [11], Zhang and Ives [12], Brackenbury et al. [13], Zhu et al. [9]
arXiv	2	0 paper
Google Scholar	2	1 paper: Dabbechi et al. [14]`
IEEE Xplore	11	3 papers: Alserafi et al. [5], Dong et al. [15], Yang et al. [16]
Science Direct	132	2 papers: Dhayne et al. [17], Quinn et al. [18]
Scopus	50	4 papers: Alili et al. [19], Reziget al. [20], Hai, Quix, and Zhou [3], Koutras [21]
Springer	16	2 papers: Jovanovic et al. [22], Beyan et al. [23]
Web of Science	17	5 papers: Alrehamy and Walker [24], Pomp et al. [25], Endris et al. [26], Haller and Lenz [27], Kathiravelu and Sharma [28]

Zhang and Ives [12] developed an architecture based on Jupyter notebooks using the model they developed (JUNEAU) as a backend and to extend the user interface. The backend integrates key-value stores and relational databases to capture and index data of interest. For data integration, the user can select a table, and the system returns a ranked list of conceptually related tables that could be joined. Dhayne et al. [17] developed EMR2vec, a platform to link clinical trial data and patient data, to help find the most suitable patients for clinical trials based on the eligibility criteria by querying an integrated data lake. Quinn et al. [18] their solution is an integration technique to map the time-series data from a building Internet of Things (IoT) sensor network to Facility Management-enabled Building Information Models (FMBIMs), using Apache Cassandra as storage where the data can be queried.

C. Data profiling-based

Alserafi et al. [5] developed a prototype for a metadata management system called Content Metadata for Data Lakes (CM4DL), which detects joinable data attributes between datasets. They also use data profiling techniques to describe each attribute and its data type. The input for their algorithm is the files, a JSON containing metadata features created by the data profiling, and the threshold for matching datasets and attributes, and the output is the discovered relationships. Helal et al. [11] developed a model named KGLac, that bases on a data profiling system on top of Apache Spark. KGLac uses embedding similarity search to reveal columns or tables with similar representation, enabling joining tables. They use data profiling to detect relations based on data content, such as inclusion dependency and primary and foreign key discovery.

D. Schema matching

Brackenbury et al. [13] developed a similarity-based approach to find related datasets in data swamps based on schema matching and discovering techniques. Dabbechi et al.` [14] use ELT jobs in ”Talend open studio for big data” for schema mapping and integration of the different data sources: Facebook, Youtube, and Twitter. Data is stored in MongoDB and Cassandra NoSQL databases. Reziget al. [20] developed DICE (Data Discovery by Example), where the user provides examples of records in a data lake. Then the system suggests Primary Key/Foreign Key join paths that can relate to other tables, based on exact or similarity matching. The candidates for PK/FK joins are then validated by the user. Dong et al. [15] developed the PEXESO framework for joinable table discovery in data lakes. PEXESO uses pre-trained models to help transform textual attributes in high-dimensional vectors, so they join the tables using semantics.

E. Set similarity-based

Kathiravelu and Sharma [28] propose Data Cafe, a data´ warehouse platform to create, integrate, and manage biomedical data lakes. They store the data in HDFS, MongoDB, and MySQL, the data schema is stored in Apache Hive, they use Apache Drill as SQL search engine, and they use Hazelcast as an in-memory data grid. For data integration, they identify join-attributes, which are the indexes that could lead from dataset A to dataset B, and then they create a graph-based on the intersection of the datasets. Yang et al. [16] developed a model for the top-k set similarity joins (SSJOIN). They focus on the step size, which is the number of elements that are accessed in each algorithm's iteration. They developed a fixed size (l-ssjoin) and an adaptative size step algorithm (A-ssjoin). Zhu et al. [9] developed JOSIE (JOining Search using Intersection Estimation), an overlap set similarity search algorithm that uses a search model to adapt according to the data distribution. They receive a table column as input, and they return the tables in the data lake that could be joined with the given columns, based on the largest number of distinct values.

F. Layered architectures

Beyan et al. [23] proposed a data value chain based on five layers: Data Acquisition; Data Interpretation and Multilingual Interoperability; Data Analysis and Curation; Data Storage; and Data Usage. Pomp et al. [25] developed ESKAPE: a data ingestion, integration, and processing model formed by three layers: a Hadoop-based data lake, which contains the datasets, the semantic models, which are created during data ingestion for each dataset, and the knowledge graph, which combines all the semantic models into a unified repository similar to an ontology.

VI. RQ2: Similarity metrics for data integration

From the papers we selected, twelve papers do not mention the similarity metric they use to evaluate the similarity for data integration. Table IV presents the similarity metrics most used in the papers.

Semantic similarity is the top-cited. Semantic similarity functions assign a score for the relationship between pieces of text by using a predefined metric (Alili et al. [19], Pomp et al. [25], Helal et al. [11], and Dhayne et al. [17].

The *Jaccard coefficient* is used in 3 papers. Jaccard is useful for comparing finite sets, represented by the quotient of the cardinalities of the intersection and the union of all tokens or characters in two strings [29].

TABLE IV
Most used similarity metrics

Similarity metric	N° of papers	Papers
Semantic similarity	4	Alili et al. [19], Pomp et al. [25], Helal et al. [11], Dhayne et al. [17]
Jaccard coefficient	3	Brackenbury et al. [13], Zhu et al. [9], Yang et al. [16]
MinHash-based distances	3	Zhu et al. [9], Brackenbury et al. [13], Alserafi et al. [5]
Overlap set similarity	2	Zhu et al. [9], Yang et al. [16]
Cosine distance	2	Yang et al. [16], Alrehamy and Walker [24]
String-based measures	2	Alrehamy and Walker [24], Zhang and Ives [12]
Euclidean distance	1	Dong et al. [15]
Jensen–Shannon divergence	1	Dong et al. [15]
Identity-based exact match	1	Alserafi et al. [5]
Projection similarity	1	Dhayne et al. [17]

MinHash-based approaches were used by Zhu et al. [9], Brackenbury et al. [13], and Alserafi et al. [5]. The latter states that it is an approach that makes comparisons of text n-grams, being a good approach for approximate string matching.

In its turn, the *Overlap set similarity*, present in two papers, represents the size of the intersection between two sets (Zhu et al. [9]). The Cosine distance is also used in two papers and measures the similarity between two vectors by evaluating the cosine value of the angle between them (Alrehamy and Walker [24]).

String-based measures, such as edit distance, are used by Alrehamy and Walker [24], and Zhang and Ives [12]. The edit distance measures the dissimilarity between two strings by counting the minimum number of operations that we need to perform to transform one string into the other.

Dong et al. [15] used the *Euclidean distance*, which is the distance between two points, often used to check the similarity measure between time-series, and the *Jensen–Shannon divergence*, which measures the similarity between two probability distributions.

Alserafi et al. [5] used *identity-based exact match*, where the attributes are normalized and then compared to find exact matches. It is a good approach for exact values comparison, such as numeric values.

Finally, Dhayne et al. [17] use the *projection similarity*, that computes the level of similarity of a dataset in the dimensions of the features of the other dataset.

VII. RQ3: Evaluation of Data Integration Models for Data Lakes

From the 22 papers, eight do not present an evaluation of their approaches. The papers that evaluate their studies perform the evaluations based on the following metrics:

- Scalability - 7 papers: [3], [13], [15], [16], [18], [24], [28]. It represents the ability to deal with a crescent amount of data.
- Execution time - 6 papers: [3], [5], [9], [16], [24], [26]. Those papers present the average time to run their experiments.
- Precision - 6 papers: [5], [11], [15], [17], [22], [24]. Precision measures the correct answers of the model over the total number of observations.
- Recall - 4 papers: [5], [11], [15], [22]. It measures the true positives over the sum of the true positives plus the false negatives.
- F1 - 2 papers: [5], [11]. F1 score is the harmonic mean between precision and recall.
- Accuracy - 2 papers: [13], [24]. It measures the proportion of correctly predicted observations regarding the total number of observations.

Other types of evaluation were also cited. For instance, Haller and Lenz [27] evaluated their model based on the ability to reconstruct the data schema based on the SQL queries. Jovanovic et al. [22] measured the usability and the number of times the user had to intervene to find the matches manually. Hai and Quix [10] evaluated the correctness, completeness, and performance of their model, while Endris et al. [26] evaluated the cardinality (number of answers a query returns), completeness (query results percentage compared to another engine), and *dief@t* (measures the continuous engine's efficiency). Yang et al. [16] measured the number of candidates for joining according to the threshold. Zhu et al. [9] measured the number of top results to retrieve based on other solutions; and Hai, Quix, and Zhou [3] evaluated their model's functionality.

VIII. RQ4: Type of Data They Integrate

The types of data most used in the experiments to validate the models are CSV-like and relational tables. *CSV, TSV, or other tabular formats* are presented in 8 papers [3], [10], [12], [12], [18], [21], [22], [24], [26]. Seven papers also present the use of *relational tables* [12], [15], [17], [19], [20], [27], [28] The *JSON format* is used by 4 papers [22], [24]–[26].

Three papers present the use of *social media data* [16], [23], [24]. Another two papers discuss the use of entire *data lakes*, such as OpenData, OpenML datasets, and WebTables [5], [9]. Three papers base their experiments on *XML files* [3], [17], [26], and two others use HTML files [13], [15].

Other types of data format include: *collections of documents* Dabbechi et al. [14],` *NoSQL databases* such as MongoDB (Hai, Quix, and Zhou [3]), a *file system dump* Brackenbury et al. [13], RDF Endris et al. [26] and other *domain-specific datasets* [11], [16], [23].

IX. RQ5: Challenges in Data Integration in Data Lakes

From the selected papers, nine papers present some challenges related to data integration in data lakes. Figure 2 presents a word cloud for the challenges.

- Complexity (Koutras [21]): They discuss the challenge of transforming data, including challenging in construction, incorporating information about the schema, and capturing entries from semi-structured datasets, which can contain comments or messages generated by a system.

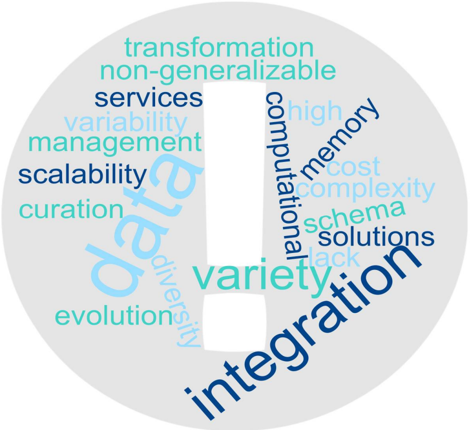

Fig. 2. Word Cloud for the challenges of Data integration in Data lakes

- Computational cost (Dong et al. [15]): To compute the similarity for high-dimensional data is expensive. Checking whether the tables are joinable or not is also time-consuming because of the large number of tables in a data lake.
- Diversity (Hai, Quix, and Zhou [3]): There is a high diversity in the data management in data lakes, with many different solutions and frameworks, but they are not always easily integrated among them.
- In-memory integration (Hai, Quix, and Zhou [3]): answering a single query based on data from several sources without creating a new structure to join the data.
- Lack of available solutions (Beyan et al. [23]): the author states that there is a lack of data integration and curation services for big data.
- Non-generalizable solutions (Kathiravelu and Sharma [28]): The solutions are more specific for a certain data source our format than generalizable, and the solutions usually expect that the users know the data schema or storage format, and it is a problem, for instance, for medical data since they usually are consisted by a huge number of small datasets.
- Scalability (Alrehamy and Walker [24]): the ability of a system to be prepared to efficiently handle more data.
- Variability (Alserafi et al. [5], Alrehamy and Walker [24]): it represents the changes that occur in data schema and structure; the schema evolution in big data.
- Variety (Dhayne et al. [17], Alserafi et al. [5], Dabbechi` et al. [14], Alrehamy and Walker [24]): the data are difficult to analyze since it is mostly unstructured or semi-structured. Because of the variety, we find syntactic and semantic complexity of the different datasets, for example, differences in the semantic concepts, which can be more generic or more specific.

X. Conclusions

This paper presented a systematic literature review to retrieve the state-of-art related to data integration on data lakes. Our initial set of papers is composed of 298 papers, and, after selection, we ended up having 22 papers accepted, published between 2018 and 2021. We identified six groups of related papers according to the models: Graph-related, Query processing-based, Data profiling-based, Schema matching, Set similarity-based, and Layered architectures. We also identified the most used similarity metrics for data integration, such as semantic similarity, Jaccard, MinHash-based, Overlap, Cosine, and String-based measures. Additionally, we investigated how they evaluate their models, and most of them perform experiments to check the scalability, execution time, and precision. Among the types of data they integrate, we found that CSV-like and relational tables are the most popular. Finally, we mapped nine challenges related to data integration in data lakes: complexity, computational cost, diversity, in-memory integration, lack of available solutions, non-generalizable solutions, scalability, variability, and the most cited: variety. We expect our results can be useful for both industry and academia by providing beginners with relevant aspects concerning big data integration in data lakes and providing directions for future research, correlated to the challenges we identified. As for directions for future research, we are working on a model for automated data integration in a Hadoop-based data lake, which could deal with the challenges we identified in the current paper.

Acknowledgment

This study was jointly financed by the Coordenação de Aperfeiçoamento de Pessoal de Nivel Superior – Brasil (CAPES) – Finance Code 001 and Dell Brazil (Law no 8.2.48, year 1991).

References

[1] J. Couto, O. Borges, D. Ruiz, S. Marczak, and R. Prikladnicki, “A mapping study about data lakes: An improved definition and possible architectures,” in *International Conference on Software Engineering and Knowledge Engineering*. Lisbon, PT: KSI Research Inc., 2019, pp. 453–458.

[2] M. Lenzerini, “Data integration: A theoretical perspective,” in *Symposium on Principles of Database Systems*. New York, US: ACM, 2002, p. 233–246.

[3] R. Hai, C. Quix, and C. Zhou, “Query rewriting for heterogeneous data lakes,” in *European Conference on Advances in Databases and Information Systems*. Budapest, HU: Springer, 2018, pp. 35–49.

[4] D. B. Searls, “Data integration: challenges for drug discovery,” *Nature reviews Drug discovery*, vol. 4, no. 1, pp. 45–58, 2005.

[5] A. Alserafi, A. Abello, O. Romero, and T. Calders, “Towards information´ profiling: Data lake content metadata management,” in *International Conference on Data Mining Workshops*. Barcelona, ES: IEEE, 2016, pp. 178–185.

[6] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, “Lessons from applying the systematic literature review process within the software engineering domain,” *Journal of Systems and Software*, vol. 80, pp. 571–583, Apr, 2007.

[7] D. L. Sackett, *Evidence-based medicine: How to practice and teach EBM*. London, UK: Churchill Livingstone, 2000.

[8] U. Murdoch, “Systematic reviews: Using PICO or PICo,” Retrieved from https://libguides.murdoch.edu.au/systematic/PICO, 2018, november, 2019.

[9] E. Zhu, D. Deng, F. Nargesian, and R. J. Miller, “Josie: overlap set similarity search for finding joinable tables in data lakes,” in *International Conference on Management of Data*. Amsterdam, NL: ACM, 2019, pp. 847–864.

[10] R. Hai and C. Quix, “Rewriting of plain so tgds into nested tgds,” *Proceedings of the VLDB Endowment*, vol. 12, p. 1526–1538, Jul, 2019.

[11] A. Helal, M. Helali, K. Ammar, and E. Mansour, “A demonstration of kglac: A data discovery and enrichment platform for data science,” *Proceedings of the VLDB Endowment*, vol. 14, p. 2675–2678, Jul, 2021.

[12] Y. Zhang and Z. G. Ives, “Juneau: Data lake management for jupyter,” *Proceedings of the VLDB Endowment*, vol. 12, p. 1902–1905, Aug, 2019.

[13] W. Brackenbury, R. Liu, M. Mondal, A. J. Elmore, B. Ur, K. Chard, and M. J. Franklin, “Draining the data swamp: A similarity-based approach,” in *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*. Houston, TX, USA: ACM, 2018.

[14] H. Dabbechi, N. Z. Haddar, H. Elghazel, and K. Haddar, “Social media` data integration: From data lake to nosql data warehouse,” in *International Conference on Intelligent Systems Design and Applications*. Online: Springer, 2020, pp. 701–710.

[15] Y. Dong, K. Takeoka, C. Xiao, and M. Oyamada, “Efficient joinable table discovery in data lakes: A high-dimensional similarity-based approach,” in *International Conference on Data Engineering*. Chania, GR: IEEE, 2021, pp. 456–467.

[16] Z. Yang, B. Zheng, G. Li, X. Zhao, X. Zhou, and C. S. Jensen, “Adaptive top-k overlap set similarity joins,” in *International Conference on Data Engineering*. Dallas, USA: IEEE, 2020, pp. 1081–1092.

[17] H. Dhayne, R. Kilany, R. Haque, and Y. Taher, “Emr2vec: Bridging the gap between patient data and clinical trial,” *Computers & Industrial Engineering*, vol. 156, p. 107236, Jun, 2021.

[18] C. Quinn, A. Z. Shabestari, T. Misic, S. Gilani, M. Litoiu, and J. McArthur, “Building automation system - bim integration using a linked data structure,” *Automation in Construction*, vol. 118, p. 16, Oct, 2020.

[19] H. Alili, K. Belhajjame, D. Grigori, R. Drira, and H. Ben Ghezala, “On enriching user-centered data integration schemas in service lakes,” *Lecture Notes in Business Information Processing*, vol. 288, pp. 3–15, Jun, 2017.

[20] E. Rezig, A. Vanterpool, V. Gadepally, B. Price, M. Cafarella, and M. Stonebraker, “Towards data discovery by example,” *Lecture Notes in Computer Science*, vol. 12633, pp. 66–71, Sep, 2021.

[21] C. Koutras, “Data as a language: A novel approach to data integration,” in *International Conference on Very Large Database - PhD Workshop*. Los Angeles, USA: Springer, 2019, pp. 1–4.

[22] P. Jovanovic, S. Nadal, O. Romero, A. Abello, and B. Bilalli, “Quarry:´ a user-centered big data integration platform,” *Information Systems Frontiers*, vol. 23, pp. 9–33, Dec, 2021.

[23] O. D. Beyan, S. Handschuh, A. Koumpis, G. Fragidis, and S. Decker, “A Framework for Applying Data Integration and Curation Pipelines to Support Integration of Migrants and Refugees in Europe,” in *Working Conference on Virtual Enterprises*. Porto, PT: HAL, Oct. 2016, pp. 588–596.

[24] H. Alrehamy and C. Walker, “Semlinker: automating big data integration for casual users,” *JOURNAL OF BIG DATA*, vol. 5, p. 26, Mar, 2018.

[25] A. Pomp, A. Paulus, A. Kirmse, V. Kraus, and T. Meisen, “Applying semantics to reduce the time to analytics within complex heterogeneous infrastructures,” *Technologies*, vol. 6, p. 29, Sep, 2018.

[26] K. M. Endris, P. D. Rohde, M.-E. Vidal, and S. Auer, “Ontario: Federated query processing against a semantic data lake,” in *International Conference on Database and Expert Systems Applications*, vol. 11706. Bratislava, SK: Springer, 2019, pp. 379–395.

[27] D. Haller and R. Lenz, “Pharos: Query-driven schema inference for the semantic web,” in *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*. Wurzburg, GE: Springer, 2020, pp. 112–124.

[28] P. Kathiravelu and A. Sharma, “A dynamic data warehousing platform for creating and accessing biomedical data lakes,” in *Very Large Data Bases Workshop on Data Management and Analytics for Medicine and Healthcare*. Munich, DE: Springer, 2017, pp. 101–120.

[29] P. Jaccard, “Distribution comparee de la flore alpine dans quelques´ regions des alpes occidentales et orientales,”´ *Bulletin de la Murithienne*, vol. XXXVII, pp. 81–92, Jan, 1902.