A systematic overview of data federation systems

Zhenzhen Gu [a], Francesco Corcoglioniti [a], Davide Lanti [a], Alessandro Mosca [a], Guohui Xiao [a,*], Jing Xiong [a] and Diego Calvanese [a,b]

[a] *KRDB Research Centre, Faculty of Computer Science, Free University of Bozen-Bolzano, Italy*
E-mail: <name>.<surname>@unibz.it
[b] *Department of Computing Science, Umeå University, Sweden*
E-mail: diego.calvanese@umu.se

Abstract. Data federation addresses the problem of uniformly accessing multiple, possibly heterogeneous data sources, by mapping them into a unified schema, such as an RDF(S)/OWL ontology or a relational schema, and by supporting the execution of queries, like SPARQL or SQL queries, over that unified schema. Data explosion in volume and variety has made data federation increasingly popular in many application domains. Hence, many data federation systems have been developed in industry and academia, and it has become challenging for users to select suitable systems to achieve their objectives. In order to systematically analyze and compare these systems, we propose an evaluation framework comprising four dimensions: *(i) federation capabilities*, *i.e.*, query language, data source, and federation technique; *(ii) data security*, *i.e.*, authentication, authorization, auditing, encryption, and data masking; *(iii) interface*, *i.e.*, graphical interface, command line interface, and application programming interface; and *(iv) development*, *i.e.*, main development language, commercial support, open source, and release. Using this framework, we thoroughly studied 48 data federation systems from the Semantic Web and Database communities. This paper shares the results of our investigation and aims to provide reference material and insights for users, developers and researchers selecting or further developing data federation systems.

Keywords: Data federation systems, Federated query answering, Data virtualization, Heterogeneous data integration, System evaluation framework

1. Introduction

The convenience of digitization, the variety of data descriptions, and the discrepancy in personal preferences have led large enterprises to store massive amounts of data in a variety of formats, ranging from structured relational databases to unstructured flat files. According to the prediction in [1], the global data volume will reach 163 zettabytes by 2025, and half of that data will be produced by enterprises.

Since data becomes more valuable if enriched and fused with other data, decision-makers need to consider data distributed in different places and with different formats in order to get valuable insights that support them in their daily activities. However, data explosion in volume, variety, and velocity — *i.e.*, the "3Vs" of Big Data [2, 3] — increases complexity and makes the traditional ways of data integration [4–6], such as data warehousing [7, 8], not only more costly in terms of time and money but also unable to guarantee the freshness of data. Integration solutions developed in a more agile way are thus demanded especially in the Big Data context. *Data federation* is a technology

*Corresponding author. E-mail: guohui.xiao@unibz.it.

that makes this possible today, that is becoming more and more appealing in both industry and academia, and that has been studied for a long time in different communities such as the Database and (more recently) the Semantic Web ones.

Data federation systems (also known as *federated database systems*) are traditionally defined as a type of meta-database management system that transparently maps multiple *autonomous database systems* into a single federated database [9, 10]. The key task of data federation systems is *federated query answering*, that is to provide users with the ability of querying multiple data sources under a uniform interface. Such interface usually consists in a *query language* over a *unified schema*, such as SQL [11] over a relational schema or SPARQL [12] over an RDF(S) / OWL [13–15] ontology, this interface being often closely related or restricting the query languages and schemas of supported data sources. Unlike in traditional pipelines for data extraction, transformation, and loading (ETL) often used in data warehouse systems, federated query answering is achieved by *data virtualization* [16, 17], *i.e.*, all the data are kept *in situ* and accessed via a common semantic layer on the fly, with no data copy, movement, or transformation. As a result, federated query answering via data virtualization reduces the risk of data errors caused by data migration and translation, decreases the costs (*e.g.*, time) of data preparation, and guarantees the freshness of data. Besides federated query answering, modern data federation systems also offer a wide range of other important capabilities for data management, such as *read-and-write data access* for enabling users to both access and modify the data in the sources, *data security* for protecting the sensitive data of users and implementing secure data access, and *data governance* for managing the availability, usability, and integrity of the data.

Data federation is an active field and many data federation systems have been and are being developed. For example, FedX [18, 19] and Teiid [20] are two systems supporting respectively SPARQL query answering over multiple SPARQL endpoints (*i.e.*, standardized HTTP services [21] that can process SPARQL queries) and SQL query answering over multiple heterogeneous data sources, like relational databases, structured files and web services, through a unified schema called *virtual database*. More generally, current data federation systems include both *industrial systems*, mostly developed by software companies and more mature, and *academic systems*, mostly developed by research organizations and providing newer functionalities. Moreover, federated query answering facilities are often included in modern data management systems aimed at heterogeneous big data. These systems include *logical data warehouses* [22–24], *data lakes* [25–28], and *polystores* [29–33], and can be seen to all intents and purposes as special cases of data federation systems. All the aforementioned systems present substantial overlap in terms of adopted techniques and extra capabilities offered to users, while differences in the exposed unified interface may be often bridged — *e.g.*, by using Ontology-Based Data Access (OBDA) [34] to adapt SQL over a federated relational schema to SPARQL over an OWL ontology — this way enabling the use of a data federation system in additional scenarios with respect to the ones it was primarily developed for — *e.g.*, use a robust industrial SQL-based data federation system to create a "virtual" knowledge graph for Linked Open Data publishing. Therefore nowadays, users have access to a large number of data federation systems to choose among, but selecting the right system for a specific task requires collecting, analyzing, and comparing the capabilities and techniques of many systems, which is very time-consuming: for industrial systems, the information needed is usually fragmented and scattered, and the official documents often consist of hundreds of pages; for academic systems, conversely, end-user documentation is typically poor or unavailable, and system features are described in academic publications, when available.

This survey tries to shed some light on this complex matter by analyzing 48 state-of-the-art data federation systems, jointly covering systems from the Semantic Web and the Database communities thanks to their substantial interchangeability and their commonalities in implemented techniques and features. The considered systems comprise 31 industrial systems under active development and with public official documentation, and 17 academic systems. Grounded on our experience in selecting suitable data federation systems for heterogeneous data integration [35–41], our work has a twofold goal: help end users in identifying the systems best suited to their applications and tasks, and allow researchers and developers to gain more insights into the capabilities, techniques, strengths, and weaknesses of current systems, this way informing further work in the field.

In order to compare the considered systems from the perspective of data federation in a uniform way, this survey proposes a *qualitative evaluation framework* consisting of four dimensions further refined into several sub-dimensions, which we defined by considering and classifying the aspects that play crucial roles in the users' choice of a system for employment in their applications and tasks:

- The *federation capabilities* dimension concerns the federated query answering features offered by a system over multiple data sources, both homogeneous and heterogeneous in type. It is further refined into three closely related sub-dimensions: *data source*, *query language*, and *federation technique*.
- The *data security* dimension concerns the capabilities of a system of safeguarding the data in the sources participating in the federation from unwanted actions by unauthorized users, especially when such data is sensitive or private. It is refined into five sub-dimensions: *authentication*, *authorization*, *auditing*, *encryption*, and *data masking*.
- The *interface dimension* concerns the usability of the systems. It is further divided into the three sub-dimensions of *graphical interface*, *command line interface*, and *application programming interface*, so as to measure the ability of supporting users in fully appreciating, accessing, and exploiting the features implemented by a system.
- The *development dimension*, finally, concerns the development, release and support practices adopted by system vendors. Its four sub-dimensions of *main development language*, *commercial support*, *open source*, and *release*, aim overall at assessing the maturity of the systems and the possibilities for users to get help from vendors, and to maintain and improve the systems by themselves, if needed.

For all the 48 considered data federation systems, we follow the proposed four dimensions by consulting the official documentation of each system, as well as its related publications. Note that since not all the features of these systems are properly documented, our analysis is conducted using our best efforts.

This survey adds to an existing body of literature [42–48] that reviews the approaches and systems for federated query answering under multiple perspectives. For example, the work in [43] evaluates seven SPARQL federation query engines by focusing on their query evaluation techniques, while the work in [47] studies the modern data federation systems (including BigDAWG [30], CloudMdsQL [32], Myria [31], and Apache Drill [49]) by focusing on their features, owners, goals, and main components. Compared with all these works and summing up, we make the following contributions:

- We carried out an extensive review of academic literature and documentation about industrial solutions to identify a large number of data federation systems from the Semantic Web and the Database communities.
- We provide a framework for investigating data federation systems in a uniform and qualitative way by taking into account aspects of interest for data federation end users, developers and researchers.
- We analyze the identified systems through the proposed framework, this work amounting to an extensive analysis covering 48 systems and 4 main evaluation dimensions overall divided into 15 sub-dimensions. To the best of our knowledge, this is the most extensive analysis on data federation so far in terms of investigated systems and considered aspects.
- As a by-product of our analysis, we make explicit the common capabilities of current data federation systems, such as the capability of handling heterogeneous data sources, or the query optimization techniques used.
- We discuss remaining open problems and challenges and point out the research directions that are interesting and valuable for pursuit.

The remainder of the survey is organized as follows. Section 2 presents an outline of data federation. Section 3 illustrates the overall methodology of the survey work. Section 4 describes the proposed framework for systems assessment and comparison. Section 5 lists and provides a summary of the selected systems. Section 6 thoroughly analyzes the capabilities of these systems according to the proposed framework. Section 7 discusses related work. Finally, Section 8 concludes by discussing open problems and challenges as well as giving directions for further work.

2. Outline of data federation

This section provides an overview of the main concepts underlying data federation that are addressed in this paper, for readers not already familiar with them.

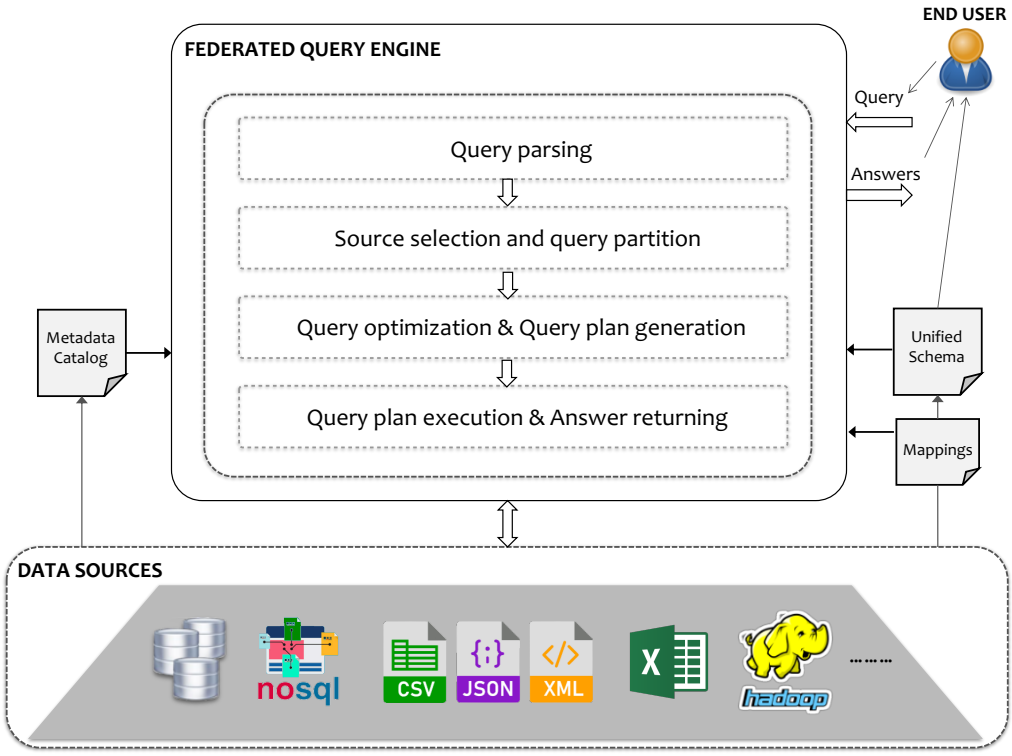

Fig. 1. Typical architecture of a federated query engine.

The core task of data federation is federated query answering [42–46]. For a set of autonomous and possibly heterogeneous data sources, the goal of federated query answering is to provide a uniform interface, typically as a unified query language over a unified schema, to access the data of these sources *in situ*, *i.e.*, without first copying the data to a centralize storage. Given a user query over the unified schema, this task is carried out by issuing and orchestrating the evaluation of native *sub-queries* targeting the data sources of the federation.

Figure 1 depicts the typical architecture of a *federated query engine* providing federated query answering. *Unified schema*, *mappings*, *metadata catalog* are key components, which respectively provides a unified schema of the data sources participating in the federation, map the data in the sources to the unified schema, and provides the statistic information of the data sources as well as the information of how these data sources can be accessed. For example, for a relational database, if the unified schema is an RDF ontology, then there exist mappings that map the tables of this database to the classes and properties of the ontology, and the metadata catalog could list the relevant content statistics, such as the number of rows of the referred tables, used in federated query optimization. Formally, a data federation instance usually consists of three components $(\mathcal{S}, \mathcal{V}, \mathcal{M})$, where $\mathcal{S}$ is a set of data sources $S_1, \ldots, S_n$ which can be relational databases, NoSQL databases, structured files, data warehouse, and so on; $\mathcal{V}$ is the *unified schema* for these n sources, such as an RDF(S) ontology or relational schema; and $\mathcal{M}$ is a set of mappings that map the data of the sources participated in the federation into the elements of the unified schema $\mathcal{V}$. Then accessing multiple data sources staying in situ simultaneously is carried out by evaluating queries Q expressed in terms of the unified schema $\mathcal{V}$ (such as SPARQL queries when $\mathcal{V}$ is an RDF ontology, and SQL queries when $\mathcal{V}$ is a relational schema) via the following steps:

1. The *query parsing* step deals with the syntactic issues of Q, i.e., checking whether the input queries are syntactically correct w.r.t. the query languages as well as the unified schema. Some engines also transform Q into an algebraic form, such as a tree structure using internal nodes to denote operations (*e.g.*, join, union, or projection) and leaf nodes to denote accessed relations.

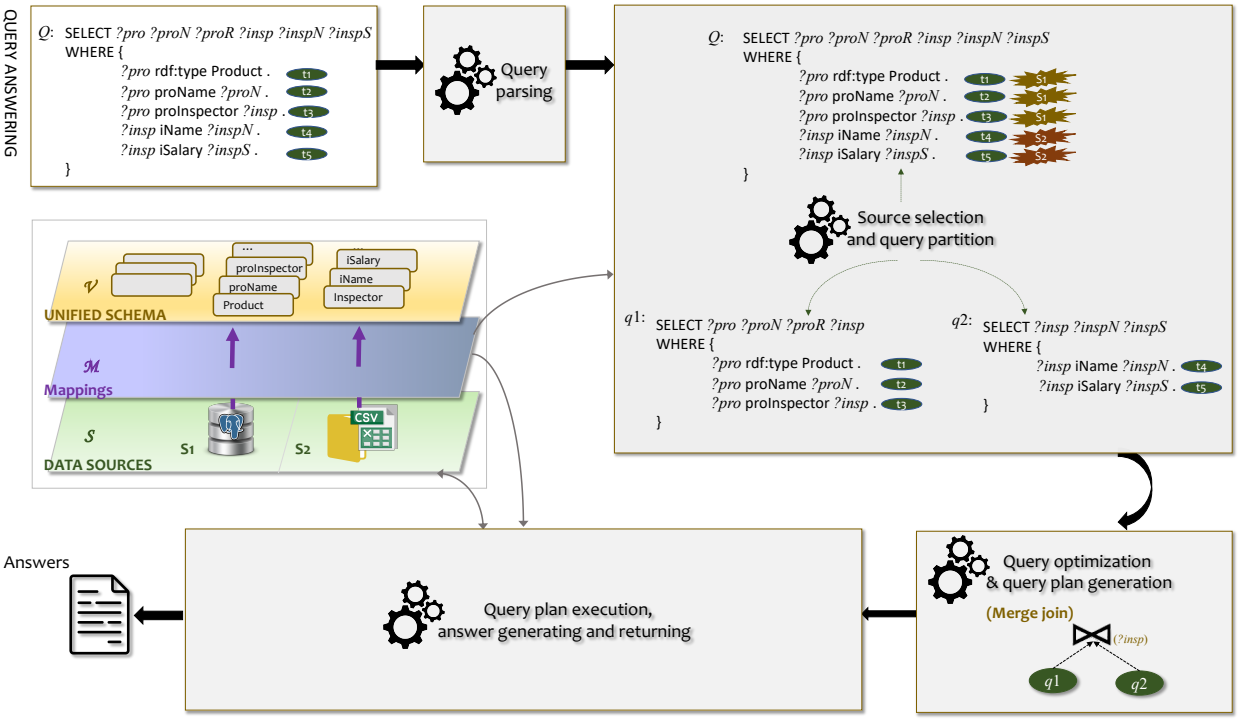

Fig. 2. An example of federated query answering.

2. The *source selection and query partition* step selects suitable data sources for each algebraic component of Q, and partitions Q into smaller sub-queries $q_1, \cdots, q_m$ (*i.e.*, query chunks) accordingly, based on the mappings from the data sources to the unified schema $\mathcal{V}$. Most of the approaches for source selection are index-based, such as the "triple pattern-wise source selection" for SPARQL queries [50, 51]. The dominant way for query partitioning is to try to "push down" the evaluation of the operators to the sources, rather than perform such evaluation at the level of the federation engine.
3. The *query optimization & query plan generation* step computes an execution plan of the partitioned sub-queries $q_1, \cdots, q_m$, establishing in which order to evaluate the sub-queries, and which algorithms to use for joining their answers (*e.g.*, bind join, hash join, etc), based on the metadata catalog. Existing approaches are mostly rule-based (*i.e.*, via predefined and deterministic heuristic rules) or cost-based (*i.e.*, choose the lowest-cost execution plan according to some heuristic cost function).
4. The *query plan execution & answer returning* step, finally, evaluates the decomposed sub-queries $q_1, \cdots, q_m$ over the corresponding data sources via the mappings and the metadata catalog as well as generates the answers of the original query Q. Note that, if the query language of the data source supporting is different from the query language of the federation engine, a translation based on the mappings is needed to translate the sub-query into the one supported by the data source.

Next, we use an example to further enhance readers' understanding of federated query answering.

Example 1. *Suppose we have a data federation instance* $(\{S_1, S_2\}, \mathcal{V}, \mathcal{M})$ *modeling information about a large enterprise, as the one in Fig. 2. Here* S_1 *and* S_2 *are two data sources storing information about two different departments. Concretely,* S_1 *is a relational database from the Sales department storing the information about products being sold, whereas* S_2 *is a set of CSV files from the Human Resources department storing information about each employee of the enterprise. The unified schema* $\mathcal{V}$ *of the federation instance is an RDF ontology including the classes* Product, Reviewer, *and* Inspector, *as well as the properties* proName, proInspector, iName, *and* iSalary. *The set* $\mathcal{M}$ *contains mappings from the data in* S_1 *to the terminology* Product, proName, *and* proInspector *of* $\mathcal{V}$, *as well as the mappings from the data in* S_2 *to the terminology* Inspector, iName, *and* iSalary.

Suppose we want to retrieve the names of inspected products as well as the names and salary of their relative inspectors. For this purpose, we formulate a SPARQL query such as Q from Fig. 2, consisting of five triple patterns $t_1, \ldots, t_5$. We send Q to the federation engine for evaluation over the data federation instance. As the first step, the engine checks the syntax of Q w.r.t. the syntax of SPARQL and the classes and properties declared in $\mathcal{V}$. After the syntactic check, the engine identifies the sources of each triple pattern in Q, and further partition Q into sub-queries according to some query partition strategy. In our example, by exploiting the mapping set $\mathcal{M}$, the federation engine infers that triple patterns t_1, t_2, and t_3 only refer source $S1$, and $t_4 - t_5$ only refer source $S2$. Then, by adopting exclusive groups*, i.e., a well known/classical query partition and optimization strategy originally proposed in the works [18, 19], the engine computes a partition $\mathcal{Q} = \{q_1, q_2\}$ of Q, via putting together the triple patterns t_i and t_j such that t_i and t_j refer to the same source*[1]*. After that, the engine computes a plan for evaluating $\mathcal{Q}$. A possible plan is the following: reformulate query q_1 into a SQL query q'_1 and query q_2 into a CSV query q'_2, according to the mappings definitions in $\mathcal{M}$; dispatch q'_1 to $S1$ and q'_2 to $S2$, and evaluate them in a parallel way; merge the returned answers for q'_1 and q'_2 to generate the answers of the initial query Q.*

As mentioned earlier, beyond the core feature of federated query answering, data federation has evolved to offer a wide range of additional capabilities supporting more powerful and intelligent forms of data consumption and management. Next, we list some of these capabilities, which may be of interest for researchers, students, as well as users who pursue extra functionalities of federation systems:

- *Data security.* It provides techniques for protecting users' privacy and sensitive data from leakage. Take the data federation platform Denodo as an example. The "unified security management" of Denodo offers a single point to control the access to any piece of information. Different users of Denodo are only allowed to access either filtered or masked data by using the Denodo role-based security model. Interested readers can refer to the official documents for more details;
- *Data update.* It provides the capability of enabling users to both read and write the data of the sources participating in the federation. For example, the SPARQL federation engine FedX [2] supports SPARQL updates[3] so as to make users able to modify the data of the SPARQL endpoints, and the SQL federation engine Denodo supports SQL data management language (SQL DML) with the motivation of making users able to modify the data stored in the source databases;
- *Data quality.* It provides the techniques for guaranteeing the correctness and consistency of data. Take the SAS Federation Server [4] as an example. Data quality on SAS Federation Server is implemented through a "SAS Quality Knowledge Base (QKB)", allowing for the specification of a set of methods and rules for data quality, such as rules to cleanse the data.

3. Survey methodology

This survey work spurs from our needs of selecting suitable data federation systems for heterogeneous data integration. Collecting, analyzing, and comparing the existing systems on data federation is a very time-consuming process. Sharing the results of our study can benefit readers interested in data federation solutions, such as end-users (consumers), developers, researchers and students. In this section, we present the overall methodology used for our study. Fig. 3 provides a snapshot of our methodology, which consists in the *identification of the considered systems*, the *design of the system evaluation framework*, and the *evaluation of the systems through the framework*.

[1] In this situation, t_i and t_j can be evaluated together in a single sub-query rather than being evaluated separately, because the join of the two triple patterns can be performed at the level of the single data source.

[2] https://rdf4j.org/documentation/programming/federation/

[3] https://www.w3.org/TR/sparql11-update/

[4] https://manualzz.com/doc/o/pcxi2/sas%C2%AE-federation-server-4.2–administrator-s-guide-data-quality-on-sas-federation-server

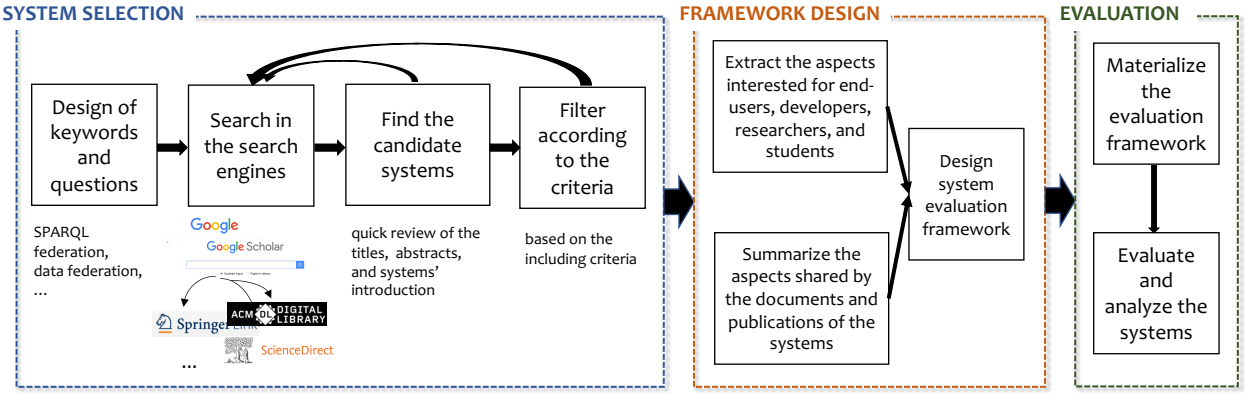

Fig. 3. The overall methodology of the survey work.

3.1. The methodology for system selection

As shown in Fig. 3, the systems considered in our survey are mainly identified through a four stage process: *designing keywords and questions*, *searching in the search engines*, *finding the candidate data federation systems*, and *filtering according to the including criteria*. This process required more than three months of work (2021/10-2021/12). Although sharing the same stages, the criteria for selecting academic systems and industrial ones are a bit different. For clarity, in the following, we describe the selection of academic systems and industrial systems separately.

The selection of academic systems. The considered academic systems are selected by reviewing the academic publications found via surfing the Google Scholar Search Engine. As a first step, we designed the following keywords to find the potential systems:

> "SPARQL federation", "SQL federation", "query federation", "federated query answering", "database federation", "federated database", "data federation", "data virtualization", "virtual data integration"

Note that for obtaining "more" results, we did not use any operator, like "AND" and "OR", in the search phrases. After searching these keywords, we speed-read the titles and abstracts of more than 2000 academic publications which are located in libraries such as *SpringerLink*, *IEEE Xplore*, *ACM Digital Library*, and so on. By evaluating these titles and abstracts, we downloaded 268 academic publications (we ignored the ones without free access), including papers, technical reports and phd (master) theses, for further in-depth reading, so as to find candidate data federation systems. We provide the full bibliography for these publications in Appendix B.

After reviewing these publications, we finally selected 17 academic systems identified from the ones satisfying the following *including criteria*:

- formal publications, including in conference proceedings or journal volumes, not preprints;
- focusing on the problem of query federation, or introducing a data federation system;
- providing the links to the source code or official websites of the systems, or source code findable from the authors' GitHub repositories (like *SPLENDID*).

We selected all the systems whose corresponding publications satisfied the above three criteria and were published after 2018, like [25] and [52], with the motivation of including more recent systems. For the systems identified from the publications satisfying the above criteria and published before, we choose some representative/classical ones, measured by the factors including the venue of the publications, the citations (found from Google Scholar and used to measure how much the systems are considered by other works), and whether they are mentioned in other survey work on data federation.

The selection of industrial systems. To find candidate industrial systems we adopted the Google Search Engine. The keywords used were the following:

> "data federation", "data virtualization", "query federation systems",
> "SPARQL query federation systems/tools/platforms/engines",
> "SQL query federation systems/tools/platforms/engines",
> "data federation systems/tools/platforms/engines",
> "data virtualization systems/tools/platforms/engines",
> "the systems like X",

where *X* denotes an already known data federation system. After this search, we opened the web-pages of the returned items, and ignored the same items returned by searching different keywords. Some of the web-pages were the official websites of some systems, some were talking about data federation/virtualization, and others were recommending/listing/comparing systems referring data federation, virtualization or integration. After quickly reviewing the introductions of the systems in the web-pages, we collected 72 candidate systems that may provide the capability of data federation. We then consulted the official websites of these systems and read their system review and documentation carefully, and eventually selected 31 industrial systems for our survey work, strictly following the including criteria below:

- having official website,
- providing the capability of query/data federation,
- having public official documents that introduce to the system, and
- having a community,

and the following excluding criteria:

- without updates in the last five years (from 2020-10, i.e., the time we started this survey).

The concrete information, as their names, owners, and description, can be found in Section 5.

3.2. The methodology for designing the evaluation framework

In this subsection, we describe the methodology of designing the framework for evaluating the selected systems in a uniform and qualitative way. The framework should be able to provide information that can benefit end-users (data federation consumers), developers, researchers and students maximally. Thus, as shown in Fig. 3, our idea is to extract the aspects that are of interest for them. These aspects were derived by answering the following questions:

Q1. What aspects are relevant for a data federation consumer;
Q2. What aspects are relevant for developers that want to implement new data federation systems or integrate existing systems to support more complex data consumption;
Q3. What aspects are relevant to researchers or students carrying out data federation related studies or research;
Q4. What aspects have been considered important by other survey works on data federation, or are shared by academic publications referring these systems;
Q5. What aspects are shared by official documents referring these systems;
Q6. What aspects are usually considered in web-pages that list or compare systems[5].

The system evaluation framework, consisting of four dimensions with sub-dimensions, is generated by combining, classifying, and specifying these aspects. The full process is depicted in Fig. 4. In such figure, clouds contain the answers to questions Q1, ..., Q6. We remark that answers to questions Q1, Q2, and Q3 were not obtained by actually interviewing these categories of people, but rather by relying on our expertise as developers and researchers, as well as our own experience on the data federation task and systems. We next provide a brief justification for our answers to these three questions.

[5]Like, for instance, https://en.wikipedia.org/wiki/Comparison_of_relational_database_management_systems

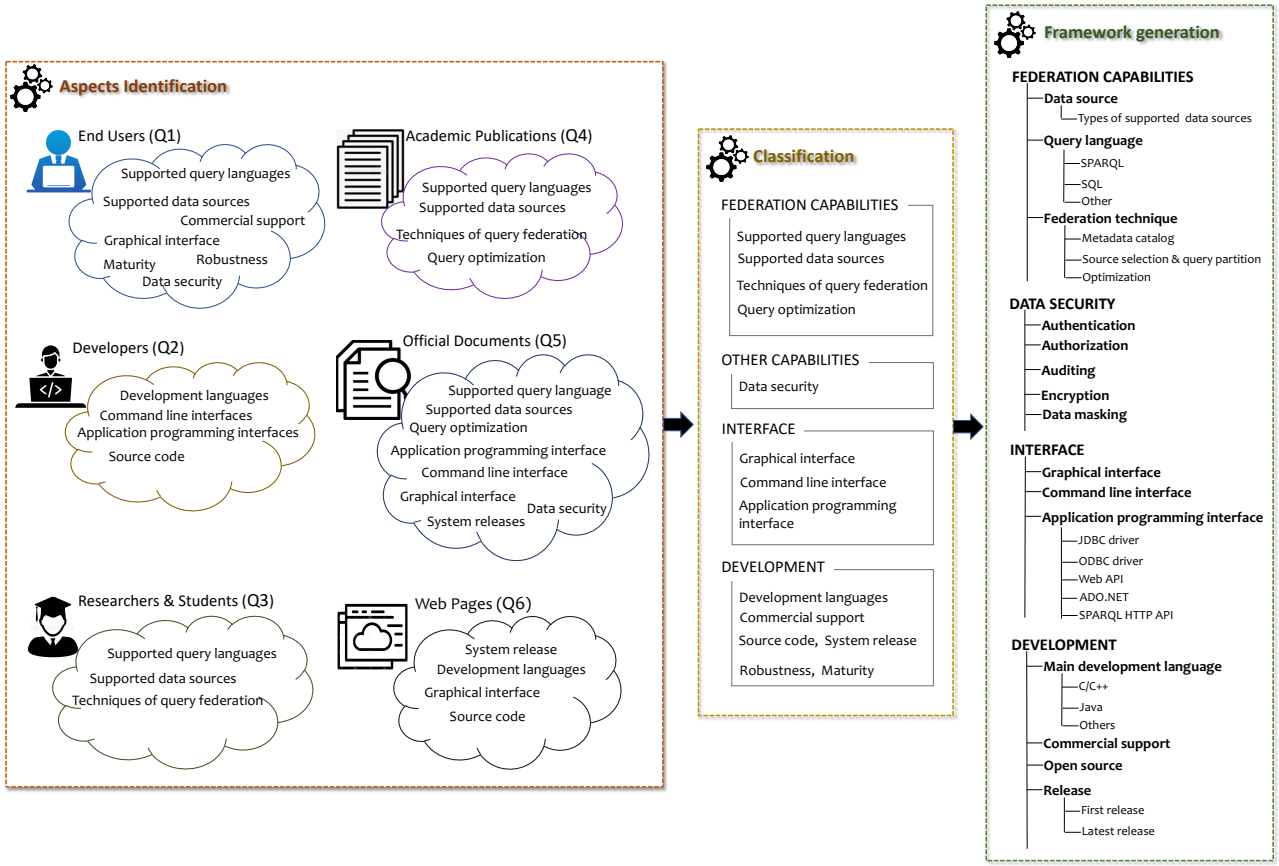

Fig. 4. The generation of the system evaluation framework.

- *End-users:* they have the concrete need of integrating and federating data sources, and might lack technical skills like programming. Hence, aspects relevant to them are whether the system is able of handling their data sources, whether it provides a query language that they are familiar with, whether it offers graphical interface to help them set up a data federation instance easily, whether it provides the services for solve the problem they may encounter, whether it provides the techniques of protecting their data from leakage, and whether it is robust enough so as to reduce the technique problems they may encounter.
- *Developers:* Their need is to work with the systems at a lower-level than end-users, for instance through programming interfaces, so as to enrich the functionalities delivered by their own applications. Other developers might also be interested in the source code of the systems themselves, for the purpose of extending it with new functionalities.
- *Researchers & Students:* They usually plan to carry out innovative research referring data/query federation. Thus, aspects of interest for them relate to the knowledge of the capabilities of the systems, or of the strategies they adopt.

After identifying all the aspects, we classified them into *four categories*, as shown in the middle part of Fig. 4. Starting from this rough classification, we obtained our framework through a series of refining steps. The framework itself, shown in the right part of Fig. 4, will be presented in Section 4.

3.3. The methodology of system evaluation

After identifying the considered systems and the evaluation framework, we use our framework to investigate and analyze the capabilities, strengths, and weakness of the considered systems, e.g., the capability of handling data heterogeneity. Finally, we point out some open problems and challenges that might be addressed by further research.

4. The framework for system evaluation and comparison

In this section, we present our framework for analyzing and comparing the selected systems under a user and application perspective in a uniform and qualitative way. Our framework, shown in the right part of Fig. 4, consists of four dimensions: *federation capabilities*, *data security*, *interface*, and *development*. Each dimension is further characterized by sub-dimensions (15 in total). In the remainder of this section we discuss each of these dimensions, and relative sub-dimensions, in detail.

4.1. Federation capabilities dimension

This dimension evaluates the main task of data federation systems, i.e., federated query answering, in terms of data source, query language, and federation technique.

Data source sub-dimension. The types of supported data sources usually play a key role when choosing a data federation system. For example, if a company has massive CSV files that need to be virtually integrated with data stored in MySQL, then, obviously, the systems not supporting CSV files and MySQL at the same time will not be taken into consideration. The information of this sub-dimension also permits distinguishing whether a system focuses on heterogeneous vs. homogeneous data sources. Roughly speaking, the more different types of data sources a system supports, the more powerful the system is in accessing heterogeneous data virtually. By reviewing the data sources supported by the considered systems, we design six types of data sources, like *relational* and *graph-based*, to inspect this sub-dimension. The concrete information will be introduced in Section 6.1.

Query language sub-dimension. We consider the query language(s) provided to users for accessing and managing the data in the federated sources. Generally speaking, a system should better adopt a standard query language that is familiar to most people, such as SQL or SPARQL. In this way, users do not need to learn a new query language when using the system, and existing tools and resources for the adopted language can be reused. Besides, a system that supports a larger subset of a language (or even a superset of it), such as the whole SPARQL, is more powerful than a system supporting a smaller subset of the language, such as the Basic Graph Patterns (BGP) of SPARQL [12], from the perspective of query answering. As mentioned before, we mainly considered the systems developed in the Semantic Web community and database community, but not limited to these two kinds. Thus this sub-dimension is further characterized into *SPARQL*, *SQL*, and *Other*.

Federation technique sub-dimension. We refer to the four-step architecture for federated query answering described in Section 2, and assess the main techniques adopted by a system. We mainly focus on metadata catalog, source selection and query partition, and query optimization and query plan generation. The motivation is to help readers in forming a general idea of the techniques used.

4.2. Data security dimension

As a data-centric application, data federation offers a single logical point to integrate data from multiple sources that may contain sensitive and private data (*e.g.*, financial transactions, users' contact information, medical procedures). The protection of such data represents a crucial problem for obtaining the trust of users and data providers. This problem is further complicated by the risk of leaking sensitive information through analysis and correlation of otherwise non-sensitive data from separate sources. Therefore, the data security dimension considers whether a data federation system has the ability of safeguarding data from unwanted actions of unauthorized users, and it is further organized in sub-dimensions according to the system's support for the most common data security mechanisms.

Authentication sub-dimension. Authentication refers specifically to accurately identifying users before they have access to data. It is the act of validating that users are whom they claim to be, and is the first step in any data security process. The most common authentication mechanism is a username and password combination. Other common authentication mechanisms use shared keys, PIN numbers, or security tokens.

Authorization sub-dimension. Authorization is a mechanism for granting or denying access to a resource based on identity. More generally, it consists in defining an access policy, and is usually implemented through a set of declarative security roles which can be associated to users. Authorization is different from authentication, and usually happens after authentication.

Auditing sub-dimension. Data auditing logs and reports on events like users' accesses, modifications, changes of ownership, or permissions regarding sensitive data. Audit procedures increase visibility on data operations and are instrumental to the investigation and prevention of data breaches and other data security incidents.

Encryption sub-dimension. Data encryption algorithms transform the original data into an unreadable format so that only authorized users having the corresponding key can decrypt and read the information. Encryption is commonly employed on data transiting between the system and the user, and possibly on data stored, cached, or otherwise materialized within the system as well, to protect them from unauthorized low level accesses.

Data masking sub-dimension. Data masking[6] is the process of masking (obscuring, encrypting, deleting, or otherwise scrambling) specific pieces of accessed data, so as to ensure that sensitive information is not exposed to unauthorized parties (*e.g.*, users, developers, system administrators).

4.3. Interface dimension

The ultimate goal of system development is to support users in fully appreciating, accessing, and exploiting the features implemented by the system. Its achievement largely depends on the interface(s) offered to users for interacting with the system, which ultimately determine the ease of use, *i.e.*, the usability, of a system. Such interfaces are the subject of this dimension, whose sub-dimensions are organized according to the different types of interfaces commonly offered by systems.

Graphical interface sub-dimension. Setting up a data federation system is typically a complex task involving an extensive amount of configuration, *e.g.*, for connecting the federated data sources, acquiring their necessary metadata, and setting up the system components. For example, Teiid supports the use of a complex XML configuration file[7] to define a federated database, there called a Virtual Database (VDB). Without fully understanding the syntax and components of this file, building a VDB is hard for users, especially for the less-technical ones. A graphical user interface may greatly ease the configuration process, as well as other administration and operation tasks, and thus largely affects the user friendliness of a system.

Command line interface sub-dimension. Data federation systems are typically used as components of larger information systems, where they need to be integrated with other components, such as business intelligence (BI), customized dashboards, or machine learning tools, to support or handle much more complex applications and tasks. To that respect, a command line interface provides a first, simple solution for automatically invoking the functionalities of a data federation system in other programs or scripts of a larger information system.

Application programming interface sub-dimension. A further, more flexible integration mechanism is represented by application programming interfaces (APIs) offered by the data federation system, such as web APIs or client libraries in various programming languages (*e.g.*, ODBC/JDBC drivers). Such APIs make it easier for developers to connect, configure, and operate an instance of the system at run-time within other applications.

[6] https://en.wikipedia.org/wiki/Data_masking
[7] http://teiid.github.io/teiid-documents/16.0.x/content/reference/r_xml-deployment-mode

4.4. Development dimension

This dimension considers the development, release, and support practices of a system, with its sub-dimensions capturing the aspects that are most relevant when matching the non-functional requirements of a user (in terms of, *e.g.*, performance, robustness, flexibility, sustainability).

Main development language sub-dimension. The main programming language(s) used to develop the core functionalities of a system, along with their runtimes (*e.g.*, the Java Virtual Machine for the Java language), influence system requirements, performance, customization, and integration options (*e.g.*, embedding the system as a library), and consequently affect the system fitness for use in an intended user application.

Commercial support sub-dimension. Learning how to best use an unfamiliar and complex system and dealing with any issue preventing its normal operation are time-consuming activities, which may result in additional costs or even in economic losses due to system downtimes. Therefore, the availability of commercial support, *e.g.*, in form of training, timely bug fixes, and installation and customization services, plays a keys role when choosing a system.

Open source sub-dimension. Systems whose source code is made freely available for modification and redistribution offer users more options for integrating the system while matching specific application requirements, for improving the system itself, and for maintaining the system even if it is no more supported by authors.

Release sub-dimension. We consider the release history and practices of a system, focusing on the number of releases and the time between the first and the last release of the system. Generally speaking, the longer this time and the more numerous the releases, the more mature and robust the system typically is, since each new release is obtained by adding new functions or fixing some issues in the previous one. For example, the first release (v1.0) of the Denodo platform was in 2002 and the last (v8.0) was in 2020. Thus, Denodo development has been active for almost 20 years, which makes it potentially more robust than some other younger systems.

5. Overview of the selected data federation systems

Before reporting on the application of the framework of Section 4, we provide here the list and a brief overview of the selected systems involved in our evaluation and comparison, also to help readers familiarize with the current offer on data federation, both industrial and academic, as a whole. For the data federation systems developed in the context of the Semantic Web community, more academic ones and less industrial ones were found. On the contrary, for the systems developed within the context of the relational databases community, more industrial ones and less academic ones were identified.

Table 1 lists the selected systems alphabetically, reporting for each one its name with relevant references where to gather detailed information, academic (name in *italics*) or industrial nature, provider, and a one sentence description introducing the system (in its latest version) and complementing the detailed information reported in the next sections. Note that here and in the following, the information for industrial systems (31 in total) was mainly extracted from their official websites, while for the academic systems (17 in total), information was mostly extracted and summarized from their academic publications.

On the whole, the table exhibits a substantial variability in terms of system provider, nature, and their main characteristics. Providers range from university and research institutions for academic systems, to open source organizations, specialized companies, and major corporations for industrial systems. Systems range from database engines (RDBMS, Graph DBs, triple stores, polystores, and other multi-model systems) whose storage services are augmented with data federation capabilities, to purely mediator systems specifically focusing on data federation, possibly complemented with accessory functionalities (*e.g.*, security). Some industrial systems can be accessed only as cloud services.

Table 1

Summary of the selected data federation systems. Academic systems in *italics*.

System	Provider	Description
AllegroGraph [53]	Franz Inc.	Distributed graph & document DB supporting OWL, SPARQL, SHACL and federation
Amazon Athena [54]	Amazon.com, Inc.	Inter. cloud query service for Amazon S3 data, based on Presto [55]
Amazon Neptune [56]	Amazon.com, Inc.	Fully-managed cloud graph DB (property graph, RDF), part of Amazon AWS
AnzoGraph DB [57]	Cambridge Semantics	Massively-parallel distributed graph DB (property-graph, RDF) for large-scale analytics
Apache Drill [49, 58]	Apache Software Foundation	Distributed schema-free engine for interactive SQL queries on heterogeneous & nested data, inspired by Dremel [59]
Apache Jena (ARQ) [60]	Apache Software Foundation	SPARQL query engine of Jena framework and TDB triple store, supporting federation
Apache Spark [61, 62]	Apache Software Foundation	Multi-lang. (incl. SQL) distributed engine for large-scale data processing & analytics
BigDAWG [30, 63]	Intel Science & Technology Center for Big data	Polystore with heterogeneous storage engines for time series (SciDB), text (Accumulo) and relational data (PostgreSQL)
CloudMdsQL [32, 64]	Inria & LIRMM	Polystore integrating heterogeneous storage engines (incl. RDBMS, NoSQL, HDFS)
CostFed [65]	Univ. Leipzig	Index-assisted, cost-based data federation system for SPARQL endpoints
DARQ [50]	Univ. HU Berlin	Earliest data federation system for SPARQL endpoints, cost-based
Data Virtuality [22]	Data Virtuality GmbH	Heterogeneous data integration solution combining data virtualization and ETL
Denodo [66]	Denodo Technologies Inc.	Data virtualization solution for heterogeneous sources, also available as cloud service
Dremio [67]	Dremio Corporation	Data “lakehouse” (lake + warehouse) solution supporting heterogeneous data sources
FEDRA [68]	Univ. Nantes (LINA lab.)	Data federation system for SPARQL endpoints exploiting data replication
FedX (RDF4J) [18, 19]	fluid Operations AG	On-demand (no statistics, query-time) data federation system for SPARQL endpoints
GraphDB [69]	Ontotext	Triple store featuring OWL reasoning, SPARQL federated queries & RDBMS access
HiBISCuS [70]	Univ. Leipzig	Source selection for SPARQL data federation (DARQ, FedX & SPLENDID extension)
IBM Cloud Pak for Data [71]	IBM	Data federation system with data discovery, governance, security and privacy solutions, also available as cloud service (formerly IBM Cloud Private for Data)
IBM Db2 Big SQL [72]	IBM	Massively-parallel Hadoop SQL engine for heterogeneous sources (formerly IBM SQL)
IBM InfoSphere Federation Server [73]	IBM	SQL-based data federation system for heterogeneous sources (formerly WebSphere Federation Server)
JBoss Data Virtualization [74]	Red Hat, Inc.	Data federation system based on Teiid and providing read/write access to heterogeneous sources, data security, and multiple user interfaces / APIs
Lusail [75]	Univ. KAUST	Data federation system for SPARQL endpoints using schema & instances statistics
Metaphactory [76, 77]	metaphacts GmbH	KG platform on top of SPARQL endpoints with two federation engines (Ephedra, FedX)
Myria [31]	Univ. Washington	Cloud service for big data management/analytics with parallel & federated query engine
Neo4j (Fabric) [78]	Neo4j, Inc.	Federation solution of Neo4J graph DB (Cypher [79] queries on property graph model)
Obi-Wan [80, 81]	Inria & Polytechnic Institute of Paris	Ontology Based Data Access (OBDA) [37] system on top of Tatooine [82] mediator for heterogeneous sources
Odyssey [83]	Univ. Aalborg & Univ. Nantes	Statistics & cost-based optimizer for SPARQL data federation (FedX extension)
Ontario [25]	L3S Research Center	Heuristics-based data federation system using RDF Molecule Templates [84] to describe/map source contents as star-shaped RDF instance descriptions
Onto-KIT [85]	Univ. Toulouse	Data federation system focusing on Earth Observation data with hypergraph-based data model and query processing techniques
Oracle Big Data SQL [86]	Oracle Corporation	Data federation system for Oracle DB that accesses Hadoop storage & processing
Oracle DB (Spatial & Graph) [87, 88]	Oracle Corporation	Oracle DB component for semantic technologies with data federation capabilities (RDF views) over relational, graph, and RDF (SPARQL) sources
PolyWeb [29, 89]	Univ. NUI Galway	SPARQL-based data federation system for different sources on the Web (RDF & CSV data, RDBMS), focusing on source selection, query optimization & execution
Presto [55, 90]	Presto Foundation	SQL-based distributed query engine suitable to interactive (big) data analytics
Querona Data Virtualization [91]	YouNeedIT Sp. z o.o. Sp. k.	Data federation system for a variety of heterogeneous sources, based on Apache Spark and targeting big data analytics with the support of main BI tools
SAFE [92]	Insight SFI Research Centre for Data Analytics	Data federation system for SPARQL endpoints exposing RDF data cubes with sensitive data, featuring access policy-aware data summaries, source selection & query execution
SAP HANA [93]	SAP SE	In-memory DB targeting with data federation capabilities, also available as cloud service
SAS Federation Server [94]	SAS Institute	Data federation system featuring data caches, masking, encryption & quality functions
SemaGrow [95]	IIT NCSR ‘Demokritos’	Data federation system for SPARQL endpoints with statistics-based query optimization
SPLENDID [51]	Univ. Koblenz-Landau	Data federation system for SPARQL endpoints that provide VOID [96] data statistics
SQL Server (PolyBase) [97]	Microsoft Corporation	SQL Server component for data federation supporting Hadoop and Azure storage
Squerall [52]	Univ. Bonn	Data federation system for heterogeneous sources built on Spark & Presto and following the OBDA framework
Starburst [98]	Starburst Data, Inc.	Commercial distribution of Trino, extra security features, available on-premise/on-cloud
Stardog [99]	Stardog Union	KG platform including data federation of heterogeneous sources & query-time inference
Teiid [20]	Red Hat, Inc.	SQL-based engine for data federation of heterogeneous sources
TIBCO Data Virtualization [100]	TIBCO Software Inc.	Data federation system for heterogeneous sources, with data caching & security, massively parallel processing & GUI tools (formerly Composite, then Cisco Data Virtualization)
Trino [101]	Trino Software Foundation	SQL-based query distributed engine for interactive big data analytics, forked from Presto
Virtuoso [102–104]	OpenLink Software	Multi-model DB (object-relational, RDF, XML) with data federation facilities

6. System evaluation and analysis

In this section, we investigate and analyze in more detail each of the systems overviewed in Section 5, while applying the four dimensions of the proposed framework. The main goal is to better understand the main characteristics of each system and to reveal its strengths and weaknesses with respect to the main task of data federation. Notice that all the systems we investigated have been considered as in their latest version (last update on November 20th, 2021).

6.1. Federation capabilities dimension

In this subsection, we evaluate the selected systems with a special attention to their capabilities to support federated query answering. In doing this, we will highlight the query languages that are supported, the data sources each system is able to manage, and the adopted federation techniques. Concerning the first two aspects, a synthetic overview of the query languages and the types of data sources supported by the investigated systems is presented in Table 2. The concrete data source implementations (*e.g.*, MySQL) supported by each system are instead listed in Table 7 (in the appendix).

Query language. According to columns 2–4 of Table 2, we can make the following three observations:

1. With no significant distinction between industrial or academic implementations, the standard and popular query languages SQL and SPARQL are adopted by most of these systems to query the data involved in the federation. Notice also that BigDAWG, CloudMds, Myria, and SAS Federation Server use alternative languages inspired by SQL to support the required capabilities in the distributed federation environment. Instead, Neo4j adopts the declarative graph language Cypher [79] as its underlying query language, with the motivation of making graph data querying easy to learn, understand, and use by the final users.
2. There exist very few systems that adopt multiple query languages at the same time. Among them, for instance, AllegroGraph supports SPARQL and Prolog simultaneously; GraphDB provides the capability of processing SPARQL, SQL, and Cypher queries; and Virtuoso takes both SPARQL and SQL as its query languages. This situation can be explained by taking into account that *(i)* the importance or necessity of supporting multiple query languages is unknown or ignored, and *(ii)* supporting multiple query languages within the very same system requires a lot of work from an engineering and development point of view.
3. Most of the academic SPARQL-based systems (*e.g.*, [25, 51, 65, 75, 80, 83, 92]) are limited to BGPs, that is, a sub-language of SPARQL, and ignore operators like `UNION` and `OPTIONAL`, which play a key role in expressing complex queries.

In summary, columns 2–4 of Table 2 suggest that it is clearly an added value for a system to provide support for popular and standard query languages, so as to prevent users from spending too much time learning an unfamiliar language for using the system. Moreover, this choice definitely eases the integration of the system with other possible interacting applications. Additionally, systems supporting multiple query languages are rare, as well as systems supporting the full SPARQL language specification.

Data source. Uniformly evaluating and analyzing systems in terms of supported data sources is a challenging task for two main reasons. Firstly, system providers usually adopt different standards and granularity to describe the data sources they support. Some systems classify supported data sources differently and possibly in incompatible ways. For example, relational sources all go under the *databases* class in Teiid [8], while Denodo [9] distinguishes between the classes of *JDBC databases*, *ODBC sources*, and *multidimensional databases*. Instead, Apache Drill [10] and Trino [11] list all the data sources they support without any classification, and IBM Cloud Pak for Data Virtualization [12] solely

[8] http://teiid.github.io/teiid-documents/16.0.x/content/reference/r_data-sources.html
[9] https://community.denodo.com/docs/html/browse/8.0/en//vdp/vql/generating_wrappers_and_data_sources/creating_data_sources/creating_data_sources
[10] https://drill.apache.org/docs/connect-a-data-source-introduction/
[11] https://trino.io/docs/current/connector.html
[12] https://www.ibm.com/docs/en/cloud-paks/cp-data/3.5.0?topic=data-supported-sources

Table 2

Evaluation of query language and data source sub-dimensions. Academic systems in *italics*. "–" denotes feature/information not found in the systems' official documentation, websites, or academic publications, to the best of our efforts.

System	Query language			Data source					
	SPARQL	SQL	Other	Relational	Graph-based	Aggregate-oriented	Structured Files	Web Service Paradigms	Other
AllegroGraph	✔	–	Prolog	–	✔	–	–	–	–
Amazon Athena	–	✔	–	✔	✔	✔	✔	–	✔
Amazon Neptune	✔	–	–	–	✔	–	–	–	–
AnzoGraph DB	✔	–	Cypher	✔	–	–	✔	✔	–
Apache Drill	–	✔	–	✔	–	✔	✔	✔	✔
Apache Jena (ARQ)	✔	–	–	–	✔	–	–	–	–
Apache Spark	–	✔	–	✔	–	–	✔	–	–
BigDAWG	–	–	BigDAWG Query	✔	–	✔	–	–	✔
CloudMdsQL	–	–	CloudMdsQL	✔	✔	✔	–	–	–
CostFed	✔	–	–	–	✔	–	–	–	–
DARQ	✔	–	–	–	✔	–	–	–	–
Data Virtuality	–	✔	–	✔	✔	✔	✔	✔	✔
Denodo	–	✔	–	✔	–	✔	✔	✔	✔
Dremio	–	✔	–	✔	–	✔	✔	–	–
FEDRA	✔	–	–	–	✔	–	–	–	–
FedX (RDF4J)	✔	–	–	–	✔	–	–	–	–
GraphDB	✔	✔	Cypher	✔	✔	–	–	–	–
HiBISCuS	✔	–	–	–	✔	–	–	–	–
IBM Cloud Pak for Data	–	✔	–	✔	–	✔	✔	✔	✔
IBM Db2 Big SQL	–	✔	–	✔	–	✔	✔	–	✔
IBM InfoSphere Federation Server	–	✔	–	✔	–	–	✔	✔	✔
JBoss Data Virtualization	–	✔	–	✔	–	✔	✔	✔	✔
Lusail	✔	–	–	–	✔	–	–	–	–
Metaphactory	✔	–	–	✔	✔	✔	–	✔	–
Myria	–	✔	MyriaL	–	✔	✔	✔	–	✔
Neo4j (Fabric)	–	–	Cypher	–	✔	–	–	–	–
Obi-Wan	✔	–	–	✔	✔	✔	–	–	–
Odyssey	✔	–	–	–	✔	–	–	–	–
Ontario	✔	–	–	✔	✔	✔	✔	–	–
Onto-KIT	✔	–	–	–	–	–	✔	–	–
Oracle Big Data SQL	–	✔	–	✔	–	✔	✔	–	✔
Oracle DB (Spatial & Graph)	✔	–	–	✔	✔	–	–	–	–
PolyWeb	✔	–	–	✔	✔	–	✔	–	–
Presto	–	✔	–	✔	–	✔	–	–	✔
Querona Data Virtualization	–	✔	–	✔	–	✔	✔	–	✔
SAFE	✔	–	–	–	✔	–	–	–	–
SAP HANA	–	✔	–	✔	–	–	–	–	✔
SAS Federation Server	–	–	FedSQL	✔	–	–	–	–	✔
SemaGrow	✔	–	–	–	✔	–	–	–	–
SPLENDID	✔	–	–	–	✔	–	–	–	–
SQL Server (PolyBase)	–	✔	–	✔	–	✔	✔	–	–
Squerall	✔	–	–	✔	–	✔	✔	–	–
Starburst	–	✔	–	✔	–	✔	✔	–	✔
Stardog	✔	–	–	✔	✔	✔	✔	–	✔
Teiid	–	✔	–	✔	–	✔	✔	✔	✔
TIBCO Data Virtualization	–	✔	–	✔	–	✔	✔	✔	✔
Trino	–	✔	–	✔	–	✔	–	–	✔
Virtuoso	✔	✔	–	✔	–	–	–	–	–
Number	**24**	**22**	**8**	**32**	**25**	**24**	**23**	**10**	**20**

classifies the supported data sources into IBM data sources, third-party data sources, and files. Secondly, systems may list as supported both a generic data access interface (*e.g.*, JDBC, ODBC, ADO.NET, OLE DB, SPARQL HTTP protocol, etc) and some data sources available through that interface, with different meanings. Often, the listed sources are just examples or special cases for which additional capabilities are implemented, and additional sources may be configured (e.g., by tuning the employed SQL dialect) and connected through the interface. In some cases, however, the listed sources are the only ones supported through the interface, which we thus disregard in our assessment. These two factors make it difficult to assess the supported data source sub-dimension uniformly and precisely.

In order to understand the *status quo* of handling the variety dimension of big data in the data federation setting, after inspecting the data sources supported by each system, we take the following 6 types of sources into consideration: *(i) Relational*, including SQL-based RDBMS, (federated) query engines, and distributed/cloud stores; *(ii) Graph-based*, including SPARQL endpoints, RDF triple stores and property graphs; *(iii) Aggregate-oriented*, including key-value stores, wide-column stores, document stores and other NoSQL stores and search engines that organize data as "aggregates" [105], ranging from opaque values to arbitrarily complex nested documents;[13] *(iv) Structured Files* such as CSV, JSON and XML; *(v) Web Service Paradigms* to access arbitrary Web sources, such as HTTP/REST and SOAP/WSDL (vs. specific Web APIs like Facebook one); and *(vi) Other*. We manually classified each specific data source (*e.g.*, MySQL, MongoDB) supported by a system under one of the considered 6 data source types (*e.g.*, relational and aggregate-oriented, respectively), also relying on established system classifications (*e.g.*, DB-Engines [106] and Database of Databases [107] catalogs). We use "Other" as a container for all those infrequently supported sources not covered by the former 5 types, such as directory services, streaming and event data processing systems, specialized databases (*e.g.*, for time series) and protocols (*e.g.*, IMAP), and various specialized Web APIs.

By combining Table 2 and Table 7, we can observe the following:

1. Industrial systems usually support more data sources than academic systems (respectively, 3.3 vs 1.9 distinct source types per system on average). Consider for example Data Virtuality, which covers all the source types we considered. It is an unsurprising conclusion, since industrial systems usually focus more on coverage.
2. As for the systems covering multiple, possibly heterogeneous, types of data sources, no matter whether industrial or academic, relational sources have been considered extensively, and most of the mainstream RDBMS implementations have been supported (cf. second column of Table 7). This may be caused by the dominant role of relational sources in organizing data. Besides, the well-formalized syntax and semantics of SQL makes it much easier to interface it with other query languages possibly supported by the data sources participating in the federation.
3. Structured files like JSON, XML, and CSV, because of their importance and wide use, are also directly supported as native data sources by many systems considered in this survey (23 out of 48, *i.e.*, 48%). Other systems not directly supporting structured files may instead support the database systems commonly used for storing and indexing the kind of data of these files (*e.g.*, MongoDB and Elasticsearch for JSON data).
4. Aggregate-oriented sources mostly consist of NoSQL systems (cf. the fourth column of Table 7), exhibit overall support (24 systems out of 48, *i.e.*, 50%) similar to the one for graph-based sources and structured files, and are present both in industrial systems (18 out of 31, *i.e.*, 58%) and, marginally less, in academic systems (6 out of 17, *i.e.*, 35%).
5. Web service paradigms, although important (many sources are available only as web services), are relatively less considered (10 systems out of 48, *i.e.*, 21%). This may be caused by the difficulty of implementing federated query answering over such kind of data, as their data models (where defined) and access patterns (usually restricted) are very dissimilar from the ones exposed by the data federation system to its users.
6. Other sources in our classification consist mostly of specialized Web APIs (cf. last column of Table 7) and are supported by industrial systems (18 out of 31, *i.e.*, 58%) more than academic systems (2 out of 17, *i.e.*, 12%).

[13] We use the broad "aggregate-oriented" category due to the difficulty of classifying many NoSQL stores into a single fine-grained category (*e.g.*, Amazon DynamoDB is independently classified as key-value, wide-column, or document store by different academic and web sources).

Table 3
Summary of the main techniques used in federated query answering. Academic systems in *italics*.

	Federation techniques	Example systems
Metadata catalog	Self-defined (*e.g.*, Virtual DB, Virtual Table, Remote Table)	Denodo, TIBCO Data Virtualization, Teiid, *HIBISCUS*, *CostFed*
	Standards (*e.g.*, RML, R2RML, VoID)	*Squerall*, *PolyWeb*, *SPLENDID*, *SemaGrow*
	No-metadata	Apache Drill, RDF4J (FedX)
Source selection and query partition	Index-based	Teiid, Denodo
	Rule-based	Teiid, *Onto-KIT*, *Ontario*
	Query-based	RDF4J (FedX)
	Push down	Teiid, Denodo, Dremio, RDF4J (FedX)
Query optimization and query plan generation	Costed-based models	Denodo, Data Virtuality, Presto, *DARQ*
	Rule-based models	Teiid, Apache Drill, *Myria*, RDF4J (FedX)
	Bind join	Teiid, *CloudMdsQL*, *DARQ*, *SAFE*
	Nested loop join	Denodo, Data Virtuality, *DARQ*, *SAFE*
	Hash join	Denodo, TIBCO Data Virtualization
	Broadcast join	Apache Drill, Apache Spark
	Merge join	Denodo, TIBCO Data Virtualization
	Optional join	Teiid, JBoss Data Virtualization
	Cache	Dremio, Apache Drill, SAS Federation Server
	Parallel	Denodo, Teiid, Dremio, *Myria*, Lusia, *CostFed*

7. Systems supporting SQL queries focus on relational sources (21 systems out of 22, *i.e.*, 95%) while graph-based sources have rarely been taken into account (4 out of 22, *i.e.*, 18%). Conversely, systems supporting SPARQL queries focus on graph-based sources (20 systems out of 24, *i.e.*, 83%) but support relational sources more frequently (10 out of 24, *i.e.*, 42%) than SQL systems do with graph-based sources.

Federation techniques. Besides the supported query languages and data sources, we also investigated the techniques used by each of the selected systems for federated query answering. We mainly focused our attention on the metadata catalog, source selection and partition, and optimization and plan generation. Information about these technical aspects is often covered scarcely or not covered at all in systems' documentation (especially for closed-source industrial systems), hence its collection for all the considered systems results not feasible in general. Instead, we focus on identifying and exemplifying the alternative approaches implemented by the selected systems, as emerging from their available documentation. Our results are summarized in Table 3, which aims at capturing the aspects of interest for researchers when developing innovative federated query answering strategies.

1. Most of the systems use self-defined dialects, such as the virtual databases of Teiid and the RDF molecule template of Ontario, to describe the metadata of the data sources participating in the federation. Few systems, like Squerall for instance, adopt standard languages, such as the Semantic Web RML and R2RML [108] mapping languages, or the VoID [96] vocabulary for Linked Data [109] datasets. Very few systems, mainly Apache Drill and RDF4J (FedX), do not require any predefined metadata catalog.
2. Index-based approaches are popular in identifying data sources for the atomic components of the input query, such as basic triple patterns in SPARQL queries and tables in SQL queries. Few systems, like Ontario and HIBISCuS, take rules to identify the data sources for a component of the input query. Only a small minority of the systems identify the sources via evaluating queries, such as RDF4J (FedX), which evaluates SPARQL ASK queries over SPARQL endpoints to check whether an endpoint contains data satisfying a given triple pattern. Moreover, *push down* is a strategy adopted by nearly all of the systems to generate sub-queries with the motivation of pushing as much computation to data sources as possible, so as to reduce intermediate results and overloading at the data federation system level.
3. Cost-based models usually combined with dynamic programming are adopted by most of the systems to compute low cost query plans. Some systems, like Teiid, enhance cost-based models with rules to improve their overall performance. There are also systems which fully rely on rule-based models. Additionally, *bind*, *nested loop*, and *hash joins* are adopted by most systems to merge the results of the partitioned sub-queries, as well as *broadcast*, *merge*, and *optional joins*. Finally, caching the results of specific queries, as well as evaluating sub-queries over different data sources in parallel, are widely used strategies to improve the overall query answering performance.

6.2. Data security dimension

We evaluate here the data security dimension. The concrete investigation results are shown in Table 4, organized according to the sub-dimensions of authentication, authorization, auditing, encryption, and data masking. In particular, by analyzing the information we synthesized in the table, the following can be observed:

1. All of the considered 31 industrial systems provide security mechanisms, such as authentication and authorization, to protect against unauthorized data access and leaking. This shows that the importance of data security is actually recognized by system providers in the data federation setting, where integrating multiple data sources via a unified virtual layer has the potential of making the private and sensitive data contained in federated sources more likely to be revealed.
2. Among the inspected mechanisms, authentication and authorization are definitely the most frequently adopted ones (see total counts in Table 4) and are implemented by almost all the industrial systems to identify users and control their access to data. For example, the Denodo Platform supports role-based authentication[14] and enforces strict and fine-grained row and column level access control.
3. Besides authentication and authorization, the other three mechanisms, i.e., auditing, encryption, and data masking, are adopted by some industrial (only) systems to enhance security by auditing the actions of users and encoding and hiding sensitive information. Take again Denodo as an example. The Denodo Platform provides an audit trail of all the information about the queries and other actions executed on the system. It also supports the application of strategies on a per-view basis to guarantee secure access to sensitive data through encryption/decryption at different levels, and it masks (hides) sensitive data to ensure they are not accessed by unauthorized users.
4. Data security has rarely been mentioned in the systems developed by academic and research institutions. Among the 17 systems we have evaluated in this category, just one system, *i.e.*, SAFE, takes data security into consideration. SAFE is a SPARQL query federation engine that enables policy-aware access to sensitive, distributed statistical data sources represented as RDF data cubes.

6.3. Interface dimension

Table 5 reports on the evaluation of the interface dimension, which is used to qualitatively evaluate the usability of the systems from both the end-user and the developer perspectives. As mentioned in Section 4 and reflected in the table, the interface dimension comprises the graphical, command line, and application programming interface sub-dimensions. Here, we focus on analyzing which of these interfaces are made available to the users, further identifying the different types of exposed application programming interfaces (*e.g.*, JDBC drivers, Web APIs). Systems not associated to any interface in the table are typically distributed as libraries whose intended use is to be embedded/extended as part of a larger system. We do not consider effectiveness and ease of use, whose evaluation is largely subjective as, for any given interface, user experience is affected by individual user's preferences and habits. In summary, from Table 5 we can derive the following observations:

1. Nearly all of the industrial systems (30 out of 31, *i.e.*, 97%) provide graphical interfaces, which consist mainly in web consoles or web interfaces, and command-line interfaces (all 31 industrial systems), which are usually exposed to help users to deploy and manage data federation instances. For example, AllegroGraph provides the AllegroGraph Web View,[15] which is a browser-based graphical interface for exploring, querying, and managing AllegroGraph databases, and Teiid provides users with Teiid Console,[16] a web-based administration and monitoring tool.

[14] https://community.denodo.com/kb/view/document/Denodo%20Security%20Overview
[15] https://allegrograph.com/products/agwebview/
[16] http://teiid.github.io/teiid-documents/16.0.x/content/admin/Teiid_Console.html

Table 4

Evaluation of the data security dimension. Academic systems in *italics*. "–" denotes feature/information not found in the systems' official documentation, websites, or academic publications, to the best of our efforts.

System	Data security				
	Authentication	Authorization	Auditing	Encryption	Data masking
AllegroGraph	✔	✔	–	–	–
Amazon Athena	✔	✔	✔	✔	–
Amazon Neptune	✔	✔	✔	✔	–
AnzoGraph DB	✔	✔	–	–	–
Apache Drill	✔	✔	–	✔	–
Apache Jena (ARQ)	✔	–	–	–	–
Apache Spark	✔	✔	–	✔	–
BigDAWG	–	–	–	–	–
CloudMdsQL	–	–	–	–	–
CostFed	–	–	–	–	–
DARQ	–	–	–	–	–
Data Virtuality	✔	✔	–	–	–
Denodo	✔	✔	✔	✔	✔
Dremio	✔	✔	–	✔	✔
FEDRA	–	–	–	–	–
FedX (RDF4J)	✔	–	–	–	–
GraphDB	✔	✔	✔	✔	–
HiBISCuS	–	–	–	–	–
IBM Cloud Pak for Data	✔	✔	✔	✔	✔
IBM Db2 Big SQL	✔	✔	✔	✔	–
IBM InfoSphere Federation Server	✔	✔	–	✔	–
JBoss Data Virtualization	✔	✔	✔	✔	–
Lusail	–	–	–	–	–
Metaphactory	✔	✔	–	–	–
Myria	–	–	–	–	–
Neo4j (Fabric)	✔	✔	–	–	–
Obi-Wan	–	–	–	–	–
Odyssey	–	–	–	–	–
Ontario	–	–	–	–	–
Onto-KIT	–	–	–	–	–
Oracle Big Data SQL	✔	✔	–	–	–
Oracle DB (Spatial & Graph)	✔	✔	–	✔	✔
PolyWeb	–	–	–	–	–
Presto	✔	✔	✔	–	–
Querona Data Virtualization	✔	✔	–	✔	✔
SAFE	✔	✔	–	–	–
SAP HANA	✔	–	–	–	–
SAS Federation Server	✔	✔	–	✔	✔
SemaGrow	–	–	–	–	–
SPLENDID	–	–	–	–	–
SQL Server (PolyBase)	✔	✔	✔	✔	–
Squerall	–	–	–	–	–
Starburst	✔	✔	✔	✔	–
Stardog	✔	✔	–	–	–
Teiid	✔	✔	–	✔	–
TIBCO Data Virtualization	✔	✔	–	✔	–
Trino	✔	✔	–	✔	–
Virtuoso	✔	✔	–	✔	–
Number	**32**	**29**	**10**	**20**	**6**

Table 5

Evaluation of the *interface* dimension. Academic systems in *italics*. "–" denotes feature/information not found in the systems' official documentation, websites, or academic publications, to the best of our efforts.

System	Graphical interface	Command line interface	Application programming interface				
			JDBC Driver	ODBC Driver	Web API	ADO.NET	SPARQL HTTP API
AllegroGraph	✔	✔	–	–	✔	–	✔
Amazon Athena	✔	✔	✔	✔	–	–	–
Amazon Neptune	✔	✔	✔	–	✔	–	–
AnzoGraph DB	✔	✔	–	–	✔	–	✔
Apache Drill	✔	✔	✔	✔	✔	–	–
Apache Jena (ARQ)	–	✔	✔	–	–	–	✔
Apache Spark	✔	✔	✔	✔	–	–	–
BigDAWG	–	–	–	–	✔	–	–
CloudMdsQL	–	–	–	–	–	–	–
CostFed	–	–	–	–	–	–	–
DARQ	–	–	–	–	–	–	–
Data Virtuality	✔	✔	✔	✔	✔	–	–
Denodo	✔	✔	✔	✔	✔	✔	–
Dremio	✔	✔	✔	✔	✔	–	–
FEDRA	–	–	–	–	–	–	–
FedX (RDF4J)	✔	✔	–	–	✔	–	✔
GraphDB	✔	✔	✔	–	✔	–	✔
HiBISCuS	–	–	–	–	–	–	–
IBM Cloud Pak for Data	✔	✔	–	–	✔	–	–
IBM Db2 Big SQL	✔	✔	✔	✔	–	–	–
IBM InfoSphere Federation Server	✔	✔	✔	–	✔	–	–
JBoss Data Virtualization	✔	✔	✔	✔	✔	–	–
Lusail	–	–	–	–	–	–	–
Metaphactory	✔	✔	–	–	✔	–	✔
Myria	✔	✔	–	–	✔	–	–
Neo4j (Fabric)	✔	✔	✔	–	✔	–	–
Obi-Wan	–	–	–	–	–	–	–
Odyssey	–	–	–	–	–	–	–
Ontario	–	–	–	–	–	–	–
Onto-KIT	–	–	–	–	–	–	–
Oracle Big Data SQL	✔	✔	–	–	–	–	–
Oracle DB (Spatial & Graph)	✔	✔	–	–	✔	–	✔
PolyWeb	–	–	–	–	–	–	–
Presto	✔	✔	✔	✔	✔	–	–
Querona Data Virtualization	✔	✔	✔	✔	–	✔	–
SAFE	–	–	–	–	–	–	–
SAP HANA	✔	✔	✔	✔	✔	✔	–
SAS Federation Server	✔	✔	✔	✔	✔	–	–
SemaGrow	–	–	–	–	–	–	–
SPLENDID	–	–	–	–	–	–	–
SQL Server (PolyBase)	✔	✔	✔	✔	–	✔	–
Squerall	–	–	–	–	–	–	–
Starburst	✔	✔	✔	✔	✔	–	–
Stardog	✔	✔	–	–	✔	–	✔
Teiid	✔	✔	✔	✔	✔	✔	–
TIBCO Data Virtualization	✔	✔	✔	✔	✔	✔	–
Trino	✔	✔	✔	✔	✔	–	–
Virtuoso	✔	✔	✔	✔	✔	✔	✔
Number	**31**	**32**	**23**	**18**	**26**	**7**	**9**

2. Besides graphical and command-line interfaces, most industrial systems like Denodo and Teiid also provide JDBC and ODBC drivers (respectively, 23 and 18 systems out of 31, *i.e.*, 74% and 58%) to enable users to access and interact with them as standard relational sources. Web APIs (mainly RESTful) are also very frequent among industrial systems (25 out of 31, *i.e.*, 81%), while there is less support for ADO.NET and the SPARQL HTTP API. The latter is exclusively provided by systems supporting the SPARQL query language (see Table 2) that also directly implement the associated SPARQL HTTP query protocol (instead of relying on other non-standard means for receiving a SPARQL query and returning its results). Furthermore, few systems, such as AllegroGraph, Presto and Stardog provide also multiple client libraries to facilitate users in interfacing with these systems programmatically via the most popular programming languages, like C, Go, Java, Python, R, and Ruby.

6.4. Development dimension

Table 6 reports on the evaluation of the development dimension and its sub-dimensions, which all together deliver information relevant to developers for integrating the system with other applications or for patching, extending, or otherwise modifying the system itself, if possible. Note that for the industrial systems, the information of the first release, *i.e.*, the year and version number of the first version made available, is actually the information of the oldest versions we have been able to gather from their official websites. Note also that the academic ones often do not follow well-defined release cycles with proper versioning, e.g., CostFed[17] and Lusail[18]. In such situations, we leave their versions as blank, and fill the years from their commit histories on their GitHub projects. The following are the main insights we can get from Table 6:

1. Java is the most used programming language, for both industrial and academic systems (see total counts in Table 6). There exist few systems, such as AnzoGraph DB and SAP HANA, that make use of C/C++, while only Squerall adopts Python, and Apache Spark and Ontario rely on Scala.
2. Among the industrial systems, the majority are closed source (21 out of 31, *i.e.*, 68%), and most of these come with commercial support services (19 systems out of 21, *i.e.*, 90%). Similarly, most of the open source industrial systems offer the option of commercial support (7 systems out of 10, *i.e.*, 70%). Academic systems are all open source without commercial support.
3. In comparison with academic systems, it is easy to see that industrial ones typically feature a much more active development. Some of these industrial systems have been developed, maintained, and improved for many years, such as Denodo and Teiid. Unfortunately, for the academic systems, despite the fact that all of them are open source initiatives, it is common that they are not enhanced or maintained after the publication of the respective academic papers.

6.5. Overall discussion and analysis

Based on the above reported evaluation and analysis, and after having reviewed the official documentation and academic publications of each of the systems considered in this survey, in the following we summarize the most crucial and interesting lessons we learned.

Background theory and standards. Data federation, especially over heterogeneous data sources, is currently a very active field in both industry and academia. However, the overall development of data federation systems still seems to lack background theory and standards. Let us note, for instance, that different systems force users to adopt their own dialects to develop and model the logical or meta-data layer of the target data sources. This strategy drastically hinders information reuse, in particular the information produced for one system cannot be directly used in other systems. In addition to that, when relying on new systems, users cannot leverage their acquired technical background and instead have to learn new dialects, both for metadata modeling and for query formulation. Most importantly, we know that the choice to not adopt already existing standards, makes it very hard to evaluate and compare the

[17] https://github.com/dice-group/CostFed
[18] https://github.com/Lusail/lusail

Table 6

Evaluation of *development* dimension. Academic systems in *italics*. “F." and “L." denote “First" and “Latest" respectively. “–” denotes feature/information not found in the systems’ official documentation, websites, or academic publications, to the best of our efforts.

System	Main development language			Commercial support	Open source	Release			
	C/C++	Java	Others			F. Version	F. Year	L. Version	L. Year
AllegroGraph	–	✔	Lisp	✔	–	v6.4.0	2004	v7.2.0	2021
Amazon Athena	–	✔	–	✔	–	–	2017	–	2021
Amazon Neptune	–	✔	–	✔	–	v1.0.1.0	2018	v1.0.5.1	2021
AnzoGraph DB	✔	–	–	✔	–	v2.0	-	v2.3	2021
Apache Drill	–	✔	–	–	✔	vM1	2012	v1.19	2021
Apache Jena (ARQ)	–	✔	–	–	✔	v2.7.0	2012	v4.2.0	2021
Apache Spark	–	–	Scala	–	✔	v1.0	2014	v3.2.1	2021
BigDAWG	–	✔	–	–	✔	–	2015	v0.0.5	2017
CloudMdsQL	–	✔	–	–	✔	–	2017	–	2017
CostFed	–	✔	–	–	✔	–	2016	–	2018
DARQ	–	✔	–	–	✔	–	2006	–	2008
Data Virtuality	–	–	–	✔	–	–	–	v2.4	2021
Denodo	–	–	–	✔	–	v1.0	2002	v8.0	2020
Dremio	–	✔	–	✔	✔	v1.1	2017	v19.0	2021
FEDRA	–	✔	–	–	✔	–	2015	–	2015
FedX (RDF4J)	–	✔	–	✔	✔	–	2011	v3.7.4	2021
GraphDB	–	✔	–	✔	–	v6.2	2015	v9.10	2021
HiBISCuS	–	✔	–	–	✔	v1	2014	v1	2014
IBM Cloud Pak for Data	–	–	–	✔	–	v2.1.0	2018	v4.0	2021
IBM Db2 Big SQL	–	✔	–	✔	–	–	2017	v7.1.0	2020
IBM InfoSphere Federation Server	–	–	–	✔	–	–	–	v10.5.0	2019
JBoss Data Virtualization	–	✔	–	✔	✔	v6.0.0	2014	v6.4.0	2018
Lusail	–	✔	–	–	✔	v1	2017	v1	2019
Metaphactory	–	–	–	–	–	–	2015	v4.3.0	2021
Myria	–	✔	–	–	✔	v1	2014	v1	2017
Neo4j (Fabric)	–	✔	–	✔	✔	v4.0.11	2020	v4.3.7	2021
Obi-Wan	–	✔	–	–	–	–	2020	–	2020
Odyssey	–	✔	–	–	✔	–	2016	–	2019
Ontario	–	–	Scala	–	✔	–	2018	–	2021
Onto-KIT	–	✔	–	–	✔	–	2020	–	2020
Oracle Big Data SQL	–	–	–	–	–	–	–	–	–
Oracle DB (Spatial & Graph)	–	–	–	✔	–	–	2016	v21c	2021
PolyWeb	–	✔	–	–	✔	–	2017	–	2017
Presto	–	✔	–	✔	✔	v0.54	2013	v0.265.1	2021
Querona Data Virtualization	–	–	–	✔	–	–	2015	–	2020
SAFE	–	✔	–	–	✔	v-	2017	v-	2017
SAP HANA	✔	–	–	✔	–	v1.0.SPS12	2018	v2.0.SPS05	2020
SAS Federation Server	✔	–	–	✔	–	v3.2	2013	v4.4	2021
SemaGrow	–	✔	–	–	✔	v1.0	2014	v2.2.1	2021
SPLENDID	–	✔	–	–	✔	–	2011	–	2011
SQL Server (PolyBase)	✔	–	–	✔	–	v2016	2016	v2019	2019
Squerall	–	–	Python	–	✔	v0.1	2018	v0.2	2019
Starburst	–	✔	–	✔	–	v0.188-e	2019	v364-e LTS	2021
Stardog	–	✔	–	✔	–	v0.7.3	2011	v7.7.3	2021
Teiid	–	✔	–	✔	✔	v6.0.0	2009	v16.0.0	2020
TIBCO Data Virtualization	–	–	–	✔	–	v7.0.5	2007	v8.4.0	2021
Trino	–	✔	–	✔	✔	v0.54	–	v364	2021
Virtuoso	✔	–	–	✔	–	–	–	v8.3	2020
Number	**5**	**31**	**4**	**26**	**26**				

performance of different systems in a fair way: for a given data federation system and a set of data sources, in fact, there may exist multiple, possibly incompatible and performance-impacting, ways to model the metadata layer, hence an in-depth knowledge of each system and its specific dialect is needed to achieve its full performance in a given evaluation setting.

Data modification and data quality dimensions. Besides the dimensions of our framework here analyzed in depth, we have also considered the capabilities in *data modification* and *data quality* of the compared systems. There exist few systems that support data modification over the federated data sources, such as Teiid and Denodo supporting[19] `INSERT` and `DELETE` operators, and RDF4J (FedX) supporting[20] SPARQL `UPDATE` over the SPARQL endpoints participating in the federation. Even if we found systems supporting SPARQL `UPDATE` or SQL DML, it is unclear from the systems' documentation whether these updates can be performed on the data sources in the federation, rather than on data stored locally by the system itself (e.g., for database systems extended with federation facilities). On the other hand, data quality represents an aspect that seems scarcely considered. Of all the systems we selected, only 3 mention data quality. *Taking it by and large*, in comparison with supporting more data sources, the aspects of data modification, data quality, and partly (from a research perspective) data security have been examined less thoroughly in the data federation scenario, despite the fact that the continued explosion of data scale and variety makes these aspects more important than ever.

Interrelationships between data sources. Most of the time, the data sources that are subject to a data integration initiative are not fully independent from each other. Indeed, there may exist interrelationships among the integrated data sources, such as information overlapping, complementarity, and conflicts. Automatically discovering such interrelationships may help developing data federation systems with higher efficiency. As a simple example, if a data source S_1 is part of a data source S_2 with respect to the metadata layer (both schema and content), then in the query evaluation procedure S_1 may be sometimes ignored (e.g., when querying for the union of the content of S_1 and S_2) and the overall performance improved. However, the current methods and systems are usually limited to virtually accumulating all the considered data sources, while ignoring the relationships among them.

Ontology-based data integration. Ontologies, providing a shared abstraction of a domain of interest, play a key role in handling the heterogeneity of concepts in data integration. The so-called *Ontology-Based Data Access and Data Integration* (OBDA/I) approach has been studied intensively in the last two decades, but mostly for relational sources [34–38, 110–112]. However, ontology-based integration of heterogeneous data sources in a virtual way has rarely been discussed and still represents an open research line. To the best of our knowledge, there exists only one system, namely Obi-Wan [80, 81], that integrates heterogeneous data sources based on an ontology (which is expressed in RDFS). Obi-Wan adopts the classical framework of OBDA/I by using the mediator system Tatooine [82] to realize query answering over multiple and heterogeneous data sources.Using domain ontologies to *virtually* integrate heterogeneous data sources combines the difficulties of ontology reasoning with the ones of integrating heterogeneous data, and this negatively affects performance. Further investigations and possibly innovative approaches are required to obtain systems that would exhibit a performance that is adequate to real-world application needs. The use of ontology-based techniques — and, more generally, of Semantic Web methods and standards — to address data quality, update, and security aspects of data federation systems also appears promising and deserves further research.

7. Related work

In this survey, we have investigated and analyzed a total of 48 data federation systems. Considering data federation in the broader context of data integration, in the following we situate this survey among other works in the Database and the Semantic Web literature that survey existing approaches, techniques, and systems for both virtual and materialized data integration.

[19] https://community.denodo.com/docs/html/browse/7.0/vdp/vql/inserts_updates_and_deletes_over_views/inserts_updates_and_deletes_over_views

[20] https://rdf4j.org/documentation/programming/federation/

Database community. The authors of [6] discuss some of the most important results in the data integration field before 2006, and outline some challenges for data integration research. The survey in [113] reports on the techniques for managing uncertainty in data integration, and the survey in [114] investigates the approaches focusing on semi-structured data. Finally, the works in [115–117] mostly address the issues emerging when techniques and systems are meant to be applied to integrate big data.

Readers that are interested in knowing more about existing approaches and implemented systems for integrating data virtually can refer to [47, 48, 118, 119]. In particular, the survey in [118] discusses data federation systems. The authors first define a "reference architecture" for distributed database management systems with the main aim of providing a framework in which to understand, categorize, and compare different architectural options for developing federated database systems. Additionally, they introduce a methodology for developing tightly coupled federated database systems with multiple federations and processors (that is, software modules that manipulate commands and data). The authors of [119] investigate multistore systems by first introducing the currently available cloud data management and query processing solutions, then describing and analyzing some representative multistore systems according to their architecture, data model, query languages, and query processing techniques. They finally classify these systems into three categories, *i.e.*, loosely-coupled, tightly-coupled, and hybrid. The survey in [48] focuses on query processing over heterogeneous data sources by first introducing a taxonomy that categorizes the solutions into data federation systems, polyglot systems, multistore systems, and polystore systems. On top of this categorization, the authors propose an evaluation framework, largely inspired by [118], incorporating the axes of "Heterogeneity", "Autonomy", "Transparency", "Flexibility" and "Optimality". The survey finally compares and analyzes four specific systems — BigDAWG, CloudMdsQL, Myria, and Apache Drill — according to the introduced evaluation framework. The work in [47] focuses on new generation data federation systems addressing the manipulation of structured and unstructured data, usually in high volume, over distributed and heterogeneous data sources. The authors first survey the literature aiming at giving an overview of state-of-the-art modern data federation systems and then analyze the four aforementioned systems — BigDAWG, CloudMdsQL, Myria, and Apache Drill — by reporting on their "Definition", "Owners", "Goals", "Query Specification and Execution", "Main Components", and other significant dimensions.

Semantic Web community. The works in [120–122] provide general surveys of those solutions for integrating data that are based on Semantic Web technologies and that follow the so-called OBDA/I approach. Other works concentrate instead on specific subdomains in which semantic technologies have been applied to integrate data. In particular, the authors of [123] focus on analyzing and comparing the existing approaches for ontology-driven geographic information integration. An investigation of the approaches and techniques for integrating biological data developed in the ontology community is presented in [124]. The survey in [125] investigates the works that have been done in the area of Linked Data integration, covering both materialized and virtual integration approaches. This work provides a concise overview of the issues, methods, tools, and systems for semantic integration of data, and gives emphasis on the methods that provide support for the integration of large numbers of datasets.

As for the virtual approach to data integration, some literature can be found [42–46] surveying, in particular, approaches and systems for federated SPARQL query answering. To summarize, the survey in [42] gives an overview of SPARQL federation frameworks — *i.e.*, frameworks supporting *(i)* SPARQL 1.1 federation extension, *(ii)* federation over SPARQL 1.0 endpoints, and *(iii)* federation over SPARQL 1.1 endpoints — and classifies and analyzes 14 existing SPARQL federation approaches. The authors of [43] evaluate 7 federation engines by first providing a detailed and clear insight on data source selection, join, and query optimization methods. They also introduce a qualitative comparison of these engines according to the following criteria: "No Preprocessing per Query", "Unbound Predicate Queries", "Parallelization", and "Adaptive Query Processing". The work in [44] provides an overview of current challenges and opportunities of federated query processing as well as summarizes the results of recent state-of-the-art studies. In [45], the authors first provide a survey of 14 federated SPARQL query engines according to: "Code Availability", "Implementation Language", "Licensing", "Source Selection Type", "Join Type", "Cache", and "Index/Catalog Update". They then compare 5 SPARQL endpoint federation systems by using the performance evaluation framework FedBench [126] and by considering the dimensions of query runtime, number of sources selected, total number of SPARQL ASK requests used, completeness of answers, and source selection time. Finally, the work in [46] first proposes some metrics to measure the errors in cardinality estimations of cost-based federation

engines and the correlation of the values of these metrics with the overall query runtimes. It then presents an empirical evaluation of 5 cost-based SPARQL federation engines on LargeRDFBench [127] according to the proposed metrics.

Comparison. The key difference between our work and the aforementioned surveys is mainly reflected in the following two aspects. First, we have analyzed and investigated a larger number of systems, including among them both industrial and academic initiatives and systems adopting different data models, *i.e.*, SQL-based and SPARQL-based. Second, we have introduced here as a novel contribution a framework to inspect, analyze, and then classify the main characteristics of each system. The framework has been developed by taking into consideration the requirements of the end-users, as well as those of the developers and of the scholars, this way trying to deliver the information that they need when making choices for their respective data federation activities and projects. Our main motivation is to assess the techniques and capabilities of the existing systems for data federation, so as to reveal their strengths and weaknesses in relation to the plurality of evaluation dimensions we consider, rather than classifying the systems along one single dimension or according to the requirements of one single category of prototypical users.

8. Concluding remarks and future work

In this paper, we provided a systematic overview of 48 data federation systems, with the motivation of evaluating their capabilities as well as the strengths and weaknesses of the employed techniques for integrating heterogeneous data sources uniformly and virtually. To do so, we have proposed a framework with four major dimensions and additional sub-dimensions to classify systems from the end-user, the developer, and the scholar perspectives, in a uniform and qualitative way. We think that the evaluation framework we have proposed can be extremely valuable for all these target personas: it helps end-users in finding the system that most suits their application requirements and, at the same time, it drives decision making by developers and researchers in further improving the currently available solutions and in designing more powerful federation systems. Besides that, our work also aims at providing up-to-date reference information for all those interested in dipping their toes in the data federation water.

Integrating and managing heterogeneous data "uniformly and virtually" still have a long way to go both at the theoretical and at the practical application levels. Our future work will mainly focus on the following two aspects. In our current evaluation, efficiency of the investigated systems remains an ignored dimension. Therefore, one direction for future work is to design extensive experiments to evaluate the performance and assess the restrictions of each system in integrating and managing heterogeneous data virtually. On the other hand, it is well known that the Semantic Web provides standards for both knowledge and data representation and management. However, integrating heterogeneous data virtually by relying on semantic technologies and Semantic Web standards still represents an open and promising research field. The second main direction we want to take is indeed to develop innovative approaches for ontology-based heterogeneous data integration and management, covering federated query answering, data updates, security, and data quality assurance, where automated logic-based reasoning techniques play a central role.

Acknowledgements

This research has been partially supported by the EU H2020 project INODE (grant agreement No. 863410), by the Italian PRIN project HOPE (2019-2022), by the European Regional Development Fund (ERDF) Investment for Growth and Jobs Programme 2014-2020 through the project IDEE (FESR1133), by the Free University of Bozen-Bolzano through the project MP4OBDA, and by the "Fusion Grant" project HIVE sponsored by Fondazione Cassa di Risparmio di Bolzano and Ontopic s.r.l. in coordination with NOI Techpark, Südtiroler Wirtschaftsring and Rete Economia Alto Adige. D. Calvanese is supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. We thank our colleagues, in particular Julien Corman, for their discussions and feedback.

References

[1] D. Reinsel, J. Gantz and J. Rydning, The digitization of the world from edge to core, Technical Report, International Data Corporation, Framingham, MA, 2018.

[2] A. Labrinidis and H.V. Jagadish, Challenges and Opportunities with Big Data, *Proc. of VLDB Endowment* **5**(12) (2012), 2032–2033. doi:10.14778/2367502.2367572.

[3] S. Sagiroglu and D. Sinanc, Big data: A review, in: *Proc. of Int. Conf. on Collaboration Technologies and Systems (CTS)*, IEEE, 2013, pp. 42–47. doi:10.1109/CTS.2013.6567202.

[4] M. Lenzerini, Data Integration: A Theoretical Perspective, in: *Proc. of ACM Symp. on Principles of Database Systems (PODS)*, ACM, 2002, pp. 233–246. doi:10.1145/543613.543644.

[5] A. Doan, A.Y. Halevy and Z.G. Ives, *Principles of Data Integration*, Morgan Kaufmann, 2012. ISBN 978-0-12-416044-6. doi:10.1016/C2011-0-06130-6.

[6] A.Y. Halevy, A. Rajaraman and J.J. Ordille, Data Integration: The Teenage Years, in: *Proc. of Int. Conf. on Very Large Data Bases (VLDB)*, ACM, 2006, pp. 9–16.

[7] J. Widom, Research Problems in Data Warehousing, in: *Proc. of Int. Conf. on Information and Knowledge Management (CIKM)*, ACM, 1995, pp. 25–30. doi:10.1145/221270.221319.

[8] S. Chaudhuri and U. Dayal, An Overview of Data Warehousing and OLAP Technology, *SIGMOD Record* **26**(1) (1997), 65–74. doi:10.1145/248603.248616.

[9] A.P. Sheth and J.A. Larson, Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases, *ACM Computing Surveys* **22**(3) (1990), 183–236. doi:10.1145/96602.96604.

[10] L.M. Haas, E.T. Lin and M.T. Roth, Data integration through database federation, *IBM Systems J.* **41**(4) (2002), 578–596. doi:10.1147/sj.414.0578.

[11] C.J. Date and H. Darwen, *A Guide to the SQL Standard*, 4th edn, Addison-Wesley, 1996.

[12] S. Harris and A. Seaborne, SPARQL 1.1 Query Language, W3C Recommendation, W3C, 2013. http://www.w3.org/TR/2013/REC-sparql11-query-20130321/.

[13] M. Lanthaler, R. Cyganiak and D. Wood, RDF 1.1 Concepts and Abstract Syntax, W3C Recommendation, W3C, 2014. http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/.

[14] D. Brickley and R. Guha, RDF Schema 1.1, W3C Recommendation, W3C, 2014. http://www.w3.org/TR/2014/REC-rdf-schema-20140225/.

[15] M. Krötzsch, P. Patel-Schneider, S. Rudolph, B. Parsia and P. Hitzler, OWL 2 Web Ontology Language Primer (Second Edition), W3C Recommendation, W3C, 2012. https://www.w3.org/TR/2012/REC-owl2-primer-20121211/.

[16] R. van der Lans, *Data Virtualization for Business Intelligence Systems: Revolutionizing Data Integration for Data Warehouses*, 1st edn, Morgan Kaufmann Publishers, San Francisco, CA, USA, 2012. ISBN 0123944252.

[17] A. Bogdanov, A. Degtyarev, N. Shchegoleva, V. Korkhov and V. Khvatov, Big Data Virtualization: Why and How?, in: *Proc. of 4th Int. Workshop on Data Life Cycle in Physics (DLC)*, CEUR Workshop Proceedings, Vol. 2679, 2020, pp. 11–21.

[18] A. Schwarte, P. Haase, K. Hose, R. Schenkel and M. Schmidt, FedX: A Federation Layer for Distributed Query Processing on Linked Open Data, in: *Proc. of Extended Semantic Web Conference (ESWC)*, LNCS, Vol. 6644, Springer, 2011, pp. 481–486. doi:10.1007/978-3-642-21064-8_39.

[19] A. Schwarte, P. Haase, K. Hose, R. Schenkel and M. Schmidt, FedX: Optimization Techniques for Federated Query Processing on Linked Data, in: *Proc. of Int. Semantic Web Conf. (ISWC)*, LNCS, Vol. 7031, Springer, 2011, pp. 601–616. doi:10.1007/978-3-642-25073-6_38.

[20] Teiid, Accessed 16 November 2021. https://teiid.io/.

[21] K. Clark, E. Torres, G. Williams and L. Feigenbaum, SPARQL 1.1 Protocol, W3C Recommendation, W3C, 2013. https://www.w3.org/TR/2013/REC-sparql11-protocol-20130321/.

[22] Data Virtuality, Accessed 17 November 2021. https://datavirtuality.com/.

[23] M.N.M. Nazri, S.A. Noah and Z. Hamid, Using Lexical Ontology for Semi-automatic Logical Data Warehouse Design, in: *Proc. of Int. Conf. on Rough Set and Knowledge Technology (RSKT)*, LNCS, Vol. 6401, Springer, 2010, pp. 257–264. doi:10.1007/978-3-642-16248-0_39.

[24] S. Bouarar, L. Bellatreche, S. Jean and M. Baron, Do Rule-Based Approaches Still Make Sense in Logical Data Warehouse Design?, in: *Proc. of East European Conf. on Advances in Databases and Information Systems (ADBIS)*, LNCS, Vol. 8716, Springer, 2014, pp. 83–96. doi:10.1007/978-3-319-10933-6_7.

[25] K.M. Endris, P.D. Rohde, M.-E. Vidal and S. Auer, Ontario: Federated Query Processing Against a Semantic Data Lake, in: *Proc. of Int. Conf. on Database and Expert Systems Applications (DEXA)*, LNCS, Vol. 11706, Springer, 2019, pp. 379–395. doi:10.1007/978-3-030-27615-7_29.

[26] F. Ravat and Y. Zhao, Data Lakes: Trends and Perspectives, in: *Proc. of Int. Conf. on Database and Expert Systems Applications (DEXA)*, LNCS, Vol. 11706, Springer, 2019, pp. 304–313. doi:10.1007/978-3-030-27615-7_23.

[27] R. Hai, S. Geisler and C. Quix, Constance: An Intelligent Data Lake System, in: *Proc. of ACM SIGMOD Int. Conf. on Management of Data (SIGMOD)*, ACM, 2016, pp. 2097–2100. doi:10.1145/2882903.2899389.

[28] R. Hai, C. Quix and C. Zhou, Query Rewriting for Heterogeneous Data Lakes, in: *Proc. of European Conf. on Advances in Databases and Information Systems (ADBIS)*, LNCS, Vol. 11019, Springer, 2018, pp. 35–49. doi:10.1007/978-3-319-98398-1_3.

[29] Y. Khan, A. Zimmermann, A. Jha, V. Gadepally, M. d'Aquin and R. Sahay, One Size Does Not Fit All: Querying Web Polystores, *IEEE Access* **7** (2019), 9598–9617. doi:10.1109/ACCESS.2018.2888601.
[30] J. Duggan, A.J. Elmore, M. Stonebraker, M. Balazinska, B. Howe, J. Kepner, S. Madden, D. Maier, T. Mattson and S.B. Zdonik, The BigDAWG Polystore System, *SIGMOD Record* **44**(2) (2015), 11–16. doi:10.1145/2814710.2814713.
[31] J. Wang, T. Baker, M. Balazinska, D. Halperin, B. Haynes, B. Howe, D. Hutchison, S. Jain, R. Maas, P. Mehta, D. Moritz, B. Myers, J. Ortiz, D. Suciu, A. Whitaker and S. Xu, The Myria Big Data Management and Analytics System and Cloud Services, in: *Proc. of Biennial Conf. on Innovative Data Systems Research (CIDR)*, www.cidrdb.org, 2017.
[32] B. Kolev, C. Bondiombouy, P. Valduriez, R. Jiménez-Peris, R. Pau and J. Pereira, The CloudMdsQL Multistore System, in: *Proc. of ACM SIGMOD Int. Conf. on Management of Data (SIGMOD)*, ACM, 2016, pp. 2113–2116. doi:10.1145/2882903.2899400.
[33] R. Alotaibi, B. Cautis, A. Deutsch, M. Latrache, I. Manolescu and Y. Yang, ESTOCADA: Towards Scalable Polystore Systems, *Proc. of VLDB Endowment* **13**(12) (2020), 2949–2952. doi:10.14778/3415478.3415516.
[34] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini and R. Rosati, Tractable Reasoning and Efficient Query Answering in Description Logics: The *DL-Lite* Family, *J. Automated Reasoning* **39**(3) (2007), 385–429. doi:10.1007/s10817-007-9078-x.
[35] D. Calvanese, B. Cogrel, S. Komla-Ebri, R. Kontchakov, D. Lanti, M. Rezk, M. Rodriguez-Muro and G. Xiao, Ontop: Answering SPARQL queries over relational databases, *Semantic Web* **8**(3) (2017), 471–487. doi:10.3233/SW-160217.
[36] G. Xiao, D. Lanti, R. Kontchakov, S. Komla-Ebri, E.G. Kalayci, L. Ding, J. Corman, B. Cogrel, D. Calvanese and E. Botoeva, The Virtual Knowledge Graph System Ontop, in: *Proc. of Int. Semantic Web Conf. (ISWC)*, LNCS, Vol. 12507, Springer, 2020, pp. 259–277. doi:10.1007/978-3-030-62466-8_17.
[37] G. Xiao, D. Calvanese, R. Kontchakov, D. Lembo, A. Poggi, R. Rosati and M. Zakharyaschev, Ontology-Based Data Access: A Survey, in: *Proc. of 27th Int. Joint Conf. on Artificial Intelligence (IJCAI)*, ijcai.org, 2018, pp. 5511–5519. doi:10.24963/ijcai.2018/777.
[38] G. Xiao, L. Ding, B. Cogrel and D. Calvanese, Virtual Knowledge Graphs: An Overview of Systems and Use Cases, *Data Intelligence* **1**(3) (2019), 201–223. doi:10.1162/dint_a_00011.
[39] E. Kharlamov, D. Hovland, M.G. Skjæveland, D. Bilidas, E. Jiménez-Ruiz, G. Xiao, A. Soylu, D. Lanti, M. Rezk, D. Zheleznyakov, M. Giese, H. Lie, Y.E. Ioannidis, Y. Kotidis, M. Koubarakis and A. Waaler, Ontology Based Data Access in Statoil, *J. Web Semant.* **44** (2017), 3–36. doi:10.1016/j.websem.2017.05.005.
[40] M. Giese, A. Soylu, G. Vega-Gorgojo, A. Waaler, P. Haase, E. Jiménez-Ruiz, D. Lanti, M. Rezk, G. Xiao, Ö.L. Özçep and R. Rosati, Optique: Zooming in on Big Data, *Computer* **48**(3) (2015), 60–67. doi:10.1109/MC.2015.82.
[41] D. Calvanese, M. Giese, P. Haase, I. Horrocks, T. Hubauer, Y. Ioannidis, E. Jiménez-Ruiz, E. Kharlamov, H. Kllapi, J. Klüwer et al., Optique: OBDA solution for big data, in: *Proc. of Extended Semantic Web Conf. (ESWC)*, Springer, 2013, pp. 293–295.
[42] N.A. Rakhmawati, J. Umbrich, M. Karnstedt, A. Hasnain and M. Hausenblas, A Comparison of Federation over SPARQL Endpoints Frameworks, in: *Proc. of 4th Int. Conf. on Knowledge Engineering and the Semantic Web (KESW)*, CCIS, Vol. 394, Springer, 2013, pp. 132–146. doi:10.1007/978-3-642-41360-5_11.
[43] D. Oguz, B. Ergenc, S. Yin, O. Dikenelli and A. Hameurlain, Federated query processing on Linked Data: a qualitative survey and open challenges, *Knowledge Engineering Review* **30**(5) (2015), 545–563. doi:10.1017/S0269888915000107.
[44] A.-C. Ngonga Ngomo and M. Saleem, Federated Query Processing: Challenges and Opportunities, in: *Proc. of Int. Workshop on Dataset Profiling and Federated Search for Linked Data (PROFILES)*, CEUR Workshop Proceedings, Vol. 1597, CEUR-WS.org, 2016.
[45] M. Saleem, Y. Khan, A. Hasnain, I. Ermilov and A.-C. Ngonga Ngomo, A fine-grained evaluation of SPARQL endpoint federation systems, *Semantic Web* **7**(5) (2016), 493–518. doi:10.3233/SW-150186.
[46] U. Qudus, M. Saleem, A.-C. Ngonga Ngomo and Y.-k. Lee, An Empirical Evaluation of Cost-based Federated SPARQL Query Processing Engines, *Semantic Web* **0**(1) (2019), 1–26. doi:10.3233/SW-200420.
[47] L.G. Azevedo, E.F. de Souza Soares, R. Souza and M.F. Moreno, Modern Federated Database Systems: An Overview, in: *Proc. of 22nd Int. Conf. on Enterprise Information Systems (ICEIS)*, SCITEPRESS, 2020, pp. 276–283. doi:10.5220/0009795402760283.
[48] R. Tan, R. Chirkova, V. Gadepally and T.G. Mattson, Enabling query processing across heterogeneous data models: A survey, in: *Proc. of Int. Conf. on Big Data (BigData)*, IEEE Computer Society, 2017, pp. 3211–3220. doi:10.1109/BigData.2017.8258302.
[49] Apache Drill, Accessed 18 November 2021. https://drill.apache.org/.
[50] B. Quilitz and U. Leser, Querying Distributed RDF Data Sources with SPARQL, in: *Proc. of European Semantic Web Conf. (ESWC)*, LNCS, Vol. 5021, Springer, 2008, pp. 524–538. doi:10.1007/978-3-540-68234-9_39.
[51] O. Görlitz and S. Staab, SPLENDID: SPARQL Endpoint Federation Exploiting VOID Descriptions, in: *Proc. of 2nd Int. Workshop on Consuming Linked Data (COLD)*, CEUR Workshop Proceedings, Vol. 782, CEUR-WS.org, 2011.
[52] M.N. Mami, D. Graux, S. Scerri, H. Jabeen, S. Auer and J. Lehmann, Squerall: Virtual Ontology-Based Access to Heterogeneous and Large Data Sources, in: *Proc. of Int. Semantic Web Conf. (ISWC)*, LNCS, Vol. 11779, Springer, 2019, pp. 229–245. doi:10.1007/978-3-030-30796-7_15.
[53] AllegroGraph, Accessed 18 November 2021. https://allegrograph.com/.
[54] Amazon Athena, Accessed 18 November 2021. https://docs.aws.amazon.com/athena/latest/ug/work-with-data-stores.html.
[55] Presto, Accessed 18 November 2021. https://prestodb.io/.
[56] Amazon Neptune, Accessed 18 November 2021. https://aws.amazon.com/neptune/.
[57] Anzograph, Accessed 17 November 2021. https://www.cambridgesemantics.com/anzograph/.
[58] M. Hausenblas and J. Nadeau, Apache Drill: Interactive Ad-Hoc Analysis at Scale, *Big Data* **1**(2) (2013), 100–104. doi:10.1089/big.2013.0011.
[59] S. Melnik, A. Gubarev, J.J. Long, G. Romer, S. Shivakumar, M. Tolton and T. Vassilakis, Dremel: interactive analysis of web-scale datasets, *Communications of the ACM* **54**(6) (2011), 114–123. doi:10.1145/1953122.1953148.

[60] Jena, Accessed 18 November 2021. https://jena.apache.org/documentation/query/.
[61] Spark SQL, Accessed 18 November 2021. https://spark.apache.org/sql/.
[62] M. Armbrust, R.S. Xin, C. Lian, Y. Huai, D. Liu, J.K. Bradley, X. Meng, T. Kaftan, M.J. Franklin, A. Ghodsi and M. Zaharia, Spark SQL: Relational Data Processing in Spark, in: *Proc. of ACM SIGMOD Int. Conf. on Management of Data (SIGMOD)*, ACM, 2015, pp. 1383–1394. doi:10.1145/2723372.2742797.
[63] V. Gadepally, K. O'Brien, A. Dziedzic, A.J. Elmore, J. Kepner, S. Madden, T. Mattson, J. Rogers, Z. She and M. Stonebraker, BigDAWG version 0.1, in: *Proc. of IEEE High Performance Extreme Computing Conf. (HPEC)*, IEEE, 2017, pp. 1–7. doi:10.1109/HPEC.2017.8091077.
[64] B. Kolev, P. Valduriez, C. Bondiombouy, R. Jiménez-Peris, R. Pau and J. Pereira, CloudMdsQL: querying heterogeneous cloud data stores with a common language, *Distributed Parallel Databases* **34**(4) (2016), 463–503. doi:10.1007/s10619-015-7185-y.
[65] M. Saleem, A. Potocki, T. Soru, O. Hartig and A.-C. Ngonga Ngomo, CostFed: Cost-Based Query Optimization for SPARQL Endpoint Federation, in: *Proc. of Int. Conf. on Semantic Systems (SEMANTICS)*, Procedia Computer Science, Vol. 137, Elsevier, 2018, pp. 163–174. doi:10.1016/j.procs.2018.09.016.
[66] Denodo, Accessed 17 Novemebr 2021. https://www.denodo.com/en.
[67] Dremio, Accessed 17 November 2021. https://www.dremio.com/.
[68] G. Montoya, H. Skaf-Molli, P. Molli and M.-E. Vidal, Federated SPARQL Queries Processing with Replicated Fragments, in: *Proc. of Int. Semantic Web Conf. (ISWC)*, LNCS, Vol. 9366, Springer, 2015, pp. 36–51. doi:10.1007/978-3-319-25007-6_3.
[69] GraphDB, Accessed 17 November 2021. https://graphdb.ontotext.com/.
[70] M. Saleem and A.-C. Ngonga Ngomo, HiBISCuS: Hypergraph-Based Source Selection for SPARQL Endpoint Federation, in: *Proc. of European Semantic Web Conf. (ESWC)*, LNCS, Vol. 8465, Springer, 2014, pp. 176–191. doi:10.1007/978-3-319-07443-6_13.
[71] IBM Cloud Pak for Data, Accessed 17 November 2021. https://www.ibm.com/products/cloud-pak-for-data.
[72] IBM Db2 Big SQL, Accessed 18 November 2021. https://www.ibm.com/products/db2-big-sql.
[73] IBM InfoSphere Federation Server, Accessed 18 November 2021. https://www.ibm.com/docs/en/iis/11.7?topic=components-infosphere-federation-server.
[74] JBoss Data Virtualization, Accessed 17 November 2021. https://developers.redhat.com/products/datavirt/overview.
[75] I. Abdelaziz, E. Mansour, M. Ouzzani, A. Aboulnaga and P. Kalnis, Lusail: A System for Querying Linked Data at Scale, *Proc. of VLDB Endowment* **11**(4) (2017), 485–498. doi:10.1145/3186728.3164144.
[76] Metaphactory, Accessed 18 November 2021. https://metaphacts.com/product.
[77] P. Haase, D.M. Herzig, A. Kozlov, A. Nikolov and J. Trame, metaphactory: A platform for knowledge graph management, *Semantic Web* **10**(6) (2019), 1109–1125. doi:10.3233/SW-190360.
[78] Neo4j, Accessed 17 November 2021. https://neo4j.com/.
[79] N. Francis, A. Green, P. Guagliardo, L. Libkin, T. Lindaaker, V. Marsault, S. Plantikow, M. Rydberg, P. Selmer and A. Taylor, Cypher: An Evolving Query Language for Property Graphs, in: *Proc. of ACM SIGMOD Int. Conf. on Management of Data (SIGMOD)*, ACM, 2018, pp. 1433–1445. doi:10.1145/3183713.3190657.
[80] M. Buron, F. Goasdoué, I. Manolescu and M.-L. Mugnier, Ontology-Based RDF Integration of Heterogeneous Data, in: *Proc. of 23rd Int. Conf. on Extending Database Technology (EDBT)*, OpenProceedings.org, 2020, pp. 299–310. doi:10.5441/002/edbt.2020.27.
[81] M. Buron, F. Goasdoué, I. Manolescu and M.-L. Mugnier, Obi-Wan: Ontology-Based RDF Integration of Heterogeneous Data, *Proc. of VLDB Endowment* **13**(12) (2020), 2933–2936. doi:10.14778/3415478.3415512.
[82] R. Bonaque, T.D. Cao, B. Cautis, F. Goasdoué, J. Letelier, I. Manolescu, O. Mendoza, S. Ribeiro, X. Tannier and M. Thomazo, Mixed-instance querying: a lightweight integration architecture for data journalism, *Proc. of VLDB Endowment* **9**(13) (2016), 1513–1516. doi:10.14778/3007263.3007297.
[83] G. Montoya, H. Skaf-Molli and K. Hose, The Odyssey Approach for Optimizing Federated SPARQL Queries, in: *Proc. of Int. Semantic Web Conf. (ISWC)*, LNCS, Vol. 10587, Springer, 2017, pp. 471–489. doi:10.1007/978-3-319-68288-4_28.
[84] K.M. Endris, M. Galkin, I. Lytra, M.N. Mami, M.-E. Vidal and S. Auer, Querying Interlinked Data by Bridging RDF Molecule Templates, *Trans. Large Scale Data Knowledge Centered Systems* **39** (2018), 1–42. doi:10.1007/978-3-662-58415-6_1.
[85] M. Masmoudi, S.B.A.B. Lamine, H.B. Zghal, B. Archimède and M.-H. Karray, Knowledge hypergraph-based approach for data integration and querying: Application to Earth Observation, *Future Generation Computer Systems* **115** (2021), 720–740. doi:10.1016/j.future.2020.09.029.
[86] Oracle Big Data SQL, Accessed 18 November 2021. https://www.oracle.com/database/technologies/datawarehouse-bigdata/bigdata-sql.html.
[87] Oracle Spatial and Graph, Accessed 16 November 2021. https://www.oracle.com/database/technologies/spatialandgraph.html.
[88] L. Jayapalan, Oracle Spatial and Graph RDF Knowledge Developer's Guide, Technical Report, Oracle, 2021. https://docs.oracle.com/en/database/oracle/oracle-database/19/rdfrm/spatial-and-graph-rdf-knowledge-graph-developers-guide.pdf.
[89] Y. Khan, A. Zimmermann, A. Jha, D. Rebholz-Schuhmann and R. Sahay, Querying web polystores, in: *Proc. of IEEE Int. Conf. on Big Data (IEEE BigData)*, IEEE Computer Society, 2017, pp. 3190–3195. doi:10.1109/BigData.2017.8258299.
[90] R. Sethi, M. Traverso, D. Sundstrom, D. Phillips, W. Xie, Y. Sun, N. Yegitbasi, H. Jin, E. Hwang, N. Shingte and C. Berner, Presto: SQL on Everything, in: *Proc. of 35th Int. Conf. on Data Engineering (ICDE)*, IEEE, 2019, pp. 1802–1813. doi:10.1109/ICDE.2019.00196.
[91] Querona Data Virtualization, Accessed 17 November 2021. https://www.querona.io/.
[92] Y. Khan, M. Saleem, M. Mehdi, A. Hogan, Q. Mehmood, D. Rebholz-Schuhmann and R. Sahay, SAFE: SPARQL Federation over RDF Data Cubes with Access Control, *J. Biomedical Semantics* **8**(1) (2017), 5:1–5:22. doi:10.1186/s13326-017-0112-6.

[93] SAP HANA (Smart Data Access), Accessed 18 November 2021. https://help.sap.com/viewer/6b94445c94ae495c83a19646e7c3fd56/1.0.12/en-US/a07c7ff25997460bbcb73099fb59007d.html.
[94] SAS Federation Server, Accessed 18 November 2021. https://support.sas.com/en/software/federation-server-support.html.
[95] A. Charalambidis, A. Troumpoukis and S. Konstantopoulos, SemaGrow: optimizing federated SPARQL queries, in: *Proc. of 11th Int. Conf. on Semantic Systems (SEMANTICS)*, ACM, 2015, pp. 121–128. doi:10.1145/2814864.2814886.
[96] K. Alexander, R. Cyganiak, M. Hausenblas and J. Zhao, Describing Linked Datasets, in: *Proc. of Int. Workshop on Linked Data on the Web (LDOW)*, CEUR Workshop Proceedings, Vol. 538, CEUR-WS.org, 2009.
[97] SQL Server (PolyBase), Accessed 18 November 2021. https://docs.microsoft.com/en-us/sql/relational-databases/polybase/polybase-guide?view=sql-server-ver15.
[98] Starburst, Accessed 18 December 2021. https://www.starburst.io/.
[99] Stardog, Accessed 17 November 2021. https://www.stardog.com/.
[100] TIBCO Data Virtualization, Accessed 17 November 2021. https://www.tibco.com/products/data-virtualization.
[101] Trino, Accessed 18 November 2021. https://trino.io/.
[102] Virtuoso, Accessed 17 November 2021. https://virtuoso.openlinksw.com/.
[103] O. Erling, Virtuoso, a Hybrid RDBMS/Graph Column Store, *IEEE Data Engineering Bull.* **35**(1) (2012), 3–8.
[104] O. Erling and I. Mikhailov, RDF Support in the Virtuoso DBMS, in: *Proc. of Conf. on Social Semantic Web (CSSW)*, LNI, Vol. P-113, GI, 2007, pp. 59–68.
[105] P.J. Sadalage and M. Fowler, *NoSQL Distilled: a Brief Guide to the Emerging World of Polyglot Persistence*, Pearson Education, 2013.
[106] DB-Engines, Accessed 16 February 2022. https://db-engines.com/en/.
[107] Database of Databases, Accessed 16 February 2022. https://dbdb.io/.
[108] S. Das, R. Cyganiak and S. Sundara, R2RML: RDB to RDF Mapping Language, W3C Recommendation, W3C, 2012. http://www.w3.org/TR/2012/REC-r2rml-20120927/.
[109] C. Bizer, T. Heath and T. Berners-Lee, Linked Data - The Story So Far, *Int. J. Semantic Web and Information Systems* **5**(3) (2009), 1–22. doi:10.4018/jswis.2009081901.
[110] A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini and R. Rosati, Linking Data to Ontologies, *J. Data Semantics* **10** (2008), 133–173. doi:10.1007/978-3-540-77688-8_5.
[111] C. Civili, M. Console, G. De Giacomo, D. Lembo, M. Lenzerini, L. Lepore, R. Mancini, A. Poggi, R. Rosati, M. Ruzzi, V. Santarelli and D.F. Savo, MASTRO STUDIO: Managing Ontology-Based Data Access Applications, *Proc. of VLDB Endowment* **6**(12) (2013), 1314–1317.
[112] D. Lanti, G. Xiao and D. Calvanese, Cost-Driven Ontology-Based Data Access, in: *Proc. of Int. Semantic Web Conf. (ISWC)*, LNCS, Vol. 10587, Springer, 2017, pp. 452–470. doi:10.1007/978-3-319-68288-4_27.
[113] M. Magnani and D. Montesi, A Survey on Uncertainty Management in Data Integration, *J. Data Information Quality* **2**(1) (2010), 5:1–5:33. doi:10.1145/1805286.1805291.
[114] N. Bikakis, C. Tsinaraki, N. Gioldasis, I. Stavrakantonakis and S. Christodoulakis, The XML and Semantic Web Worlds: Technologies, Interoperability and Integration: A Survey of the State of the Art, in: *Semantic Hyper/Multimedia Adaptation - Schemes and Applications*, SCI, Vol. 418, Springer, 2013, pp. 319–360. doi:10.1007/978-3-642-28977-4_12.
[115] B. Arputhamary and L. Arockiam, A review on Big Data Integration, *Int. J. Computer Applications* **22**(3) (2015), 21–26.
[116] X.L. Dong and D. Srivastava, *Big Data Integration*, Synthesis Lectures on Data Management, Morgan & Claypool Publishers, 2015. doi:10.2200/S00578ED1V01Y201404DTM040.
[117] J. Hui, L. Li and Z. Zhang, Integration of Big Data: A Survey, in: *Proc. of 4th Int. Conf. or Pioneering Computer Scientists, Engineers and Educators (ICPCSEE)*, CCIS, Vol. 901, Springer, 2018, pp. 101–121. doi:10.1007/978-981-13-2203-7_9.
[118] A.P. Sheth and J.A. Larson, Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases, *ACM Computing Surveys* **22**(3) (1990), 183–236. doi:10.1145/96602.96604.
[119] C. Bondiombouy and P. Valduriez, Query processing in multistore systems: an overview, *Int. J. Cloud Computing* **5**(4) (2016), 309–346. doi:10.1504/IJCC.2016.10001884.
[120] H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann and S. Hübner, Ontology-Based Integration of Information - A Survey of Existing Approaches, in: *Proc. of Workshop on Ontologies and Information Sharing*, CEUR Workshop Proceedings, Vol. 47, CEUR-WS.org, 2001.
[121] N.F. Noy, Semantic Integration: A Survey Of Ontology-Based Approaches, *SIGMOD Record* **33**(4) (2004), 65–70. doi:10.1145/1041410.1041421.
[122] F.J. Ekaputra, M. Sabou, E. Serral, E. Kiesling and S. Biffl, Ontology-Based Data Integration in Multi-Disciplinary Engineering Environments: A Review, *Open J. Information Systems* **4**(1) (2017), 1–26.
[123] A. Buccella, A. Cechich and P.R. Fillottrani, Ontology-driven geographic information integration: A survey of current approaches, *Computers and Geosciences* **35**(4) (2009), 710–723. doi:10.1016/j.cageo.2008.02.033.
[124] B. Hassan, R. Fissoune and C. Messaoudi, A Survey of Semantic Integration Approaches in Bioinformatics, *Int. J. Computer, Electrical, Automation, Control and Information Engineering* **10**(12) (2016), 1968–1973.
[125] M. Mountantonakis and Y. Tzitzikas, Large-scale Semantic Integration of Linked Data: A Survey, *ACM Computing Surveys* **52**(5) (2019), 103:1–103:40. doi:10.1145/3345551.
[126] M. Schmidt, O. Görlitz, P. Haase, G. Ladwig, A. Schwarte and T. Tran, FedBench: A Benchmark Suite for Federated Semantic Data Query Processing, in: *Proc. of Int. Semantic Web Conf. (ISWC)*, LNCS, Vol. 7031, Springer, 2011, pp. 585–600. doi:10.1007/978-3-642-25073-6_37.

[127] M. Saleem, A. Hasnain and A.-C. Ngonga Ngomo, LargeRDFBench: A billion triples benchmark for SPARQL endpoint federation, *J. Web Semantics* **48** (2018), 85–125. doi:10.1016/j.websem.2017.12.005.

Appendix A. Specific data sources supported by the selected systems

Table 7 lists the specific sources supported by each investigated data federation system, obtained from available systems' documentation and publications. Sources are classified along the source types defined in Section 6.1, with additional source information — such as the specific kind(s) of relational, graph-based or aggregate-oriented system — reported next to the source name via subscript letters (see table caption for legend). We remark the following:

- Some sources correspond to data access interfaces that can be configured to connect additional systems beyond the ones explicitly listed in the table. In particular, companies such as CData[21] and Progress[22] commercialize *connectors* for the relational SQL-based JDBC, ODBC, ADO.NET and OLE DB interfaces that can be used to access a myriad of heterogeneous data sources, possibly using a different data model that is transparently adapted to the relational one by the connector (*e.g.*,, via flattening of nested data). In Table 7, besides the supported data access interfaces, we explicitly list only the sources that are directly and natively supported by a system without relying on such third party connectors / adapters.
- Structured files are distinguished from other source types with the same data model (*e.g.*, relational sources for CSV files, aggregate-oriented — specifically, document-based — for JSON files) by virtue of direct access to raw file contents by the data federation system. In some cases, however, access to structured files stored may require metadata services external to the filesystem (*e.g.*, Hive Metadata Store) for locating and interpreting file contents, or may leverage processing services (*e.g.*, from Hadoop) co-located with the nodes storing the file in a distributed filesystem (*e.g.*, HDFS), for instance to *push down* data access operations and computations (*e.g.*, filtering, sorting) close to where raw file data reside, this way reducing communication costs.
- Some of the data federation systems investigated in this survey are also listed as supported sources (marked with $*$ subscript) of other systems in Table 7, reflecting the fact that the virtual data sources obtained through data federation can be used themselves in downstream federations. As a limit case (*e.g.*, AllegroGraph), a system may list only itself as supported data source, which occurs when the system offers both storage and data federation capabilities, and the latter are restricted to instances of the same system.
- Test sources (e.g., emulating `/dev/null`) and system-specific connectors used to access configuration, performance or log data of the system itself are omitted in Table 7, for simplicity.

Table 7: Supported data sources of the investigated systems. Academic systems in *italics*. Additional source information in subscript position: $*$ = investigated system; a = specialized web API; r = RDF triple store; g = property graph store; k = key-value store; w = wide-column store; d = document store; s = search engine; h = hardware + software appliance; m = MDX (MultiDimensional eXpressions) support. SPARQLp denotes the SPARQL protocol.

System	Relational	Graph-based	Aggregate-oriented	Structured Files	Web Service Paradigms	Other
AllegroGraph		Allegro-Graph$_{r*}$				
Amazon Athena	Amazon Redshift, MySQL, PostgreSQL, Vertica	Amazon Neptune$_{rg*}$	Amazon DocumentDB$_d$, Amazon DynamoDB$_d$, Amazon OpenSearch$_s$, HBase$_w$, Redis$_k$	Common Log Format, CSV, JSON, ORC, Parquet		Amazon AWS System Manager Inventory$_a$, Amazon CloudWatch$_a$, Amazon Timestream
Amazon Neptune		SPARQLp				

[21] https://www.cdata.com/drivers/
[22] https://www.progress.com/connectors

System	Relational	Graph-based	Aggregate-oriented	Structured Files	Web Service Paradigms	Other
AnzoGraph DB	Derby, Google BigQuery, Hive, HSQLDB, IBM DB2, Impala, JDBC, MariaDB, MS SQL Server$_*$, MySQL, PostgreSQL, SAP ASE			CSV, JSON, Parquet, SAS7BDAT, SAS XPT, XML	HTTP / REST	
Apache Drill	Derby, Druid, Hive, H2, MS SQL Server$_*$, MySQL, Oracle DB$_*$, PostgreSQL		Cassandra$_w$, Elasticsearch$_s$, HBase$_w$, MapR-DB$_w$, MongoDB$_d$, Splunk$_s$	Avro, Common Log Format, CSV, Excel, JSON, Parquet, SequenceFile, XML	HTTP / REST	Kafka, OpenTSDB
Apache Jena (ARQ)		Jena API, SPARQLp				
Apache Spark	Hive, JDBC			any file (content field + metadata), Avro, CSV, JSON, ORC, Parquet		
BigDAWG	PostgreSQL		Accumulo$_w$			SciDB
CloudMdsQL	Derby	Sparksee$_g$	MongoDB$_d$			
CostFed		SPARQLp				
DARQ		SPARQLp				
Data Virtuality	Amazon Redshift, ClickHouse, Data Virtuality$_*$, Derby, Exasol, Google BigQuery, Greenplum, Hive, HSQLDB, H2, IBM DB2, IBM Informix, IBM Netezza$_h$, Ingres, JDBC, MDX$_m$, MetaMatrix$_*$, MS SQL Server$_*$, MySQL, Oracle DB$_*$, PostgreSQL, SAP ASE, SingleStore, Snowflake, Teradata	Neo4j$_{g*}$	MongoDB$_d$, Redis$_k$	CSV, Excel, JSON, XML	HTTP / REST	DHL Track & Trace$_a$, Google Ads$_a$, Google Analytics$_a$, InterSystems Caché, Kdb+, LDAP, ModeShape, Salesforce$_a$
Denodo	Amazon Athena$_*$, Amazon Redshift, Databricks, Denodo$_*$, Derby, Google BigQuery, Greenplum, Hive, IBM DB2, IBM Informix, IBM Netezza$_h$, Impala, JDBC, MS Analysis Service$_m$, MS Azure SQL Database, MS SQL Server$_*$, MS Azure Synapse Analytics, Mondrian$_m$, MySQL, Oracle DB$_*$, Oracle Essbase$_m$, Oracle TimesTen, PostgreSQL, Presto$_*$, SAP ASE, SAP Business Warehouse$_m$, SAP HANA$_*$, Snowflake, Teradata, Trino$_*$, Vertica, Yellowbrick$_h$		Amazon OpenSearch$_s$, Cassandra$_w$, Elasticsearch$_s$, MongoDB$_d$	CSV, Excel, JSON, XML	SOAP / WSDL	ITPilot (website wrapper generator), LDAP, Salesforce$_a$, SAP Business$_a$
Dremio	Amazon Redshift, Hive, MS SQL Server$_*$, MySQL, Oracle DB$_*$, PostgreSQL, Teradata		Amazon OpenSearch$_s$, Elasticsearch$_s$, HBase$_w$, MongoDB$_d$	CSV, Excel, JSON, Parquet		
FEDRA		SPARQLp				
FedX (RDF4J)		RDF4J API, SPARQLp				
GraphDB	IBM DB2, MS SQL Server$_*$, MySQL, Oracle DB$_*$, PostgreSQL	GraphDB$_r$, SPARQLp				
HiBISCuS		SPARQLp				
IBM Cloud Pak for Data	Amazon Redshift, Derby, Google BigQuery, Greenplum, Hive, IBM DB2, IBM Db2 Big SQL$_*$, IBM Db2 Warehouse, IBM DVM, IBM Informix, IBM Netezza$_h$, Impala, MariaDB, MS SQL Server$_*$, MySQL, Oracle DB$_*$, PostgreSQL, SAP ASE, SAP HANA$_*$, Snowflake, Teradata		MongoDB$_d$	CSV, Excel	OData	IBM Db2 Event Store, Salesforce$_a$, SAP Gateway OData$_a$
IBM Db2 Big SQL	Amazon Athena$_*$, Amazon Redshift, Derby, Google BigQuery, Greenplum, Hive, IBM DB2, IBM Db2 Big SQL$_*$, IBM Db2 Warehouse, IBM DVM, IBM Informix, IBM Integrated Analytics System$_h$, IBM Netezza$_h$, IBM PureData$_h$, Impala, MariaDB, MS Azure SQL Database, MS SQL Server$_*$, MySQL, Oracle DB$_*$, PostgreSQL, SAP ASE, SAP HANA$_*$, Teradata		Amazon OpenSearch$_s$, CouchDB$_d$, MongoDB$_d$	Parquet		IBM MQ, Salesforce$_a$
IBM InfoSphere Federation Server	IBM DB2, IBM Informix, MS SQL Server$_*$, Oracle DB$_*$, SAP ASE, Datacom/DB, Teradata, IBM Netezza$_h$			Excel, XML	SOAP / WSDL	BioRS, IBM MQ, IDMS, IMS

System	Relational	Graph-based	Aggregate-oriented	Structured Files	Web Service Paradigms	Other
JBoss Data Virtualization	Actian Vector, Amazon Redshift, Exasol, Greenplum, Hive, Hive, IBM DB2, IBM Informix, IBM Netezza$_h$, Impala, Ingres, JBoss Data Virtualization$_*$, MariaDB, MetaMatrix$_*$, MS Access, MS SQL Server$_*$, Mondrian$_m$, MySQL, Oracle DB$_*$, PostgreSQL, Presto$_*$, SAP ASE, SAP HANA$_*$, SAP IQ, Teradata, Vertica		Accumulo$_w$, Amazon OpenSearch$_s$, Cassandra$_w$, Couchbase$_d$, HBase$_w$, MongoDB$_d$, Red Hat Data Grid$_k$, Solr$_s$	CSV, Excel, XML	HTTP / REST, OData, SOAP / WSDL	Google Sheets$_a$, LDAP, ModeShape, OSIsoft PI, Red Hat Directory Server, Salesforce$_a$, SAP Gateway OData$_a$
Lusail		SPARQLp				
Metaphactory	JDBC	Amazon Neptune$_{rg*}$, GraphDB$_r$, SPARQLp, Stardog$_{r*}$, Virtuoso$_{r*}$	Elasticsearch$_s$		HTTP / REST	
Myria		SPARQLp	Amazon OpenSearch$_s$	CSV		SciDB
Neo4j (Fabric)		Neo4j$_{g*}$				
Obi-Wan	PostgreSQL	Jena TDB$_r$	MongoDB$_d$, Redis$_k$			
Odyssey		SPARQLp				
Ontario	MySQL	Neo4j$_{g*}$, SPARQLp	MongoDB$_d$	CSV, XML		
Onto-KIT				CSV, ENVI, JSON		
Oracle Big Data SQL	Hive		HBase$_w$, Oracle NoSQL$_k$	Avro, CSV, JSON, ORC, Parquet, XML		Kafka
Oracle DB (Spatial & Graph)	Oracle DB$_*$, Oracle DB$_*$	SPARQLp				
PolyWeb	MySQL	SPARQLp		CSV		
Presto	Amazon Redshift, Druid, Google BigQuery, Hive, Iceberg, Kudu, MS SQL Server$_*$, MySQL, Oracle DB$_*$, Pinot, PostgreSQL		Accumulo$_w$, Cassandra$_w$, Elasticsearch$_s$, MongoDB$_d$, Redis$_k$			Kafka, Prometheus
Querona Data Virtualization	Actian Matrix, Actian Vector, ADO.NET, Alibaba AnalyticDB for MySQL, Alibaba Data Lake Analytics, Amazon Athena$_*$, Amazon Aurora, Amazon Redshift, ClickHouse, Databricks, dBASE, Denodo$_*$, Drill$_*$, Exasol, Google BigQuery, IBM DB2, JDBC, MariaDB, MS Access, MS SQL Server$_*$, MS Azure Synapse Analytics, MySQL, ODBC, OLE DB, Oracle DB$_*$, PostgreSQL, SAP HANA$_*$, SAS Scalable Performance Data Server, Spark$_*$, Teradata, Teradata Aster, Vertica		Amazon OpenSearch$_s$, DataStax$_w$	CSV, Excel, MSG/EML (email), PDF (metadata)		Kafka
SAFE		SPARQLp				
SAP HANA	Amazon Athena$_*$, Google BigQuery, IBM DB2, IBM Netezza$_h$, MS SQL Server$_*$, Oracle DB$_*$, SAP ASE, SAP HANA$_*$, SAP IQ, SAP MaxDB, Teradata					SAP HANA Streaming Analytics
SAS Federation Server	dBASE, Greenplum, Hive, IBM DB2, IBM Informix, IBM Netezza$_h$, Impala, MS Access, MS SQL Server$_*$, MySQL, Oracle DB$_*$, Paradox, PostgreSQL, Progress OpenEdge RDBMS, SAP ASE, SAP HANA$_*$, SAS Federation Server$_*$, SAS Scalable Performance Data Server, Teradata					Btrieve, Salesforce$_a$, SAP RFC$_a$
SemaGrow		SPARQLp				
SPLENDID		SPARQLp				
SQL Server (PolyBase)	MS SQL Server$_*$, ODBC, Oracle DB$_*$, Teradata		MongoDB$_d$	CSV, JSON, ORC, Parquet, RCFile		
Squerall	MySQL		Cassandra$_w$, Couchbase$_d$, Elasticsearch$_s$, MongoDB$_d$	CSV, Parquet		

System	Relational	Graph-based	Aggregate-oriented	Structured Files	Web Service Paradigms	Other
Starburst	Amazon Redshift, ClickHouse, Druid, Google BigQuery, Greenplum, Hive, IBM DB2, IBM Netezza$_h$, Iceberg, JDBC, Kudu, MS SQL Server$_*$, MS Azure Synapse Analytics, MySQL, Oracle DB$_*$, Pinot, PostgreSQL, SAP HANA$_*$, SingleStore, Snowflake, Starburst$_*$, Teradata, Vertica		Accumulo$_w$, Amazon DynamoDB$_d$, Cassandra$_w$, Elasticsearch$_s$, HBase$_w$, MongoDB$_d$, Redis$_k$, Splunk$_s$	Avro, CSV, JSON, ORC, Parquet, RCFile, SequenceFile		Amazon Kinesis, Google Sheets$_a$, Kafka, Prometheus, Salesforce$_a$
Stardog	Amazon Athena$_*$, Amazon Aurora, Amazon Redshift, Derby, Exasol, Google BigQuery, Hive, H2, IBM DB2, Impala, MariaDB, MS SQL Server$_*$, MySQL, Oracle DB$_*$, PostgreSQL, SAP ASE, SAP HANA$_*$, Snowflake, Teradata	SPARQLp, Stardog$_{r*}$	Amazon OpenSearch$_s$, Cassandra$_w$, DataStax$_w$, Elasticsearch$_s$, MS Azure Cosmos DB$_d$, MongoDB$_d$, Splunk$_s$	CSV, JSON		Google Sheets$_a$, Jira$_a$, LDAP, Salesforce$_a$
Teiid	Actian Vector, Amazon Athena$_*$, Amazon Redshift, Derby, Exasol, Greenplum, Hive, HSQLDB, H2, IBM DB2, IBM Informix, IBM Netezza$_h$, Impala, Ingres, JDBC, MariaDB, MDX$_m$, MetaMatrix$_*$, MS Access, MS SQL Server$_*$, Mondrian$_m$, MySQL, Oracle DB$_*$, PostgreSQL, Presto$_*$, SAP ASE, SAP HANA$_*$, SAP IQ, Teiid$_*$, Teradata, Vertica		Accumulo$_w$, Amazon OpenSearch$_s$, Amazon SimpleDB$_{kw}$, Cassandra$_w$, Couchbase$_d$, HBase$_w$, Infinispan$_k$, MongoDB$_d$, Solr$_s$	CSV, Excel, JSON, XML	HTTP / REST, OData, OpenAPI, SOAP / WSDL	Google Sheets$_a$, InterSystems Caché, JPA/JPQL sources, LDAP, MS Active Directory, ModeShape, OSIsoft PI, Red Hat Directory Server, Salesforce$_a$, SAP Gateway OData$_a$
TIBCO Data Virtualization	Amazon Redshift, Drill$_*$, Google BigQuery, Greenplum, Hive, HP Neoview$_h$, HSQLDB, IBM DB2, IBM Informix, IBM Netezza$_h$, MS Access, MS SQL Server$_*$, MySQL, Oracle DB$_*$, PostgreSQL, SAP ASE, SAP Business Warehouse$_m$, SAP Business Warehouse$_m$, SAP HANA$_*$, Snowflake, Teradata, Tibco ComputeDB, TibcoDataVirtualization$_*$, Vertica		Amazon DynamoDB$_d$, Amazon OpenSearch$_s$, Cassandra$_w$, Couchbase$_d$, Elasticsearch$_s$, HBase$_w$, MarkLogic$_d$, MS Azure Cosmos DB$_d$, MongoDB$_d$, Splunk$_s$	CSV, Excel, JSON, XML	HTTP / REST, OData, SOAP / WSDL	Eloqua$_a$, Facebook$_a$, Google Ads$_a$, Google Analytics$_a$, Google Calendar$_a$, Google Contacts$_a$, Google Sheets$_a$, HubSpot$_a$, IMAP, Marketo$_a$, MS Sharepoint$_a$, MS Sharepoint Excel Services$_a$, NetSuite$_a$, RSS, Salesforce$_a$, SAP RFC$_a$, Twitter$_a$
Trino	Amazon Redshift, ClickHouse, Druid, Google BigQuery, Hive, Iceberg, Kudu, MS SQL Server$_*$, MySQL, Oracle DB$_*$, Pinot, PostgreSQL, SingleStore		Accumulo$_w$, Cassandra$_w$, Elasticsearch$_s$, HBase$_w$, MongoDB$_d$, Redis$_k$			Amazon Kinesis, Google Sheets$_a$, Kafka, Prometheus
Virtuoso	Firebird, IBM DB2, IBM Informix, Ingres, MS SQL Server$_*$, MySQL, Oracle DB$_*$, PostgreSQL, Progress OpenEdge RDBMS, SAP ASE					

Appendix B. Selection of academic systems bibliography

[128] A.A. Algosaibi, High-Performance Computing Based Approach for Improving Semantic-Based Federated Data Processing, *Computer Science* **16**(1) (2021), 287–309.

[129] P. Amanpartap Singh, J.S. Khaira et al., A comparative review of extraction, transformation and loading tools, *Database Systems Journal* **42** (2013).

[130] B. Arputhamary and L. Arockiam, A review on big data integration, *Int. J. Comput. Appl* (2014), 21–26.

[131] J. Duggan, A.J. Elmore, M. Stonebraker, M. Balazinska, B. Howe, J. Kepner, S. Madden, D. Maier, T. Mattson and S.B. Zdonik, The BigDAWG Polystore System, *SIGMOD Record* **44**(2) (2015), 11–16. doi:10.1145/2814710.2814713.

[132] A.-R. Bologa and R. Bologa, A Perspective on the Benefits of Data Virtualization Technology., *Informatica Economica* **15**(4) (2011).

[133] M. Butenuth, G.v. Gösseln, M. Tiedge, C. Heipke, U. Lipeck and M. Sester, Integration of heterogeneous geospatial data in a federated database, *ISPRS Journal of Photogrammetry and Remote Sensing* **62**(5) (2007), 328–346.

[134] D. Chaves-Fraga, F. Priyatna, A. Alobaid and O. Corcho, Exploiting declarative mapping rules for generating graphQL servers with morph-graphQL, *International Journal of Software Engineering and Knowledge Engineering* **30**(06) (2020), 785–803.

[135] Y. Khan, A. Zimmermann, A. Jha, V. Gadepally, M. d'Aquin and R. Sahay, One Size Does Not Fit All: Querying Web Polystores, *IEEE Access* **7** (2019), 9598–9617. doi:10.1109/ACCESS.2018.2888601.

[136] W. Shen, Q. Hao, H. Mak, J. Neelamkavil, H. Xie, J. Dickinson, R. Thomas, A. Pardasani and H. Xue, Systems integration and collaboration in architecture, engineering, construction, and facilities management: A review, *Adv. Eng. Informatics* **24**(2) (2010), 196–207. doi:10.1016/j.aei.2009.09.001.

[137] C. Lazar, S. Meganck, J. Taminau, D. Steenhoff, A. Coletta, C. Molter, D.Y.W. Sol'ıs, R. Duque, H. Bersini and A. Now'e, Batch effect removal methods for microarray gene expression data integration: a survey, *Briefings Bioinform.* **14**(4) (2013), 469–490. doi:10.1093/bib/bbs037.

[138] S. Jupp, J. Malone, J.T. Bolleman, M. Brandizi, M. Davies, L.J. Garcia, A. Gaulton, S. Gehant, C. Laibe, N. Redaschi, S.M. Wimalaratne, M.J. Martin, N.L. Novère, H.E. Parkinson, E. Birney and A.M. Jenkinson, The EBI RDF platform: linked open data for the life sciences, *Bioinformatics* **30**(9) (2014), 1338–1339. doi:10.1093/bioinformatics/btt765.

[139] A. Hasnain, Q. Mehmood, S.S. e Zainab, M. Saleem, C.N.W. Jr., D. Zehra, S. Decker and D. Rebholz-Schuhmann, BioFed: federated query processing over life sciences linked open data, *J. Biomed. Semant.* **8**(1) (2017), 13:1–13:19. doi:10.1186/s13326-017-0118-0.

[140] Y. Khan, M. Saleem, M. Mehdi, A. Hogan, Q. Mehmood, D. Rebholz-Schuhmann and R. Sahay, SAFE: SPARQL Federation over RDF Data Cubes with Access Control, *J. Biomed. Semant.* **8**(1) (2017), 5:1–5:22. doi:10.1186/s13326-017-0112-6.

[141] M. Saleem, S.S. Padmanabhuni, A.N. Ngomo, A. Iqbal, J.S. Almeida, S. Decker and H.F. Deus, TopFed: TCGA Tailored Federated Query Processing and Linking to LOD, *J. Biomed. Semant.* **5** (2014), 47. doi:10.1186/2041-1480-5-47.

[142] K. Cheung, H.R. Frost, M.S. Marshall, E. Prud'hommeaux, M. Samwald, J. Zhao and A. Paschke, A journey to Semantic Web query federation in the life sciences, *BMC Bioinform.* **10**(S–10) (2009), 10. doi:10.1186/1471-2105-10-S10-S10.

[143] D. Gomez-Cabrero, I. Abugessaisa, D. Maier, A.E. Teschendorff, M. Merkenschlager, A. Gisel, E. Ballestar, E. Bongcam-Rudloff, A. Conesa and J. Tegnér, Data integration in the era of omics: current and future challenges, *BMC Syst. Biol.* **8**(S–2) (2014), I1. doi:10.1186/1752-0509-8-S2-I1.

[144] C. Parent and S. Spaccapietra, Issues and Approaches of Database Integration, *Commun. ACM* **41**(5) (1998), 166–178. doi:10.1145/276404.276408.

[145] X. Zhang, M. Zhang, P. Peng, J. Song, Z. Feng and L. Zou, gSMat: A Scalable Sparse Matrix-based Join for SPARQL Query Processing, *CoRR* **abs/1807.07691** (2018).

[146] D. Chaves-Fraga, E. Ruckhaus, F. Priyatna, M. Vidal and Ó. Corcho, Enhancing OBDA Query Translation over Tabular Data with Morph-CSV, *CoRR* **abs/2001.09052** (2020).

[147] K. Bereta, G. Papadakis and M. Koubarakis, OBDA for the Web: Creating Virtual RDF Graphs On Top of Web Data Sources, *CoRR* **abs/2005.11264** (2020).

[148] W. Ali, M. Saleem, B. Yao, A. Hogan and A.N. Ngomo, Storage, Indexing, Query Processing, and Benchmarking in Centralized and Distributed RDF Engines: A Survey, *CoRR* **abs/2009.10331** (2020).

[149] L. Heling and M. Acosta, A Framework for Federated SPARQL Query Processing over Heterogeneous Linked Data Fragments, *CoRR* **abs/2102.03269** (2021).

[150] G. Montoya, H. Skaf-Molli, P. Molli and M. Vidal, Fedra: Query Processing for SPARQL Federations with Divergence, *CoRR* **abs/1407.2899** (2014).

[151] G. Montoya, H. Skaf-Molli, P. Molli and M. Vidal, Efficient Query Processing for SPARQL Federations with Replicated Fragments, *CoRR* **abs/1503.02940** (2015).

[152] N.A. Rakhmawati, J. Umbrich, M. Karnstedt, A. Hasnain and M. Hausenblas, Querying over Federated SPARQL Endpoints - A State of the Art Survey, *CoRR* **abs/1306.1723** (2013).

[153] I.F. Ilyas, G. Beskales and M.A. Soliman, A survey of top-*k* query processing techniques in relational database systems, *ACM Comput. Surv.* **40**(4) (2008), 11:1–11:58. doi:10.1145/1391729.1391730.

[154] M. Mountantonakis and Y. Tzitzikas, Large-scale Semantic Integration of Linked Data: A Survey, *ACM Comput. Surv.* **52**(5) (2019), 103:1–103:40. doi:10.1145/3345551.

[155] X. Wang, L.M. Haas and A. Meliou, Explaining Data Integration, *IEEE Data Eng. Bull.* **41**(2) (2018), 47–58.

[156] I. Mountasser, B. Ouhbi, F. Hdioud and B. Frikh, Semantic-based Big Data integration framework using scalable distributed ontology matching strategy, *Distributed Parallel Databases* **39**(4) (2021), 891–937. doi:10.1007/s10619-021-07321-6.

[157] M.T. "Ozsu, A survey of RDF data management systems, *Frontiers Comput. Sci.* **10**(3) (2016), 418–432. doi:10.1007/s11704-016-5554-y.

[158] M. Masmoudi, S.B.A.B. Lamine, H.B. Zghal, B. Archimède and M. Karray, Knowledge hypergraph-based approach for data integration and querying: Application to Earth Observation, *Future Gener. Comput. Syst.* **115** (2021), 720–740. doi:10.1016/j.future.2020.09.029.

[159] Q.M. Ilyas, M. Ahmad, S. Rauf and D. Irfan, RDF Query Path Optimization Using Hybrid Genetic Algorithms: Semantic Web vs. Data-Intensive Cloud Computing, *Int. J. Cloud Appl. Comput.* **12**(1) (2022), 1–16. doi:10.4018/IJCAC.2022010101.

[160] C. Avila-Garzon, Applications, Methodologies, and Technologies for Linked Open Data: A Systematic Literature Review, *Int. J. Semantic Web Inf. Syst.* **16**(3) (2020), 53–69. doi:10.4018/IJSWIS.2020070104.

[161] A.G. Anadiotis, O. Balalau, C. Conceição, H. Galhardas, M.Y. Haddad, I. Manolescu, T. Merabti and J. You, Graph integration of structured, semistructured and unstructured data for data journalism, *Inf. Syst.* **104** (2022), 101846. doi:10.1016/j.is.2021.101846.

[162] Ö. Ulusoy, Research Issues in Real-Time Database Systems, *Inf. Sci.* **87**(1–3) (1995), 123–151. doi:10.1016/0020-0255(95)00130-1.

[163] F. Gandon, A survey of the first 20 years of research on semantic Web and linked data, *Ing'enierie des Systèmes d Inf.* **23**(3–4) (2018), 11–38. doi:10.3166/isi.23.3-4.11-38.

[164] D. Oguz, S. Yin, B. Ergenç, A. Hameurlain and O. Dikenelli, Extended Adaptive Join Operator with Bind-Bloom Join for Federated SPARQL Queries, *Int. J. Data Warehous. Min.* **13**(3) (2017), 47–72. doi:10.4018/IJDWM.2017070103.

[165] D.R.B. Cunha and B.F. Lóscio, An Approach for Query Decomposition on Federated SPARQL Query Systems, *J. Inf. Data Manag.* **6**(2) (2015), 106–117.

[166] R. Ramakrishnan and J.D. Ullman, A survey of deductive database systems, *J. Log. Program.* **23**(2) (1995), 125–149. doi:10.1016/0743-1066(94)00039-9.

[167] D. Oguz, B. Ergenc, S. Yin, O. Dikenelli and A. Hameurlain, Federated query processing on linked data: a qualitative survey and open challenges, *Knowl. Eng. Rev.* **30**(5) (2015), 545–563. doi:10.1017/S0269888915000107.

[168] S. Gama-Castro, H. Salgado, A. Santos-Zavaleta, D. Ledezma-Tejeida, L. Muñiz-Rascado, J.S. García-Sotelo, K. Alquicira-Hernández, I. Martínez-Flores, L. Pannier, J.A. Castro-Mondragón, A. Medina-Rivera, H. Solano-Lira, C. Bonavides-Martínez, E. Pérez-Rueda, S. Alquicira-Hernández, L. Porrón-Sotelo, A. López-Fuentes, A. Hernández-Koutoucheva, V. del Moral-Chávez, F. Rinaldi and J. Collado-Vides, RegulonDB version 9.0: high-level integration of gene regulation, coexpression, motif clustering and beyond, *Nucleic Acids Res.* **44**(Database–Issue) (2016), 133–143. doi:10.1093/nar/gkv1156.

[169] F.J. Ekaputra, M. Sabou, E. Serral, E. Kiesling and S. Biffl, Ontology-Based Data Integration in Multi-Disciplinary Engineering Environments: A Review, *Open J. Inf. Syst.* **4**(1) (2017), 1–26.

[170] G. Fusco and L. Aversano, An approach for semantic integration of heterogeneous data sources, *PeerJ Comput. Sci.* **6** (2020), e254. doi:10.7717/peerj-cs.254.

[171] I. Abdelaziz, E. Mansour, M. Ouzzani, A. Aboulnaga and P. Kalnis, Lusail: A System for Querying Linked Data at Scale, *Proc. VLDB Endow.* **11**(4) (2017), 485–498. doi:10.1145/3186728.3164144.

[172] R. Alotaibi, B. Cautis, A. Deutsch, M. Latrache, I. Manolescu and Y. Yang, ESTOCADA: Towards Scalable Polystore Systems, *Proc. VLDB Endow.* **13**(12) (2020), 2949–2952. doi:10.14778/3415478.3415516.

[173] R. Bonaque, T.D. Cao, B. Cautis, F. Goasdoué, J. Letelier, I. Manolescu, O. Mendoza, S. Ribeiro, X. Tannier and M. Thomazo, Mixed-instance querying: a lightweight integration architecture for data journalism, *Proc. VLDB Endow.* **9**(13) (2016), 1513–1516. doi:10.14778/3007263.3007297.

[174] M. Buron, F. Goasdoué, I. Manolescu and M. Mugnier, Obi-Wan: Ontology-Based RDF Integration of Heterogeneous Data, *Proc. VLDB Endow.* **13**(12) (2020), 2933–2936. doi:10.14778/3415478.3415512.

[175] A. Schätzle, M. Przyjaciel-Zablocki, S. Skilevic and G. Lausen, S2RDF: RDF Querying with SPARQL on Spark, *Proc. VLDB Endow.* **9**(10) (2016), 804–815. doi:10.14778/2977797.2977806.

[176] J. Tan, T.M. Ghanem, M. Perron, X. Yu, M. Stonebraker, D.J. DeWitt, M. Serafini, A. Aboulnaga and T. Kraska, Choosing A Cloud DBMS: Architectures and Tradeoffs, *Proc. VLDB Endow.* **12**(12) (2019), 2170–2182. doi:10.14778/3352063.3352133.

[177] B. Arsic, M. Đokic-Petrovic, P.C. Spalevic, I.Z. Milentijevic, D.D. Rancic and M. Zivanovic, SpecINT: A framework for data integration over cheminformatics and bioinformatics RDF repositories, *Semantic Web* **10**(4) (2019), 795–813. doi:10.3233/SW-180327.

[178] D. Chaves-Fraga, E. Ruckhaus, F. Priyatna, M. Vidal and Ó. Corcho, Enhancing virtual ontology based access over tabular data with Morph-CSV, *Semantic Web* **12**(6) (2021), 869–902. doi:10.3233/SW-210432.

[179] U. Qudus, M. Saleem, A.N. Ngomo and Y. Lee, An empirical evaluation of cost-based federated SPARQL query processing engines, *Semantic Web* **12**(6) (2021), 843–868. doi:10.3233/SW-200420.

[180] M. Saleem, Y. Khan, A. Hasnain, I. Ermilov and A.N. Ngomo, A fine-grained evaluation of SPARQL endpoint federation systems, *Semantic Web* **7**(5) (2016), 493–518. doi:10.3233/SW-150186.

[181] A. Stolpe, A logical characterisation of SPARQL federation, *Semantic Web* **6**(6) (2015), 565–584. doi:10.3233/SW-140160.

[182] N.F. Noy, Semantic Integration: A Survey Of Ontology-Based Approaches, *SIGMOD Rec.* **33**(4) (2004), 65–70. doi:10.1145/1041410.1041421.

[183] M. Lissandrini, T.B. Pedersen, K. Hose and D. Mottin, Knowledge graph exploration: where are we and where are we going?, *SIGWEB Newsl.* **2020**(Summer) (2020), 4:1–4:8. doi:10.1145/3409481.3409485.

[184] P. Peng, Q. Ge, L. Zou, M.T. Özsu, Z. Xu and D. Zhao, Optimizing Multi-Query Evaluation in Federated RDF Systems, *IEEE Trans. Knowl. Data Eng.* **33**(4) (2021), 1692–1707. doi:10.1109/TKDE.2019.2947050.

[185] K. Stefanidis, G. Koutrika and E. Pitoura, A survey on representation, composition and application of preferences in database systems, *ACM Trans. Database Syst.* **36**(3) (2011), 19:1–19:45. doi:10.1145/2000824.2000829.

[186] Z. Kaoudi and I. Manolescu, RDF in the clouds: a survey, *VLDB J.* **24**(1) (2015), 67–91. doi:10.1007/s00778-014-0364-z.

[187] S. Kruse, Z. Kaoudi, B. Contreras-Rojas, S. Chawla, F. Naumann and J. Quiané-Ruiz, RHEEMix in the data jungle: a cost-based optimizer for cross-platform systems, *VLDB J.* **29**(6) (2020), 1287–1310. doi:10.1007/s00778-020-00612-x.

[188] P. Peng, L. Zou, M.T. Özsu, L. Chen and D. Zhao, Processing SPARQL queries over distributed RDF graphs, *VLDB J.* **25**(2) (2016), 243–268. doi:10.1007/s00778-015-0415-0.

[189] M. Saleem, A. Hasnain and A.N. Ngomo, LargeRDFBench: A billion triples benchmark for SPARQL endpoint federation, *J. Web Semant.* **48** (2018), 85–125. doi:10.1016/j.websem.2017.12.005.

[190] C.B. Aranda, M. Arenas, Ó. Corcho and A. Polleres, Federating queries in SPARQL 1.1: Syntax, semantics and evaluation, *J. Web Semant.* **18**(1) (2013), 1–17. doi:10.1016/j.websem.2012.10.001.

[191] C. Basca and A. Bernstein, Querying a messy web of data with Avalanche, *J. Web Semant.* **26** (2014), 1–28. doi:10.1016/j.websem.2014.04.002.

[192] J. Halvorsen and A. Stolpe, On the size of intermediate results in the federated processing of SPARQL BGPs, *J. Web Semant.* **51** (2018), 20–38. doi:10.1016/j.websem.2018.06.001.

[193] G. Montoya, H. Skaf-Molli, P. Molli and M. Vidal, Decomposing federated queries in presence of replicated fragments, *J. Web Semant.* **42** (2017), 1–18. doi:10.1016/j.websem.2016.12.001.

[194] O. Görlitz and S. Staab, Federated Data Management and Query Optimization for Linked Open Data, in: *New Directions in Web Data Management 1*, Studies in Computational Intelligence, Vol. 331, 2011, pp. 109–137. doi:10.1007/978-3-642-17551-0_5.

[195] P. Fafalios and Y. Tzitzikas, Answering SPARQL queries on the web of data through zero-knowledge link traversal, *ACM SIGAPP Applied Computing Review* **19**(3) (2019), 18–32.

[196] O. Golovnin, Data federation through on-demand queries in intelligent transport systems **1694**(1) (2020), 012030, IOP Publishing.

[197] L. Heling and M. Acosta, A Framework for Federated SPARQL Query Processing over Heterogeneous Linked Data Fragments, *arXiv preprint arXiv:2102.03269* (2021).

[198] S. Huang, K. Chaudhary and L.X. Garmire, More is better: recent progress in multi-omics data integration methods, *Frontiers in genetics* **8** (2017), 84.
[199] A.Z.E. Qutaany, A.H.E. Bastawissy and O. Hegazi, V-DIF: Virtual data integration framework, *International Journal of Computer Systems* **05**(05) (2018).
[200] Y. Khan and R. Sahay, SPARQL Query Federation over Biomedical Data, *Insight Centre for Data Analytics, National University Literature* (2019).
[201] V. Lapatas, M. Stefanidakis, R.C. Jimenez, A. Via and M.V. Schneider, Data integration in biological research: an overview, *Journal of Biological Research-Thessaloniki* **22**(1) (2015), 1–16.
[202] M. Saleem, A. Hasnain and A.-C. Ngonga Ngomo, LargeRDFBench: A Billion Triples Benchmark for SPARQL Endpoint Federation, *SSRN Electronic Journal* (2018). doi:10.2139/ssrn.3199316.
[203] S. Mathivanan and P. Jayagopal, A big data virtualization role in agriculture: a comprehensive review, *Walailak Journal of Science and Technology (WJST)* **16**(2) (2019), 55–70.
[204] C. Messaoudi, R. Fissoune and H. Badir, A survey of semantic integration approaches in bioinformatics, *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering* **10**(12) (2016), 1924–1929.
[205] M. Mountantonakis and Y. Tzitzikas, Large-scale semantic integration of linked data: A survey, *ACM Computing Surveys (CSUR)* **52**(5) (2019), 1–40.
[206] F. Prasser, O. Kohlbacher, U. Mansmann, B. Bauer and K.A. Kuhn, Data integration for future medicine (DIFUTURE), *Methods of information in medicine* **57**(S 01) (2018), e57–e65.
[207] N.A. Rakhmawati, An Holistic Evaluation of Federated SPARQL Query Engine, *Information Systems International Conference* **2013** (2013).
[208] N.A. Rakhmawati et al., How interlinks influence federated over sparql endpoints, *International Journal of Internet and Distributed Systems* **1**(01) (2013), 1.
[209] N.A. Rakhmawati and L.N. Fadzilah, Dataset characteristics identification for federated sparql query, *Scientific Journal of Informatics* **6**(1) (2019), 23–33.
[210] V. Ranjan, A comparative study between ETL (Extract, Transform, Load) and ELT (Extract, Load and Transform) approach for loading data into data warehouse, *MS Candidate in Computer Science at California State University, Chico, CA* **95929** (2009).
[211] P. Sernadela, P. Lopes and J.L. Oliveira, A knowledge federation architecture for rare disease patient registries and biobanks, *J. Inf. Syst. Eng. Manag* **1**(1) (2016), 83–90.
[212] Y. Shi, Y. Tong, Y. Zeng, Z. Zhou, B. Ding and L. Chen, Efficient Approximate Range Aggregation over Large-scale Spatial Data Federation, *IEEE Transactions on Knowledge and Data Engineering* (2021).
[213] I. Subramanian, S. Verma, S. Kumar, A. Jere and K. Anamika, Multi-omics data integration, interpretation, and its application, *Bioinformatics and biology insights* **14** (2020), 1–24. doi:10.1177/1177932219899051.
[214] P. Szabo, Data Virtualization and Federation, *Stone Bond Technologies* (2014).
[215] R. Van Der Lans, *Data Virtualization for business intelligence systems: revolutionizing data integration for data warehouses*, Elsevier, 2012.
[216] M.-E. Vidal, S. Castillo, M. Acosta, G. Montoya and G. Palma, On the selection of SPARQL endpoints to efficiently execute federated SPARQL queries, in: *Transactions on large-scale data-and knowledge-centered systems XXV*, Springer, 2016, pp. 109–149.
[217] C. Wu, F. Zhou, J. Ren, X. Li, Y. Jiang and S. Ma, A selective review of multi-level omics data integration using variable selection, *High-throughput* **8**(1) (2019), 4.
[218] M. Wylot, M. Hauswirth, P. Cudré-Mauroux and S. Sakr, RDF data storage and query processing schemes: A survey, *ACM Computing Surveys (CSUR)* **51**(4) (2018), 1–36.
[219] Z. Zhang, V.B. Bajic, J. Yu, K.-H. Cheung and J.P. Townsend, Data integration in bioinformatics: current efforts and challenges, *Bioinformatics-Trends and Methodologies* (2011), 41–56.
[220] G. Montoya, H. Skaf-Molli, P. Molli and M. Vidal, Federated SPARQL Query Processing with Replicated Fragments, *BDA 2016 Gestion de Données–Principes, Technologies et Applications 32 e anniversaire 15-18 novembre 2016, Poitiers, Futuroscope* **421**(170,078) (2016), 39.
[221] M. Karpathiotakis, I. Alagiannis and A. Ailamaki, Fast Queries Over Heterogeneous Data Through Engine Customization, *Proc. VLDB Endow.* **9**(12) (2016), 972–983. doi:10.14778/2994509.2994516.
[222] B. Kolev, P. Valduriez, C. Bondiombouy, R. Jiménez-Peris, R. Pau and J. Pereira, CloudMdsQL: querying heterogeneous cloud data stores with a common language, *Distributed and parallel databases* **34**(4) (2016), 463–503.
[223] M. Acosta and M.-E. Vidal, Evaluating adaptive query processing techniques for federations of sparql endpoints, in: *10th International Semantic Web Conference (ISWC) Demo Session*, Citeseer, 2011.
[224] M. Armbrust, R.S. Xin, C. Lian, Y. Huai, D. Liu, J.K. Bradley, X. Meng, T. Kaftan, M.J. Franklin, A. Ghodsi et al., Spark SQL: Relational data processing in spark, in: *Proceedings of the 2015 ACM SIGMOD international conference on management of data*, 2015, pp. 1383–1394.
[225] A. Bogdanov, A. Degtyarev, N. Shchegoleva, V. Korkhov and V. Khvatov, Big Data Virtualization: Why and How?, in: *CEUR Workshop Proceedings (2679)*, 2020, pp. 11–21.
[226] M. Buron, F. Goasdoué, I. Manolescu and M.-L. Mugnier, Rewriting-Based Query Answering for Semantic Data Integration Systems, in: *BDA: Gestion de Données–Principes, Technologies et Applications*, 2018.

[227] S. Cheng and O. Hartig, FedQPL: A Language for Logical Query Plans over Heterogeneous Federations of RDF Data Sources, in: *Proceedings of the 22nd International Conference on Information Integration and Web-based Applications & Services*, 2020, pp. 436–445.

[228] A. Katasonov, DataBearings: An Efficient Semantic Approach to Data Virtualization and Federation, in: *The Ninth International Conference on Advances in Semantic Processing, SEMAPRO 2015*, 2015.

[229] R. Hai, C. Quix and C. Zhou, Query Rewriting for Heterogeneous Data Lakes, in: *Advances in Databases and Information Systems - 22nd European Conference, ADBIS 2018, Budapest, Hungary, September 2-5, 2018, Proceedings*, Lecture Notes in Computer Science, Vol. 11019, Springer, 2018, pp. 35–49. doi:10.1007/978-3-319-98398-1_3.

[230] P.N. Sawadogo, É. Scholly, C. Favre, É. Ferey, S. Loudcher and J. Darmont, Metadata Systems for Data Lakes: Models and Features, in: *New Trends in Databases and Information Systems, ADBIS 2019 Short Papers, Workshops BBIGAP, QAUCA, SemBDM, SIMPDA, M2P, MADEISD, and Doctoral Consortium, Bled, Slovenia, September 8-11, 2019, Proceedings*, Communications in Computer and Information Science, Vol. 1064, Springer, 2019, pp. 440–451. doi:10.1007/978-3-030-30278-8_43.

[231] C.B. Aranda and A. Polleres, Towards Equivalences for Federated SPARQL Queries, in: *Proceedings of the 8th Alberto Mendelzon Workshop on Foundations of Data Management, Cartagena de Indias, Colombia, June 4-6, 2014*, CEUR Workshop Proceedings, Vol. 1189, CEUR-WS.org, 2014.

[232] C.B. Aranda, M. Ugarte, M. Arenas and M. Dumontier, A Preliminary Investigation into SPARQL Query Complexity and Federation in Bio2RDF, in: *Proceedings of the 9th Alberto Mendelzon International Workshop on Foundations of Data Management, Lima, Peru, May 6 - 8, 2015*, CEUR Workshop Proceedings, Vol. 1378, CEUR-WS.org, 2015.

[233] F. Yang, A. Crainiceanu, Z. Chen and D. Needham, Cluster-Based Join for Geographically Distributed Big RDF Data, in: *2019 IEEE International Congress on Big Data, BigData Congress 2019, Milan, Italy, July 8-13, 2019*, IEEE, 2019, pp. 170–178. doi:10.1109/BigDataCongress.2019.00037.

[234] E. Kharlamov, T.P. Mailis, K. Bereta, D. Bilidas, S. Brandt, E. Jiménez-Ruiz, S. Lamparter, C. Neuenstadt, Ö.L. Özçep, A. Soylu, C. Svingos, G. Xiao, D. Zheleznyakov, D. Calvanese, I. Horrocks, M. Giese, Y.E. Ioannidis, Y. Kotidis, R. Möller and A. Waaler, A semantic approach to polystores, in: *2016 IEEE International Conference on Big Data (IEEE BigData 2016), Washington DC, USA, December 5-8, 2016*, IEEE Computer Society, 2016, pp. 2565–2573. doi:10.1109/BigData.2016.7840898.

[235] R. Tan, R. Chirkova, V. Gadepally and T.G. Mattson, Enabling query processing across heterogeneous data models: A survey, in: *2017 IEEE International Conference on Big Data (IEEE BigData 2017), Boston, MA, USA, December 11-14, 2017*, IEEE Computer Society, 2017, pp. 3211–3220. doi:10.1109/BigData.2017.8258302.

[236] M. Karpathiotakis, I. Alagiannis, T. Heinis, M. Branco and A. Ailamaki, Just-In-Time Data Virtualization: Lightweight Data Management with ViDa, in: *Seventh Biennial Conference on Innovative Data Systems Research, CIDR 2015, Asilomar, CA, USA, January 4-7, 2015, Online Proceedings*, www.cidrdb.org, 2015.

[237] A. Quamar, J. Straube and Y. Tian, Enabling Rich Queries Over Heterogeneous Data From Diverse Sources In HealthCare, in: *10th Conference on Innovative Data Systems Research, CIDR 2020, Amsterdam, The Netherlands, January 12-15, 2020, Online Proceedings*, www.cidrdb.org, 2020.

[238] J. Wang, T. Baker, M. Balazinska, D. Halperin, B. Haynes, B. Howe, D. Hutchison, S. Jain, R. Maas, P. Mehta, D. Moritz, B. Myers, J. Ortiz, D. Suciu, A. Whitaker and S. Xu, The Myria Big Data Management and Analytics System and Cloud Services, in: *8th Biennial Conference on Innovative Data Systems Research, CIDR 2017, Chaminade, CA, USA, January 8-11, 2017, Online Proceedings*, www.cidrdb.org, 2017.

[239] J. Lu, I. Holubová and B. Cautis, Multi-model Databases and Tightly Integrated Polystores: Current Practices, Comparisons, and Open Challenges, in: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, ACM, 2018, pp. 2301–2302. doi:10.1145/3269206.3274269.

[240] B. Kolev, C. Bondiombouy, O. Levchenko, P. Valduriez, R. Jiménez-Peris, R. Pau and J. Pereira, Design and Implementation of the CloudMdsQL Multistore System, in: *CLOSER 2016 - Proceedings of the 6th International Conference on Cloud Computing and Services Science, Volume 1, Rome, Italy, April 23-25, 2016*, SciTePress, 2016, pp. 352–359. doi:10.5220/0005923803520359.

[241] L.E. Bertossi and L. Bravo, Consistent Query Answers in Virtual Data Integration Systems, in: *Inconsistency Tolerance [result from a Dagstuhl seminar]*, Lecture Notes in Computer Science, Vol. 3300, Springer, 2005, pp. 42–83. doi:10.1007/978-3-540-30597-2_3.

[242] P. Peng, L. Zou, M.T. Özsu and D. Zhao, Multi-query Optimization in Federated RDF Systems, in: *Database Systems for Advanced Applications - 23rd International Conference, DASFAA 2018, Gold Coast, QLD, Australia, May 21-24, 2018, Proceedings, Part I*, Lecture Notes in Computer Science, Vol. 10827, Springer, 2018, pp. 745–765. doi:10.1007/978-3-319-91452-7_48.

[243] K.M. Endris, M. Galkin, I. Lytra, M.N. Mami, M. Vidal and S. Auer, MULDER: Querying the Linked Data Web by Bridging RDF Molecule Templates, in: *Database and Expert Systems Applications - 28th International Conference, DEXA 2017, Lyon, France, August 28-31, 2017, Proceedings, Part I*, Lecture Notes in Computer Science, Vol. 10438, Springer, 2017, pp. 3–18. doi:10.1007/978-3-319-64468-4_1.

[244] K.M. Endris, P.D. Rohde, M. Vidal and S. Auer, Ontario: Federated Query Processing Against a Semantic Data Lake, in: *Database and Expert Systems Applications - 30th International Conference, DEXA 2019, Linz, Austria, August 26-29, 2019, Proceedings, Part I*, Lecture Notes in Computer Science, Vol. 11706, Springer, 2019, pp. 379–395. doi:10.1007/978-3-030-27615-7_29.

[245] F. Hacques, H. Skaf-Molli, P. Molli and S.E. Hassad, PFed: Recommending Plausible Federated SPARQL Queries, in: *Database and Expert Systems Applications - 30th International Conference, DEXA 2019, Linz, Austria, August 26-29, 2019, Proceedings, Part II*, Lecture Notes in Computer Science, Vol. 11707, Springer, 2019, pp. 184–197. doi:10.1007/978-3-030-27618-8_14.

[246] F. Ravat and Y. Zhao, Data Lakes: Trends and Perspectives, in: *Database and Expert Systems Applications - 30th International Conference, DEXA 2019, Linz, Austria, August 26-29, 2019, Proceedings, Part I*, Lecture Notes in Computer Science, Vol. 11706, Springer, 2019, pp. 304–313. doi:10.1007/978-3-030-27615-7_23.

[247] K.M. Endris, M. Vidal and S. Auer, FedSDM: Semantic Data Manager for Federations of RDF Datasets, in: *Data Integration in the Life Sciences - 13th International Conference, DILS 2018, Hannover, Germany, November 20-21, 2018, Proceedings*, Lecture Notes in Computer Science, Vol. 11371, Springer, 2018, pp. 85–90. doi:10.1007/978-3-030-06016-9_8.

[248] A. Nolle and G. Nemirovski, ELITE: An Entailment-Based Federated Query Engine for Complete and Transparent Semantic Data Integration, in: *Informal Proceedings of the 26th International Workshop on Description Logics, Ulm, Germany, July 23 - 26, 2013*, CEUR Workshop Proceedings, Vol. 1014, CEUR-WS.org, 2013, pp. 854–867.

[249] M. Buron, F. Goasdoué, I. Manolescu and M. Mugnier, Ontology-Based RDF Integration of Heterogeneous Data, in: *Proceedings of the 23rd International Conference on Extending Database Technology, EDBT 2020, Copenhagen, Denmark, March 30 - April 02, 2020*, OpenProceedings.org, 2020, pp. 299–310. doi:10.5441/002/edbt.2020.27.

[250] K. Makris, N. Bikakis, N. Gioldasis and S. Christodoulakis, SPARQL-RW: transparent query access over mapped RDF data sources, in: *15th International Conference on Extending Database Technology, EDBT '12, Berlin, Germany, March 27-30, 2012, Proceedings*, ACM, 2012, pp. 610–613. doi:10.1145/2247596.2247678.

[251] P.D. Rohde and M. Vidal, Optimizing Federated Queries Based on the Physical Design of a Data Lake, in: *Proceedings of the Workshops of the EDBT/ICDT 2020 Joint Conference, Copenhagen, Denmark, March 30, 2020*, CEUR Workshop Proceedings, Vol. 2578, CEUR-WS.org, 2020.

[252] J. Umbrich, M. Karnstedt, A. Hogan and J.X. Parreira, Freshening up while Staying Fast: Towards Hybrid SPARQL Queries, in: *Knowledge Engineering and Knowledge Management - 18th International Conference, EKAW 2012, Galway City, Ireland, October 8-12, 2012. Proceedings*, Lecture Notes in Computer Science, Vol. 7603, Springer, 2012, pp. 164–174. doi:10.1007/978-3-642-33876-2_16.

[253] R. Hai, C. Quix and D. Wang, Relaxed Functional Dependency Discovery in Heterogeneous Data Lakes, in: *Conceptual Modeling - 38th International Conference, ER 2019, Salvador, Brazil, November 4-7, 2019, Proceedings*, Lecture Notes in Computer Science, Vol. 11788, Springer, 2019, pp. 225–239. doi:10.1007/978-3-030-33223-5_19.

[254] P. Fafalios, T. Yannakis and Y. Tzitzikas, Querying the Web of Data with SPARQL-LD, in: *Research and Advanced Technology for Digital Libraries - 20th International Conference on Theory and Practice of Digital Libraries, TPDL 2016, Hannover, Germany, September 5-9, 2016, Proceedings*, Lecture Notes in Computer Science, Vol. 9819, Springer, 2016, pp. 175–187. doi:10.1007/978-3-319-43997-6_14.

[255] C.B. Aranda, M. Arenas and Ó. Corcho, Semantics and Optimization of the SPARQL 1.1 Federation Extension, in: *The Semanic Web: Research and Applications - 8th Extended Semantic Web Conference, ESWC 2011, Heraklion, Crete, Greece, May 29 - June 2, 2011, Proceedings, Part II*, Lecture Notes in Computer Science, Vol. 6644, Springer, 2011, pp. 1–15. doi:10.1007/978-3-642-21064-8_1.

[256] O. Hartig and G. Pirrò, A Context-Based Semantics for SPARQL Property Paths Over the Web, in: *The Semantic Web. Latest Advances and New Domains - 12th European Semantic Web Conference, ESWC 2015, Portoroz, Slovenia, May 31 - June 4, 2015. Proceedings*, Lecture Notes in Computer Science, Vol. 9088, Springer, 2015, pp. 71–87. doi:10.1007/978-3-319-18818-8_5.

[257] A. Hasnain, S.S. e Zainab, D. Zehra, Q. Mehmood, M. Saleem and D. Rebholz-Schuhmann, Federated Query Formulation and Processing through BioFed, in: *Proceedings of the Workshop on Semantic Web Solutions for Large-scale Biomedical Data Analytics co-located with 14th Extended Semantic Web Conference, SeWeBMeDA@ESWC 2017, Portoroz, Slovenia, May 28, 2017*, CEUR Workshop Proceedings, Vol. 1948, CEUR-WS.org, 2017, pp. 16–19.

[258] L. Heling, Quality-Driven Query Processing over Federated RDF Data Sources, in: *The Semantic Web: ESWC 2019 Satellite Events - ESWC 2019 Satellite Events, Portorož, Slovenia, June 2-6, 2019, Revised Selected Papers*, Lecture Notes in Computer Science, Vol. 11762, Springer, 2019, pp. 209–219. doi:10.1007/978-3-030-32327-1_40.

[259] D. Ibragimov, K. Hose, T.B. Pedersen and E. Zimányi, Processing Aggregate Queries in a Federation of SPARQL Endpoints, in: *The Semantic Web. Latest Advances and New Domains - 12th European Semantic Web Conference, ESWC 2015, Portoroz, Slovenia, May 31 - June 4, 2015. Proceedings*, Lecture Notes in Computer Science, Vol. 9088, Springer, 2015, pp. 269–285. doi:10.1007/978-3-319-18818-8_17.

[260] A.L. Jakobsen, G. Montoya and K. Hose, How Diverse Are Federated Query Execution Plans Really?, in: *The Semantic Web: ESWC 2019 Satellite Events - ESWC 2019 Satellite Events, Portorož, Slovenia, June 2-6, 2019, Revised Selected Papers*, Lecture Notes in Computer Science, Vol. 11762, Springer, 2019, pp. 105–110. doi:10.1007/978-3-030-32327-1_21.

[261] K. Kjernsmo, Sharing Statistics for SPARQL Federation Optimization, with Emphasis on Benchmark Quality, in: *The Semantic Web: Research and Applications - 9th Extended Semantic Web Conference, ESWC 2012, Heraklion, Crete, Greece, May 27-31, 2012. Proceedings*, Lecture Notes in Computer Science, Vol. 7295, Springer, 2012, pp. 828–832. doi:10.1007/978-3-642-30284-8_65.

[262] C. Kostopoulos, G. Mouchakis, A. Troumpoukis, N. Prokopaki-Kostopoulou, A. Charalambidis and S. Konstantopoulos, KOBE: Cloud-Native Open Benchmarking Engine for Federated Query Processors, in: *The Semantic Web - 18th International Conference, ESWC 2021, Virtual Event, June 6-10, 2021, Proceedings*, Lecture Notes in Computer Science, Vol. 12731, Springer, 2021, pp. 664–679. doi:10.1007/978-3-030-77385-4_40.

[263] K. Kurniawan, Semantic Query Federation for Scalable Security Log Analysis, in: *The Semantic Web: ESWC 2018 Satellite Events - ESWC 2018 Satellite Events, Heraklion, Crete, Greece, June 3-7, 2018, Revised Selected Papers*, Lecture Notes in Computer Science, Vol. 11155, Springer, 2018, pp. 294–303. doi:10.1007/978-3-319-98192-5_48.

[264] A. Langegger, W. Wöß and M. Blöchl, A Semantic Web Middleware for Virtual Data Integration on the Web, in: *The Semantic Web: Research and Applications, 5th European Semantic Web Conference, ESWC 2008, Tenerife, Canary Islands, Spain, June 1-5, 2008, Proceedings*, Lecture Notes in Computer Science, Vol. 5021, Springer, 2008, pp. 493–507. doi:10.1007/978-3-540-68234-9_37.

[265] M. Lefrançois, A. Zimmermann and N. Bakerally, A SPARQL Extension for Generating RDF from Heterogeneous Formats, in: *The Semantic Web - 14th International Conference, ESWC 2017, Portorož, Slovenia, May 28 - June 1, 2017, Proceedings, Part I*, Lecture Notes in Computer Science, Vol. 10249, 2017, pp. 35–50. doi:10.1007/978-3-319-58068-5_3.

[266] M.N. Mami, I. Grangel-González, D. Graux, E. Elezi and F. Lösch, Semantic Data Integration for the SMT Manufacturing Process Using SANSA Stack, in: *The Semantic Web: ESWC 2020 Satellite Events - ESWC 2020 Satellite Events, Heraklion, Crete, Greece, May 31 - June 4, 2020, Revised Selected Papers*, Lecture Notes in Computer Science, Vol. 12124, Springer, 2020, pp. 307–311. doi:10.1007/978-3-030-62327-2_47.

[267] M. Martin, J. Unbehauen and S. Auer, Improving the Performance of Semantic Web Applications with SPARQL Query Caching, in: *The Semantic Web: Research and Applications, 7th Extended Semantic Web Conference, ESWC 2010, Heraklion, Crete, Greece, May 30 - June 3, 2010, Proceedings, Part II*, Lecture Notes in Computer Science, Vol. 6089, Springer, 2010, pp. 304–318. doi:10.1007/978-3-642-13489-0_21.

[268] T. Minier, G. Montoya, H. Skaf-Molli and P. Molli, PeNeLoop: Parallelizing Federated SPARQL Queries in Presence of Replicated Fragments, in: *Joint Proceedings of the 2nd RDF Stream Processing (RSP 2017) and the Querying the Web of Data (QuWeDa 2017) Workshops co-located with 14th ESWC 2017 (ESWC 2017), Portoroz, Slovenia, May 28th - to - 29th, 2017*, CEUR Workshop Proceedings, Vol. 1870, CEUR-WS.org, 2017, pp. 37–50.

[269] T. Minier, G. Montoya, H. Skaf-Molli and P. Molli, Parallelizing Federated SPARQL Queries in Presence of Replicated Data, in: *The Semantic Web: ESWC 2017 Satellite Events - ESWC 2017 Satellite Events, Portorož, Slovenia, May 28 - June 1, 2017, Revised Selected Papers*, Lecture Notes in Computer Science, Vol. 10577, Springer, 2017, pp. 181–196. doi:10.1007/978-3-319-70407-4_33.

[270] A.N. Ngomo and M. Saleem, Federated Query Processing: Challenges and Opportunities, in: *Proceedings of the 3rd International Workshop on Dataset PROFIling and fEderated Search for Linked Data (PROFILES '16) co-located with the 13th ESWC 2016 Conference, Anissaras, Greece, May 30, 2016*, CEUR Workshop Proceedings, Vol. 1597, CEUR-WS.org, 2016.

[271] E.C. Ozkan, M. Saleem, E. Dogdu and A.N. Ngomo, UPSP: Unique Predicate-based Source Selection for SPARQL Endpoint Federation, in: *Proceedings of the 3rd International Workshop on Dataset PROFIling and fEderated Search for Linked Data (PROFILES '16) co-located with the 13th ESWC 2016 Conference, Anissaras, Greece, May 30, 2016*, CEUR Workshop Proceedings, Vol. 1597, CEUR-WS.org, 2016.

[272] F. Priyatna, C.B. Aranda and Ó. Corcho, Applying SPARQL-DQP for Federated SPARQL Querying over Google Fusion Tables, in: *The Semantic Web: ESWC 2013 Satellite Events - ESWC 2013 Satellite Events, Montpellier, France, May 26-30, 2013, Revised Selected Papers*, Lecture Notes in Computer Science, Vol. 7955, Springer, 2013, pp. 189–193. doi:10.1007/978-3-642-41242-4_22.

[273] B. Quilitz and U. Leser, Querying Distributed RDF Data Sources with SPARQL, in: *The Semantic Web: Research and Applications, 5th European Semantic Web Conference, ESWC 2008, Tenerife, Canary Islands, Spain, June 1-5, 2008, Proceedings*, Lecture Notes in Computer Science, Vol. 5021, Springer, 2008, pp. 524–538. doi:10.1007/978-3-540-68234-9_39.

[274] M. Saleem and A.N. Ngomo, HiBISCuS: Hypergraph-Based Source Selection for SPARQL Endpoint Federation, in: *The Semantic Web: Trends and Challenges - 11th International Conference, ESWC 2014, Anissaras, Crete, Greece, May 25-29, 2014. Proceedings*, Lecture Notes in Computer Science, Vol. 8465, Springer, 2014, pp. 176–191. doi:10.1007/978-3-319-07443-6_13.

[275] A. Schwarte, P. Haase, K. Hose, R. Schenkel and M. Schmidt, FedX: A Federation Layer for Distributed Query Processing on Linked Open Data, in: *The Semanic Web: Research and Applications - 8th Extended Semantic Web Conference, ESWC 2011, Heraklion, Crete, Greece, May 29 - June 2, 2011, Proceedings, Part II*, Lecture Notes in Computer Science, Vol. 6644, Springer, 2011, pp. 481–486. doi:10.1007/978-3-642-21064-8_39.

[276] V. Thost and J. Dolby, QED: Out-of-the-Box Datasets for SPARQL Query Evaluation, in: *The Semantic Web - 16th International Conference, ESWC 2019, Portorož, Slovenia, June 2-6, 2019, Proceedings*, Lecture Notes in Computer Science, Vol. 11503, Springer, 2019, pp. 491–506. doi:10.1007/978-3-030-21348-0_32.

[277] T. Yannakis, P. Fafalios and Y. Tzitzikas, Heuristics-based Query Reordering for Federated Queries in SPARQL 1.1 and SPARQL-LD, in: *Proceedings of the 3rd International Workshop on Geospatial Linked Data and the 2nd Workshop on Querying the Web of Data co-located with 15th Extended Semantic Web Conference (ESWC 2018), Heraklion, Greece, June 3, 2018*, CEUR Workshop Proceedings, Vol. 2110, CEUR-WS.org, 2018, pp. 74–88.

[278] G. Gombos and A. Kiss, Federated Query Evaluation Supported by SPARQL Recommendation, in: *Human Interface and the Management of Information: Information, Design and Interaction - 18th International Conference, HCI International 2016 Toronto, Canada, July 17-22, 2016, Proceedings, Part I*, Lecture Notes in Computer Science, Vol. 9734, Springer, 2016, pp. 263–274. doi:10.1007/978-3-319-40349-6_25.

[279] V. Gadepally, P. Chen, J. Duggan, A.J. Elmore, B. Haynes, J. Kepner, S. Madden, T. Mattson and M. Stonebraker, The BigDAWG polystore system and architecture, in: *2016 IEEE High Performance Extreme Computing Conference, HPEC 2016, Waltham, MA, USA, September 13-15, 2016*, IEEE, 2016, pp. 1–6. doi:10.1109/HPEC.2016.7761636.

[280] M. Saleem, A. Potocki, T. Soru, O. Hartig and A.N. Ngomo, CostFed: Cost-Based Query Optimization for SPARQL Endpoint Federation, in: *Proceedings of the 14th International Conference on Semantic Systems, SEMANTiCS 2018, Vienna, Austria, September 10-13, 2018*, Procedia Computer Science, Vol. 137, Elsevier, 2018, pp. 163–174. doi:10.1016/j.procs.2018.09.016.

[281] A. Charalambidis, A. Troumpoukis and S. Konstantopoulos, SemaGrow: optimizing federated SPARQL queries, in: *Proceedings of the 11th International Conference on Semantic Systems, SEMANTiCS 2015, Vienna, Austria, September 15-17, 2015*, ACM, 2015, pp. 121–128. doi:10.1145/2814864.2814886.

[282] M. Galkin, K.M. Endris, M. Acosta, D. Collarana, M. Vidal and S. Auer, SMJoin: A Multi-way Join Operator for SPARQL Queries, in: *Proceedings of the 13th International Conference on Semantic Systems, SEMANTiCS 2017, Amsterdam, The Netherlands, September 11-14, 2017*, ACM, 2017, pp. 104–111. doi:10.1145/3132218.3132220.

[283] P. Haase, T. Mathäß and M. Ziller, An evaluation of approaches to federated query processing over linked data, in: *Proceedings the 6th International Conference on Semantic Systems, I-SEMANTICS 2010, Graz, Austria, September 1-3, 2010*, ACM International Conference Proceeding Series, ACM, 2010. doi:10.1145/1839707.1839713.

[284] S. Jaiswal and M. Lefrançois, Towards Federated Queries for Web of Things Devices, in: *Joint Proceedings of SEMANTiCS 2017 Workshops co-located with the 13th International Conference on Semantic Systems (SEMANTiCS 2017), Amsterdam, Netherlands, September 11 and 14, 2017*, CEUR Workshop Proceedings, Vol. 2063, CEUR-WS.org, 2017.

[285] R. Singhal, N. Zhang, L. Nardi, M. Shahbaz and K. Olukotun, Polystore++: Accelerated Polystore System for Heterogeneous Workloads, in: *39th IEEE International Conference on Distributed Computing Systems, ICDCS 2019, Dallas, TX, USA, July 7-10, 2019*, IEEE, 2019, pp. 1641–1651. doi:10.1109/ICDCS.2019.00163.

[286] J.A. Blakeley, C. Cunningham, N. Ellis, B. Rathakrishnan and M. Wu, Distributed/Heterogeneous Query Processing in Microsoft SQL Server, in: *Proceedings of the 21st International Conference on Data Engineering, ICDE 2005, 5-8 April 2005, Tokyo, Japan*, IEEE Computer Society, 2005, pp. 1001–1012. doi:10.1109/ICDE.2005.51.

[287] W. Le, A. Kementsietsidis, S. Duan and F. Li, Scalable Multi-query Optimization for SPARQL, in: *IEEE 28th International Conference on Data Engineering (ICDE 2012), Washington, DC, USA (Arlington, Virginia), 1-5 April, 2012*, IEEE Computer Society, 2012, pp. 666–677. doi:10.1109/ICDE.2012.37.

[288] K. Schlegel, F. Stegmaier, S. Bayerl, M. Granitzer and H. Kosch, Balloon Fusion: SPARQL rewriting based on unified co-reference information, in: *Workshops Proceedings of the 30th International Conference on Data Engineering Workshops, ICDE 2014, Chicago, IL, USA, March 31 - April 4, 2014*, IEEE Computer Society, 2014, pp. 254–259. doi:10.1109/ICDEW.2014.6818335.

[289] M. Kaminski and E.V. Kostylev, Beyond Well-designed SPARQL, in: *19th International Conference on Database Theory, ICDT 2016, Bordeaux, France, March 15-18, 2016*, LIPIcs, Vol. 48, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016, pp. 5:1–5:18. doi:10.4230/LIPIcs.ICDT.2016.5.

[290] L.G. Azevedo, E.F. de Souza Soares, R. Souza and M.F. Moreno, Modern Federated Database Systems: An Overview, in: *Proceedings of the 22nd International Conference on Enterprise Information Systems, ICEIS 2020, Prague, Czech Republic, May 5-7, 2020, Volume 1*, SCITEPRESS, 2020, pp. 276–283. doi:10.5220/0009795402760283.

[291] M.L. Mouhoub, D. Grigori and M. Manouvrier, A Framework for Searching Semantic Data and Services with SPARQL, in: *Service-Oriented Computing - 12th International Conference, ICSOC 2014, Paris, France, November 3-6, 2014. Proceedings*, Lecture Notes in Computer Science, Vol. 8831, Springer, 2014, pp. 123–138. doi:10.1007/978-3-662-45391-9_9.

[292] C.G. Neto, L. Salgado, V. Ströele and D. de Oliveira, SigniFYIng APIs in the context of polystore systems: a case study with BigDAWG, in: *IHC '20: XIX Brazilian Symposium on Human Factors in Computing Systems, Online Event / Diamantina, Brazil, October 26-30, 2020*, ACM, 2020, pp. 57:1–57:6. doi:10.1145/3424953.3426654.

[293] S. Cheng and O. Hartig, FedQPL: A Language for Logical Query Plans over Heterogeneous Federations of RDF Data Sources, in: *iiWAS '20: The 22nd International Conference on Information Integration and Web-based Applications & Services, Virtual Event / Chiang Mai, Thailand, November 30 - December 2, 2020*, ACM, 2020, pp. 436–445. doi:10.1145/3428757.3429120.

[294] J. Lorey, SPARQL Endpoint Metrics for Quality-Aware Linked Data Consumption, in: *The 15th International Conference on Information Integration and Web-based Applications & Services, IIWAS '13, Vienna, Austria, December 2-4, 2013*, ACM, 2013, p. 319. doi:10.1145/2539150.2539240.

[295] M.N. Mami, D. Graux, S. Scerri, H. Jabeen, S. Auer and J. Lehmann, Uniform Access to Multiform Data Lakes using Semantic Technologies, in: *Proceedings of the 21st International Conference on Information Integration and Web-based Applications & Services, iiWAS 2019, Munich, Germany, December 2-4, 2019*, ACM, 2019, pp. 313–322. doi:10.1145/3366030.3366054.

[296] N.A. Rakhmawati, M. Saleem, S. Lalithsena and S. Decker, QFed: Query Set For Federated SPARQL Query Benchmark, in: *Proceedings of the 16th International Conference on Information Integration and Web-based Applications & Services, Hanoi, Vietnam, December 4-6, 2014*, ACM, 2014, pp. 207–211. doi:10.1145/2684200.2684321.

[297] H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann and S. Hübner, Ontology-Based Integration of Information - A Survey of Existing Approaches, in: *Proceedings of the IJCAI-01 Workshop on Ontologies and Information Sharing Seattle, USA, August 4-5, 2001*, CEUR Workshop Proceedings, Vol. 47, CEUR-WS.org, 2001.

[298] A. Valdestilhas, T. Soru and M. Saleem, More Complete Resultset Retrieval from Large Heterogeneous RDF Sources, in: *Proceedings of the 10th International Conference on Knowledge Capture, K-CAP 2019, Marina Del Rey, CA, USA, November 19-21, 2019*, ACM, 2019, pp. 223–230. doi:10.1145/3360901.3364436.

[299] A. Nikolov, P. Haase, J. Trame and A. Kozlov, Ephedra: Efficiently Combining RDF Data and Services Using SPARQL Federation, in: *Knowledge Engineering and Semantic Web - 8th International Conference, KESW 2017, Szczecin, Poland, November 8-10, 2017, Proceedings*, Communications in Computer and Information Science, Vol. 786, Springer, 2017, pp. 246–262. doi:10.1007/978-3-319-69548-8_17.

[300] N.A. Rakhmawati, J. Umbrich, M. Karnstedt, A. Hasnain and M. Hausenblas, A Comparison of Federation over SPARQL Endpoints Frameworks, in: *Knowledge Engineering and the Semantic Web - 4th International Conference, KESW 2013, St. Petersburg, Russia, October 7-9, 2013. Proceedings*, Communications in Computer and Information Science, Vol. 394, Springer, 2013, pp. 132–146. doi:10.1007/978-3-642-41360-5_11.

[301] R. Kontchakov and E.V. Kostylev, On Expressibility of Non-Monotone Operators in SPARQL, in: *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifteenth International Conference, KR 2016, Cape Town, South Africa, April 25-29, 2016*, AAAI Press, 2016, pp. 369–379.

[302] B. Golshan, A.Y. Halevy, G.A. Mihaila and W. Tan, Data Integration: After the Teenage Years, in: *Proceedings of the 36th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2017, Chicago, IL, USA, May 14-19, 2017*, ACM, 2017, pp. 101–106. doi:10.1145/3034786.3056124.

[303] M. Arenas and J. Pérez, Federation and Navigation in SPARQL 1.1, in: *Reasoning Web. Semantic Technologies for Advanced Query Answering - 8th International Summer School 2012, Vienna, Austria, September 3-8, 2012. Proceedings*, Lecture Notes in Computer Science, Vol. 7487, Springer, 2012, pp. 78–111. doi:10.1007/978-3-642-33158-9_3.

[304] P. Fafalios and Y. Tzitzikas, How many and what types of SPARQL queries can be answered through zero-knowledge link traversal?, in: *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, SAC 2019, Limassol, Cyprus, April 8-12, 2019*, ACM, 2019, pp. 2267–2274. doi:10.1145/3297280.3297505.

[305] J. Fink, M. Gobert and A. Cleve, Adapting Queries to Database Schema Changes in Hybrid Polystores, in: *20th IEEE International Working Conference on Source Code Analysis and Manipulation, SCAM 2020, Adelaide, Australia, September 28 - October 2, 2020*, IEEE, 2020, pp. 127–131. doi:10.1109/SCAM51674.2020.00019.

[306] A.M. Rinaldi and C. Russo, A Matching Framework for Multimedia Data Integration Using Semantics and Ontologies, in: *12th IEEE International Conference on Semantic Computing, ICSC 2018, Laguna Hills, CA, USA, January 31 - February 2, 2018*, IEEE Computer Society, 2018, pp. 363–368. doi:10.1109/ICSC.2018.00074.

[307] M. Acosta, M. Vidal, F. Flöck, S. Castillo, C.B. Aranda and A. Harth, SHEPHERD: A Shipping-Based Query Processor to Enhance SPARQL Endpoint Performance, in: *Proceedings of the ISWC 2014 Posters & Demonstrations Track a track within the 13th International Semantic Web Conference, ISWC 2014, Riva del Garda, Italy, October 21, 2014*, CEUR Workshop Proceedings, Vol. 1272, CEUR-WS.org, 2014, pp. 453–456.

[308] M. Acosta, M. Vidal, T. Lampo, J. Castillo and E. Ruckhaus, ANAPSID: An Adaptive Query Processing Engine for SPARQL Endpoints, in: *The Semantic Web - ISWC 2011 - 10th International Semantic Web Conference, Bonn, Germany, October 23-27, 2011, Proceedings, Part I*, Lecture Notes in Computer Science, Vol. 7031, Springer, 2011, pp. 18–34. doi:10.1007/978-3-642-25073-6_2.

[309] M.I. Ali, Q. Mehmood and M. Saleem, Assessing, Monitoring and Analyzing Linked Data Quality in Public SPARQL Endpoints, in: *Proceedings of the QuWeDa 2019: 3rd Workshop on Querying and Benchmarking the Web of Data co-located with 18th International Semantic Web Conference (ISWC 2019), Auckland, New Zealand, October 26-30, 2019*, CEUR Workshop Proceedings, Vol. 2496, CEUR-WS.org, 2019, pp. 37–50.

[310] C.B. Aranda, A. Hogan, J. Umbrich and P. Vandenbussche, SPARQL Web-Querying Infrastructure: Ready for Action?, in: *The Semantic Web - ISWC 2013 - 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part II*, Lecture Notes in Computer Science, Vol. 8219, Springer, 2013, pp. 277–293. doi:10.1007/978-3-642-41338-4_18.

[311] C.B. Aranda, A. Polleres and J. Umbrich, Strategies for Executing Federated Queries in SPARQL1.1, in: *The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part II*, Lecture Notes in Computer Science, Vol. 8797, Springer, 2014, pp. 390–405. doi:10.1007/978-3-319-11915-1_25.

[312] C. Basca and A. Bernstein, Avalanche: Putting the Spirit of the Web back into Semantic Web Querying, in: *Proceedings of the ISWC 2010 Posters & Demonstrations Track: Collected Abstracts, Shanghai, China, November 9, 2010*, CEUR Workshop Proceedings, Vol. 658, CEUR-WS.org, 2010.

[313] M. Buron, F. Goasdoué, I. Manolescu, T. Merabti and M. Mugnier, Revisiting RDF storage layouts for efficient query answering, in: *Proceedings of the 12th International Workshop on Scalable Semantic Web Knowledge Base Systems co-located with 19th International Semantic Web Conference (ISWC 2020), Athens, Greece, November 2, 2020*, CEUR Workshop Proceedings, Vol. 2757, CEUR-WS.org, 2020, pp. 17–32.

[314] S. Campinas, Live SPARQL Auto-Completion, in: *Proceedings of the ISWC 2014 Posters & Demonstrations Track a track within the 13th International Semantic Web Conference, ISWC 2014, Riva del Garda, Italy, October 21, 2014*, CEUR Workshop Proceedings, Vol. 1272, CEUR-WS.org, 2014, pp. 477–480.

[315] S. Castillo, G. Palma and M. Vidal, SILURIAN: a Sparql vIsuaLizer for UndeRstanding querIes And federatioNs, in: *Proceedings of the ISWC 2013 Posters & Demonstrations Track, Sydney, Australia, October 23, 2013*, CEUR Workshop Proceedings, Vol. 1035, CEUR-WS.org, 2013, pp. 137–140.

[316] D. Chaves-Fraga, C. Gutiérrez and Ó. Corcho, On the Role of the GRAPH Clause in the Performance of Federated SPARQL Queries, in: *Proceedings of the 4th International Workshop on Dataset PROFIling and fEderated Search for Web Data (PROFILES 2017) co-located with The 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, October 22, 2017*, CEUR Workshop Proceedings, Vol. 1927, CEUR-WS.org, 2017.

[317] P. Fafalios and Y. Tzitzikas, SPARQL-LD: a SPARQL Extension for Fetching and Querying Linked Data, in: *Proceedings of the ISWC 2015 Posters & Demonstrations Track co-located with the 14th International Semantic Web Conference (ISWC-2015), Bethlehem, PA, USA, October 11, 2015*, CEUR Workshop Proceedings, Vol. 1486, CEUR-WS.org, 2015.

[318] O. Görlitz and S. Staab, SPLENDID: SPARQL Endpoint Federation Exploiting VOID Descriptions, in: *Proceedings of the Second International Workshop on Consuming Linked Data (COLD2011), Bonn, Germany, October 23, 2011*, CEUR Workshop Proceedings, Vol. 782, CEUR-WS.org, 2011.

[319] O. Görlitz, M. Thimm and S. Staab, SPLODGE: Systematic Generation of SPARQL Benchmark Queries for Linked Open Data, in: *The Semantic Web - ISWC 2012 - 11th International Semantic Web Conference, Boston, MA, USA, November 11-15, 2012, Proceedings, Part I*, Lecture Notes in Computer Science, Vol. 7649, Springer, 2012, pp. 116–132. doi:10.1007/978-3-642-35176-1_8.

[320] T. Grubenmann, A. Bernstein, D. Moor and S. Seuken, Challenges of Source Selection in the WoD, in: *The Semantic Web - ISWC 2017 - 16th International Semantic Web Conference, Vienna, Austria, October 21-25, 2017, Proceedings, Part I*, Lecture Notes in Computer Science, Vol. 10587, Springer, 2017, pp. 313–328. doi:10.1007/978-3-319-68288-4_19.

[321] M. Gueroussova, A. Polleres and S.A. McIlraith, SPARQL with Qualitative and Quantitative Preferences, in: *Proceedings of the 2nd International Workshop on Ordering and Reasoning, OrdRing 2013, Co-located with the 12th International Semantic Web Conference (ISWC 2013), Sydney, Australia, October 22nd, 2013*, CEUR Workshop Proceedings, Vol. 1059, CEUR-WS.org, 2013, pp. 2–8.

[322] O. Hartig, C. Bizer and J.C. Freytag, Executing SPARQL Queries over the Web of Linked Data, in: *The Semantic Web - ISWC 2009, 8th International Semantic Web Conference, ISWC 2009, Chantilly, VA, USA, October 25-29, 2009. Proceedings*, Lecture Notes in Computer Science, Vol. 5823, Springer, 2009, pp. 293–309. doi:10.1007/978-3-642-04930-9_19.

[323] A. Hasnain, M. Saleem, A.N. Ngomo and D. Rebholz-Schuhmann, Extending LargeRDFBench for Multi-Source Data at Scale for SPARQL Endpoint Federation, in: *Emerging Topics in Semantic Technologies - ISWC 2018 Satellite Events [best papers from 13 of the workshops co-located with the ISWC 2018 conference]*, Studies on the Semantic Web, Vol. 36, IOS Press, 2018, pp. 203–218. doi:10.3233/978-1-61499-894-5-203.

[324] D. Hernández, A. Hogan, C. Riveros, C. Rojas and E. Zerega, Querying Wikidata: Comparing SPARQL, Relational and Graph Databases, in: *The Semantic Web - ISWC 2016 - 15th International Semantic Web Conference, Kobe, Japan, October 17-21, 2016, Proceedings, Part II*, Lecture Notes in Computer Science, Vol. 9982, 2016, pp. 88–103. doi:10.1007/978-3-319-46547-0_10.

[325] S. Jozashoori, D. Chaves-Fraga, E. Iglesias, M. Vidal and Ó. Corcho, FunMap: Efficient Execution of Functional Mappings for Knowledge Graph Creation, in: *The Semantic Web - ISWC 2020 - 19th International Semantic Web Conference, Athens, Greece, November 2-6, 2020, Proceedings, Part I*, Lecture Notes in Computer Science, Vol. 12506, Springer, 2020, pp. 276–293. doi:10.1007/978-3-030-62419-4_16.

[326] S. Konstantopoulos, A. Charalambidis, G. Mouchakis, A. Troumpoukis, J. Jakobitsch and V. Karkaletsis, Semantic Web Technologies and Big Data Infrastructures: SPARQL Federated Querying of Heterogeneous Big Data Stores, in: *Proceedings of the ISWC 2016 Posters & Demonstrations Track co-located with 15th International Semantic Web Conference (ISWC 2016), Kobe, Japan, October 19, 2016*, CEUR Workshop Proceedings, Vol. 1690, CEUR-WS.org, 2016.

[327] M.N. Mami, D. Graux, S. Scerri, H. Jabeen, S. Auer and J. Lehmann, How to Feed the Squerall with RDF and Other Data Nuts?, in: *Proceedings of the ISWC 2019 Satellite Tracks (Posters & Demonstrations, Industry, and Outrageous Ideas) co-located with 18th International Semantic Web Conference (ISWC 2019), Auckland, New Zealand, October 26-30, 2019*, CEUR Workshop Proceedings, Vol. 2456, CEUR-WS.org, 2019, pp. 293–296.

[328] M.N. Mami, D. Graux, S. Scerri, H. Jabeen, S. Auer and J. Lehmann, Squerall: Virtual Ontology-Based Access to Heterogeneous and Large Data Sources, in: *The Semantic Web - ISWC 2019 - 18th International Semantic Web Conference, Auckland, New Zealand, October 26-30, 2019, Proceedings, Part II*, Lecture Notes in Computer Science, Vol. 11779, Springer, 2019, pp. 229–245. doi:10.1007/978-3-030-30796-7_15.

[329] G. Montoya, H. Skaf-Molli and K. Hose, The Odyssey Approach for Optimizing Federated SPARQL Queries, in: *The Semantic Web - ISWC 2017 - 16th International Semantic Web Conference, Vienna, Austria, October 21-25, 2017, Proceedings, Part I*, Lecture Notes in Computer Science, Vol. 10587, Springer, 2017, pp. 471–489. doi:10.1007/978-3-319-68288-4_28.

[330] G. Montoya, H. Skaf-Molli, P. Molli and M. Vidal, Federated SPARQL Queries Processing with Replicated Fragments, in: *The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part I*, Lecture Notes in Computer Science, Vol. 9366, Springer, 2015, pp. 36–51. doi:10.1007/978-3-319-25007-6_3.

[331] G. Montoya, M. Vidal and M. Acosta, A Heuristic-Based Approach for Planning Federated SPARQL Queries, in: *Proceedings of the Third International Workshop on Consuming Linked Data, COLD 2012, Boston, MA, USA, November 12, 2012*, CEUR Workshop Proceedings, Vol. 905, CEUR-WS.org, 2012.

[332] G. Montoya, M. Vidal, Ó. Corcho, E. Ruckhaus and C.B. Aranda, Benchmarking Federated SPARQL Query Engines: Are Existing Testbeds Enough?, in: *The Semantic Web - ISWC 2012 - 11th International Semantic Web Conference, Boston, MA, USA, November 11-15, 2012, Proceedings, Part II*, Lecture Notes in Computer Science, Vol. 7650, Springer, 2012, pp. 313–324. doi:10.1007/978-3-642-35173-0_21.

[333] A. Nikolov, P. Haase, J. Trame and A. Kozlov, Ephedra: SPARQL Federation over RDF Data and Services, in: *Proceedings of the ISWC 2017 Posters & Demonstrations and Industry Tracks co-located with 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, October 23rd - to - 25th, 2017*, CEUR Workshop Proceedings, Vol. 1963, CEUR-WS.org, 2017.

[334] A. Nikolov, A. Schwarte and C. Hütter, FedSearch: Efficiently Combining Structured Queries and Full-Text Search in a SPARQL Federation, in: *The Semantic Web - ISWC 2013 - 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part I*, Lecture Notes in Computer Science, Vol. 8218, Springer, 2013, pp. 427–443. doi:10.1007/978-3-642-41335-3_27.

[335] A. Potocki, M. Saleem, T. Soru, O. Hartig, M. Voigt and A.N. Ngomo, Federated SPARQL Query Processing Via CostFed, in: *Proceedings of the ISWC 2017 Posters & Demonstrations and Industry Tracks co-located with 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, October 23rd - to - 25th, 2017*, CEUR Workshop Proceedings, Vol. 1963, CEUR-WS.org, 2017.

[336] M. Saleem, M.I. Ali, A. Hogan, Q. Mehmood and A.N. Ngomo, LSQ: The Linked SPARQL Queries Dataset, in: *The Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, Bethlehem, PA, USA, October 11-15, 2015, Proceedings, Part II*, Lecture Notes in Computer Science, Vol. 9367, Springer, 2015, pp. 261–269. doi:10.1007/978-3-319-25010-6_15.

[337] M. Saleem, A.N. Ngomo, J.X. Parreira, H.F. Deus and M. Hauswirth, DAW: Duplicate-AWare Federated Query Processing over the Web of Data, in: *The Semantic Web - ISWC 2013 - 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part I*, Lecture Notes in Computer Science, Vol. 8218, Springer, 2013, pp. 574–590. doi:10.1007/978-3-642-41335-3_36.

[338] F. Schmedding, Incremental SPARQL Evaluation for Query Answering on Linked Data, in: *Proceedings of the Second International Workshop on Consuming Linked Data (COLD2011), Bonn, Germany, October 23, 2011*, CEUR Workshop Proceedings, Vol. 782, CEUR-WS.org, 2011.

[339] M. Schmidt, O. Görlitz, P. Haase, G. Ladwig, A. Schwarte and T. Tran, FedBench: A Benchmark Suite for Federated Semantic Data Query Processing, in: *The Semantic Web - ISWC 2011 - 10th International Semantic Web Conference, Bonn, Germany, October 23-27, 2011, Proceedings, Part I*, Lecture Notes in Computer Science, Vol. 7031, Springer, 2011, pp. 585–600. doi:10.1007/978-3-642-25073-6_37.

[340] A. Schwarte, P. Haase, K. Hose, R. Schenkel and M. Schmidt, FedX: Optimization Techniques for Federated Query Processing on Linked Data, in: *The Semantic Web - ISWC 2011 - 10th International Semantic Web Conference, Bonn, Germany, October 23-27, 2011, Proceedings, Part I*, Lecture Notes in Computer Science, Vol. 7031, Springer, 2011, pp. 601–616. doi:10.1007/978-3-642-25073-6_38.

[341] R. Taelman, J.V. Herwegen, M.V. Sande and R. Verborgh, Comunica: A Modular SPARQL Query Engine for the Web, in: *The Semantic Web - ISWC 2018 - 17th International Semantic Web Conference, Monterey, CA, USA, October 8-12, 2018, Proceedings, Part II*, Lecture Notes in Computer Science, Vol. 11137, Springer, 2018, pp. 239–255. doi:10.1007/978-3-030-00668-6_15.

[342] A. Troumpoukis, S. Konstantopoulos, G. Mouchakis, N. Prokopaki-Kostopoulou, C. Paris, L. Bruzzone, D. Pantazi and M. Koubarakis, GeoFedBench: A Benchmark for Federated GeoSPARQL Query Processors, in: *Proceedings of the ISWC 2020 Demos and Industry Tracks: From Novel Ideas to Industrial Practice co-located with 19th International Semantic Web Conference (ISWC 2020), Globally online, November 1-6, 2020 (UTC)*, CEUR Workshop Proceedings, Vol. 2721, CEUR-WS.org, 2020, pp. 228–232.

[343] J. Umbrich, M. Karnstedt, A. Hogan and J.X. Parreira, Hybrid SPARQL Queries: Fresh vs. Fast Results, in: *The Semantic Web - ISWC 2012 - 11th International Semantic Web Conference, Boston, MA, USA, November 11-15, 2012, Proceedings, Part I*, Lecture Notes in Computer Science, Vol. 7649, Springer, 2012, pp. 608–624. doi:10.1007/978-3-642-35176-1_38.

[344] H. Wu, A. Yamaguchi and J. Kim, Dynamic Join Order Optimization for SPARQL Endpoint Federation, in: *Proceedings of the 11th International Workshop on Scalable Semantic Web Knowledge Base Systems co-located with 14th International Semantic Web Conference (ISWC 2015), Bethlehem, PA, USA, October 11, 2015*, CEUR Workshop Proceedings, Vol. 1457, CEUR-WS.org, 2015, pp. 48–62.

[345] R. Alotaibi, D. Bursztyn, A. Deutsch, I. Manolescu and S. Zampetakis, Towards Scalable Hybrid Stores: Constraint-Based Rewriting to the Rescue, in: *Proceedings of the 2019 International Conference on Management of Data, SIGMOD Conference 2019, Amsterdam, The Netherlands, June 30 - July 5, 2019*, ACM, 2019, pp. 1660–1677. doi:10.1145/3299869.3319895.

[346] R. Hai, S. Geisler and C. Quix, Constance: An Intelligent Data Lake System, in: *Proceedings of the 2016 International Conference on Management of Data, SIGMOD Conference 2016, San Francisco, CA, USA, June 26 - July 01, 2016*, ACM, 2016, pp. 2097–2100. doi:10.1145/2882903.2899389.

[347] D. Halperin, V.T. de Almeida, L.L. Choo, S. Chu, P. Koutris, D. Moritz, J. Ortiz, V. Ruamviboonsuk, J. Wang, A. Whitaker, S. Xu, M. Balazinska, B. Howe and D. Suciu, Demonstration of the Myria big data management service, in: *International Conference on Management of Data, SIGMOD 2014, Snowbird, UT, USA, June 22-27, 2014*, ACM, 2014, pp. 881–884. doi:10.1145/2588555.2594530.

[348] K. Hose and R. Schenkel, Towards benefit-based RDF source selection for SPARQL queries, in: *Proceedings of the 4th International Workshop on Semantic Web Information Management, SWIM 2012, Scottsdale, AZ, USA, May 20, 2012*, ACM, 2012, p. 2. doi:10.1145/2237867.2237869.

[349] L. Xu, R.L. Cole and D. Ting, Learning to optimize federated queries, in: *Proceedings of the Second International Workshop on Exploiting Artificial Intelligence Techniques for Data Management, aiDM@SIGMOD 2019, Amsterdam, The Netherlands, July 5, 2019*, ACM, 2019, pp. 2:1–2:7. doi:10.1145/3329859.3329873.

[350] S. Bouarar, L. Bellatreche and A. Roukh, Eco-Data Warehouse Design Through Logical Variability, in: *SOFSEM 2017: Theory and Practice of Computer Science - 43rd International Conference on Current Trends in Theory and Practice of Computer Science, Limerick, Ireland, January 16-20, 2017, Proceedings*, Lecture Notes in Computer Science, Vol. 10139, Springer, 2017, pp. 436–449. doi:10.1007/978-3-319-51963-0_34.

[351] I. Megdiche, F. Ravat and Y. Zhao, Metadata Management on Data Processing in Data Lakes, in: *SOFSEM 2021: Theory and Practice of Computer Science - 47th International Conference on Current Trends in Theory and Practice of Computer Science, SOFSEM 2021, Bolzano-Bozen, Italy, January 25-29, 2021, Proceedings*, Lecture Notes in Computer Science, Vol. 12607, Springer, 2021, pp. 553–562. doi:10.1007/978-3-030-67731-2_40.

[352] A. Stolpe, J. Halvorsen and B.J. Hansen, Supporting Evacuation Missions with Ontology-Based SPARQL Federation, in: *Proceedings of the Eighth Conference on Semantic Technologies for Intelligence, Defense, and Security, Fairfax VA, USA, November 12-15, 2013*, CEUR Workshop Proceedings, Vol. 1097, CEUR-WS.org, 2013, pp. 141–148.

[353] A.P. Sheth, Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases, in: *17th International Conference on Very Large Data Bases, September 3-6, 1991, Barcelona, Catalonia, Spain, Proceedings*, Morgan Kaufmann, 1991, p. 489.

[354] F. Michel, C. Faron-Zucker and J. Montagnat, A Generic Mapping-based Query Translation from SPARQL to Various Target Database Query Languages, in: *Proceedings of the 12th International Conference on Web Information Systems and Technologies, WEBIST 2016, Volume 2, Rome, Italy, April 23-25, 2016*, SciTePress, 2016, pp. 147–158. doi:10.5220/0005905401470158.

[355] N.A. Rakhmawati, M. Karnstedt, M. Hausenblas and S. Decker, On Metrics for Measuring Fragmentation of Federation over SPARQL Endpoints, in: *WEBIST 2014 - Proceedings of the 10th International Conference on Web Information Systems and Technologies, Volume 1, Barcelona, Spain, 3-5 April, 2014*, SciTePress, 2014, pp. 119–126. doi:10.5220/0004760101190126.

[356] M. Saleem, G. Szárnyas, F. Conrads, S.A.C. Bukhari, Q. Mehmood and A.N. Ngomo, How Representative Is a SPARQL Benchmark? An Analysis of RDF Triplestore Benchmarks, in: *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, ACM, 2019, pp. 1623–1633. doi:10.1145/3308558.3313556.

[357] M. Acosta, E. Simperl, F. Flöck and M. Vidal, HARE: An Engine for Enhancing Answer Completeness of SPARQL Queries via Crowdsourcing, in: *Companion of the The Web Conference 2018 on The Web Conference 2018, WWW 2018, Lyon , France, April 23-27, 2018*, ACM, 2018, pp. 501–505. doi:10.1145/3184558.3186241.

[358] Z. Akar, T.G. Halaç, E.E. Ekinci and O. Dikenelli, Querying the Web of Interlinked Datasets using VOID Descriptions, in: *WWW2012 Workshop on Linked Data on the Web, Lyon, France, 16 April, 2012*, CEUR Workshop Proceedings, Vol. 937, CEUR-WS.org, 2012.

[359] A. Charalambidis, S. Konstantopoulos and V. Karkaletsis, Dataset Descriptions for Optimizing Federated Querying, in: *Proceedings of the 24th International Conference on World Wide Web Companion, WWW 2015, Florence, Italy, May 18-22, 2015 - Companion Volume*, ACM, 2015, pp. 17–18. doi:10.1145/2740908.2742779.

[360] M. Kaminski, E.V. Kostylev and B.C. Grau, Semantics and Expressive Power of Subqueries and Aggregates in SPARQL 1.1, in: *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, ACM, 2016, pp. 227–238. doi:10.1145/2872427.2883022.

[361] M.N. Mami, D. Graux, S. Scerri, H. Jabeen and S. Auer, Querying Data Lakes using Spark and Presto, in: *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, ACM, 2019, pp. 3574–3578. doi:10.1145/3308558.3314132.

[362] F. Michel, C. Faron-Zucker and F. Gandon, SPARQL Micro-Services: Lightweight Integration of Web APIs and Linked Data, in: *Workshop on Linked Data on the Web co-located with The Web Conference 2018, LDOW@WWW 2018, Lyon, France April 23rd, 2018*, CEUR Workshop Proceedings, Vol. 2073, CEUR-WS.org, 2018.

[363] A. Gaignard, J. Montagnat, C.F. Zucker and O. Corby, Semantic Federation of Distributed Neurodata, in: *MICCAI Workshop on Data-and Compute-Intensive Clinical and Translational Imaging Applications*, 2012, pp. 41–50.

[364] L. Golubchik, S. Khuller, K. Mukherjee and Y. Yao, To send or not to send: Reducing the cost of data transmission, in: *2013 Proceedings IEEE INFOCOM*, IEEE, 2013, pp. 2472–2478.

[365] B. Moreau and P. Serrano-Alvarado, Ensuring License Compliance in Federated Query Processing, in: *36ème Conférence sur la Gestion de Données–Principes, Technologies et Applications (BDA 2020)*, 2020.

[366] R. Mukherjee and P. Kar, A comparative review of data warehousing ETL tools with new trends and industry insight, in: *2017 IEEE 7th International Advance Computing Conference (IACC)*, IEEE, 2017, pp. 943–948.

[367] M. Gobert, Schema Evolution in Hybrid Databases Systems, in: *proceedings of the 46th International Conference on Very Large Data Bases: PhD workshop track*, 2020.

[368] R. Sethi, M. Traverso, D. Sundstrom, D. Phillips, W. Xie, Y. Sun, N. Yegitbasi, H. Jin, E. Hwang, N. Shingte et al., Presto: SQL on everything, in: *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, IEEE, 2019, pp. 1802–1813.

[369] E. Begoli, J. Camacho-Rodríguez, J. Hyde, M.J. Mior and D. Lemire, Apache calcite: A foundational framework for optimized query processing over heterogeneous data sources, in: *Proceedings of the 2018 International Conference on Management of Data*, 2018, pp. 221–230.

[370] S. Kim and B. Moon, Federated database system for scientific data, in: *Proceedings of the 30th International Conference on Scientific and Statistical Database Management*, 2018, pp. 1–4.

[371] X.-S. Vu, A. Ait-Mlouk, E. Elmroth and L. Jiang, Graph-based interactive data federation system for heterogeneous data retrieval and analytics, in: *The World Wide Web Conference*, 2019, pp. 3595–3599.

[372] A. Abel, Faster SPARQL Federated Queries, PhD thesis, Université Rennes1, 2019.

[373] C. Basca, Federated SPARQL Query Processing Reconciling Diversity, Flexibility and Performance on the Web of Data, PhD thesis, University of Zurich, 2015.

[374] D. Bilidas, Database techniques for ontology-based data access, PhD thesis, National and Kapodistrian University of Athens, 2020.

[375] M. Buron, Efficient reasoning on large-scale heterogeneous data, PhD thesis, Institut Polytechnique de Paris, 2020.

[376] 2020 Gartner Magic Quadrant for Data Integration Tools, Accessed 31 November 2020. https://www.bmc.com/blogs/gartner-magic-quadrant-data-integration-tools/.

[377] K.M. Endris, Federated Query Processing over Heterogeneous Data Sources in a Semantic Data Lake, PhD thesis, University of Bonn, Germany, 2020. http://hdl.handle.net/20.500.11811/8347.

[378] M. Saleem, Efficient source selection and benchmarking for SPARQL endpoint query federation, PhD thesis, Leipzig University, Germany, 2018. ISBN 978-3-89838-732-3. https://d-nb.info/1162645547.

[379] A. Valdestilhas, Identifying, Relating, Consisting and Querying Large Heterogeneous RDF Sources, PhD thesis, Leipzig University, Germany, 2021. https://nbn-resolving.org/urn:nbn:de:bsz:15-qucosa2-732931.

[380] R. Ayed, Aggregated search in Distributed Graph Databases. (Recherche d'information agrégative dans des bases de graphes distribuées), PhD thesis, University of Lyon, France, 2019. https://tel.archives-ouvertes.fr/tel-02520460.

[381] M. Buron, Efficient reasoning on large and heterogeneous graphs. (Raisonnement efficace sur des grandsgraphes hétérogènes), PhD thesis, École Polytechnique, Palaiseau, France, 2020. https://tel.archives-ouvertes.fr/tel-03107689.

[382] R. Hai, R. Miller, M. Jarke and C.J. Quix, Data Integration and Metadata Management in Data Lakes, Technical Report, Lehrstuhl für Informatik 5 (Informationssysteme und Datenbanken), 2020.

[383] S.M.A. Hasnain, Cataloguing and linking publicly available biomedical SPARQL endpoints for federation-addressing aPosteriori data integration, PhD thesis, National University of IReland, Galway, 2017.

[384] P. Molli, H. Skaf-Molli and A. Grall, SemCat: Source Selection Services for Linked Data, PhD thesis, université de Nantes, 2020.

[385] J. Pålsson, Querying Federations of Eiffel Event Data Repositories, 2020.

[386] N.A. Rakhmawati, Evaluating and benchmarking the performance of federated SPARQL endpoints and their partitioning using selected metrics and specific query types, PhD thesis, National University of Ireland, Galway, 2017.

[387] P.D. Rohde, Query Optimization Techniques For Scaling Up To Data Variety, Master's thesis, Hannover: Institutionelles Repositorium der Leibniz Universität Hannover, 2019.

[388] C.R. da Silva Teixeira, Implementation of a data virtualization laver applied to insurance data, Master's thesis, University of Porto, 2016.

[389] M. Wigham, State of the art in federated querying in SPARQL, Technical Report, Wageningen UR, 2014.

[390] L. Xu, New capabilities for large-scale exploratory data analysis, PhD thesis, University of Illinois at Urbana-Champaign, 2020.

[391] D. Clunie, H. Hickman, W. Ver Hoef, S. Hastak, J. Evans, J. Neville and U. Wagner, Observations from the Data Integration and Imaging Informatics (DI-Cubed) Project, Technical Report, 2020.
[392] P. Serrano-Alvarado, Protecting user data in distributed systems, PhD thesis, Université de Nantes (UN), 2020.